

2

AD-A277 313



Unc
SECURI

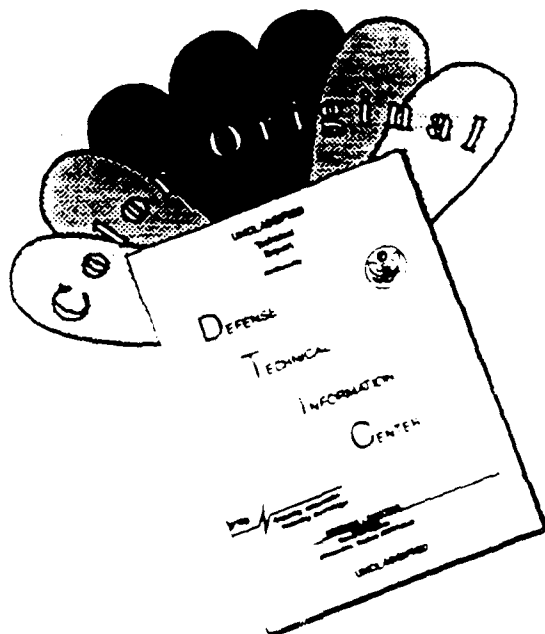
ATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release: distribution unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR. 94 0073	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research	
6a. NAME OF PERFORMING ORGANIZATION Carnegie Mellon University	6b. OFFICE SYMBOL (If applicable) NE	7b. ADDRESS (City, State, and ZIP Code) Program Manager, Neural Networks Bolling AFB, D.C. 20332-6448	
6c. ADDRESS (City, State, and ZIP Code) 5000 Forbes Avenue Pittsburgh, PA 15213		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-89-0551	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) NE	10. SOURCE OF FUNDING NUMBERS	
8c. ADDRESS (City, State, and ZIP Code) Building 410 Bolling AFB, D.C. 20332-6448		PROGRAM ELEMENT NO. 61102F	TASK NO. 2305
		WORK UNIT B3	ACCESSION NO.
11. TITLE (Include Security Classification) A Differential Theory of Learning for Efficient Statistical Pattern Recognition			
12. PERSONAL AUTHOR(S) John Hampshire and B.V.K. Vijaya Kumar			
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 9-30-89-4-24-93	14. DATE OF REPORT (Year, Month, Day) 12/15/93	15. PAGE COUNT 455
16. SUPPLEMENTARY NOTATION None			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		Learning, Pattern Recognition, Classification, Neural Networks	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Probabilistic learning strategies currently used are inefficient, requiring high classifier complexity and large training samples. In this report, we introduce and analyze an asymptotically efficient differential learning strategy. It guarantees the best generalization allowed by the chosen classifier paradigm. Differential learning also requires the classifier with minimal complexity. The theory is demonstrated in several real-world machine learning/pattern recognition tasks.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Steven Supparth		22b. TELEPHONE (Include Area Code) 202/762-5028	22c. OFFICE SYMBOL NM

Original document color
plotted: All DTIC reproductions
will be in black and
white.

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

Contents

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	

1 Overview	1
1.1 Abstract	1
1.2 Intended Audience	2
1.3 Outline of the Text	2
1.3.1 Summary of Findings	3
1.3.2 Profile of the Chapters	5
I Theory	11
2 Probabilistic and Differential Strategies for Learning the Bayesian Discriminant Function	13
2.1 Introduction	13
2.2 Bayesian Discrimination	14
2.2.1 The Classifier and the Bayesian Discriminant Function	16
2.2.2 Probabilistic and Differential forms of the Bayesian Discriminant Function	18
2.2.3 Learning Paradigms for the Bayesian Discriminant Function	23
2.2.4 The Link Between Objective Function and Learning Strategy	28
2.3 Probabilistic Learning Λ_P	31
2.3.1 The General Error Measure	32
2.3.2 Specific Strictly Probabilistic Error Measures	36
2.3.3 Minkowski-r Power Metrics and Other Common Error Measures	41
2.4 Differential Learning Λ_Δ	43
2.4.1 Further Constraints Imposed on ψ by the Discriminator	50
2.5 Summary	51
3 Differential Learning is Asymptotically Efficient	53
3.1 Introduction	53
3.2 Discriminant Error, the Efficient Classifier, and the Efficient Learning Strategy	54
3.2.1 Learning and Expectation	55
3.2.2 Discriminant Error and the Efficient Classifier	58
3.2.3 Efficient Learning	62



3.3	Differential Learning is Asymptotically Efficient	66
3.3.1	Differential Learning Generates Consistent Classifiers	69
3.3.2	A Word Regarding "Agnostic" Learning	70
3.4	Discriminant Error Versus Functional Error, and the Inefficiency of Probabilistic Learning	71
3.5	Differential Learning Requires the Minimum-Complexity Classifier	72
3.6	The Case for Probabilistic Learning	76
3.6.1	Assessing the Asymptotic Relative Efficiency (ARE) of a non-Differential Learning Strategy	79
3.6.2	A Word Regarding "Proper Models"	81
3.7	Summary	81
4	The Robust Beauty of Differentially-Generated Improper Parametric Models	83
4.1	Introduction	83
4.2	Analysis of a Proper Parametric Model	85
4.2.1	The Proper Parametric Model	85
4.2.2	Probabilistic Learning for the Asymptotically Large Training Sample	89
4.2.3	Differential Learning via CFM for the Asymptotically Large Training Sample	90
4.2.4	Results of Differential and Probabilistic Learning for Asymptotically Large and Small Training Samples	92
4.3	Analysis of an Improper Parametric Model	99
4.3.1	The Improper Parametric Model	100
4.3.2	Probabilistic Learning via MSE for the Asymptotically Large Training Sample	102
4.3.3	Differential Learning via CFM for the Asymptotically Large Training Sample	104
4.3.4	Results of Differential and Probabilistic Learning for Asymptotically Large and Small Training Samples	105
4.4	Summary	111
5	Properties of the CFM Objective Function	113
5.1	Introduction	113
5.2	Discriminator Output Space	114
5.2.1	The Discriminant Differential δ_τ , the Reduced Discriminant Continuum, and the Reduced Discriminant Boundary	120
5.3	Objective Function Monotonicity and Learning Efficiency	122
5.3.1	MAE is Non-Monotonic	127
5.3.2	MSE is Non-Monotonic	132
5.3.3	The Kullback-Leibler Information Distance is Non-Monotonic	136
5.3.4	The General Error Measure is Non-Monotonic	140
5.3.5	The Link Between Objective Function Monotonicity and Learning Efficiency	141

5.3.6	CFM is Monotonic	142
5.4	Training Example Types	148
5.5	The Convergence Properties of Differential Learning via CFM	150
5.5.1	Differential Learning via the Synthetic Form of CFM is Reasonably Fast	153
5.5.2	Differential Learning via the Original Forms of CFM is Unreasonably Slow and/or Inefficient	155
5.6	Summary	156
6	An Information-Theoretic View of Stochastic Concept Learning	159
6.1	Introduction	159
6.2	Probabilistic versus Differential Complexity	161
6.3	Exploring the Curious Relationship Between Winning a Rigged Game of Dice and Building an Efficient Classifier	163
6.3.1	The Differential Mechanism by which the Most Likely Die Face Becomes Empirically Evident	164
6.3.2	The Probabilistic Mechanism by which the Most Likely Die Face Becomes Empirically Evident	171
6.3.3	Discriminant Information versus Probabilistic Information	175
6.4	Bounds on the Training Sample Size Requirements of the Differential and Probabilistic Learning Strategies	176
6.4.1	A Greatest Lower Bound on n_{Δ}	177
6.4.2	A Greatest Lower Bound on n_P	178
6.5	Extending the Rigged Die Paradigm to the General C -class Learning/Pattern Recognition Task	179
6.6	Summary	181
II	Applications	183
7	Implementing Differential Learning	185
7.1	Introduction	185
7.2	Three Hypothesis Classes	186
7.2.1	The Linear Hypothesis Class	186
7.2.2	The Logistic Linear Hypothesis Class	186
7.2.3	The Gaussian Radial Basis Hypothesis Class	187
7.3	Learning to Identify the Irises of the Gaspé Peninsula	187
7.4	Controlling the Confidence Parameter	191
7.5	Focussing on the Un-Learned Examples	193
7.6	Rejecting the Classification	202
7.7	The Importance of Representational Choices	207

7.8	Minimizing the Classifier's Complexity	215
7.9	Summary	217
8	Optical Character Recognition with Differential Learning	219
8.1	Introduction	219
8.1.1	A Word Regarding Training and Test Samples	220
8.2	Test and Evaluation Protocols	221
8.2.1	Estimating Error Rates	221
8.2.2	Estimating a Classifier's MSDE	222
8.2.3	Graphical Statistical Summaries	224
8.3	Compressing the Data to Improve Generalization	225
8.4	Recognition Results	230
8.4.1	Experiments with the Logistic Linear Hypothesis Class	232
8.4.2	Experiments with Alternative Hypothesis Classes	239
8.4.3	Interpretation of Results	243
8.4.4	Rejecting Classifications After Learning	246
8.5	Recognition Results in the Presence of Noise	248
8.5.1	Signal-to-Noise Ratio (SNR) Computations	248
8.5.2	The Compressed Noisy Feature Vector is Approximately Homoscedastic Gaussian	250
8.5.3	Recognition Results for a Moderate SNR	253
8.5.4	Recognition Results for a Low SNR	261
8.6	Summary	266
9	Medical Diagnosis with Differential Learning	271
9.1	Introduction	271
9.2	Recognition Results	277
9.3	Summary	284
10	Remote Sensing with Differential Learning	285
10.1	Introduction	285
10.2	Training Data	287
10.3	Experimental Results	289
10.3.1	Interpretation of Test Results	298
10.4	Summary	301
11	Conclusions	303
11.1	Scientific Contributions	303
11.2	Philosophical Implications of Differential Learning	304

11.3 Future Research	307
A Glossary of Notation	311
B Notes on Convergence	319
C The Box Plot Statistical Summary	321
C.1 How to Read a Box Plot	321
C.2 How to Construct a Box Plot	323
D A Synthetic Functional Form of the Classification Figure of Merit	327
D.1 Specifications for the Synthetic CFM Objective Function	328
D.2 The Computational Cost of the Synthetic CFM Objective Function	334
D.3 The Convergence Properties of Differential Learning via the CFM Objective Function	334
D.3.1 The Convergence Properties Differential Learning via the Original Logistic Sigmoidal Form of CFM	336
D.3.2 The Convergence Properties of Differential Learning via the Synthetic Form of CFM	340
D.4 A Proof Relating to Synthetic CFM and Chapter 2	346
D.5 Modifying Backpropagation for use with CFM	348
D.6 Source Code for the Synthetic CFM Objective Function	351
E Differential Learning via CFM Viewed as a Generalization of Learning via Rosenblatt's Perceptron Criterion Function	359
F Proper Parametric Models of the Homoscedastic Gaussian Feature Vector	363
F.1 The Fully-Parametric Proper Model	364
F.2 The Partially-Parametric Proper Model	365
F.2.1 $C = 2$: Logistic Regression	365
F.2.2 $C > 2$: Logistic Discriminant Analysis	367
F.3 The Asymptotic Relative Efficiency of Logistic Discriminant Analysis Versus Normal-Based Linear Discriminant Analysis	368
F.4 The Proper Parametric Model Constraints are Severe	369
G Error Rate Computations for the Classifiers of Chapter 4	371
G.1 Error Rate Computations for the Proper Parametric Model	371
G.2 Error Rate Computations for the Improper Parametric Model	372

H Asymptotic Parameterizations for the Probabilistically-Generated Improper Parametric Models of Chapter 4	375
H.1 Distribution-Independent Expressions for the Parameterization of Low-Order Polynomial Discriminant Functions	377
H.2 Distribution-Dependent Expressions for the Parameterization of Polynomial Discriminant Functions	379
I Monotonic Fractions Generated by Three Error Measures	383
I.1 MAE Monotonic Fractions	383
I.2 MSE Monotonic Fractions	385
I.3 Kullback-Leibler Monotonic Fractions	387
J Tabulated Die Casting Bounds	391
K A Modified Radial Basis Function Classifier	405
L Anderson & Fisher's Iris Data	407
L.1 Original Iris Data	407
L.2 Normalized Iris Data	409
M Complexity Reduction Techniques	413
M.1 Weight Decay	413
M.1.1 Parametric Entropy	415
M.2 Weight Smoothing	416
M.3 Linear Non-Invertible Feature Vector Compression	417
M.3.1 A Brief Argument Against Principal Components Analysis	419
Index	431

List of Tables

2.1	Some well-known classification paradigms and the learning strategies they employ.	25
4.1	The minimum-MSE parameterizations for the minimum-, low-, and high-complexity polynomial classifiers of x when the training sample size n is asymptotically large (i.e., $n \rightarrow \infty$).	104
8.1	Estimated MSDE for the high and low-complexity logistic linear classifiers employing differential and probabilistic learning.	232
8.2	Estimated discriminant bias, discriminant variance, and MSDE for 650-parameter classifiers generated from the linear, logistic linear, and modified RBF hypothesis classes by differential and probabilistic learning.	245
8.3	Empirical benchmark test sample error rates for 650-parameter classifiers produced from the linear, logistic linear, and modified RBF hypothesis classes by differential and probabilistic learning.	247
8.4	Benchmark test sample rejection/empirical error rate statistics for 650-parameter classifiers produced from the linear, logistic linear, and modified RBF hypothesis classes by differential and probabilistic learning.	247
8.5	Estimated relative efficiency of differential versus probabilistic learning for the linear, logistic linear, and modified Gaussian RBF hypothesis classes.	258
9.1	Estimated discriminant bias, discriminant variance, and MSDE for the 257-parameter logistic linear classifiers generated by differential and probabilistic learning. Estimates are also shown for Manduca, Christy, and Ehman's 2050-parameter logistic linear and 6164-parameter multi-layer perceptron (MLP) classifiers, both generated probabilistically with the MSE objective function [90]. Finally, estimates are shown for 10 Human subjects [90]: each performed one 2-fold cross validation trial.	282
10.1	The training sample sizes for both the maximum-likelihood and DRBF classifiers.	288

10.2	Left: Class labels assigned to the 11 ground truth classes. Right: Top ten confusions made by the DRBF classifier over the civill site.	291
10.3	Confusion matrix for the DRBF classifier over the civill site.	291
10.4	Left: Class labels assigned to the 11 ground truth classes. Right: Top ten confusions made by the maximum-likelihood (ML) classifier over the civill site.	292
10.5	Confusion matrix for the maximum-likelihood (ML) classifier over the civill site.	292
10.6	Left: Class labels assigned to the 11 ground truth classes. Right: Top ten confusions made by the DRBF classifier over the gao1 site.	295
10.7	Confusion matrix for the DRBF classifier over the gao1 site.	295
10.8	Left: Class labels assigned to the 11 ground truth classes. Right: Top ten confusions made by the maximum-likelihood (ML) classifier over the gao1 site.	296
10.9	Confusion matrix for the maximum-likelihood (ML) classifier over the gao1 site.	296
10.10A	summary of the empirical test sample error rates for both the maximum-likelihood and DRBF classifiers.	298
C.1	A listing of indices used to compute box plot 5-number summaries for various sample sizes.	324

List of Figures

2.1	Left: The class-conditional density — class prior probability products $\rho_{\mathbf{x} \mathbf{w}}(x \omega_i) \cdot P_{\mathbf{w}}(\omega_i)$ for a three-class random scalar x . Right: The associated <i>a posteriori</i> probabilities $P_{\mathbf{w} \mathbf{x}}(\omega_i x)$ for each of the three-classes.	15
2.2	A diagrammatic view of the classifier and its associated functional mappings.	17
2.3	The <i>a posteriori</i> class differentials $\Delta_{\mathbf{w} \mathbf{x}}(\omega_i x)$ for the three-class random variable x depicted in figure 2.1.	20
2.4	Left: The <i>a posteriori</i> class probabilities $P_{\mathbf{w} \mathbf{x}}(\omega_i x)$ of the three-class random variable depicted in figures 2.1 and 2.3, with a differential form of the Bayesian discriminant function $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$ superimposed. Right: The <i>a posteriori</i> class differentials $\Delta_{\mathbf{w} \mathbf{x}}(\omega_i x)$ of the same three-class random variable, with the discriminant differentials of $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$ superimposed.	24
2.5	A diagrammatic comparison of error measure (EM) and classification figure-of-merit (CFM) objective functions.	28
2.6	A synthetic asymmetric sigmoidal form of the classification figure-of-merit (CFM).	28
2.7	The discriminator output's minimum-error value for the Minkowski- r power metric ($r = 1.25, 2, 9$; binary output target values).	42
3.1	The <i>discriminant bias</i> , <i>discriminant variance</i> , and <i>mean-squared discriminant error</i> (MSDE) of three different classifier paradigms.	61
4.1	A two-class scalar feature discrimination task. The single feature is a homoscedastic, Gaussian-distributed random variable.	86
4.2	The proper parametric model of x . The logistic linear hypothesis class follows from both the partially-parametric and fully-parametric proper models of x	86
4.3	A comparison of error rates for differentially (CFM) and probabilistically (MSE, CE, and ML) generated logistic linear classifiers.	94

4.4	The empirical class-conditional pdfs of x multiplied by their empirical class prior probabilities for a training sample size of 10 examples.	95
4.5	The empirical <i>a posteriori</i> class probabilities of x , shown in histogram form for the 10 examples of figure 4.4. The discriminant functions of the CE-generated logistic linear classifier (i.e., the partially-parametric model of x) are superimposed in black.	96
4.6	The same empirical <i>a posteriori</i> class probabilities shown in figure 4.5. The discriminant functions of the CFM-generated logistic linear classifier are superimposed in black.	96
4.7	A comparison of the approximated mean-squared discriminant error ($\sim$ MSDE) for the differentially (CFM) and probabilistically (MSE, CE, and ML) generated classifiers.	98
4.8	A comparison of the approximated discriminant bias ($\sim$ DBias) for the differentially (CFM) and probabilistically (MSE) generated logistic linear classifiers.	98
4.9	A three-class scalar feature discrimination task. The single feature is a heteroscedastic, uniformly-distributed random variable.	101
4.10	The polynomial classifier of x depicted as a neural network paradigm.	102
4.11	Discriminant functions of probabilistically (MSE) and differentially (CFM) generated polynomial classifiers of x for an asymptotically large training sample size.	107
4.12	Discriminant functions of probabilistically (MSE) and differentially (CFM) generated polynomial classifiers of x for a training sample of size $n = 100$	108
4.13	A comparison of error rates for differentially (CFM) and probabilistically (MSE) generated polynomial classifiers.	110
4.14	A comparison of the approximated mean-squared discriminant error ($\sim$ MSDE) for differentially (CFM) and probabilistically (MSE) generated polynomial classifiers.	110
5.1	Reduced discriminator output space for a hypothetical classifier with C outputs that take on values between zero and one.	119
5.2	An illustration of the discriminant differential and its relationship to reduced discriminator output space.	121
5.3	The MSE objective function is non-monotonic.	127
5.4	The MAE objective function can be monotonic if and only if the number of classes is $C = 2$	130
5.5	The MAE objective function is increasingly non-monotonic as C increases.	131
5.6	The MSE objective function is increasingly non-monotonic as C increases.	135
5.7	The Kullback-Leibler information distance (CE objective function) is non-monotonic.	139
5.8	The CE objective function is increasingly non-monotonic as C increases.	140
5.9	The simple two-class scalar feature discrimination task for which CFM is monotonic if and only if $\psi \lesssim .05$	145

5.10	The CFM generated by an asymptotically large training sample of the two-class random feature x described by (5.76) — (5.78).	146
5.11	Details of the maximum CFM generated by an asymptotically large training sample of the two-class random feature x	146
5.12	Three types of training examples: un-learned examples, transition examples, and learned examples.	149
5.13	The synthetic CFM objective function, given a confidence parameter of $\psi = .05$	153
5.14	Old forms of the CFM objective function, described in [55].	156
7.1	Two of the four features (petal length and width) E. Anderson measured on the Irises of the Gaspe Peninsula [3].	189
7.2	The confusable examples of figure 7.1, plotted as a function of the other two features (sepal length and width).	190
7.3	The 15-parameter differential logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris data with high confidence.	194
7.4	The differentially-generated logistic linear classifier's output state after attempting to learn the Iris data with moderate confidence.	195
7.5	The differentially-generated logistic linear classifier's output state after attempting to learn the Iris data with low confidence.	196
7.6	Left: Histograms of the output states for the classifier in figure 7.4 after 350 learning epochs: $\psi = 0.6$. Right: Histograms of the output states for the same classifier (figure 7.5) after 350 learning epochs: ψ is reduced from 0.6 to 0.1 over the first 200 learning epochs.	197
7.7	The empirical error rates (training sample with all 150 examples) for the 15-parameter logistic linear classifier shown in figure 7.5 as it learns differentially (CFM). The classifier's empirical error rate after 350 learning epochs is 1.3 (+2.5/-1.3)%.	198
7.8	The empirical error rates (training sample with all 150 examples) for the 15-parameter logistic linear classifier as it learns probabilistically (MSE)). The classifier's empirical error rate after 350 learning epochs is 2.0 (+2.9/-2.0)%.	199
7.9	The 15-parameter logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (MSE — see figure 7.8).	200
7.10	The empirical error rates (training sample with all 150 examples) for the 15-parameter logistic linear classifier shown in figure 7.3 as it learns probabilistically (Kullback-Leibler — CE). The classifier's empirical error rate after 350 learning epochs is 2.0 (+2.9/-2.0)%.	201

7.11 The 15-parameter logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (CE — see figure 7.10).	202
7.12 Figure 7.4 shown with a rejection threshold of $\delta_{reject} = 0.35$	203
7.13 The differentially-generated logistic linear classifier's output state after attempting to learn the Iris data with lower confidence (0.26), shown with a rejection threshold of $\delta_{reject} = 0.15$	204
7.14 Figures 7.9 (MSE, top) and 7.11 (CE, bottom) shown with a rejection threshold of $\delta_{reject} = 0.35$	205
7.15 The empirical error rates (training sample with all 150 examples) for the 15-parameter linear classifier as it learns differentially (CFM). The classifier's empirical error rate after 350 learning epochs is 1.3 (+2.5/-1.3)%.	207
7.16 The 15-parameter differentially-generated linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris with low confidence. The classifier cannot learn examples 83 and 133 (see figure 7.2).	208
7.17 The empirical error rates (training sample with all 150 examples) for the 15-parameter linear classifier as it learns probabilistically (MSE)). The classifier's empirical error rate after 350 learning epochs is 14.7 (+6.2/-5.8)%.	209
7.18 The 15-parameter linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (MSE). The classifier cannot learn 22 of the examples.	210
7.19 The empirical error rates (training sample with all 150 examples) for the 15-parameter modified RBF classifier as it learns differentially (CFM). The classifier's empirical error rate after 350 learning epochs is 2.0 (+2.9/-2.0)%.	211
7.20 The 15-parameter differential modified RBF classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris with low confidence. The classifier cannot learn examples 70, 83, and 133 (see figure 7.2).	212
7.21 The empirical error rates (training sample with all 150 examples) for the 15-parameter modified RBF classifier as it learns probabilistically (MSE)). The classifier's empirical error rate after 350 learning epochs is 4.7 (+4.0/-3.4)%.	213
7.22 The 15-parameter modified RBF classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (MSE). The classifier cannot learn 7 of the examples.	214
7.23 The empirical error rates (training sample with all 150 examples) for the 15-parameter modified RBF classifier as it learns probabilistically (Kullback-Leibler — CE)). The classifier's empirical error rate after 350 learning epochs is 6.7 (+4.6/-4.0)%.	215

7.24	The 15-parameter modified RBF classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (Kullback-Leibler — CE). The classifier cannot learn 10 of the examples.	216
8.1	Forty digits randomly chosen from the AT&T DB1 database.	220
8.2	Parameters or <i>weights</i> of the logistic linear classifier after learning the DB1 database's benchmark training sample differentially.	225
8.3	Left: Test sample classification summaries for the 2570-parameter logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). Right: The difference between the probabilistically-generated models' empirical error rates and the differentially-generated model's rate on a trial-by-trial basis.	226
8.4	The distribution of parameter values in the 257-parameter logistic linear discriminant function representing the digit "3" (cf. figure 8.2).	227
8.5	The same digits shown in figure 8.1, linearly compressed from 256- to 64-pixel images. . .	228
8.6	Parameters or <i>weights</i> of the 650-parameter logistic linear classifier after learning the DB1 database's benchmark training sample differentially.	228
8.7	The distribution of parameter values in the 65-parameter logistic linear discriminant function representing the digit "3".	229
8.8	Test sample empirical error rates with 95% confidence intervals for the DB1 database's benchmark split of training/testing examples.	231
8.9	Left: Test sample classification summaries for the 650-parameter logistic linear classifier employing differential and probabilistic learning. Right: The difference between the probabilistically-generated models' empirical error rate and the differentially-generated model's rate on a trial-by-trial basis.	231
8.10	The empirical error rates for the 650-parameter logistic linear classifier as it learns the benchmark training sample differentially.	233
8.11	The 650-parameter logistic linear classifier's output state — as projected onto reduced discriminator output space — after learning the 600 benchmark training examples differentially.	234
8.12	The empirical error rates (training sample in gray and test sample in black) for the 650-parameter logistic linear classifier as it learns the benchmark training sample probabilistically (MSE objective function).	235
8.13	The 650-parameter logistic linear classifier's output state — as projected onto reduced discriminator output space — after it attempts to learn the 600 benchmark training examples probabilistically.	236

8.14	The empirical error rates for the 650-parameter logistic linear classifier as it learns the benchmark training sample probabilistically (CE objective function).	237
8.15	The 650-parameter logistic linear classifier's output state — as projected onto reduced discriminator output space — after it attempts to learn the 600 benchmark training examples probabilistically.	238
8.16	The empirical error rates for the 650-parameter linear classifier as it learns the benchmark training sample differentially.	240
8.17	The 650-parameter linear classifier's output state — as projected onto reduced discriminator output space — after learning the 600 benchmark training examples differentially.	241
8.18	The empirical error rates for the 650-parameter linear classifier as it learns the benchmark training sample probabilistically (MSE objective function).	242
8.19	The 650-parameter linear classifier's output state — as projected onto the reduced discriminator output space — after it attempts to learn the 600 benchmark training examples probabilistically (MSE objective function).	243
8.20	Left: Test sample classification summaries for the 650-parameter linear classifier employing differential learning (Λ_{Δ}) and the MSE form of probabilistic learning (Λ_P). Right: The difference between the probabilistically-generated models' empirical error rate and the differentially-generated model's rate on a trial-by-trial basis.	244
8.21	Left: Test sample classification summaries for the 650-parameter modified RBF classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). Right: The difference between the probabilistically-generated models' empirical error rate and the differentially-generated model's rate on a trial-by-trial basis.	244
8.22	The probability density function for a noisy compressed DB1 image pixel when the probability of pixel inversion is 0.2.	253
8.23	Moderately noisy versions of the digits shown in figure 8.5.	254
8.24	Parameters of the differential logistic linear classifier generated with the first of 25 random DB1 database training/testing splits with moderate noise.	254
8.25	Left: Test sample classification summaries for the 650-parameter logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). Right: The increase in the discriminant error of the two probabilistically-generated models over the differentially-generated model on a trial-by-trial basis.	257

8.26	Left: The test sample classification summaries shown in figure 8.25 for the 650-parameter (64-pixels/digit) logistic linear classifier employing differential and probabilistic learning Right: Classification summaries for fifteen human subjects asked to classify the 40 64-pixel examples shown in figure 8.23. Far Right: Classification summaries for fifteen different human subjects asked to classify the analogous 40 256-pixel examples shown in figure 8.33.	257
8.27	Left: Test sample classification summaries for the 650-parameter linear classifier employing differential learning (Λ_{Δ}) and the MSE form of probabilistic learning (Λ_P). Right: The increase in the discriminant error of the probabilistic model over the differentially-generated model on a trial-by-trial basis.	260
8.28	Very noisy versions of the digits shown in figure 8.5.	262
8.29	Parameters of the differential logistic linear classifier generated with the first of 25 random DB1 database training/testing splits with strong noise.	262
8.30	Left: Test sample classification summaries for the 650-parameter logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). Right: The increase in the discriminant error of the two probabilistically-generated models over the differentially-generated model on a trial-by-trial basis.	265
8.31	Left: The test sample classification summaries shown in figure 8.30 for the 650-parameter (64-pixels/digit) logistic linear classifier employing differential and probabilistic learning. Right: Classification summaries for fifteen human subjects asked to classify the 40 64-pixel examples shown in figure 8.28. Far Right: Classification summaries for fifteen different human subjects asked to classify the analogous 40 256-pixel examples shown in figure 8.34.	265
8.32	Left: Test sample classification summaries for the 650-parameter linear classifier employing differential learning and the MSE form of probabilistic learning. Right: The increase in the discriminant error of the probabilistic model over the differentially-generated model on a trial-by-trial basis.	266
8.33	Moderately noisy 256-pixel digits formed by flipping the binary pixels of the original DB1 database with probability 0.1.	269
8.34	Very noisy 256-pixel digits formed by flipping the binary pixels of the original DB1 database with probability 0.2.	269
8.35	Correct labels for the digits in figures 8.23.	270
8.36	Correct labels for the digits in figures 8.28.	270
8.37	Correct labels for the digits in figures 8.33.	270
8.38	Correct labels for the digits in figures 8.34.	270

9.1	Fourteen 1024-pixel examples of healthy (bottom row) and AVN compromised (top row) femoral heads.	273
9.2	Left: The parameters of a 1024-pixel logistic linear classifier, obtained by differentially learning all 125 example images. Right: A histogram of the weights in the left figure. . . .	273
9.3	The images in figure 9.1, linearly compressed to 256 pixels.	275
9.4	Left: The parameters of a 256-pixel logistic linear classifier, obtained by differentially learning all 125 example images. Right: A histogram of the weights in the left figure. . . .	275
9.5	The 257-parameter differentially-generated logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after learning all 125 AVN examples.	276
9.6	The 257-parameter differentially-generated logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after a typical learning trial for which the training sample size is 58 AVN images selected randomly from the pool of 125 total images.	278
9.7	Left: The differentially-generated logistic linear classifier's sensitivity and specificity for the trial depicted in figure 9.6. Right: The associated receiver operator characteristic for detecting an AVN-compromised femoral head.	279
9.8	The three test images that the differentially-generated logistic linear classifier of figure 9.6 misclassifies.	280
9.9	Left: Test sample classification summaries for the low-complexity logistic linear classifier employing differential learning and two forms of probabilistic learning. Right: The increase in the discriminant error of the two probabilistic models over the differentially-generated model on a trial-by-trial basis.	281
9.10	Test sample classification summaries for the low-complexity (257-parameter) differentially-generated logistic linear classifier (far left), Manduca, Christy, and Ehman's [90] probabilistically-generated linear classifier (middle left), and their best probabilistically-generated non-linear classifier (middle right). Each of ten human subjects performed one 2-fold cross validation trial (far right); the differentially-generated logistic linear classifier compares favorably with the humans.	281
10.1	Top Left: Panchromatic image of the <i>civil1</i> site (1.2 meter resolution). Top Right: Composite of the multi-spectral data for the <i>civil1</i> site (8 meter resolution), which the classifiers interpret.	289
10.2	Top: The DRBF classifier's interpretation of the <i>civil1</i> site. Middle The ground truth for the <i>civil1</i> site. Bottom The maximum-likelihood (ML) classifier's interpretation of the <i>civil1</i> site.	290
10.3	Top Left: Panchromatic image of the <i>gao1</i> site (1.2 meter resolution). Top Right: Composite of the multi-spectral data for the <i>gao1</i> site (8 meter resolution), which the classifiers interpret.	293

10.4	Top: The DRBF classifier's interpretation of the gaol site. Middle The ground truth for the gaol site. Bottom The maximum-likelihood (ML) classifier's interpretation of the gaol site.	294
10.5	Interpretation of the White House site.	299
10.6	Interpretation of the Bureau of Engraving site.	300
11.1	A simplified view of efficient autonomous learning.	308
C.1	A box plot for a sample of the random variable x	322
D.1	Details of the synthetic asymmetric sigmoidal form of the classification figure-of-merit (CFM).	329
D.2	Three types of training examples: un-learned examples, transition examples, and learned examples.	335
D.3	Left: The original logistic sigmoidal form of the CFM objective function for four values of the steepness parameter β (figure adapted from [55]). Right: The function's first derivative with respect to the discriminant differential δ for the same four values of β	337
D.4	The logistic sigmoidal form of the CFM objective function has a first derivative that decreases exponentially with increasing steepness parameter β	339
D.5	The ratio of the differential learning rate for transition examples to that for un-learned examples with a nominal discriminant differential value of $\delta = -.7$	341
D.6	The slope of the synthetic CFM objective function's lower leg, as a function of the confidence parameter ψ	343
D.7	The slope of the synthetic CFM objective function's transition region, as a function of the confidence parameter ψ	343
D.8	Equivalent logistic and synthetic CFM functional forms.	345
D.9	A diagrammatic view of backpropagation with the CFM objective function.	349
E.1	A classifier comprising a single linear discriminant function is equivalent to Rosenblatt's perceptron when generated with this modified form of the CFM objective function.	360
M.1	Left: The parameters of a 1024-pixel differentially-generated logistic linear classifier described in chapter 9. Right: A histogram of the weights in the left figure.	414
M.2	Left: The parameters of the logistic linear classifier shown in figure M.1, generated by differential learning with weight decay. Right: A histogram of the weights in the left figure.	415
M.3	The moving average filter kernel used for weight smoothing.	417

- M.4 Left:** The parameters of the logistic linear classifier shown in figure M.1, generated by differential learning with weight smoothing. **Right:** A histogram of the weights in the left figure. 418
- M.5 Top:** Linear non-invertible compression with a compression ratio of 4:1. **Bottom:** Linear non-invertible compression with a compression ratio of 2.25:1. 419

Chapter 1

Overview

1.1 Abstract

There is more to learning stochastic concepts for robust statistical pattern recognition than the learning itself: computational resources must be allocated and information must be obtained. Therein lies the key to a learning strategy that is efficient, requiring the fewest resources and the least information necessary to produce classifiers that generalize well. Probabilistic learning strategies currently used with connectionist (as well as most traditional) classifiers are often inefficient, requiring high classifier complexity and large training sample sizes to ensure good generalization. An asymptotically efficient *differential learning strategy* is set forth. It guarantees the best generalization allowed by the choice of classifier paradigm as long as the training sample size is large; this guarantee also holds for small training sample sizes when the classifier is an “improper parametric model”¹ of the data (as it often is). Differential learning requires the classifier with the minimum functional complexity necessary — under a broad range of accepted complexity measures — for Bayesian (i.e., minimum probability-of-error) discrimination.

The theory is demonstrated in several real-world machine learning/pattern recognition tasks associated with optical character recognition, medical diagnosis, and airborne remote sensing imagery interpretation. These applications focus on the implementation of differential learning and illustrate its advantages and limitations in a series of experiments that complement the theory. The experiments demonstrate that differentially-generated classifiers consistently generalize better than their probabilistically-generated counterparts across a wide range of real-world learning-and-classification tasks. The discrimination improvements range from moderate to significant, depending on the statistical nature of the learning task and its relationship to the functional basis of the classifier used.

¹ See definition 3.14.

1.2 Intended Audience

The material in this text is intended for researchers in the areas of statistical pattern recognition and machine learning. At the very least this includes researchers from the fields of statistics, electrical and computer engineering, and computer science. Within each of these broad fields there are disciplines that have generated their own culture and technical jargon: the terminology of one culture does not always match that of another, so there are inherent problems associated with any attempt to reach a wide audience with a single message. Nevertheless, we shall try.

To this end we combine elements and notation of estimation theory, statistical pattern recognition, information theory, and computational learning theory; we exploit the more expressive aspects of each discipline in order to articulate our message clearly. We therefore employ a mixture of the notational conventions of [15, 45, 29, 117, 100], among others; appendix A provides a glossary of notation.

Although we have endeavored to make the material accessible to a broad audience, the text assumes that the reader has a basic understanding of probability and statistics and a familiarity with the terminology commonly used in the pattern recognition literature (e.g., [29]). This terminology is sometimes at odds with that of other disciplines. The most notable example is the word *class*: in real analysis, measure theory, and computational learning theory, the term is synonymous with “set”; in the pattern recognition literature, the term is synonymous with “concept”. We generally mean “concept” (not “set”) when we use the term *class*. At the same time, we use the term *hypothesis class* — computational learning theory jargon — when referring to the *set* of all possible classifiers that might be generated by a learning strategy. Computer scientists will note that we use the term “search” to denote “numerical optimization procedure”. Strictly speaking, a numerical optimization procedure is a specific type of search (i.e., one that takes place over a continuous, differentiable function on an uncountable space of independent variables); thus, our terminology is somewhat imprecise. We have kept these kinds of inconsistencies to a minimum, committing them only when we feel economy of words and/or the historical precedence of a particular research field warrants.

1.3 Outline of the Text

Up through the first half of this century, statistical pattern recognition was generally done with so-called “parametric” models. The parametric model assumes that the feature vector (or attribute vector) has a particular probabilistic form, so the process of learning is simply one of choosing the set of parameters that maximizes the likelihood that the observed data could have been generated by the model; logistic regression is arguably the best-known example. Parametric models are generally simple paradigms that lend themselves to detailed analysis. Their simplicity is appealing in terms of the analytical tractability it affords, but it enforces a restrictive probabilistic view of the world that is not always valid.

The computer age has brought us the power to explore less restrictive “non-parametric” models, which

make no *explicit* assumptions regarding the underlying probabilistic nature of the world. Parzen windows, decision trees, and neural network classifiers are three popular examples of non-parametric models. The adjective “non-parametric” is ironic (and in our opinion unfortunate) because it falsely implies that such models have no parameters. Of course *all* models have parameters to the extent that they encode a definition of the classifier in some tangible form. This leads us to view all models as parametric and, furthermore, to make a clear distinction between *proper* and *improper* parametric models.² A proper parametric model has a specific probabilistic form; when parameterized correctly, it is an exact expression of the feature vector’s probabilistic nature. An improper parametric model is *not* a valid expression of the feature vector’s probabilistic nature, whether or not such an expression exists.

This text is motivated by three convictions. The first is that it is not necessary to estimate probabilities in order to perform robust statistical pattern recognition. The second conviction is that many real-world pattern recognition tasks do not have a proper parametric model; among those that do, the proper parametric model may not be readily discernible. The third conviction is expressed in Occam’s razor, the celebrated folk theorem³ that asserts, “the simplest model of the data is the best one.” Ultimately then, the task of learning to perform robust statistical pattern recognition becomes a process of making the most of the least: we strive to generate the best classifier we can from the simplest model that will do. Differential learning is a theoretically defensible means of achieving this goal consistently — an assertion we support with proofs and illustrative experiments.

1.3.1 Summary of Findings

All of our findings follow from two premises:

Probabilistic versus discriminative learning — There are at least two approaches to learning stochastic concepts for statistical pattern recognition: probabilistic and discriminative. *Probabilistic learning* strategies seek to learn the *a posteriori* class probabilities of the feature vector over its domain, whereas *discriminative learning* strategies seek only to learn the identity of the most likely class at each point on the feature vector’s domain (equivalently, discriminative learning strategies seek only to learn the Bayes-optimal class boundaries on the feature vector’s domain). Both of these strategies can be employed with *differentiable supervised classifiers*, which form their input-to-output mappings by adjusting a set of internal parameters via an iterative search aimed at optimizing a differentiable objective function (or empirical risk measure). The objective function is a metric that evaluates how well the classifier’s evolving mapping reflects the empirical relationship between the input patterns of the training sample and their class membership, modeled by the classifier’s discriminant functions. Error measure objective functions and the classification figure-of-merit

²The term “proper model” probably originates with Dawes [26]; see section 3.6.2.

³Occam’s razor is formalized in the notion of universal probability (e.g., [21, pg. 160]) and in VC theory [137, 136].

(CFM objective function) [55] — both described in chapter 2 — induce different kinds of learning. Error measures induce probabilistic learning, whereas the CFM objective function induces *differential learning*, a form of discriminative learning appropriate for the differentiable supervised classifier.

The efficient classifier and efficient learning — We provide rigorous estimation-theoretic definitions of the *efficient classifier*: in simple terms, it consistently exhibits the lowest error rate possible for a given learning/classification task; no other classifier generalizes better. The *relatively efficient classifier* is analogous to the efficient classifier with one qualification: the relatively efficient classifier generalizes better than any other classifier drawn from a limited set of possibilities. We refer to this limited set of possibilities as the classifier's *hypothesis class* (e.g., [100]). The differentiable supervised classifier can be viewed as a Bayesian learning paradigm because its discriminator's initial (or prior) parameterization is transformed to a posterior parameterization during learning. Given a particular training sample size, a particular choice of discriminant functions (i.e., *hypothesis class*), and a particular initial parameterization, the transformation depends entirely on the learning strategy employed. An *efficient learning strategy* generates the relatively efficient classifier described above for both small and large training sample sizes. An *asymptotically efficient learning strategy* requires large training sample sizes to guarantee the relatively efficient classifier.

Principal Theoretical Findings

We prove the following:

- Classifiers that learn by minimizing error measure objective functions (e.g., mean-squared error, the Kullback-Leibler information distance — a.k.a. “cross entropy” — [82, 81], etc.) learn probabilistically. Again, the classifier that learns probabilistically attempts to learn the *a posteriori* probabilities of the feature vector over its domain.
- Learning probabilistically by minimizing error measure objective functions rarely generates the relatively efficient classifier. As a result, probabilistic learning is usually inefficient.
- Classifiers that learn by maximizing the CFM objective function learn differentially. Again, the classifier that learns differentially attempts to learn only the most likely class of the feature vector over its domain.
- Learning differentially by maximizing the synthetic CFM objective function described herein always generates the relatively efficient classifier for large training sample sizes. As a result, differential learning is asymptotically efficient.
- Learning differentially by maximizing the synthetic CFM objective function usually generates the relatively efficient classifier for small training sample sizes as well. As a result, differential learning is usually efficient.

- Learning differentially via the CFM objective function requires discriminant functions with the least functional complexity (e.g., the fewest parameters) necessary for Bayesian (i.e., minimum probability-of-error) discrimination.
- Information-theoretic analysis proves that the training sample sizes necessary to guarantee a specified level of generalization via differential learning are typically orders of magnitude smaller than those necessary to estimate probabilities with a specified level of precision. This indicates that current probabilistic extensions of the PAC learning paradigm [133] to stochastic concepts on uncountable feature vector domains (e.g., [59, 60, 146]) are likely to over-estimate the training sample sizes necessary for good generalization when the learning objective is merely pattern classification.

Experimental Findings

We apply differential learning to several real-world machine learning/pattern recognition tasks associated with optical character recognition, medical diagnosis, and airborne remote sensing imagery interpretation. In each task the differentially generated classifier generalizes better than its probabilistically generated counterpart. The discrimination improvements range from moderate to significant, depending on the statistical nature of the learning task and its relationship to the functional basis of the classifier used. In general, differential learning exhibits the following characteristics:

- Differential learning allows classifiers with $1/2 - 1/10$ the number of parameters used in the best independently-developed models for each task.
- The error rates of differentially-generated classifiers are 20% — 50% less than those of the best independently-developed models.
- The error rates of differentially-generated classifiers are 30% — 80% less than those of probabilistically-generated control models.
- Differentially-generated classifiers are between two and ten times more efficient than their probabilistically-generated counterparts.

1.3.2 Profile of the Chapters

The preceding findings are organized in three basic parts: theoretical findings are described in part I; experimental findings are described in part II; supporting material is detailed in the appendices. We profile the contents of parts I and II in the following sections.

Part I: Theory

Chapter 2: We define the differentiable supervised classifier in terms of its discriminator, the set of discriminant functions that map the feature vector to a set of possible classifications. We discuss the Bayes-optimal classifier: we refer to its discriminator as the Bayesian discriminant function (BDF). We show that the BDF has two fundamental forms that correspond to two fundamental approaches to learning: probabilistic learning seeks to learn the *a posteriori* class probabilities of the feature vector over its domain; differential learning merely seeks to learn the identity of the most likely class over the feature vector's domain. We prove that both strategies generate the Bayes-optimal classifier, given a sufficiently large training sample size and a classifier with sufficient functional complexity to learn the appropriate form of the BDF.⁴

Chapter 3: We characterize the classifier as an estimator of the Bayes-optimal classifier. We define the efficient classifier in terms of three metrics, discriminant bias, discriminant variance, and mean-squared discriminant error (MSDE). These metrics are shown to be quite different from the *functional* bias, variance, and mean-squared error metrics that are commonly discussed in the literature. The efficient classifier generalizes better than any other classifier, exhibiting the lowest possible MSDE for a given training sample size. The relatively efficient classifier is analogous, with one caveat: the relatively efficient classifier generalizes better than any other classifier drawn from a limited set of possibilities (i.e., the hypothesis class). The relatively efficient classifier is identically the efficient classifier if the latter is contained in the hypothesis class; otherwise, the relatively efficient classifier is the best approximation to the efficient classifier allowed by the hypothesis class. An efficient learning strategy always generates the relatively efficient classifier, regardless of the training sample size. An asymptotically efficient learning strategy always generates the relatively efficient classifier, but requires large training sample sizes to do so. We prove that differential learning is asymptotically efficient and that probabilistic learning is inefficient. We then prove that differential learning generates the Bayes-optimal classifier from the hypothesis class with the minimum functional complexity necessary for the task. Probabilistic learning generally requires a hypothesis class with greater functional complexity to generate the Bayes-optimal classifier. We conclude the chapter by outlining the special conditions under which probabilistic learning is more efficient than differential learning. These conditions can exist only if the hypothesis class constitutes a proper parametric model of the feature vector.

Chapter 4: We illustrate the proofs of chapter 3 with two simple learning/classification tasks that lend themselves to closed-form analysis. The first illustration involves a proper parametric model; it shows the special circumstances under which probabilistic learning generates a more efficient classifier than differential learning does when the training sample size is small. The second illustration shows the typical learning/classification scenario wherein the hypothesis class constitutes an improper parametric model. In

⁴We address the issue of classifier complexity in chapter 3.

this circumstance, differential learning generates the relatively efficient classifier for both small and large training sample sizes, whereas probabilistic learning generally fails to do so for any training sample size.

Chapter 5: We discuss the CFM objective function in terms of its monotonicity and the convergence properties it engenders in the differential learning strategy. Monotonicity proves to be an essential characteristic of any objective function associated with an efficient learning strategy. In describing this property we invoke a differential view of the discriminator's output state (i.e., the state of the classifier's discriminant functions), which we employ frequently in the experiments of part II. The view is a 2-dimensional representation that is consistent with the differential form of the BDF described in chapter 2. It leads to a simple taxonomy of training examples and an equally simple geometric explanation of the difference between differential and probabilistic learning strategies. We conclude the chapter by proving that differential learning via the synthetic CFM objective function is reasonably fast. That is, the search for parameters that maximize the synthetic CFM objective function converges in reasonable time.

Chapter 6: We make a clear distinction between the probabilistic information content and the discriminant information content of a randomly-selected training sample. We show that a simple unfair (or "rigged") game of dice forms the basis of all learning/statistical pattern recognition tasks. We analyze this game in order to prove that a random sample's discriminant information content is always at least as great as its probabilistic information content. The information-theoretic argument relies on Rissanen's notion of stochastic complexity (e.g., [115]) and can be viewed as an extension of the chapter 3 proof that differential learning requires the least functional complexity necessary to learn the Bayes-optimal classifier. We derive tight distribution-dependent bounds on the training sample size and (information-theoretic) functional complexity requirements of the differential and probabilistic learning strategies. We show that the rigged game of dice extends naturally to pattern recognition tasks for which the feature vector exists on a finite countable domain. A further extension of the paradigm brings us to the general case in which the feature vector exists on a potentially infinite uncountable domain. We conclude by discussing the limitations of the rigged-die paradigm when it is generalized to the uncountable feature vector space.

Part II: Applications

Chapter 7: We discuss the pragmatic issues that must be addressed in order to implement differential learning successfully. We do this with the aid of the celebrated Iris data, collected by E. Anderson [3] and subsequently used by R. A. Fisher in his seminal paper on linear discriminants [34]. Differential learning allows a natural partitioning of the training sample into three sub-sets: un-learned examples, learned examples, and transition examples. The first two categories are self-explanatory; the third category comprises those examples that are neither un-learned nor learned, but are "in between" those two states. We describe

the nature of confidence as it pertains to differential learning. We show that the differential paradigm focuses on un-learned examples without dwelling on learned examples. We contrast differential learning with two forms of probabilistic learning associated with the mean-squared error (MSE) objective function and the Kullback-Leibler information distance [82, 81]. We view the impact of differential learning on the classifier's ability to detect and reject specious classifications. We conclude by demonstrating the efficiency of differential learning by learning/classifying the Iris with three substantially different hypothesis classes: the differentially-generated classifiers consistently generalize better than their probabilistic counterparts, as predicted by the proofs of part I.

Chapter 8: We expand upon the findings of chapter 7 with an optical character recognition (OCR) task involving the AT&T DB1 handwritten digit database. The database has been studied extensively by other researchers, so it provides a good benchmark for evaluating differential learning. We begin with a description of the controlled experimental protocols we use throughout the text when comparing differential learning with probabilistic learning. We show that classifier's generated differentially from three substantially different hypothesis classes generalize better than their probabilistically-generated counterparts; the disparity increases significantly when the OCR images are compressed in order to reduce the classifiers' complexities. The differentially-generated classifiers' error rates are typically 2% to 4%; their probabilistically-generated counterparts' error rates range from 3% to 12%. By adding noise to the OCR images prior to compression, we induce conditions under which one particular choice of hypothesis class approximates the proper parametric model of the noisy digits. As predicted by chapter 3, the classifier generated probabilistically from this proper parametric model generalizes better than its differentially-generated counterpart for small training samples of very noisy digits.

Chapter 9: We repeat the experiments of Manduca, Christy, and Ehman [90] in which avascular necrosis (AVN) of the femoral head (a debilitating hip joint disorder) is diagnosed from magnetic resonance images (MRIs) using neural network classifiers. We compare the diagnostic accuracy of a simple differentially-generated classifier and two probabilistically-generated controls (including the logistic regression model) with their original results. When presented with approximately sixty training images and subsequently evaluated on the same number of test images, the differentially-generated classifier discriminates between healthy and AVN compromised femoral heads with a 5.9% error rate. This error rate is slightly lower than the 7.5% error rate of humans without formal training in radiology, reported in [90]. The differentially-generated logistic linear classifier generalizes better than the probabilistic controls and the best previous neural network classifier, a multi-layer perceptron having approximately 24 times the number of parameters (6,164, versus 257 for our classifier) [90].

Chapter 10: We describe a series of remote sensing experiments conducted in collaboration with the Digital Mapping Laboratory, School of Computer Science, Carnegie Mellon University. We use a modified RBF classifier employing differential learning (DRBF) to interpret multi-spectral imagery from the Daedalus airborne remote sensing system. The interpretation procedure involves classifying individual image pixels, which represent 64 square meters of earth surface material, into eleven categories of natural and man-made materials — a preliminary step in automated map generation and various environmental analysis tasks. The DRBF classifier has 132 parameters and exhibits a 29% error rate on the interpretation task. The maximum-likelihood (probabilistic) model currently used for this task has 847 parameters and exhibits a 46% error rate.

Chapter 11: We state our contributions to the fields of machine learning and statistical pattern recognition. We then discuss the philosophical implications of our research. We conclude with an outline of future research that follows naturally from what we have accomplished to date.

Appendices

Most of the appendices are explanations of issues that are not essential to the main text, but a few appendices contain essential material worth mentioning here. Appendix A provides a glossary of notation used throughout the text. Terminology is explained throughout the text. The reader can find references to terms via the index; boldface page numbers indicate the page on which a term is defined or explained most thoroughly. Appendix D provides details of the synthetic CFM objective function, including ANSI C source code for the function and its first two derivatives. This appendix also contains a tutorial explanation of how the backpropagation algorithm [119, 120] can be modified for use with CFM. Appendix E explores the similarities and differences between differential learning via the CFM objective function and learning via Rosenblatt's perceptron criterion function [116]. The reader familiar with learning via the perceptron criterion function might find this material a helpful introduction to differential learning via the CFM objective function since the latter can be viewed as a generalization of the former.

Part I

Theory

Chapter 2

Probabilistic and Differential Strategies for Learning the Bayesian Discriminant Function¹

Outline

We describe two learning strategies by which a broad category of pattern classifiers (including, but not limited to, multi-layer perceptron and radial basis function neural networks) can learn to perform Bayesian discrimination (i.e., minimum-error statistical pattern recognition). The *probabilistic learning strategy* is associated with error measure objective functions such as mean squared error and the Kullback-Leibler information distance; it engenders Bayesian discrimination by estimating probabilities. The *differential learning strategy* is associated with classification figure-of-merit objective functions [55]; because it is a discriminative strategy, it engenders Bayesian discrimination without estimating probabilities directly. We describe each strategy in detail as a preliminary step in proving that differential learning is efficient, whereas probabilistic learning is not.

2.1 Introduction

This chapter describes two supervised strategies for learning stochastic concepts in order to perform statistical pattern recognition. Each of these strategies can be applied to any computational model (hereafter called the *classifier*) that forms an input-to-output mapping by adjusting a set of internal parameters via an iterative search aimed at optimizing an objective function (or empirical risk measure). The objective function is a metric that evaluates how well the classifier's evolving mapping reflects the empirical relationship between the input patterns of the training sample and their class membership, modeled by the classifier's outputs. Optimizing the objective function via iterative search on the classifier's parameter space is therefore a

¹This chapter is a revised and extended version of work first published in [54].

mathematically defensible approach to machine learning.

Our principal objective in this chapter is to describe the specific nature of two supervised learning strategies — *probabilistic learning* and *differential learning* — that lead to minimum-error Bayesian discrimination when applied to classifiers of the form described above. Because many neural networks learn in a supervised fashion, it is our hope that the following proofs will be of general interest to the connectionist community. Our presentation begins with an overview of Bayesian discrimination, which provides the framework upon which the main proofs are built. Our secondary objective is to show that these proofs apply to a broad spectrum of machine learning paradigms — not all of which are typically associated with statistical pattern recognition. We do this by introducing them within an historical context that views connectionist models as natural extensions of more traditional classifier paradigms.

In this chapter and all that follow, we combine the elements and notation of statistical pattern recognition and computational learning theory in order to exploit the more expressive aspects of each discipline and present a succinct set of proofs. We employ a mixture of the notational conventions of [45, 29, 117, 100]; appendix A provides a glossary of notation.

We define and contrast probabilistic and differential learning in terms of the functional forms of the Bayesian discriminant function (e.g., [29, sec. 2.5]) they generate. Simply stated, probabilistic learning yields classifiers that estimate the class probabilities for a given input pattern, whereas differential learning yields classifiers that merely identify the most likely class for a given input pattern. Each learning strategy is associated with a family of objective functions. Proofs of varying generality and rigor linking specific objective functions to what we call probabilistic learning are not new. Many authors have shown this linkage for the mean-squared-error (MSE) objective function² [103, 29, 7, 142, 86, 42, 118, 17, 138, 125, 70], while others have shown it for the Kullback-Leibler information distance (a.k.a. “cross entropy”: CE) and/or closely related error measures [82, 81, 67, 11, 127, 142, 64, 42, 31]. Simulations demonstrating the validity of those proofs for neural network classifiers can be found in [111]. We prove that both of these objective functions belong to a broad family of error (or distance) measures that engender probabilistic learning. We then prove that classification figure-of-merit (CFM) objective functions [55] engender differential learning.

2.2 Bayesian Discrimination

Consider a random vector (RV) $\mathbf{X}$, which exists on the domain of real-valued N -dimensional vectors: $\mathbf{X} \in \mathcal{X} = \mathbb{R}^N$. $\mathbf{X}$ is the *feature vector* (or *attribute vector*), which can represent any one of C classes (or concepts)³ on feature vector space $\mathcal{X}$. Feature vector space $\mathcal{X}$ is paired with a *class label* (or *classification*) space $\Omega = \{\omega_1, \dots, \omega_C\}$. We denote the i th class of $\mathbf{X}$ by ω_i . The feature vector $\mathbf{X}$ is always

²The MSE objective function is also called the least-mean-squared (LMS) and the least-squared error (LSE) objective function.

³This uncountably infinite definition of feature vector space includes more restrictive definitions, such as the set of all integer-valued N -dimensional vectors $\mathbb{Z}^N$ and the set of all N -tuples $\{0, 1\}^N$. The following proofs therefore apply to more restrictive definitions of the feature vector, involving finite and/or countable spaces, without loss of generality.

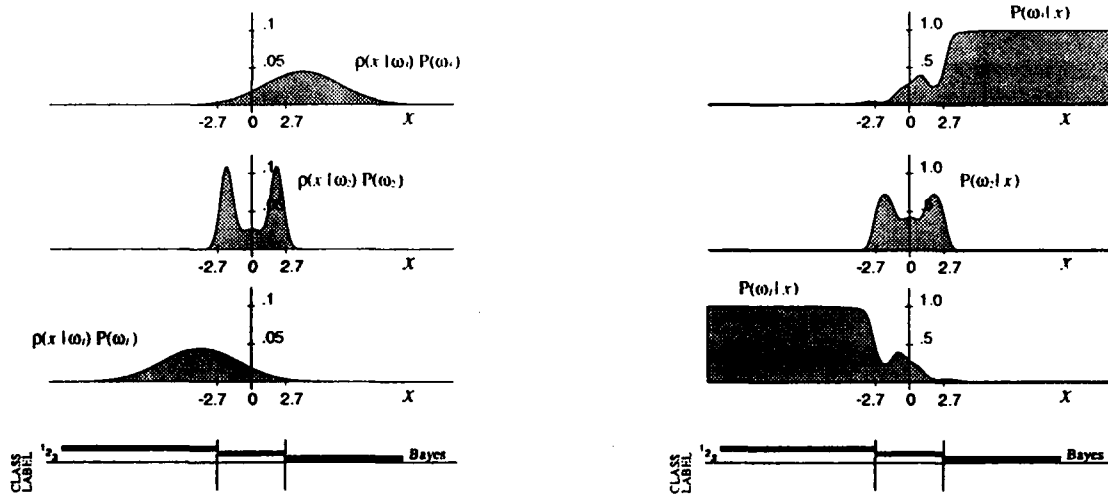


Figure 2.1: **Left:** The class-conditional density — class prior probability products $\rho_{X|W}(x|\omega_i) \cdot P_W(\omega_i)$ for a three-class random scalar x . **Right:** The associated *a posteriori* probabilities $P_{W|X}(\omega_i|x)$ for each of the three-classes, plotted over the effective domain of x . These constitute the strictly probabilistic form of the Bayesian discriminant function for x (see definition 2.3).

paired with a *class label* $W \in \Omega$; we denote the pairing by $\langle X, W \rangle$. Because the classes defined on $\mathcal{X}$ are stochastic, W for a given X is not deterministic; rather it is a random variable, examples of which are generated according to the conditional distribution $P_{W|X}(W|X)$ over Ω . Thus, the probability that an example of X constitutes an example of the i th class ω_i is $P_{W|X}(\omega_i|X)$. We refer to $P_{W|X}(\omega_i|X)$ as the, “*a posteriori* probability of the i th class (given X).”

There exists some means of obtaining examples of X , which are generated according to the probability density function (pdf) $\rho_X(X)$ over $\mathcal{X}$. Examples of X representing the i th class are generated according to the class-conditional pdf $\rho_{X|W}(X|\omega_i)$ over $\mathcal{X}$ such that

$$\rho_X(X) = \sum_{j=1}^c \rho_{X|W}(X|\omega_j) \cdot P_W(\omega_j) \quad (2.1)$$

The “prior” distribution of classes on Ω , denoted by $P_W(W)$ in (2.1), is obtained by integrating the joint pdf $\rho_{X,W}(X, W)$, which is over the joint space $\mathcal{X} \times \Omega$, over $\mathcal{X}$:

$$P_W(W) = \int_{\mathcal{X}} \rho_{X,W}(X, W) dX \quad (2.2)$$

By Bayes’ rule, the *a posteriori* probability of the i th class is

$$P_{W|X}(\omega_i|X) = \frac{\rho_{X,W}(X, \omega_i)}{\rho_X(X)}$$

$$= \frac{\rho_{\mathbf{X}|\mathbf{W}}(\mathbf{X}|\omega_i) \cdot P_{\mathbf{W}}(\omega_i)}{\sum_{j=1}^C \rho_{\mathbf{X}|\mathbf{W}}(\mathbf{X}|\omega_j) \cdot P_{\mathbf{W}}(\omega_j)} \quad (2.3)$$

Example 2.1 Figure 2.1 provides a concrete example of a simple three-class pattern recognition task described by the mathematical formalism above. In this figure the feature vector is actually a scalar, so feature vector space is the real number line: $x \in \mathcal{X} = \mathbb{R}$. This random variable can represent one of three classes on classification space: $\mathcal{W} \in \Omega = \{\omega_1, \omega_2, \omega_3\}$. The left-hand side of the figure shows the class-conditional pdf's of x multiplied by the class prior probabilities ($\rho_{\mathbf{X}|\mathbf{W}}(x|\omega_i) \cdot P_{\mathbf{W}}(\omega_i)$; $i = 1, 2, 3$) in order of increasing class index, from bottom left to top right. The right-hand side of the figure shows the corresponding *a posteriori* class probabilities ($P_{\mathbf{W}|\mathbf{X}}(\omega_i|x)$; $i = 1, 2, 3$) over the effective domain of x . The bar-graph display at the bottom of the left and right figures is described in example 2.3.

2.2.1 The Classifier and the Bayesian Discriminant Function

Pattern recognition, discrimination, or classification is the process by which the classifier associates a class (or concept) label with each example of the feature vector presented to it. For this reason, the classifier implements a set of C deterministic functional mappings (known as the *discriminant functions*) from feature vector space $\mathcal{X}$ to *discriminator output space*⁴ $\mathcal{Y}$. This set of discriminant functions $\mathcal{G}(\mathbf{X}|\theta)$ is known collectively as the *discriminator*. The *discriminator output* $\mathbf{Y} = \langle y_1, \dots, y_C \rangle$ exists on the domain of real-valued C -dimensional vectors⁵ ($\mathbf{Y} \in \mathcal{Y} = \mathbb{R}^C$) such that

$$\begin{aligned} \mathcal{G} : \mathbf{X} &\rightarrow \mathbf{Y}; \\ \mathcal{G}(\mathbf{X}|\theta) &\triangleq \{g_1(\mathbf{X}|\theta), \dots, g_C(\mathbf{X}|\theta)\}, \end{aligned} \quad (2.4)$$

where

$$g_i : \mathbf{X} \rightarrow y_i \quad (y_i \in \mathbb{R}) \quad \forall i \quad (2.5)$$

The argument θ in $\mathcal{G}(\mathbf{X}|\theta)$ and $g_i(\mathbf{X}|\theta)$ indicates that the discriminant functional mappings depend on the *parameterization* θ of the discriminator.

⁴The following definition of a classifier and its associated functional mappings assumes a one-to-one correspondence between the number of discriminator outputs and the number of classes C . The proofs that follow rely on this conventional (rather than necessary) assumption. Other assumptions are equally valid. As an example, a classifier with $\lceil \log_2[C] \rceil$ discriminator outputs can recognize C classes. In such a case the following proofs will hold, given appropriate modifications to account for the altered discriminator output space $\mathcal{Y}$.

⁵As with feature vector space, this uncountably infinite definition of discriminator output space includes more restrictive definitions involving countable and/or finite spaces. Examples of such spaces are the finite uncountable space $\mathcal{Y} = [0, 1]^C$ associated with multi-layer perceptrons having output nodes with logistic (i.e., differentiable sigmoidal) non-linearities, and the finite countable space $\mathcal{Y} = \{0, 1\}^C$ associated with decision trees.

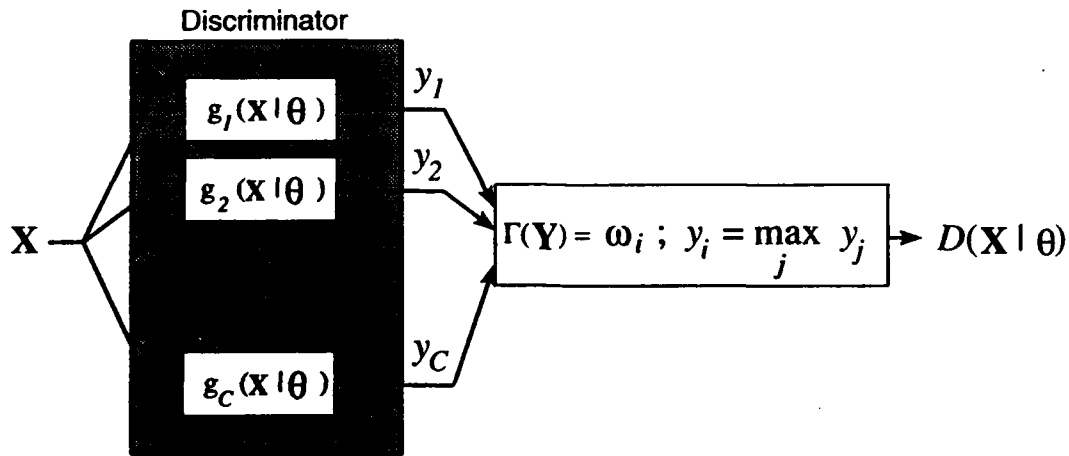


Figure 2.2: A diagrammatic view of the classifier and its associated functional mappings. The classifier input is a feature vector $\mathbf{X}$; the C discriminator outputs $y_1, \dots, y_C$ correspond to the classes that $\mathbf{X}$ can represent; the class label $\mathcal{D}(\mathbf{X}|\theta)$ assigned to the input feature vector corresponds to the discriminator's largest output. Figure based on figure 2.3 of Duda & Hart [29].

The final classification $\mathcal{D}(\mathbf{X}|\theta)$ of $\mathbf{X}$ is obtained by a mapping Γ from discriminator output space $\mathcal{Y}$ to classification space Ω . This mapping associates $\mathbf{X}$ with the class label corresponding to the discriminator's largest output.⁶ If two or more discriminator outputs are equal and larger than the rest, the mapping yields a *set* of possible classifications, corresponding to all of the top outputs:

$$\Gamma : \mathcal{Y} \rightarrow \mathcal{W}$$

$$\Gamma(\mathbf{Y}) = \begin{cases} \omega_i : y_i = \max_j y_j, & y_k < y_i \ \forall k \neq i \\ \{\omega_i : y_i = \max_j y_j\}, & \text{otherwise} \end{cases} \quad (2.6)$$

$$\text{s.t. } \mathcal{D}(\mathbf{X}|\theta) = \Gamma(\mathbf{Y}) = \Gamma(\mathcal{G}(\mathbf{X}|\theta)) : \mathcal{D}(\mathbf{X}|\theta) \in \Omega = \{\omega_1, \dots, \omega_C\} \quad (2.7)$$

In the words of Duda and Hart [29, sec. 2.5.1], "the classifier is viewed as a machine that computes C discriminant functions and selects the [class] corresponding to the largest discriminant." Figure 2.2 is based on figure 2.3 of [29], and illustrates this mathematical notion of the classifier.

Definition 2.1 The (minimum-error) Bayes-optimal classifier: *It is straightforward to prove that the classifier $\mathcal{D}(\mathbf{X})$ that minimizes the probability of an incorrect classification of $\mathbf{X}$ is the one that always maps the feature vector to its most probable class (e.g., [29, pp. 16-20]):*

⁶The following proofs are necessarily linked with this method of choosing the class label for an example. For classifiers that do not have a one-to-one correspondence between their number of discriminator outputs and the number of classes, the discriminator output state representing the classification of the example must be maximal by a measure that is appropriate for the discriminator.

$$\mathcal{D}(\mathbf{X})_{\text{Bayes}} \triangleq \omega_* : P_{\mathcal{W}|\mathbf{X}}(\omega_*|\mathbf{X}) \geq P_{\mathcal{W}|\mathbf{X}}(\omega_k|\mathbf{X}) \quad \forall \omega_k \neq \omega_* \quad (2.8)$$

Any classifier satisfying (2.8) is known as a Bayes-optimal classifier, which is said to yield (minimum-error) Bayesian discrimination.

Remark: Equation (2.8) describes a *unique* mapping from $\mathbf{X}$ to $\mathcal{W}$. However, given our definition of the classifier in (2.6) – (2.7), there are infinitely many discriminators $\mathcal{G}(\mathbf{X}|\theta)$ that implement the Bayes-optimal classifier. Indeed, as long as the discriminant function $g_*(\mathbf{X}|\theta)$ associated with the largest *a posteriori* class probability $P_{\mathcal{W}|\mathbf{X}}(\omega_*|\mathbf{X})$ in (2.8) is always largest, the classifier yields Bayesian discrimination.

Definition 2.2 The Bayesian discriminant function (BDF): *In mathematically formal terms, the discriminator $\mathcal{G}(\mathbf{X}|\theta)$ constitutes the Bayesian discriminant function — and the classifier $\mathcal{D}(\mathbf{X}|\theta)$ yields Bayesian discrimination — if the classifier's largest output always corresponds to the most likely class, given $\mathbf{X}$:*

$$\begin{aligned} \mathcal{D}(\mathbf{X}|\theta) = \Gamma(\mathbf{Y}) = \Gamma(\mathcal{G}(\mathbf{X}|\theta)) &\equiv \mathcal{D}(\mathbf{X})_{\text{Bayes}} \text{ iff} \\ g_i(\mathbf{X}|\theta) &= y_i; \\ \left\{ \begin{array}{l} y_i > y_j : P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X}) > P_{\mathcal{W}|\mathbf{X}}(\omega_j|\mathbf{X}) \\ y_i = y_j : P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X}) = P_{\mathcal{W}|\mathbf{X}}(\omega_j|\mathbf{X}) \\ y_i < \max_j y_j, \end{array} \right\}, & \left\{ \begin{array}{l} P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X}) = \max_k P_{\mathcal{W}|\mathbf{X}}(\omega_k|\mathbf{X}) \\ P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X}) < \max_j P_{\mathcal{W}|\mathbf{X}}(\omega_j|\mathbf{X}) \end{array} \right\} \quad (2.9) \\ \forall i, \quad \forall \mathbf{X} \in \mathcal{X} \end{aligned}$$

Remark: The BDF is of course a *set* of functions rather than a single function, as the name suggests. Because there are infinitely many discriminators that satisfy (2.9), there are infinitely many discriminant functions that implement the Bayes-optimal classifier of definition 2.1.

2.2.2 Probabilistic and Differential forms of the Bayesian Discriminant Function

We group all Bayesian discriminant functions into four categories. Note that the following definitions consider *all* possible forms of the BDF — not just those allowed by our choice of discriminator $\mathcal{G}(\mathbf{X}|\theta)$ (we discuss the difference at length in chapter 3). We denote the arbitrary BDF by $\mathcal{F}(\mathbf{X})_{\text{Bayes}}$ and the set of all such BDFs by $\mathbf{F}_{\text{Bayes}}$.

Definition 2.3 The strictly probabilistic form of the BDF: This form of the BDF is given by

$$\begin{aligned}\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}} &= \{f_1(\mathbf{X}), \dots, f_c(\mathbf{X})\}; \\ f_i(\mathbf{X}) &= P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}) \quad \forall i\end{aligned}\tag{2.10}$$

Remark: There is only one $\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}}$, which is uniquely specified by the *a posteriori* class probabilities of $\mathbf{X}$.

Definition 2.4 The probabilistic form of the BDF: This form of the BDF is given by

$$\begin{aligned}\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}} &= \{f_1(\mathbf{X}), \dots, f_c(\mathbf{X})\}; \\ f_i(\mathbf{X}) &= \varphi(P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X})) \quad \forall i\end{aligned}\tag{2.11}$$

where $\varphi(\cdot)$ is a strictly increasing function of its argument:

$$\frac{d}{dz} \varphi(z) > 0, \quad 0 \leq z \leq 1\tag{2.12}$$

We denote the set of all probabilistic forms of the BDF by $\mathbf{F}_{\text{Bayes-Probabilistic}}$ (i.e., every $\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}}$ is a member of $\mathbf{F}_{\text{Bayes-Probabilistic}}$).

Remark: There are innumerable probabilistic forms of the BDF, since there are innumerable strictly increasing functions $\varphi(z)$.

Example 2.2 Consider the following four probabilistic forms of the BDF:

- $\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}} = \mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}} = \{f_i(\mathbf{X}) = P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}); \quad i = 1, \dots, c\}$
- $\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}} = \{f_i(\mathbf{X}) = 3 \cdot P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}) + 1; \quad i = 1, \dots, c\}$
- $\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}} = \{f_i(\mathbf{X}) = \log(P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X})); \quad i = 1, \dots, c\}$
- $\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}} = \{f_i(\mathbf{X}) = (P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}))^2; \quad i = 1, \dots, c\}$

All of these discriminant functions preserve the rankings of the *a posteriori* class probabilities of $\mathbf{X}$ via a strictly increasing transformation of $P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}) \rightarrow f_i(\mathbf{X})$. Note that the first one is the strictly probabilistic form of the BDF $\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}}$.

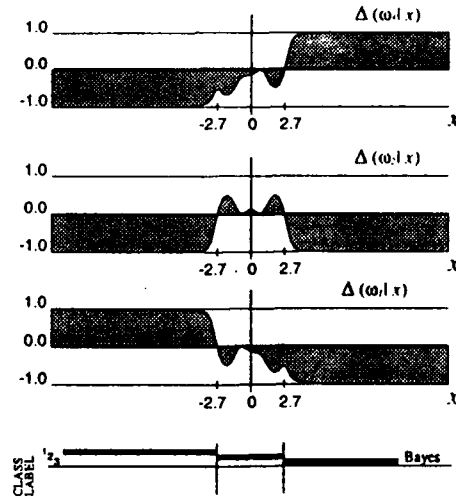


Figure 2.3: The *a posteriori* class differentials $\Delta_{\mathcal{W}|x}(\omega_i | x)$ for the three-class random variable x depicted in figure 2.1, plotted over the effective domain of x . These constitute the strictly differential form of the Bayesian discriminant function for x (see definition 2.5).

Although $\mathcal{F}(X)_{\text{Bayes-Strictly Probabilistic}}$ and $\mathcal{F}(X)_{\text{Bayes-Probabilistic}}$ are the most obvious forms of $\mathcal{F}(X)_{\text{Bayes}}$, other forms exist. One that satisfies (2.9) is manifest in any discriminant function with a top-ranked member function $f_*(X)$ corresponding to the largest *a posteriori* class probability $P_{\mathcal{W}|x}(\omega_* | X)$ in (2.8). The two categories of this *differential* form of the BDF are analogous to the two probabilistic categories defined above.

Definition 2.5 The strictly differential form of the BDF: This form of the BDF is manifest in any discriminant function $\mathcal{F}(X)_{\text{Bayes-Strictly Differential}}$ with the following property: the difference between the *i*th function $f_i(X)$ and the *k*th function $f_k(X)$ is equal to the difference between their corresponding *a posteriori* probabilities. For each $f_i(X)$, $f_k(X)$ is not chosen arbitrarily; rather it corresponds to the *a posteriori* probability $P_{\mathcal{W}|x}(\omega_k | X) = \max_{j \neq i} P_{\mathcal{W}|x}(\omega_j | X)$. Mathematically,

$$\begin{aligned} \mathcal{F}(X)_{\text{Bayes-Strictly Differential}} &= \{f_1(X), \dots, f_c(X)\} \\ \text{s.t. } \forall i, & \\ f_i(X) - f_k(X) &= \underbrace{P_{\mathcal{W}|x}(\omega_i | X) - P_{\mathcal{W}|x}(\omega_k | X)}_{\Delta_{\mathcal{W}|x}(\omega_i | x)} : P_{\mathcal{W}|x}(\omega_k | X) = \max_{j \neq i} P_{\mathcal{W}|x}(\omega_j | X) \end{aligned} \quad (2.13)$$

After some reflection it should be clear that the necessary and sufficient condition for the discriminant function to be a strictly differential form of the BDF is that its member functions be related to their corresponding

posteriori class probabilities by a constant k :

$$\mathcal{F}(\mathbf{X}) = \mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Differential}} \text{ iff } f_i(\mathbf{X}) = P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}) + k \quad \forall i \quad (2.14)$$

We denote the set of all strictly differential forms of the BDF by $\mathbf{F}_{\text{Bayes-Strictly Differential}}$.

Remark: Note that, by this and the preceding definition, $\mathbf{F}_{\text{Bayes-Strictly Differential}} \subset \mathbf{F}_{\text{Bayes-Probabilistic}}$. For this reason, $\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Differential}}$ reflects the rankings of all *a posteriori* class probabilities in the rankings of its discriminant functions. We refer to $\Delta_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X})$ in (2.13) as the, “(*a posteriori*) differential of the i th class (given $\mathbf{X}$).”

Example 2.3 Figure 2.3 illustrates the *a posteriori* differentials $\{\Delta_{\mathcal{W}|\mathbf{x}}(\omega_1|\mathbf{X}), \dots, \Delta_{\mathcal{W}|\mathbf{x}}(\omega_3|\mathbf{X})\}$ that correspond to the three-class random variable depicted in figure 2.1. A review of these two figures and definition (2.5) reveals that the Bayes-optimal class label ω_* in (2.8) is ω_i for all patterns that elicit a non-negative i th class differential $\Delta_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X})$:

$$\omega_* = \omega_i \text{ iff } \Delta_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}) \geq 0 \quad (2.15)$$

The bar-graph displays at the bottom of figures 2.1 and 2.3 denote the most probable class ω_* for x over its effective domain. The bar-graphs also mark the class boundaries at $x = +/ - 2.7$.

Note that the class boundaries on $\mathcal{X}$ are indicated by the absence of a positive differential; only zero and negative differentials exist at the boundaries (again, for example, $x = +/ - 2.7$ in figure 2.3). In such cases, the Bayes-optimal class label for a boundary value of $\mathbf{X}$ — denoted by $\mathbf{X}_{\text{boundary}}$ — is any one of the classes with a corresponding *a posteriori* differential of zero:

$$\forall \mathbf{X}_{\text{boundary}} \in \mathcal{X}, \quad \omega_* \in \{\omega_i : \Delta_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}_{\text{boundary}}) = 0\} \quad (2.16)$$

Definition 2.6 The differential form of the BDF: This form of the BDF is manifest in any discriminant function $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$ with a top-ranked member function $f_*(\mathbf{X})$ corresponding to the largest *a posteriori* class probability $P_{\mathcal{W}|\mathbf{x}}(\omega_*|\mathbf{X})$ in (2.8). Mathematically,

$$\begin{aligned} \mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}} &= \{f_1(\mathbf{X}), \dots, f_c(\mathbf{X})\} \\ \text{s.t. } \forall i, \\ \text{sign} \left[\underbrace{f_i(\mathbf{X}) - \max_{j \neq i} f_j(\mathbf{X})}_{\delta_i(\mathbf{X})} \right] &= \text{sign} \left[\underbrace{P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}) - \max_{k \neq i} P_{\mathcal{W}|\mathbf{x}}(\omega_k|\mathbf{X})}_{\Delta_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X})} \right] \end{aligned} \quad (2.17)$$

We denote the set of all differential forms of the BDF by $\mathbf{F}_{\text{Bayes-Differential}}$.

Definition 2.7 The discriminant differential: We refer to $\delta_i(\mathbf{X})$ in (2.17) as the, “*ith discriminant differential*” — that is, the difference between the *ith* discriminant function and the largest other discriminant function. We use the notation

$$\delta_i(\mathbf{X}|\theta) \triangleq g_i(\mathbf{X}|\theta) - \max_{k \neq i} g_k(\mathbf{X}|\theta) \quad (2.18)$$

when referring to the *ith* discriminant differential of the classifier with the discriminator $\mathcal{G}(\mathbf{X}|\theta) = \{g_1(\mathbf{X}|\theta), \dots, g_C(\mathbf{X}|\theta)\}$. In this context, the discriminant differential is the difference between the classifier's *ith* output and the largest other output.

Remark: Note that $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$ accurately reflects the ranking of *only the largest a posteriori* class probability in the rankings of its discriminant functions and their associated discriminant differentials. We reiterate that the *only* condition necessary for the discriminant function to be a differential form of the BDF is that its top-ranked member function $f_*(\mathbf{X})$ always correspond to the largest *a posteriori* class probability $P_{W|\mathbf{X}}(\omega_*|\mathbf{X})$ in (2.8). This is precisely the necessary and sufficient condition for Bayesian discrimination given in (2.9), which leads us to the following theorem:

Theorem 2.1 The differential form of the Bayesian discriminant function $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$ is the most general; the set of all such forms $\mathbf{F}_{\text{Bayes-Differential}}$ is equal to the set of all Bayesian discriminant functions $\mathbf{F}_{\text{Bayes}}$ by the following relationship:

$$\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}} \in \mathbf{F}_{\text{Bayes-Strictly Differential}} \subset \mathbf{F}_{\text{Bayes-Probabilistic}} \subset \mathbf{F}_{\text{Bayes-Differential}} = \mathbf{F}_{\text{Bayes}} \quad (2.19)$$

Proof : The proof follows from definitions 2.6 – 2.11 ■

Example 2.4 In order to clarify the notion of a differential form of the BDF, let us return to the three-class random variable x depicted in figures 2.1 and 2.3, which we replicate in figure 2.4.

On the left side of the figure we shown the *a posteriori* class probabilities of x ; superimposed on these are the discriminant functions

$$\begin{aligned} f_1(x) &= -0.0376x + 0.3985 \\ f_2(x) &= 0.5 \\ f_3(x) &= 0.0376x + 0.3985 \end{aligned} \quad (2.20)$$

which comprise $\mathcal{F}(x)$. Note that $f_1(-2.7) = f_2(-2.7) = 0.5$ and $f_2(2.7) = f_3(2.7) = 0.5$, so that the discriminant differentials shown in figure 2.4 (right) are given by

$$\begin{aligned}\delta_i(x) &\triangleq f_i(x) - \max_{k \neq i} f_k(x) \\ \therefore \delta_1(x) &= \begin{cases} f_1(x) - f_2(x), & x < 2.7 \\ f_1(x) - f_3(x), & \text{otherwise} \end{cases} \\ &= \begin{cases} -0.0376x - 0.1015, & x < 2.7 \\ -0.0752x, & \text{otherwise} \end{cases} \\ \delta_2(x) &= \begin{cases} f_2(x) - f_1(x), & x < 0 \\ f_2(x) - f_3(x), & \text{otherwise} \end{cases} \\ &= \begin{cases} 0.0376x + 0.1015, & x < 0 \\ -0.0376x + 0.1015, & \text{otherwise} \end{cases} \\ \delta_3(x) &= \begin{cases} f_3(x) - f_2(x), & x > -2.7 \\ f_3(x) - f_1(x), & \text{otherwise} \end{cases} \\ &= \begin{cases} 0.0376x - 0.1015, & x > -2.7 \\ 0.0752x, & \text{otherwise} \end{cases} \end{aligned} \quad (2.21)$$

Note that the largest discriminant function and, as a result, the positive discriminant differential always correspond to the largest *a posteriori* class probability of x . For this reason, (2.17) is satisfied and $\mathcal{F}(x) = \mathcal{F}(x)_{\text{Bayes-Differential}}$. By comparing the characteristics of $\mathcal{F}(x)$ against definitions 2.3 — 2.6, we find that $\mathcal{F}(x)$ does not satisfy the conditions for any other form of the BDF.

2.2.3 Learning Paradigms for the Bayesian Discriminant Function

We use the notation $\mathbf{X}^j$ to denote the j th example of the random vector $\mathbf{X}$; likewise, we use the notation $\mathcal{W}^j$ to denote the class label of that example. Supervised learning is the process by which the n example/empirical class label pairs $\{(\mathbf{X}^1, \mathcal{W}^1), \dots, (\mathbf{X}^n, \mathcal{W}^n)\}$ of the *training sample*⁷ are used to adjust the parameters of the classifier so that its labeling of the training examples matches the actual class labels as closely as possible. As the training sample size increases towards infinity, the empirical *a posteriori* class probabilities of the

⁷The training sample is the set of all pairs $\{(\mathbf{X}^1, \mathcal{W}^1), \dots, (\mathbf{X}^n, \mathcal{W}^n)\}$ used to train the classifier. The *test sample* is used to assess the classifier's discrimination, and comprises all pairs not in the training sample.

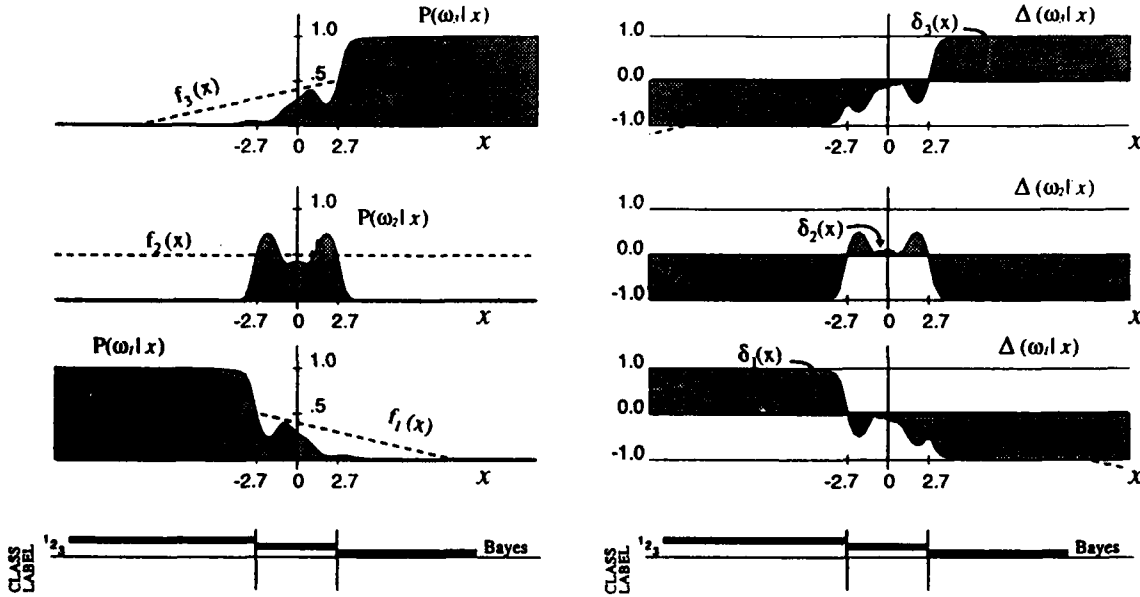


Figure 2.4: Left: The *a posteriori* class probabilities $P_{W|X}(\omega_i | x)$ of the three-class random variable depicted in figures 2.1 and 2.3, with a differential form of the Bayesian discriminant function $\mathcal{F}(X)_{\text{Bayes-Differential}}$ superimposed. Right: The *a posteriori* class differentials $\Delta_{W|X}(\omega_i | x)$ of the same three-class random variable, with the discriminant differentials of $\mathcal{F}(X)_{\text{Bayes-Differential}}$ superimposed. Note that where the i th discriminant differential $\delta_i(X)$ is positive, ω_i is the Bayes-optimal class label for x .

feature vector converge to their true underlying values. For this reason, the classifier possessing sufficient functional complexity⁸ can learn to approximate the BDF in at least one of the four forms described in the previous section.

There are two fundamental learning strategies for statistical pattern recognition. Classifiers that employ *probabilistic learning* learn to approximate the probabilistic form of the BDF. That is, they learn to estimate the *a posteriori* probabilities of the C classes over all of feature vector space — as exemplified by the gray shaded functions in figure 2.4 (left). The estimation is done either directly or by estimating the class-conditional pdf's and class prior probabilities, from which the *a posteriori* class probabilities can be computed. Classifiers that employ *discriminative learning* do not learn to estimate probabilities; they merely learn to estimate the identity of the Bayes-optimal class over all of feature vector space. In effect, discriminative learning focuses on partitioning $\mathcal{X}$ along the class boundaries, thereby identifying regions on $\mathcal{X}$ inside which all patterns represent a single class in the Bayes-optimal sense. This learning is done without explicitly estimating the *a posteriori* probabilities of each class over $\mathcal{X}$; instead it focuses on estimating a

⁸A formal definition of functional complexity is not essential to this chapter; it is sufficient to state that there is some upper bound on the intricacy of the discriminator $\mathcal{G}(X | \theta)$ that a classifier with limited functional complexity can implement.

Paradigm	Differentiable Supervised Classifier?	Learning Strategy	Parametric?
Linear Classifiers			
Rosenblatt's perceptron	yes	discriminative ^{a,b}	no
Widrow-Hoff (i.e., LMS/MSE-generated) variants	yes	probabilistic ^a	no
Ho-Kashyap	yes	probabilistic/ discriminative ^{a,b,c}	no
logistic regression	yes	probabilistic ^a	yes
Non-Linear Classifiers			
k nearest neighbors	no	probabilistic	no
Parzen windows	no	probabilistic	no
radial basis functions	yes	probabilistic ^a	no
multi-layer perceptrons	yes	probabilistic ^a	no
decision trees	no	discriminative	no
LVQ2	no	discriminative	no

Table 2.1: Some well-known classification paradigms and the learning strategies they employ. The differentiable supervised classifier is described in definition 2.8.

^aCan learn differentially.

^bGuaranteed to be minimum-error only in the case that the training sample is linearly separable.

^cFundamentally probabilistic, but discriminative if the training examples are linearly separable.

differential form of the BDF — as exemplified by the dashed lined functions in figure 2.4. By definitions 2.5 and 2.6, this is equivalent to learning $\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Differential}}$ to at least one (sign) bit precision over $\mathcal{X}$.

Differential learning is discriminative learning in which the optimal parameters of the classifier are determined by a search on *parameter space* Θ aimed at optimizing a differentiable objective function. Three definitions are relevant at this point:

Definition 2.8 The differentiable supervised classifier: *This classifier is one that forms an input-to-output mapping by adjusting a set of internal parameters via an iterative search aimed at optimizing a differentiable objective function (or empirical risk measure). The objective function is a metric that evaluates how well the classifier's evolving mapping reflects the empirical relationship between the input patterns of the training sample and their class membership, modeled by the classifier's outputs. Each one of the classifier's discriminant functions $g_i(\mathbf{X} | \theta)$ must be a differentiable function of its parameters θ .*

Definition 2.9 Probabilistic learning Λ_P : Any classifier that learns a probabilistic form of the Bayesian discriminant function $\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}}$ (definition 2.4) employs probabilistic learning. We use the notation Λ_P to denote probabilistic learning. If the classifier is a differentiable supervised classifier (defined above), it implements probabilistic learning through the use of an error measure objective function (see sections 2.2.4 and 2.3).

Definition 2.10 Differential learning Λ_Δ : This is discriminative learning performed by a differentiable supervised classifier (defined above) that employs the classification figure-of-merit (CFM) objective function (see [55], sections 2.2.4 and 2.4, and appendix D). A classifier that employs differential learning learns the differential form of the Bayesian discriminant function $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$ (definition 2.6). We use the notation Λ_Δ to denote differential learning.

Table 2.1 lists a few well-known classifier paradigms and the learning strategies they employ; it emphasizes that all classifier paradigms can be associated with either the probabilistic or the discriminative learning strategy. Classifiers characterized as "linear" are those that form (piece-wise) linear decision boundaries on $\mathcal{X}$; those characterized as "non-linear" form potentially non-linear decision boundaries on $\mathcal{X}$. Rosenblatt's perceptron, the Widrow-Hoff linear classifier, and the Ho-Kashyap linear classifier all have linear discriminant functions $g_i(\mathbf{X} | \theta)$. Reference [29] describes each of these three classifiers in detail; they differ only by the manner in which they learn. Specifically, each uses a different objective function to search iteratively for optimal parameters. Thus, they constitute differentiable supervised classifiers. Rosenblatt's perceptron criterion function [116, 29] seeks only to classify $\mathbf{X}$ correctly, not to estimate the *a posteriori* probabilities; as a result, it is a discriminative learning procedure (see appendix E). The Widrow-Hoff and Ho-Kashyap variants both minimize a mean-squared error objective function: the Ho-Kashyap model adds a constraint to the MSE-minimization procedure that guarantees class separation of the training sample if it is indeed linearly separable. As we shall see in section 2.3.2, minimizing an MSE objective function is equivalent to approximating the *a posteriori* class probabilities of $\mathbf{X}$ — a probabilistic learning strategy.

The logistic regression model replaces the preceding linear discriminant functions with logistic discriminant functions $g_i(\mathbf{X} | \theta) = \left[1 + \exp \left(-(\theta_0 + \sum_{i=1}^N \theta_i x_i) \right) \right]^{-1}$ (e.g., [91, ch. 8] [68]). Although the decision boundaries on $\mathcal{X}$ remain (piece-wise) linear, the logistic discriminant function is a better choice for approximating the *a posteriori* class probabilities of $\mathbf{X}$ — particularly when the class-conditional pdfs of $\mathbf{X}$ are Gaussian (see appendix F). The logistic regression model learns by the method of maximum-likelihood, but the logistic non-linearity makes closed-form computation of the optimal parameters impossible. For this reason, logistic regression takes the form of an iterative learning procedure in which the Kullback-Leibler information distance ([82, 81] — see section 2.3.2) between the discriminant function(s) and the empirical *a posteriori* class probabilities of $\mathbf{X}$ (manifest in the training sample's statistics) is minimized (see

appendix F). Since the logistic discriminant function it incorporates is identical to the one employed in multi-layer perceptron (MLP) classifiers [120], the logistic regression model can be viewed as the MLP in its simplest form. Both the logistic regression paradigm and MLPs are probabilistically-generated differentiable supervised classifiers.

The k nearest neighbors algorithm estimates the class of a test example by comparing it with the most likely class of the k nearest training examples. The likelihood of each class is estimated by its relative frequency among the k nearest training examples. As the training sample size grows large, these relative frequencies converge to the true *a posteriori* class probabilities of $\mathbf{X}$, so the k nearest neighbors paradigm learns probabilistically. The lack of an objective function-directed learning procedure disqualifies it as a differentiable supervised classifier.

Parzen windows attempt to estimate the class-conditional densities of $\mathbf{X}$ via a linear superposition of window functions — one for each training sample. The specific form of the window function is not particularly important, as long as it is unimodal and has a unit area under its curve. The volume of $\mathcal{X}$ that the window covers is variable. As the training sample size grows large, the linear superposition of window functions for a given class converges to the true class-conditional density of $\mathbf{X}$ [29, sec. 4.3], so the paradigm constitutes a form of probabilistic learning. Again, the lack of an objective function-directed learning procedure disqualifies it as a differentiable supervised classifier.

Radial basis function neural networks (RBFs) (e.g., [18, 95, 104, 92]) are — like MLPs mentioned earlier — discriminant functions formed by cascaded layers of non-linear basis functions. In the case of MLPs, the basis function is a logistic one that forms linear decision surfaces; in the case of RBFs, the basis function is most commonly a Gaussian one that forms radial (i.e., hyperelliptical) decision surfaces. Beyond these differences the models are quite similar. Both are differentiable supervised classifiers that typically employ probabilistic learning.

Decision trees are discriminative classifiers that form linear decision surfaces on $\mathcal{X}$ via a set of thresholds; these thresholds are expressed as a set of rules associated with each class of $\mathbf{X}$; the rules are expressed in disjunctive normal form (DNF)⁹. There are numerous methods by which the rules for dividing $\mathcal{X}$ into class regions are induced (see, for example, [139, ch. 5]). The details of rule induction are not important for our purpose, which is merely to point out that the process is fundamentally discriminative. Because the rules of the DNF are, in effect, step functions on $\mathcal{X}$, they are non-differentiable, and the resulting classifier does not satisfy the requirements for a differentiable supervised classifier.

The general family of vector quantization (VQ) classifiers contains paradigms that are probabilistic as well as discriminative.¹⁰ The k nearest neighbors paradigm described earlier can be viewed as a probabilistic VQ classifier. Discriminative variants also exist. The most notable is the LVQ2 paradigm [76], which

⁹See [139, pg. 114] for a simple definition of the DNF.

¹⁰See [89] for an extensive summary of vector quantization techniques through the early 1980's.

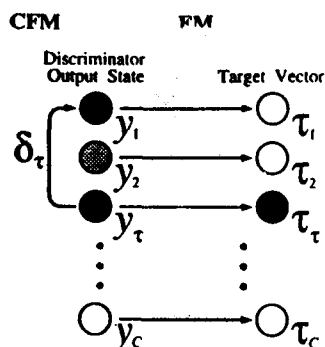


Figure 2.5: A diagrammatic comparison of error measure (EM) and classification figure-of-merit (CFM) objective functions (darker outputs have larger values than lighter ones). EMs attempt to match the discriminator output state (left) with a target vector (right); CFM does not. Instead it uses the target vector merely to identify the discriminator output y_t corresponding to the class label of the classifier's input example. CFM then seeks to maximize a function of the difference (or *discriminant differential*) δ_t between this output and the largest *other* output (in this case, y_1). The function $\sigma[\delta, \psi]$ is shown in figure 2.6.

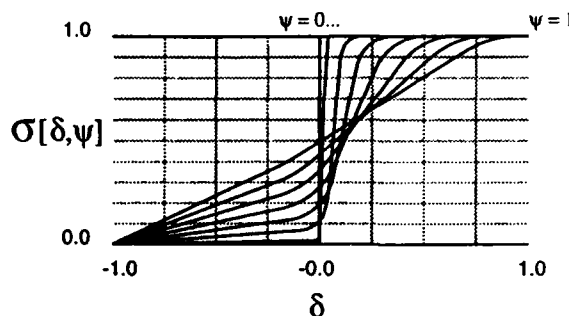


Figure 2.6: A synthetic asymmetric sigmoidal form of the classification figure-of-merit (CFM) [55] shown for discriminant differential values on the interval $-1 \leq \delta \leq 1$. The shape of the sigmoid is controlled by a confidence measure ψ on $(0, 1]$: $\sigma[\delta, \psi]$ is shown for eight different values of ψ . The differentiable function of δ is nearly linear for a confidence measure of unity. In the limit that ψ is zero, the function becomes a Heaviside step. The synthetic function and its first derivative are easily computed (see appendix D).

associates a number of prototypical vectors with each class. The vectors are initially determined by k-means clustering (e.g., see [89]). Learning is then performed by iteratively perturbing the vector locations on $\mathcal{X}$ with the goal of minimizing the number of misclassified training examples. The learning strategy is therefore discriminative; the resulting classifier does not satisfy the requirements for a differentiable supervised classifier.

Table 2.1 lists only a few classifiers. Countless others exist, but each is either a differentiable supervised classifier or it is not. The remainder of this chapter — and this text — deals with those that are, since all such classifiers can employ both probabilistic and differential learning.

2.2.4 The Link Between Objective Function and Learning Strategy

In section 2.3 we prove that differentiable supervised classifiers generated with a broad family of error measures (EMs) learn probabilistically; that is, they learn a probabilistic form of the BDF described by definitions 2.3 and 2.4. In section 2.4 we prove that these same classifiers learn differentially when generated with the CFM objective function [55]; that is, they learn a differential form of the BDF described by definition 2.6.

Figure 2.5 compares error measures to the CFM objective function diagrammatically. Error measures

such as mean-squared error (MSE) and the Kullback-Leibler information distance [82, 81] — known as “cross entropy” (CE) in the neural network literature (e.g., [64]) — compare the classifier’s discriminator output state with a *target vector* $\mathcal{T}$. Given a training example/empirical class label pair $\langle \mathbf{X}^j, \mathcal{W}^j = \omega_\tau \rangle$, the τ th element of $\mathcal{T}$ (τ_τ) is typically unity, indicating that the empirical class label of the example is ω_τ . All the other elements of $\mathcal{T}$ are typically zero.¹¹ The right-hand side of figure 2.5 illustrates how the classifier learns via an error measure: it alters its parameters in order to match the discriminator’s output state with the training example’s target vector. This is done by minimizing the error measure (EM) between these two vectors. The arrows superimposed on the gray shading between the discriminator output state and the target vector symbolize the process, which is iteratively repeated for all examples in the training sample until the average error measure converges to a small value.

Unlike its EM counterparts, the CFM objective function has no target values; this is because it is not an error measure. The left-hand side of figure 2.5 illustrates how the classifier learns via CFM. It alters its discriminator’s parameters in order to maximize the discriminant differential δ_τ between 1) the output y_τ corresponding to the class label ω_τ of the example, and 2) the largest *other* output $\bar{y}_\tau$ (note that $\bar{y}_\tau = y_1$ in the figure):¹²

$$\delta_\tau \triangleq y_\tau - \bar{y}_\tau; \quad (2.22)$$

$$\mathcal{W}^j = \omega_\tau \quad \bar{y}_\tau = \max_{k \neq \tau} y_k$$

Notational convention for the discriminant differential: We generally omit the subscript τ when referring to the discriminant differential. Absent a subscript, the notation δ always implies δ_τ .

The single curved arrow superimposed on the gray shading to the left of the discriminator output state in figure 2.5 symbolizes the computation of δ_τ , which is maximized by maximizing the measure $\sigma[\delta_\tau, \psi]$. The maximization is iteratively repeated for all examples in the training sample until the average CFM converges to a large value. Note that the target vector $\mathcal{T}$ is used only to identify y_τ .

Definition 2.11 The CFM objective function¹³: The CFM objective function for a given example is a strictly increasing function $\sigma[\delta, \psi]$ of the discriminant differential δ corresponding to the empirical class label of the training example. The function must have a sigmoidal form that spans the continuum between a linear function of δ and a step function of δ . The maximum steepness of the sigmoid is regulated by the

¹¹ $\mathcal{T}$ need not be binary. The following proofs allow for non-binary target vectors.

¹² It is important to note that the identity of the largest other output $\bar{y}_\tau$ in δ_τ is stochastic; it not only varies across examples, it may also change for a given example as learning progresses [55].

¹³ Throughout this text we refer to the form of CFM that involves the computation and use of one and only one discriminant differential. This is the form of CFM originally described as “N-monotonic CFM”: see [55, pg. 226].

confidence parameter ψ . The specific functional form of $\sigma[\cdot]$ is not important, as long as it satisfies the following sigmoidal constraints:

- The function must have finite lower and upper bounds l and h :

$$-\infty \ll l \leq \sigma[\delta, \psi] \leq h \ll \infty \quad (2.23)$$

- The function must be a strictly non-decreasing sigmoidal function of δ :

$$\left\{ \begin{array}{ll} \frac{d}{d\delta} \sigma[\delta, \psi] > 0, & \text{for small } |\delta| \\ \frac{d}{d\delta} \sigma[\delta, \psi] \geq 0, & \text{otherwise} \end{array} \right\} \quad (2.24)$$

- The function must have a maximum slope occurring in its transition region. This transition slope must be inversely proportional to the confidence parameter ψ :

$$\max_{\delta} \frac{d}{d\delta} \sigma[\delta, \psi] \propto \psi^{-1}, \quad \psi \in (0, 1] \quad (2.25)$$

This proportionality requirement ensures that learning is reasonably fast (see section D.3).

Remark: We use an asymmetric sigmoidal function for $\sigma[\cdot]$, which satisfies the constraints above as well as others imposed by theorem 2.2 of section 2.4 and chapter 5; it has lower and upper bounds of $l = 0$ and $h = 1$. This function is illustrated in figure 2.6; it is expressed by a computationally efficient mathematical form (see appendix D). The original sigmoidal form of the CFM objective function is given in [55]¹⁴, but the synthetic form described herein has a number of advantages relating to its computational efficiency and the differential learning rates it engenders (see appendix D and chapter 5). Figure 2.6 illustrates the synthetic function on the interval $-1 \leq \delta \leq 1$ for eight different confidence values ψ . Note that this synthetic function is approximately linear in δ for $\psi = 1$ and it is a step function of δ for $\psi \rightarrow 0^+$:

$$\begin{aligned} \lim_{\psi \rightarrow 1} \sigma[\delta, \psi] &\approx \frac{1}{2} (\delta + 1) \\ \lim_{\psi \rightarrow 0^+} \sigma[\delta, \psi] &= \begin{cases} 1, & \delta > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (2.26)$$

The parameter ψ is described further in section 2.4 and appendix D.

If the number of classes C is two, the discriminator $\mathcal{G}(X|\theta)$ is linear, and the classifier learns differentially via the CFM objective function, the resulting paradigm is quite similar to Rosenblatt's

¹⁴In this reference the parameter β is proportional to $1/\psi$.

perceptron [116]. Indeed, appendix C shows that one can view differential learning via CFM as a generalization of the two-class perceptron approach to discriminative learning. The generalization is such that

- there is no restriction on the number of classes C ,
- there is no restriction on the functional form of the discriminator (except that $\mathcal{G}(X|\theta)$ be differentiable on parameter space Θ),
- the learning procedure involves an iterative search on Θ aimed at maximizing a differentiable CFM objective function $\mathcal{O}[\cdot]$.

2.3 Probabilistic Learning Λ_P ¹⁵

The differentiable supervised classifier learns *probabilistically* by adjusting its parameters to minimize an error measure over the training sample. We assume that a number of favorable conditions exist prior to learning, in order to be sure that the classifier learns a probabilistic form of the BDF $\mathcal{F}(X)_{\text{Bayes-Probabilistic}} \in \mathbf{F}_{\text{Bayes-Probabilistic}}$ (see definitions 2.3 and 2.4):

- We assume that we have access to an unlimited number of training examples, so that we have sufficient data to learn $\mathcal{F}(X)_{\text{Bayes-Probabilistic}}$ precisely.
- We assume that the discriminator $\mathcal{G}(X|\theta)$ has sufficient functional complexity¹⁶ to learn $\mathcal{F}(X)_{\text{Bayes-Probabilistic}}$ precisely. Specifically, we assume that the classifier's parameter space Θ contains at least one point θ^* that both minimizes the error measure (EM) over the training set and satisfies the constraint $\mathcal{G}(X|\theta^*) \in \mathbf{F}_{\text{Bayes-Probabilistic}}$.
- We assume that the algorithm we use to search for the parameters θ^* is guaranteed to find θ^* , given sufficient time and computational resources.

In short, we assume that $\mathcal{F}(X)_{\text{Bayes-Probabilistic}}$ is learnable to the extent that we have sufficient (possibly infinite) information, computational, and temporal resources to learn it. We are not yet concerned with the efficiency of the learning procedure; we are merely concerned that it does the right thing, given enough resources. We address the issue of learnability in more realistic terms in the chapters that follow.

¹⁵Barak Pearlmutter has made important contributions to the material in this section. We note in particular his original formulation of 1) the general error measure (the present formulation is a minor extension), and 2) the strictly probabilistic constraint of (2.45).

¹⁶Again, we eschew a formal definition of functional complexity in this chapter, stating merely that there is some upper bound on the intricacy of the discriminant functions that a classifier with limited functional complexity can implement. We assume that the classifier has sufficient complexity to learn the probabilistic form of the BDF. While the proofs of this chapter are not limited to neural network classifiers per se, we note the proofs that feed-forward neural networks can learn arbitrarily complex mappings [24, 143].

2.3.1 The General Error Measure

Given a training example/class label pair $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, the target vector $\mathcal{T}$ for the discriminator outputs $\mathbf{Y}$ has elements that can assume one of two states. The target vector element corresponding to the empirical class of the training example is set to the high state D , and all the other elements are set to the low state $\neg D$ (read, "not D "):

$$\begin{aligned} \mathcal{T} &= \langle \tau_1, \dots, \tau_c \rangle; \quad \mathcal{T} \in \{\neg D, D\}^c \\ \tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) &= \begin{cases} D, & \mathcal{W}^j = \omega_i \\ \neg D, & \text{otherwise} \end{cases} \\ D &\in \mathbb{R}; \quad \neg D \in \mathbb{R}; \quad \neg D < D \end{aligned} \quad (2.27)$$

We require a particular kind of symmetry in the general error measure $\xi[\cdot]$: a discriminator output y_i that is higher than its low-state target $\neg D$ by the amount ϵ must generate the same error as a discriminator output y_k that is lower than its high-state target D by ϵ . This symmetry constraint reinforces our intuitive notion that the error measure should penalize all missed targets in a consistent manner. Mathematically,

$$\xi[y_i = \neg D + \epsilon, \tau_i = \neg D] = \xi[y_k = D - \epsilon, \tau_k = D] \quad \forall \epsilon \in \mathbb{R} \quad (2.28)$$

The general measure of error between the i th discriminator output and its target is therefore given by

$$\begin{aligned} &\xi[\underbrace{g_i(\mathbf{X}^j | \boldsymbol{\theta})}_{y_i}, \tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle)] \\ &= \begin{cases} f(\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) - g_i(\mathbf{X}^j | \boldsymbol{\theta})), & \tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) = D \\ f(g_i(\mathbf{X}^j | \boldsymbol{\theta}) - \tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle)), & \tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) = \neg D \end{cases} \\ &= \begin{cases} f(D - g_i(\mathbf{X}^j | \boldsymbol{\theta})), & \mathcal{W}^j = \omega_i \\ f(g_i(\mathbf{X}^j | \boldsymbol{\theta}) - \neg D), & \text{otherwise} \end{cases} \end{aligned} \quad (2.29)$$

The function $f(\cdot)$ in (2.29) is positive definite, with a unique minimum occurring when its argument is zero:

$$f(u = 0) < f(u \neq 0) \quad \& \quad \frac{d^2}{du^2} f(u) > 0 \quad (2.30)$$

Equations (2.28) — (2.30) are minor variants of the expressions in [54, sec. 3.1]. Miller, Goodman, and Smyth have derived similar expressions independently in [93].

The total error generated by a training sample S^n is the sum of the C error terms in (2.29), over all discriminator outputs, for each example:

$$\begin{aligned} \text{EM}(S^n | \theta) &\triangleq \sum_{i=1}^C e_i \\ e_i &\triangleq \frac{1}{n} \sum_{j=1}^n \xi[g_i(\mathbf{X}^j | \theta), \tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle)] \end{aligned} \quad (2.31)$$

Up to this point we have used the notation $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$ to denote an example $\mathbf{X}^j$ of $\mathbf{X}$ and its associated class label $\mathcal{W}^j$. Now we introduce the notation $\mathbf{X}_p^j$ to denote an example of $\mathbf{X}$ having the specific value $\mathbf{X}_p$ — a unique *pattern* or *prototype* (terms that we use synonymously) of $\mathbf{X}$ (i.e., a particular point on $\mathcal{X}$ identified by the subscript p). No two prototypes represent the same point on $\mathcal{X}$ ($\mathbf{X}_a = \mathbf{X}_b$ iff $a = b$), and there is no restriction on the number of prototypes.¹⁷ We denote the class label associated with $\mathbf{X}_p^j$ by $\mathcal{W}_p^j$, and we denote the resulting example/class label pair by $\langle \mathbf{X}_p^j, \mathcal{W}_p^j \rangle$. Using this notation, we can re-state e_i in (2.31) as

$$e_i = \frac{1}{n} \sum_{p=1}^P \sum_{j=1}^{n_p} \xi[g_i(\mathbf{X}_p^j | \theta), \tau_i(\langle \mathbf{X}_p^j, \mathcal{W}_p^j \rangle)] \quad (2.32)$$

where P denotes the total number of unique patterns and n_p denotes the number of examples of the pattern $\mathbf{X}_p$ among the n training examples. Thus, $\sum_{p=1}^P n_p = n$. If we use $n_{p,i}$ to denote the number of examples of the pattern $\mathbf{X}_p$ having the class label ω_i , such that $\sum_{i=1}^C n_{p,i} = n_p$, (2.29) can be used to simplify (2.32), replacing the notion of examples with the more general notion of prototypes:

$$e_i = \frac{1}{n} \sum_{p=1}^P [n_{p,i} \cdot f(D - g_i(\mathbf{X}_p | \theta)) + (n_p - n_{p,i}) \cdot f(g_i(\mathbf{X}_p | \theta) - \neg D)] \quad (2.33)$$

Thus

$$\text{EM}(S^n | \theta) = \sum_{i=1}^C \sum_{p=1}^P \frac{n_p}{n} \left[\frac{n_{p,i}}{n_p} \cdot f(D - g_i(\mathbf{X}_p | \theta)) + \frac{(n_p - n_{p,i})}{n_p} \cdot f(g_i(\mathbf{X}_p | \theta) - \neg D) \right] \quad (2.34)$$

As the training sample size n grows asymptotically large, the empirical frequencies converge to their underlying probabilities.¹⁸ Thus,

¹⁷Indeed, we assume the number of prototypes to be infinite for the uncountable feature vector space.

¹⁸See appendix B. Note also that we are not yet concerned with the specific rate at which this convergence takes place.

$$\begin{aligned} \lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{EM}(S^n | \theta) &= \sum_{i=1}^c \sum_{p=1}^P P_{\mathbf{X}}(\mathbf{X}_p) [P_{W|\mathbf{X}}(\omega_i | \mathbf{X}_p) \cdot f(D - g_i(\mathbf{X}_p | \theta)) \\ &\quad + P_{W|\mathbf{X}}(\neg \omega_i | \mathbf{X}_p) \cdot f(g_i(\mathbf{X}_p | \theta) - \neg D)] \end{aligned} \quad (2.35)$$

where

$$P_{W|\mathbf{X}}(\neg \omega_i | \mathbf{X}_p) \triangleq 1 - P_{W|\mathbf{X}}(\omega_i | \mathbf{X}_p) \quad (2.36)$$

Simultaneously, the number of patterns P grows asymptotically large, such that $\text{EM}(S^n | \theta)$ converges to the expected value of the error measure over $\mathcal{X}$, which we denote by $E_{\mathbf{X}}[\text{EM}(\mathbf{X} | \theta)]$:

$$\begin{aligned} \lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{EM}(S^n | \theta) &= E_{\mathbf{X}}[\text{EM}(\mathbf{X} | \theta)] \\ &= \sum_{i=1}^c \int_{\mathcal{X}} \underbrace{[f(D - g_i(\mathbf{X} | \theta)) \cdot P_{W|\mathbf{X}}(\omega_i | \mathbf{X}) \\ &\quad + f(g_i(\mathbf{X} | \theta) - \neg D) \cdot (1 - P_{W|\mathbf{X}}(\omega_i | \mathbf{X}))]}_{E_{\mathbf{X}}[e_i(\mathbf{X})]} \rho_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \end{aligned} \quad (2.37)$$

$$= \int_{\mathcal{X}} \left[\sum_{i=1}^c \underbrace{[f(D - g_i(\mathbf{X} | \theta)) \cdot P_{W|\mathbf{X}}(\omega_i | \mathbf{X}) + f(g_i(\mathbf{X} | \theta) - \neg D) \cdot (1 - P_{W|\mathbf{X}}(\omega_i | \mathbf{X}))]}_{\text{EM}(\mathbf{X} | \theta)} \right] \rho_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \quad (2.38)$$

To minimize $E_{\mathbf{X}}[\text{EM}(\mathbf{X} | \theta)]$ with respect to the parameters θ of the discriminator $\mathcal{G}(\mathbf{X} | \theta)$, we solve for θ^* such that

$$\nabla_{\theta} (E_{\mathbf{X}}[\text{EM}(\mathbf{X} | \theta^*)]) = E_{\mathbf{X}}[\nabla_{\theta} (\text{EM}(\mathbf{X} | \theta^*))] = \mathbf{0}, \quad (2.39)$$

where $\nabla_{\theta} (\text{EM}(\mathbf{X} | \theta^*))$ denotes the gradient of $\text{EM}(\mathbf{X} | \theta)$ with respect to θ , evaluated at θ^* , and $\mathbf{0}$ indicates the vector with zero magnitude. In order for (2.39) to hold for any and all $\rho_{\mathbf{X}}(\mathbf{X})$ on $\mathcal{X}$ (as a trivial example $\rho_{\mathbf{X}}(\mathbf{X}) = \delta(\mathbf{X} - \mathbf{X}_0)$, where $\delta(\cdot)$ denotes the Dirac delta function [80, pg. 266]), it is necessary that the error measure's gradient $\nabla_{\theta} (\text{EM}(\mathbf{X} | \theta^*))$ be zero for all values of $\mathbf{X}$:

$$\nabla_{\theta} \left(\underbrace{\sum_{i=1}^c \left[\underbrace{f(D - g_i(\mathbf{X}|\theta^*)) \cdot P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) + f(g_i(\mathbf{X}|\theta^*) - \neg D) \cdot (1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X}))}_{e_i(\mathbf{X})} \right]}_{EM(\mathbf{X}|\theta^*)} \right) = \underline{0} \quad \forall \mathbf{X} \in \mathcal{X} \quad (2.40)$$

This, in turn, requires that

$$\sum_{i=1}^c \frac{\partial e_i(\mathbf{X})}{\partial \theta_i^*} = \sum_{i=1}^c \frac{\partial e_i(\mathbf{X})}{\partial g_i(\mathbf{X}|\theta^*)} \cdot \frac{\partial g_i(\mathbf{X}|\theta^*)}{\partial \theta_i^*} = 0 \quad \forall \theta_i^*, \forall \mathbf{X} \in \mathcal{X} \quad (2.41)$$

Clearly, (2.41) is satisfied if the i th discriminant function's derivative with respect to the parameter θ_i is zero, but we are interested in the more general case for which this derivative is non-zero. Indeed, manipulating the parameters that affect the discriminator's functional mappings (i.e., those for which $\frac{\partial}{\partial \theta_i} g_i(\mathbf{X}|\theta) \neq 0$) is the whole point of learning. Thus, if the error in (2.37) is to be minimized independent of the values $\left\{ \frac{\partial}{\partial \theta_i} g_1(\mathbf{X}|\theta), \dots, \frac{\partial}{\partial \theta_i} g_c(\mathbf{X}|\theta) \right\}$, it is necessary that

$$\begin{aligned} \frac{\partial e_i(\mathbf{X})}{\partial g_i(\mathbf{X}|\theta^*)} &= \left[-f'(D - g_i(\mathbf{X}|\theta^*)) \cdot P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) \right. \\ &\quad \left. + f'(g_i(\mathbf{X}|\theta^*) - \neg D) \cdot (1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X})) \right] \\ &= 0 \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \end{aligned} \quad (2.42)$$

(where $f'(z)$ denotes $\frac{d}{dz} f(z)$). Rearranging terms,

$$f'(D - g_i(\mathbf{X}|\theta^*)) \cdot P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) = f'(g_i(\mathbf{X}|\theta^*) - \neg D) \cdot (1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X})) \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \quad (2.43)$$

or

$$\frac{f'(D - g_i(\mathbf{X}|\theta^*))}{f'(g_i(\mathbf{X}|\theta^*) - \neg D)} = \frac{(1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X}))}{P_{W|\mathbf{X}}(\omega_i|\mathbf{X})} \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \quad (2.44)$$

If we add the additional *strictly probabilistic constraint* on the functional form of the error measure $f(\cdot)$

$$f'(D - g_i(\mathbf{X}|\theta)) = f'(g_i(\mathbf{X}|\theta) - \neg D) \cdot \frac{(1 - g_i(\mathbf{X}|\theta))}{g_i(\mathbf{X}|\theta)} \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \quad (2.45)$$

then (2.42) becomes

$$\begin{aligned}
 f'(g_i(\mathbf{X}|\theta^*) - \neg D) \cdot \frac{(1 - g_i(\mathbf{X}|\theta^*))}{g_i(\mathbf{X}|\theta^*)} \cdot P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X}) &= f'(g_i(\mathbf{X}|\theta^*) - \neg D) \cdot (1 - P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X})) \\
 &= f'(g_i(\mathbf{X}|\theta^*) - \neg D) \left[\frac{P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X})}{g_i(\mathbf{X}|\theta^*)} - 1 \right] \\
 &= 0 \quad \forall i, \forall \mathbf{X} \in \mathcal{X}
 \end{aligned} \tag{2.46}$$

Equation (2.46) requires that

$$g_i(\mathbf{X}|\theta^*) = P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X}) \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \tag{2.47}$$

Thus, the differentiable supervised classifier learns $\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}}$ when (given our favorable assumptions) it is generated with an error measure satisfying (2.28) – (2.30) and the strictly probabilistic constraint of (2.45).

2.3.2 Specific Strictly Probabilistic Error Measures

One family of error measures satisfying the strictly probabilistic constraint of (2.45), has the following functional forms:

$$\begin{aligned}
 f(u) &= \int u^r (\neg u)^{r-1} du \\
 f'(u) &= u^r (\neg u)^{r-1}
 \end{aligned} \tag{2.48}$$

where r is a positive integer,

$$\begin{aligned}
 u &= \begin{cases} y_i - \neg D, & \tau_i = \neg D \\ D - y_i, & \tau_i = D \end{cases} \\
 \neg u &= \begin{cases} D - y_i, & \tau_i = \neg D \\ y_i - \neg D, & \tau_i = D \end{cases}
 \end{aligned} \tag{2.49}$$

and we employ the relationship $y_i = g_i(\mathbf{X}|\theta)$ to simplify our notation. Using (2.49) and the relationships

$$\frac{d}{dy_i} (y_i - \neg D) = 1 \quad \text{s.t.} \quad d(y_i - \neg D) = dy_i \tag{2.50}$$

$$\frac{d}{dy_i} (D - y_i) = -1 \quad \text{s.t.} \quad d(D - y_i) = -dy_i, \tag{2.51}$$

we can expand (2.48) via two simple transformations of variables

$$\begin{aligned}
 f(y_i - \neg D) &= \int (y_i - \neg D)^r (D - y_i)^{r-1} dy_i \\
 f(D - y_i) &= - \int (D - y_i)^r (y_i - \neg D)^{r-1} dy_i \\
 f'(y_i - \neg D) &= (y_i - \neg D)^r (D - y_i)^{r-1} \\
 f'(D - y_i) &= (D - y_i)^r (y_i - \neg D)^{r-1}
 \end{aligned} \tag{2.52}$$

Substituting (2.52) into (2.44), we find that minimizing this family of error measures leads to the following relationship between the discriminator outputs and their corresponding *a posteriori* class probabilities:

$$\frac{D - y_i}{y_i - \neg D} = \frac{1 - P_{W|X}(\omega_i | X)}{P_{W|X}(\omega_i | X)} \tag{2.53}$$

$$\therefore y_i = P_{W|X}(\omega_i | X) \cdot (D - \neg D) + \neg D \tag{2.54}$$

Note that if the low and high-state target values are binary, the discriminator outputs equal their corresponding *a posteriori* class probabilities:

$$y_i = P_{W|X}(\omega_i | X); \quad \neg D = 0, \quad D = 1 \tag{2.55}$$

However, even if the target values are not binary, the discriminator outputs remain linear functions of their corresponding *a posteriori* probabilities. Since differentiation is a linear operation, any linear combination of K functions $f_k(u)$ satisfying (2.28) – (2.30), and (2.45)

$$\phi(u) = \sum_{k=1}^K \alpha_k f_k(u) \tag{2.56}$$

will constitute a viable error measure. The linear coefficients α_k must have values that enforce the constraints of (2.30) on $\phi(u)$, such that

$$\phi(u = 0) < \phi(u \neq 0) \quad \& \quad \frac{d^2}{du^2} \phi(u) > 0 \tag{2.57}$$

The family of error measures described by (2.48) and (2.49) has two familiar members.

($r = 0$) Kullback-Leibler information distance: When $r = 0$ in (2.48),

$$f(u) = \int (\neg u)^{-1} du = -\log(\neg u) \quad \text{s.t.} \quad \xi[y_i, \tau_i] = \begin{cases} -\log(D - y_i), & \tau_i = \neg D \\ -\log(y_i - \neg D), & \tau_i = D \end{cases} \tag{2.58}$$

When the low and high-state target values are binary, the error measure is the Kullback-Leibler information distance [82, 81] — known as “cross entropy” (CE) in the connectionist literature (e.g., [64]):

$$\xi\{y_i, \tau_i\} = \begin{cases} -\log(1 - y_i), & \tau_i = \neg D = 0 \\ -\log(y_i), & \tau_i = D = 1 \end{cases} \quad (2.59)$$

Note that the discriminator's output Y must be bounded in order for $f(\cdot)$ in (2.58) to both exist and meet the constraint imposed by (2.30):

$$Y \in \mathcal{Y} = [l, h]^c; \quad l \geq \neg D, \quad h \leq D, \quad l < h \quad (2.60)$$

If $l < \neg D \cup h > D$ above, then $\neg u$ can be negative, and $-\log(\neg u)$ will be undefined. If $l > \neg D, h < D$ above, the constraints imposed by (2.30) will be violated, strictly speaking. However, for practical purposes, setting low and high-state target values that are beyond the limits of the discriminator outputs when $f(\cdot)$ is given by (2.58) is equivalent to setting $\neg D = l$ and $D = h$. Thus, the Kullback-Leibler information distance dictates discriminator outputs on any subset of the output space $\mathcal{Y} \in [0, 1]^c$. With these constraints, the Kullback-Leibler information distance-generated classifier learns $\mathcal{F}(X)_{\text{Bayes-Strictly Probabilistic}}$ by the following proof:

$$\text{CE}(S^n | \theta) = \sum_{i=1}^c e_i = - \sum_{i=1}^c \sum_{p=1}^P \frac{n_p}{n} \left[\frac{n_{p,i}}{n_p} \cdot \log(g_i(X_p | \theta)) + \frac{(n_p - n_{p,i})}{n_p} \cdot \log(1 - g_i(X_p | \theta)) \right] \quad (2.61)$$

Following the derivation for the general error measure in (2.35) – (2.38), we obtain the expected value of the Kullback-Leibler information distance (cross entropy — CE), which we denote by $E_X [\text{CE}(X | \theta)]$:

$$\begin{aligned} \lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{CE}(S^n | \theta) &= E_X [\text{CE}(X | \theta)] \\ &= \sum_{i=1}^c \underbrace{- \int_{\mathcal{X}} [\log(g_i(X | \theta)) \cdot P_{W|X}(\omega_i | X) + \log(1 - g_i(X | \theta)) \cdot (1 - P_{W|X}(\omega_i | X))] \rho_X(X) dX}_{E_X[e_i(X)]} \end{aligned} \quad (2.62)$$

$$= \int_{\mathcal{X}} \left[\underbrace{- \sum_{i=1}^c [\log(g_i(\mathbf{X}|\boldsymbol{\theta})) \cdot P_{\mathcal{W}|\mathcal{X}}(\omega_i|\mathbf{X}) + \log(1 - g_i(\mathbf{X}|\boldsymbol{\theta})) \cdot (1 - P_{\mathcal{W}|\mathcal{X}}(\omega_i|\mathbf{X}))]}_{\text{CE}(\mathbf{X}|\boldsymbol{\theta})} \right] \rho_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \quad (2.63)$$

To minimize $E_{\mathbf{X}} [\text{CE}(\mathbf{X}|\boldsymbol{\theta})]$ with respect to the parameters $\boldsymbol{\theta}$ of the discriminator $\mathcal{G}(\mathbf{X}|\boldsymbol{\theta})$, we solve for $\boldsymbol{\theta}^*$ such that $\nabla_{\boldsymbol{\theta}} (E_{\mathbf{X}} [\text{CE}(\mathbf{X}|\boldsymbol{\theta}^*)]) = E_{\mathbf{X}} [\nabla_{\boldsymbol{\theta}} (\text{CE}(\mathbf{X}|\boldsymbol{\theta}^*))] = \mathbf{0}$. Following the litany of (2.39) – (2.42), we obtain the necessary condition for minimizing $E_{\mathbf{X}} [\text{CE}(\mathbf{X}|\boldsymbol{\theta})]$:

$$\begin{aligned} \frac{\partial e_i(\mathbf{X})}{\partial g_i(\mathbf{X}|\boldsymbol{\theta}^*)} &= \frac{1}{g_i(\mathbf{X}|\boldsymbol{\theta}^*)} \cdot P_{\mathcal{W}|\mathcal{X}}(\omega_i|\mathbf{X}) - \frac{1}{1 - g_i(\mathbf{X}|\boldsymbol{\theta}^*)} \cdot (1 - P_{\mathcal{W}|\mathcal{X}}(\omega_i|\mathbf{X})) \\ &= 0 \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \end{aligned} \quad (2.64)$$

In order for this equation to hold for any and all $\rho_{\mathbf{X}}(\mathbf{X})$ on $\mathcal{X}$, it is necessary that

$$\frac{P_{\mathcal{W}|\mathcal{X}}(\omega_i|\mathbf{X})}{g_i(\mathbf{X}|\boldsymbol{\theta}^*)} = \frac{1 - P_{\mathcal{W}|\mathcal{X}}(\omega_i|\mathbf{X})}{1 - g_i(\mathbf{X}|\boldsymbol{\theta}^*)} \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \quad (2.65)$$

or

$$g_i(\mathbf{X}|\boldsymbol{\theta}^*) = P_{\mathcal{W}|\mathcal{X}}(\omega_i|\mathbf{X}) \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \quad (2.66)$$

Again, if the low and high-state target values for the Kullback-Leibler information distance are not binary but meet the constraints of (2.60), the discriminator outputs will be a linear function of their corresponding *a posteriori* probabilities: $g_i(\mathbf{X}|\boldsymbol{\theta}) = P_{\mathcal{W}|\mathcal{X}}(\omega_i|\mathbf{X}) \cdot (D - \neg D) + \neg D$.

($r = 1$) **Mean squared error:** When $r = 1$ in (2.48),

$$f(u) = \int u du = \frac{1}{2}u^2 \quad \text{s.t.} \quad \xi[y_i, \tau_i] = \begin{cases} \frac{1}{2}(y_i - \neg D)^2, & \tau_i = \neg D \\ \frac{1}{2}(D - y_i)^2, & \tau_i = D \end{cases} \quad (2.67)$$

the error measure is the mean-squared error (MSE) objective function:

$$\xi[y_i, \tau_i] = \frac{1}{2}(y_i - \tau_i)^2 \quad (2.68)$$

MSE has the particularly nice property that it satisfies the constraints of (2.28) – (2.30), and (2.45) regardless of the nature of D , $\neg D$, and $\mathcal{Y}$.

With binary target values, the MSE-generated classifier learns $\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}}$ by the following proof:

$$\text{MSE}(S^n|\theta) = \frac{1}{2} \sum_{i=1}^c e_i = \frac{1}{2} \sum_{i=1}^c \sum_{p=1}^P \frac{n_p}{n} \left[\frac{n_{p,i}}{n_p} \cdot (g_i(\mathbf{X}_p|\theta) - 1)^2 + \frac{(n_p - n_{p,i})}{n_p} \cdot g_i(\mathbf{X}_p|\theta)^2 \right] \quad (2.69)$$

Following the derivation for the general error measure in (2.35) – (2.38), we obtain the expected value of the mean-squared error, which we denote by $E_{\mathbf{X}} [\text{MSE}(\mathbf{X}|\theta)]$:

$$\begin{aligned} \lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{MSE}(S^n|\theta) &= E_{\mathbf{X}} [\text{MSE}(\mathbf{X}|\theta)] \\ &= \frac{1}{2} \sum_{i=1}^c \int_{\mathcal{X}} \underbrace{[(g_i(\mathbf{X}|\theta) - 1)^2 \cdot P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) + g_i(\mathbf{X}|\theta)^2 \cdot (1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X}))]}_{E_{\mathbf{X}}[e_i(\mathbf{X})]} \rho_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \end{aligned} \quad (2.70)$$

$$= \int_{\mathcal{X}} \left[\frac{1}{2} \sum_{i=1}^c \underbrace{[(g_i(\mathbf{X}|\theta) - 1)^2 \cdot P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) + g_i(\mathbf{X}|\theta)^2 \cdot (1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X}))]}_{\text{MSE}(\mathbf{X}|\theta)} \right] \rho_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \quad (2.71)$$

To minimize $E_{\mathbf{X}} [\text{MSE}(\mathbf{X}|\theta)]$ with respect to the parameters θ of the discriminator $\mathcal{G}(\mathbf{X}|\theta)$, we solve for θ^* such that $\nabla_{\theta} (E_{\mathbf{X}} [\text{MSE}(\mathbf{X}|\theta^*)]) = E_{\mathbf{X}} [\nabla_{\theta} (\text{MSE}(\mathbf{X}|\theta^*))] = \mathbf{0}$. Following the litany of (2.39) – (2.42), we obtain the necessary condition for minimizing $E_{\mathbf{X}} [\text{MSE}(\mathbf{X}|\theta)]$:

$$\begin{aligned} \frac{\partial e_i(\mathbf{X})}{\partial g_i(\mathbf{X}|\theta^*)} &= (g_i(\mathbf{X}|\theta^*) - 1) \cdot P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) + (g_i(\mathbf{X}|\theta^*)) \cdot (1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X})) \\ &= 0 \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \end{aligned} \quad (2.72)$$

In order for this equation to hold for any and all $\rho_{\mathbf{X}}(\mathbf{X})$ on $\mathcal{X}$, it is necessary that

$$(1 - g_i(\mathbf{X}|\theta^*)) \cdot P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) = g_i(\mathbf{X}|\theta^*) \cdot (1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X})) \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \quad (2.73)$$

or

$$g_i(\mathbf{X}|\theta^*) = P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) \quad \forall i, \forall \mathbf{X} \in \mathcal{X} \quad (2.74)$$

Again, if the low and high-state target values for MSE training are not binary, the discriminator outputs will be a linear function of their corresponding *a posteriori* probabilities:

$$g_i(\mathbf{X}|\theta) = P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) \cdot (D - \neg D) + \neg D.$$

2.3.3 Minkowski- r Power Metrics and Other Common Error Measures

The Minkowski- r power metric¹⁹ is given by

$$\xi[y_i, \tau_i] = \frac{1}{r} (|y_i - \tau_i|)^r \quad (2.75)$$

If we constrain discriminator output space as in (2.60),²⁰ the metric's functional form reduces to

$$\begin{aligned} f(u) &= \frac{1}{r} u^r \\ f'(u) &= u^{r-1} \end{aligned} \quad (2.76)$$

Given u in (2.42),

$$\xi[y_i, \tau_i] = \begin{cases} \frac{1}{r} (D - y_i)^r, & \tau_i = D \\ \frac{1}{r} (y_i - \neg D)^r, & \tau_i = \neg D \end{cases} \quad (2.77)$$

Substituting (2.75) into (2.44), we find

$$\frac{D - y_i}{y_i - \neg D} = \underbrace{r - \sqrt{\frac{1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X})}{P_{W|\mathbf{X}}(\omega_i|\mathbf{X})}}}_{\zeta} \quad (2.78)$$

$$\begin{aligned} \therefore y_i &= \frac{D + \neg D \zeta}{1 + \zeta}; \quad 1 < r < \infty \\ &= (1 + \zeta)^{-1}; \quad \neg D = 0, D = 1, 1 < r < \infty \end{aligned} \quad (2.79)$$

For binary target values, the discriminator outputs engendered by the limiting values of r are

¹⁹The Minkowski- r power metric and the L_R norm (e.g., [78, ch. 4]) are closely related, but not identical.

²⁰It can be shown that if the constraints of (2.60) are violated, Minkowski- r power metrics can require complex discriminator outputs in order to satisfy (2.44).

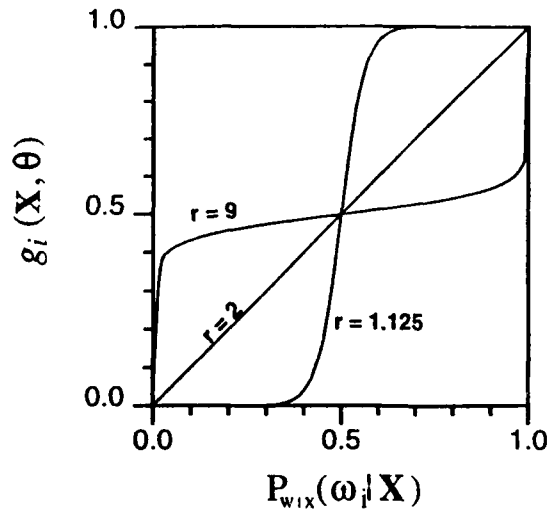


Figure 2.7: The discriminator output's minimum-error value for the Minkowski- r power metric ($r = 1.25, 2, 9$; binary output target values).

$$\lim_{r \rightarrow 1} y_i = \begin{cases} 1, & P_{w_i|X}(\omega_i|X) > .5 \\ .5, & P_{w_i|X}(\omega_i|X) = .5 \\ 0, & P_{w_i|X}(\omega_i|X) < .5 \end{cases} ; \quad \neg D = 0, D = 1 \quad (2.80)$$

$$\lim_{r \rightarrow \infty} y_i = \begin{cases} 1, & P_{w_i|X}(\omega_i|X) = 1 \\ .5, & 0 < P_{w_i|X}(\omega_i|X) < 1 \\ 0, & P_{w_i|X}(\omega_i|X) = 0 \end{cases}$$

Figure 2.7 illustrates the relationship between the discriminator output and its corresponding *a posteriori* class probability for three Minkowski- r power metrics with binary target values ($r = 1.25, 2, 9$). The $r = 2$ case corresponds to the MSE error measure described earlier.

($r = 1$) Mean absolute error: Note that when $r = 1$ in (2.75) – (2.77), $\xi[\cdot]$ is the mean absolute error (MAE) measure.²¹ Because, by (2.80), this EM engenders binary discriminator outputs, it is not a wise choice for pattern recognition tasks in which (the number of classes) $C > 2$. In such cases it is possible that all the *a posteriori* class probabilities are less than 0.5 for some values of $\mathbf{X}$. At such points on $\mathcal{X}$, the absolute error-generated classifier's discriminator outputs will all be zero by (2.80), and it will fail to identify the Bayes optimal class of $\mathbf{X}$ — a particularly undesirable characteristic. However, when $C = 2$, MAE has some desirable properties, which we discuss in section 5.3.1 (page 127).

²¹The mean absolute error measure is known by many other names; the most common are least absolute error (LAE) and least absolute deviation (LAD, e.g., [9]).

The $r = \infty$ case above engenders discriminator outputs that are all constant at 0.5 — resulting in a useless classifier. Values of r on the open interval $(1, \infty)$ engender discriminator outputs $g_i(\mathbf{X}|\theta)$ that are strictly increasing functions of their corresponding *a posteriori* class probabilities $P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X})$. Thus, classifiers generated with Minkowski- r power measures ($1 < r < \infty$) learn $\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}}$ of definition 2.4; only the $r = 2$ form leads to $\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}}$ of definition 2.3.

Other classes of error measures exist (e.g., [6, 98, 31, 43, 94]);²² in general they are variants of well-known EMs, and can be analyzed via (2.44) to determine the discriminator outputs they engender for asymptotically large training sample sizes. It is intuitively appealing that error measures — with a few notable exceptions — lead to classifiers that learn probabilistically. On the other hand it leads us to question — solely on the basis of learning efficiency (chapter 3) — the comparative advantage of choosing one error measure over another. To be sure, the statistical pattern recognition literature shows that specific error measures lead to more efficient learning, given *particular* class-conditional densities of $\mathbf{X}$. To date, however, we know of no single best-choice EM for the feature vector with *arbitrary* class-conditional densities.

2.4 Differential Learning Λ_Δ

The differentiable supervised classifier learns differentially by adjusting its parameters to maximize a classification figure-of-merit over the training sample. We assume that conditions analogous to the favorable conditions preceding probabilistic learning exist prior to differential learning, in order to be sure that the classifier learns a differential form of the BDF $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}} \in \mathbf{F}_{\text{Bayes-Differential}}$ (see definitions 2.5 and 2.6):

- We assume that we have access to an unlimited number of training examples, so that we have sufficient data to learn some $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}} \in \mathbf{F}_{\text{Bayes-Differential}}$.
- We assume that the discriminator $\mathcal{G}(\mathbf{X}|\theta)$ has sufficient functional complexity to learn $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$. Specifically, we assume that the classifier's parameter space Θ contains at least one point θ^* that both maximizes the CFM objective function (see section 2.2.4, chapter 5, and appendix D) over the training set and satisfies the constraint $\mathcal{G}(\mathbf{X}|\theta^*) \in \mathbf{F}_{\text{Bayes-Differential}}$.
- We assume that the algorithm we use to search for the parameters θ^* is guaranteed to find θ^* , given sufficient time and computational resources.

The measure of CFM generated by a training sample S^n is the sum of $\sigma[\cdot]$ in definition 2.11 for each example:

²²See [91, sec. 1.12] for an extensive list of error measures.

$$\text{CFM}(S^n | \theta) \triangleq \frac{1}{n} \sum_{j=1}^n (\sigma[\delta_\tau(X^j | \theta), \psi] : \mathcal{W}^j = \omega_\tau) \quad (2.81)$$

$$= \frac{1}{n} \sum_{p=1}^P \sum_{j=1}^{n_p} (\sigma[\delta_\tau(X_p^j | \theta), \psi] : \mathcal{W}_p^j = \omega_\tau) \quad (2.82)$$

where n is the training sample size, and P , n_p , and $n_{p,i}$ (below) are defined in section 2.3.1. Equation (2.82) can be simplified by replacing the notion of examples with the more general notion of prototypes:

$$m_i = \frac{1}{n} \sum_{p=1}^P [n_{p,i} \cdot \sigma[\delta_i(X_p | \theta), \psi]] \quad (2.83)$$

Thus

$$\text{CFM}(S^n | \theta) = \sum_{i=1}^C \sum_{p=1}^P \frac{n_p}{n} \left[\frac{n_{p,i}}{n_p} \cdot \sigma[\delta_i(X_p | \theta), \psi] \right] \quad (2.84)$$

As the training sample size n grows asymptotically large, the empirical frequencies converge to their underlying probabilities.²³ Thus,

$$\lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{CFM}(S^n | \theta) = \sum_{p=1}^P P_X(X_p) \left[\sum_{i=1}^C P_{\mathcal{W}|X}(\omega_i | X_p) \cdot \sigma[\delta_i(X_p | \theta), \psi] \right] \quad (2.85)$$

Simultaneously, the number of patterns P grows asymptotically large, such that $\text{CFM}(S^n | \theta)$ converges in probability to the expected value of the CFM objective function, which we denote by $E_X[\text{CFM}(X | \theta)]$:

$$\lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{CFM}(S^n | \theta) = E_X[\text{CFM}(X | \theta)] = \int_{\mathcal{X}} \underbrace{\left[\sum_{i=1}^C \sigma[\delta_i(X | \theta), \psi] \cdot P_{\mathcal{W}|X}(\omega_i | X) \right]}_{\text{CFM}(X | \theta)} \rho_X(X) dX \quad (2.86)$$

In order to maximize $E_X[\text{CFM}(X | \theta)]$ in (2.86) for any and all $\rho_X(X)$ on $\mathcal{X}$, we must maximize $\text{CFM}(X | \theta)$ for all $X \in \mathcal{X}$. Since the δ terms in (2.86) are not independent of one another (see (2.87) and (2.90) below) $\text{CFM}(X | \theta)$ cannot be maximized term-by-term; it must be maximized as a whole.

²³See appendix B. Note, as in the preceding probabilistic learning proofs, we are not yet concerned with the specific rate at which this convergence takes place.

Additionally, since $\frac{d}{d\delta} \sigma[\cdot]$ is non-negative, the common approach of differentiating $\text{CFM}(\mathbf{X} | \theta)$ with respect to some function of the δ terms is untenable. Instead we must take a less direct approach.

Suppose we rank the discriminator outputs, using the subscript (j) to denote the j th-ranked output (not to be confused with the subscript j , which denotes the output associated with ω_j):²⁴

$$\begin{aligned} y_{(1)} &= \max_k g_k(\mathbf{X} | \theta) \\ y_{(2)} &= \max_{k \neq (1)} g_k(\mathbf{X} | \theta) \\ y_{(3)} &= \max_{k \notin \{(1), (2)\}} g_k(\mathbf{X} | \theta) \\ &\vdots \\ y_{(c)} &= \min_k g_k(\mathbf{X} | \theta) \end{aligned} \quad (2.87)$$

Then, by

$$\delta_i \triangleq y_i - \max_{k \neq i} y_k \quad (2.88)$$

or equivalently,

$$\delta_i(\mathbf{X} | \theta) \triangleq g_i(\mathbf{X} | \theta) - \max_{k \neq i} g_k(\mathbf{X} | \theta) \quad (2.89)$$

each discriminant differential $\delta_{(j)}$ can be expressed in terms of the largest one ($\delta_{(1)}$) minus a positive rank-dependent value $\epsilon_{(j)}$:

$$\begin{aligned} \delta_{(1)} &= y_{(1)} - y_{(2)} \\ \delta_{(j)} &= y_{(j)} - y_{(1)} = -\delta_{(1)} - \epsilon_{(j)}; \quad j \geq 2 \\ \epsilon_{(j)} &\triangleq y_{(2)} - y_{(j)} \quad \forall j > 2; \quad \epsilon_{(2)} = 0, \quad \epsilon_{(j)} \geq 0 \end{aligned} \quad (2.90)$$

Using $\omega_{(i)}$ to denote the class associated with the i th-ranked output $y_{(i)}$ and discriminant differential $\delta_{(i)}$,

$$\begin{aligned} \text{CFM}(\mathbf{X} | \theta) &= \sum_{i=1}^c \sigma[\delta_{(i)}(\mathbf{X} | \theta), \psi] \cdot P_{\mathbf{W}|\mathbf{X}}(\omega_{(i)} | \mathbf{X}) \\ &= \sigma[\delta_{(1)}(\mathbf{X} | \theta), \psi] \cdot P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X}) \end{aligned}$$

²⁴We adopt a notational convention from the field of rank statistics by using subscripts in parentheses to denote rank (e.g., [49]).

$$+ \sum_{j=2}^c \sigma [-\delta_{(1)}(\mathbf{X}|\theta) - \epsilon_{(j)}(\mathbf{X}|\theta), \psi] \cdot P_{\mathbf{W}|\mathbf{X}}(\omega_{(j)}|\mathbf{X}) \quad (2.91)$$

Since $\frac{d}{d\delta}\sigma[\cdot]$ and all $\epsilon_{(j)}(\mathbf{X}|\theta)$ are non-negative, $\text{CFM}(\mathbf{X}|\theta)$ is greatest for a given $\delta_{(1)}(\mathbf{X}|\theta)$ if $\epsilon_{(j)}(\mathbf{X}|\theta) = 0 \ \forall j \geq 2$ (i.e., if all but the top-ranked output have the same value). In this case

$$\begin{aligned} \text{CFM}(\mathbf{X}|\theta) &= \\ &\sigma[\delta_{(1)}(\mathbf{X}|\theta), \psi] \cdot P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)}|\mathbf{X}) + \sigma[-\delta_{(1)}(\mathbf{X}|\theta), \psi] \cdot (1 - P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)}|\mathbf{X})) \\ &= \sigma[0, \psi] + \underbrace{\vartheta(\mathbf{X}) P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)}|\mathbf{X}) + \neg\vartheta(\mathbf{X}) (1 - P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)}|\mathbf{X}))}_{\Delta\text{CFM}(\mathbf{X}|\theta)}, \end{aligned} \quad (2.92)$$

where the perturbations ($\vartheta(\mathbf{X})$ and $\neg\vartheta(\mathbf{X})$) in the value of CFM from its equilibrium value $\sigma[0, \psi]$ — due to a non-zero discriminant differential $\delta_{(1)}(\mathbf{X}|\theta)$ — are

$$\begin{aligned} \vartheta(\mathbf{X}) &\triangleq \sigma[\delta_{(1)}(\mathbf{X}|\theta), \psi] - \sigma[0, \psi] \geq 0 \\ \neg\vartheta(\mathbf{X}) &\triangleq \sigma[-\delta_{(1)}(\mathbf{X}|\theta), \psi] - \sigma[0, \psi] \leq 0 \end{aligned} \quad (2.93)$$

Since $\vartheta(\mathbf{X})$ is non-negative and $\neg\vartheta(\mathbf{X})$ is non-positive $\Delta\text{CFM}(\mathbf{X}|\theta)$ is greatest, given specific values of $\vartheta(\mathbf{X})$ and $\neg\vartheta(\mathbf{X})$, when $\omega_{(1)} = \omega_*$ (again, by (2.8), ω_* denotes the class with the largest *a posteriori* probability). In this case, the discriminator output corresponding to the largest *a posteriori* class probability is larger than all the other outputs. We denote this Bayes-optimal output by y_* , and denote the corresponding output differential by δ_* .

The specific values of $\vartheta(\mathbf{X})$ and $\neg\vartheta(\mathbf{X})$ that maximize $\text{CFM}(\mathbf{X}|\theta)$ in (2.92) — values that we denote by $\vartheta(\mathbf{X})^*$ and $\neg\vartheta(\mathbf{X})^*$ — depend on the specific functional form of $\sigma[\delta_*(\mathbf{X}|\theta^*), \psi]$. Clearly they must satisfy

$$\begin{aligned} \vartheta(\mathbf{X})^* &= \sigma[\delta_*(\mathbf{X}|\theta^*), \psi] - \sigma[0, \psi] \\ \neg\vartheta(\mathbf{X})^* &= \sigma[-\delta_*(\mathbf{X}|\theta^*), \psi] - \sigma[0, \psi] \end{aligned}$$

$$\left(\underbrace{\left(\frac{\vartheta(\mathbf{X})^*}{-\neg\vartheta(\mathbf{X})^*} \geq \frac{1 - P_{\mathbf{W}|\mathbf{X}}(\omega_*|\mathbf{X})}{P_{\mathbf{W}|\mathbf{X}}(\omega_*|\mathbf{X})} \right)}_{\Delta\text{CFM}(\mathbf{X}|\theta^*) > 0} \cup \underbrace{(\vartheta(\mathbf{X})^* = \neg\vartheta(\mathbf{X})^* = 0)}_{\Delta\text{CFM}(\mathbf{X}|\theta^*) = 0} \right); \quad (2.94)$$

$$\omega_{(1)} = \omega_*, \text{ s.t. } \delta_{(1)}(\mathbf{X}|\theta) = \delta_*(\mathbf{X}|\theta^*)$$

for $\Delta\text{CFM}(\mathbf{X}|\theta)$ in (2.92) to be non-negative. If no $\delta_{(1)}(\mathbf{X}|\theta) = \delta_*(\mathbf{X}|\theta^*) > 0$ yields $\vartheta(\mathbf{X})^*$ and $\neg\vartheta(\mathbf{X})^*$ that satisfy the condition for $\Delta\text{CFM}(\mathbf{X}|\theta^*) > 0$ in (2.94), then $\text{CFM}(\mathbf{X}|\theta)$ is maximized when $\delta_{(i)}(\mathbf{X}|\theta) = 0 \ \forall i$ s.t. $\Delta\text{CFM}(\mathbf{X}|\theta) = 0$. That is, all the discriminator outputs have the same value, and the optimal $\text{CFM}(\mathbf{X}|\theta^*)$ is the “equilibrium” $\text{CFM}(\mathbf{X}|\theta^*) = \sigma[0, \psi]$.

Theorem 2.2 *The CFM objective function $\sigma[\delta, \psi]$ must be a bounded sigmoidal function spanning a continuum between a linear and a step function of δ in order to ensure that differential learning always generates the Bayes-optimal classifier.*

Proof : Consider two extreme scenarios:

- In the first scenario, $\mathbf{X}$ represents $\mathcal{C} = 2$ classes $\{\omega_1, \omega_2\}$. Thus, the smallest *a posteriori* class probability that the more likely class ω_* can have is $P_{\mathbf{W}|\mathbf{X}}(\omega_*|\mathbf{X}) = \frac{1}{2}$. Under these circumstances, (2.94) is satisfied if, by (2.90) and (2.93), $\delta_*(\mathbf{X}|\theta^*) > 0$ and $\sigma[\delta, \psi] = \delta$. In simple words, if the CFM objective function is linear in the discriminant differential δ , it will generate the Bayes-optimal classifier for the two-class case.

- In the second (worst-case) scenario, $\mathbf{X}$ represents $\mathcal{C} = \infty$ classes $\{\omega_1, \dots, \omega_\infty\}$. Thus, the smallest *a posteriori* class probability that the most likely class ω_* can have approaches zero $P_{\mathbf{W}|\mathbf{X}}(\omega_*|\mathbf{X}) \rightarrow 0^+$. Under these circumstances, (2.94) is satisfied if and only if, by (2.90) and

$$(2.93), \delta_*(\mathbf{X}|\theta^*) > 0 \text{ and } \sigma[\delta, \psi] = \begin{cases} h, & \delta > 0 \\ l, & \delta \leq 0 \end{cases}, \text{ where } l \text{ and } h \text{ are real constants,}$$

and $l < h$. In simple words, the CFM objective function must be a step function of the discriminant differential δ in order to generate the Bayes-optimal classifier for the malicious $\mathcal{C} \gg 1$ -class case.

For the more general case that falls between these two extremes, CFM must have a bounded sigmoidal shape in order for (2.94) to hold via (2.90) — (2.93). The lack of a finite lower bound l on $\sigma[\delta, \psi]$ in particular prevents the ratio $\frac{\vartheta(\mathbf{X})^*}{\neg\vartheta(\mathbf{X})^*}$ from being sufficiently large to satisfy (2.94) for all $P_{\mathbf{W}|\mathbf{X}}(\omega_*|\mathbf{X})$. The lack of a finite upper bound h on $\sigma[\delta, \psi]$ generates classifiers with large discriminant differentials. While this phenomenon is not fatal to Bayesian discrimination (as the lack of a finite lower bound is), it does prevent the discriminator from learning those forms of $\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$ for which $\delta_*(\mathbf{X}|\theta^*)$ is a small positive number. As we will see in chapters 3 and 6, if differential learning via CFM is to be efficient, it must allow the classifier to learn *any and all* $\mathcal{G}(\mathbf{X}|\theta) \in \mathbf{F}_{\text{Bayes-Differential}}$. Thus, CFM must have a bounded sigmoidal form. ■

Remark: Section III of [55] and section 5.4 provide a more intuitive rationale for the sigmoidal form of CFM, which might be helpful to the reader. We stress that the steepness of the CFM sigmoid need not vary across training examples; it simply needs to satisfy (2.94) for the worst-case $P_{W|X}(\omega_* | X)$ as approximated in the statistics of the training sample. Chapter 7 discusses practical approaches to setting ψ in order to ensure that (2.94) is satisfied.

We derive the specific values of $\vartheta(X)^*$ and $\neg\vartheta(X)^*$ for the two limiting forms of the CFM objective function satisfying the constraints of definition 2.11 and appendix D. The derivations assume that $l = 0$, $h = 1$, and $-1 \leq \delta \leq 1$ in (2.23) for the sake of simplicity. This constraint on δ requires that the discriminator outputs be bounded: $Y \in \mathcal{Y} = [0, 1]^C$. Since the output state of any classifier can be normalized to $[0, 1]^C$, via a simple affine transformation, the following derivations hold for the general classifier with outputs $Y \in \mathcal{Y} = \mathbb{R}^C$.

Linear CFM ($\psi = 1$): When the confidence parameter ψ assumes its maximum value of unity, the CFM objective function has the following form; the expression is approximately linear for all discriminant differentials δ not greater than one, otherwise it assumes the maximum value of unity:

$$\sigma[\delta, \psi = 1] \cong \begin{cases} \frac{1}{2}(\delta + 1), & \delta \leq 1 \\ 1, & \text{otherwise} \end{cases} \quad (2.95)$$

The perturbations ($\vartheta(X)$ and $\neg\vartheta(X)$) in the value of CFM from its equilibrium value $\sigma[0, \psi]$ — due to a non-zero discriminant differential $\delta_*(X | \theta^*)$ — are therefore

$$\begin{aligned} \sigma[0, \psi = 1] &\cong \frac{1}{2} \\ \vartheta(X) &\cong \begin{cases} \frac{1}{2}\delta_*(X | \theta^*), & \delta_*(X | \theta^*) \leq 1 \\ \frac{1}{2}, & \text{otherwise} \end{cases} \\ \neg\vartheta(X) &\cong \begin{cases} -\frac{1}{2}\delta_*(X | \theta^*), & \delta_*(X | \theta^*) \geq -1 \\ \frac{1}{2}, & \text{otherwise} \end{cases} \end{aligned} \quad (2.96)$$

and (2.94) is satisfied if and only if $P_{W|X}(\omega_* | X) \geq \frac{1}{2}$. Thus, for all non-boundary X , the discriminant differential that maximizes $\text{CFM}(X | \theta)$ is

$$\delta_*(X | \theta^*) = \begin{cases} 1 \text{ s.t. } \vartheta(X)^* \cong \frac{1}{2}, \neg\vartheta(X)^* \cong -\frac{1}{2}, & P_{W|X}(\omega_* | X) > \frac{1}{2} \\ 0 \text{ s.t. } \vartheta(X)^* = \neg\vartheta(X)^* = 0, & \text{otherwise} \end{cases} \quad (2.97)$$

$$\therefore y_* = \begin{cases} 1, & P_{W|X}(\omega_i | X) > \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \quad (2.98)$$

Differential learning via linear CFM therefore exhibits the same pathology that probabilistic learning via mean absolute error (MAE) does. If $C = 2$, then a classifier that learns via linear CFM can learn $\mathcal{F}(X)_{\text{Bayes-Differential}}$. However, if $C > 2$, then all discriminator outputs of the linear CFM-generated classifier will have the same value for all regions on $\mathcal{X}$ where $P_{W|X}(\omega_* | X) < \frac{1}{2}$. On such regions the linear CFM-generated classifier will fail to identify the Bayes-optimal class of X .

$$\text{Step CFM } (\psi = 0^+): \text{ When } \sigma[\delta, \psi = 0^+] = \begin{cases} 1, & \delta > 0 \\ 0, & \delta \leq 0 \end{cases} :$$

$$\sigma[0, \psi = 0^+] = 0$$

$$\vartheta(X) = \begin{cases} 1, & \delta_*(X | \theta^*) > 0 \\ 0, & \delta_*(X | \theta^*) \leq 0 \end{cases} \quad (2.99)$$

$$\neg\vartheta(X) = 0$$

and (2.94) is satisfied for all $P_{W|X}(\omega_* | X)$. Thus, for all non-boundary X , the discriminant differential δ_* need only be positive to maximize $\text{CFM}(X | \theta)$,

$$\delta_*(X | \theta^*) > 0 \text{ s.t. } \vartheta(X)^* = 1, \neg\vartheta(X)^* = 0, \quad (2.100)$$

the discriminator outputs all satisfy the constraints of (2.9), and the classifier that learns with the step form of CFM learns $\mathcal{F}(X)_{\text{Bayes-Differential}}$.

The general CFM ($0^+ < \psi < 1$): When CFM is neither of its limiting forms, differentiating $\sigma[\delta, \psi]$ with respect to δ does not lead to a closed-form expression for the specific value of $\delta_*(X | \theta^*)$ corresponding to $\vartheta(X)^*$ and $\neg\vartheta(X)^*$ in (2.94).

Nevertheless, the steepness of the CFM objective function's sigmoidal form — regulated by ψ — can be shown to govern $\vartheta(X)$ and $\neg\vartheta(X)$ thus:²⁵

$$\frac{\vartheta(X)}{\neg\vartheta(X)} = \mathcal{O}[\psi^{-1}] \quad (2.101)$$

Indeed, as long as²⁵

²⁵See appendix D, section D.4.

$$0 < \psi \leq 1 \quad \cap \quad \psi \leq 1.43 \frac{P_{W|X}(\omega_* | X)}{1 - P_{W|X}(\omega_* | X)} \quad (2.102)$$

both (2.9) and (2.94) are satisfied, and maximizing CFM ensures that the classifier learns a differential form of the Bayesian discriminant function:

$$\begin{aligned} \text{sign}[\delta_i(X|\theta)] &= \text{sign}[\Delta_{W|X}(\omega_i | X)] \quad \forall i, \forall X \in \mathcal{X} \\ \text{s.t.} \quad \mathcal{G}(X|\theta) &= \mathcal{F}(X)_{\text{Bayes-Differential}} \in \mathbf{F}_{\text{Bayes-Differential}} \end{aligned} \quad (2.103)$$

Maximizing CFM is tantamount to establishing a correlation coefficient of unity between the index of the discriminator's largest output $g_{(1)}(X|\theta)$ and the Bayes-optimal class label ω_* , given by (2.8). In short, differential learning is discriminative. Chapter 3 proves that for the limiting step form of the CFM objective function (i.e., $\lim_{\psi \rightarrow 0+} \sigma[\delta, \psi]$), maximizing CFM is equivalent to minimizing the classifier's error rate. That proof is central to the proof that differential learning via CFM is asymptotically efficient.

The preceding proofs make no assumptions regarding the functional form of $\mathcal{G}(X|\theta)$ beyond those stated at the beginning of this section, nor do they restrict the number of classes $\mathcal{C}$. Barnard proffers a less general but more elegant proof that differential learning leads to Bayesian discrimination in [5]; it is restricted to the two-class case in which the mappings $g_i(X|\theta)$ are linear functions of X .

2.4.1 Further Constraints Imposed on ψ by the Discriminator

Equation (2.102) specifies an upper bound on ψ , given the *a posteriori* probability of the most likely class ($P_{W|X}(\omega_* | X)$), which of course depends on X . Since ψ uniquely specifies the minimum discriminant differential $\delta_*(X|\theta^*)$ for which CFM is maximized (this value is μ_{ψ} in appendix D — see figure D.1, page 329), (2.102) implicitly assumes that the discriminator $\mathcal{G}(X|\theta)$ can generate a discriminant differential at least this large, in order to satisfy the constraint

$$\begin{aligned} \psi \text{ must satisfy (2.102) and possibly be reduced further, such that} \\ \delta_*(X|\theta^*) \geq \underbrace{\mu_{\psi}}_{(D.26)} \quad \cap \quad \sigma[\delta_*(X|\theta^*), \psi] = 1 \end{aligned} \quad (2.104)$$

In the event that the discriminator *cannot* satisfy (2.104) for the ψ specified by (2.102), ψ must be reduced to the value at which (2.104) is satisfied. In simple terms, the confidence parameter must be reduced so that the discriminator maximizes CFM for the largest positive discriminant differential $\delta_*(X|\theta^*)$ it can manage to generate, however small $\delta_*(X|\theta^*)$ might be. Under such conditions, the upper bound on ψ is

determined by the functional properties of the discriminator rather than by the *a posteriori* class probabilities of the feature vector. An illustration of this phenomenon is given in section 5.3.6.

2.5 Summary

In this chapter we have outlined the general statistical pattern recognition paradigm and defined two fundamental forms of the Bayesian discriminant function: *probabilistic* and *differential*. We have shown that each of these BDF forms is associated with a learning strategy, and that each learning strategy is, in turn, associated with a family of objective functions used to train the *differentiable supervised classifier*. We have proven that both the probabilistic and differential learning strategies generate the Bayes-optimal classifier, given sufficient information, computational, and temporal resources.

The probabilistic and differential learning strategies lead to Bayesian discrimination in substantially different ways. The probabilistic strategy has the distinct advantage of generating classifiers that reflect the *a posteriori* class probabilities of the feature vector in the outputs of the discriminator, whereas the differential strategy merely reflects the identity of the most probable class. There are clear advantages to having the classifier estimate the *a posteriori* class probabilities of the feature vector. This fact might lead the reader to wonder what advantage there is in using differential learning instead of probabilistic learning. In fact, the advantage lies in the *efficiency* of the differential learning strategy — that is, its ability to approximate Bayesian discrimination with the smallest training sample and the least complex classifier necessary for the task.

Throughout this chapter we have assumed that we have access to unlimited training data, a classifier with potentially unlimited functional complexity, a search algorithm that assures us of finding the globally optimal parameterization of our classifier, and infinite time for the algorithm to converge. In reality none of these advantages exist; we face the challenge of achieving the best discrimination possible, given limited training data, relatively simple classifier paradigms, limited time for learning, and search algorithms that become increasingly slow and subject to halting in local optima as classifier complexity increases. If the ultimate objective is estimating the *a posteriori* class probabilities of the feature vector, then we are compelled to use probabilistic learning. This, in turn, compels us to employ a sufficiently complex classifier and obtain a sufficiently large training sample if we are to have confidence in the classifier's probabilistic estimates. However, if the ultimate objective is simply to classify patterns, then differential learning is a better strategic choice, allowing us to achieve the goal of robust pattern classification efficiently. Proofs of this claim are given in chapters 3 and 6.

Chapter 3

Differential Learning is Asymptotically Efficient¹

Outline

We present a formal definition of efficiency in the statistical pattern recognition context. By viewing the classifier's discriminator as an estimator of the Bayesian discriminant function (BDF), we regard the classifier's error rate² as an estimator of the *Bayes error rate* (i.e., the Bayes-optimal classifier's minimum error rate). On this basis, we use traditional estimation-theoretic notions of bias and variance to define the *efficient classifier* and the *efficient learning strategy*. These definitions, in turn, lead to a quantitative measure of generalization — the ability of a classifier to discriminate accurately examples not encountered during learning. We prove that differential learning is asymptotically efficient and that it requires the classifier with the least functional complexity necessary for Bayesian discrimination. We prove that probabilistic learning is inefficient and that it does not guarantee Bayesian discrimination with the minimum-complexity classifier.

3.1 Introduction

The proofs of chapter 2 rely on favorable but unrealistic assumptions of learnability. Regardless of whether the Bayes-optimal classifier is simple or complex, we have only limited time and computational resources, and — perhaps most importantly — limited access to training data. In the face of such restrictions the *efficiencies* of the learning strategy and the classifier it generates become important.

In this chapter we employ classical estimation theory (e.g., [22, ch's. 32-34] [134]) to define the mean-squared discriminant error of a classifier as the expected squared difference between its error rate and the *Bayes error rate* (i.e., the minimum possible error rate, yielded by the Bayes-optimal classifier of definition 2.1). The classifier's mean-squared discriminant error is a measure of its ability to generalize

¹Sections 3.3 and 3.5 contain detailed versions of proofs first outlined in [53].

²We use the term "error rate" as a synonym for probability of error.

well to patterns not encountered during learning. The efficient classifier exhibits minimum mean-squared discriminant error, thereby guaranteeing the highest probability of good generalization. The efficient learning strategy is defined as one that guarantees the lowest mean-squared discriminant error allowed by the *a priori* choice of "hypothesis class",³ no matter what that choice is. These definitions are shown to be quite different from the definitions of *functional* bias, variance, and mean-squared error typically discussed in the connectionist, machine learning, and statistical pattern recognition literature.

We show that the differentiable supervised classifier performs general Bayesian learning (e.g., [29, sec. 3.5]) in which its discriminator's parameterization is transformed to a post-learning state from a "tabula rasa" state prior to learning. Viewed over all possible initial parameterizations, learning transforms the classifier's⁴ *a priori* parameter probability density function (pdf) to its posterior parameter pdf. As a result, we show that the classifier's expected ability to approximate Bayesian discrimination depends entirely on the posterior parameter pdf, which in turn and in part depends on the learning strategy employed.

We prove that differential learning is asymptotically efficient (i.e., efficient for asymptotically large training sample sizes) and that it requires the classifier with the least functional complexity necessary for Bayesian discrimination; we also prove that probabilistic learning is inefficient and that it does not guarantee Bayesian discrimination with the minimum-complexity classifier. We therefore argue in favor of *differential learning and against probabilistic learning (for all but a few special cases) when information and computational resources are limited.*

3.2 Discriminant Error, the Efficient Classifier, and the Efficient Learning Strategy

Consider the classifier's discriminator as an estimator of the Bayesian discriminant function — or, more precisely, consider the classifier's error rate as an estimator of the Bayes error rate (definition 3.2).⁵ From this perspective, the classifier's *discriminant efficiency* can be assessed in terms of discriminant bias and variance expressions that reflect how well and how consistently the classifier approximates the Bayes error rate for the pattern recognition task. We use the notation $P_e(\cdot)$ to denote error rate, and remind the reader that $\Gamma(\mathcal{G}(X|\theta))$ and $\mathcal{D}(X|\theta)$, which are given in (2.6) and (2.7), represent the class label that the classifier with parameterization θ assigns to X . Thus, the probability that the classifier will misclassify X is

³The term *hypothesis class* arises in PAC learning theory. In the statistical pattern recognition context it describes the set of all possible discriminators $\mathcal{G}(X|\theta)$, given our choice of classifier paradigm and parameter space. The classifier paradigm is determined by the functional basis of its discriminant functions (e.g., the logistic functional basis of multi-layer perceptrons and the Gaussian basis of Gaussian radial basis functions). The set of all C -output multi-layer perceptrons with no more than 500 total connections is therefore an example of a hypothesis class. Please see definition 3.5.

⁴The classifier's and discriminator's parameterizations are one and the same.

⁵Fukunaga takes this perspective in [40, ch. 7], although he considers the estimated error rate of the classifier, rather than the *true* error rate to which we refer (see definition 3.1). We use the term "error rate" when referring to the classifier's true probability of error, and "estimated error rate" when referring to any empirical estimate of the classifier's true probability of error. Of course, the classifier's error rate is an abstraction — a number that we cannot know with certainty for any real classifier. Nevertheless, the quantity is central to the arguments of this chapter.

$$\begin{aligned}
 P_e(\mathcal{G}(X|\theta)) &\triangleq 1 - P_{W|X}(\mathcal{D}(X|\theta) | X) \\
 &= 1 - P_{W|X}(\Gamma(\mathcal{G}(X|\theta)) | X)
 \end{aligned}
 \tag{3.1}$$

Definition 3.1 The Classifier's Error Rate (or Probability of Error): *The classifier's error rate, or probability of error, is the expected value of its probability of misclassifying X . We denote this expectation for the classifier with parameterization θ by $P_e(\mathcal{G}|\theta)$, where*

$$P_e(\mathcal{G}|\theta) \triangleq E_X [P_e(\mathcal{G}(X|\theta))] = \int_{\mathcal{X}} P_e(\mathcal{G}(X|\theta)) \rho_X(X) dX \tag{3.2}$$

Remark: Note that this error rate is the *true* error rate of the classifier, not an estimate. As such, it represents a theoretical number; knowing the number requires knowledge of the feature vector's class-conditional pdfs and its class prior probabilities. Since we do not know these (if we did, we could deterministically create the Bayes-optimal classifier), we cannot know $P_e(\mathcal{G}|\theta)$. However, this error rate is essential to our theoretical arguments, so we ask the reader to imagine that we pass our classifier with the discriminator $\mathcal{G}(X|\theta)$ to an oracle. The oracle knows the probabilistic nature of the feature vector X and can *deterministically* compute for us the value of $P_e(\mathcal{G}|\theta)$, given any and all $\mathcal{G}(X|\theta)$.

Definition 3.2 The Bayes Error Rate: *Recall from section 2.2.2 that $\mathcal{F}(X)_{\text{Bayes}}$ denotes the Bayesian discriminant function of X in any of its possible forms. The Bayes error rate, which we denote by $P_e(\mathcal{F}_{\text{Bayes}})$, is the error rate of the Bayes-optimal classifier (see definition 2.1). By definition, this is the lowest error rate possible for a classifier of X :*

$$P_e(\mathcal{F}_{\text{Bayes}}) \triangleq E_X [P_e(\mathcal{F}(X)_{\text{Bayes}})] = \int_{\mathcal{X}} P_e(\mathcal{F}(X)_{\text{Bayes}}) \rho_X(X) dX \tag{3.3}$$

where

$$P_e(\mathcal{F}(X)_{\text{Bayes}}) = 1 - P_{W|X}(\Gamma(\mathcal{F}(X)_{\text{Bayes}}) | X) \tag{3.4}$$

3.2.1 Learning and Expectation

A differentiable supervised classifier learns by transforming its initial parameterization (which we denote by θ_0) into its post-learning (or *posterior*) parameterization θ . As described in section 2.2.3, this transformation involves adjusting the differentiable supervised classifier's parameters via an iterative search aimed at optimizing an objective function; the learning strategy describes this process.

Definition 3.3 The Learning Strategy Λ : Ultimately, the learning strategy Λ reduces to a description of the mapping from the classifier's initial parameterization θ_0 to its posterior parameterization θ , given the training sample S^n (see definition 3.4) and the hypothesis class $\mathbf{G}(\Theta)$ (see definition 3.5):

$$\begin{aligned}\Lambda &: \theta_0 \rightarrow \theta \\ \theta &= \Lambda(\theta_0 | S^n, \mathbf{G}(\Theta))\end{aligned}\tag{3.5}$$

The classifier's initial parameterization θ_0 is generated according to the prior pdf $\rho_\theta(\theta_0)$ on parameter space Θ . The posterior parameterization θ is stochastic because both θ_0 and S^n are; thus, Λ can also be viewed as an "algorithm" for general Bayesian learning (e.g., [29, sec. 3.5]).

Definition 3.4 The Training Sample S^n : The training sample S^n is the set of n example/class label pairs $\{(\mathbf{X}^1, \mathcal{W}^1), \dots, (\mathbf{X}^n, \mathcal{W}^n)\}$, generated according to the (unknown) joint pdf $\rho_{\mathbf{x}^1, \mathcal{W}^1, \dots, \mathbf{x}^n, \mathcal{W}^n}(S^n)$. Note that if the training example pairs are independent and identically distributed (iid) the joint pdf of the training sample can be expressed as $\prod_{j=1}^n \rho_{\mathbf{x}, \mathcal{W}}(\mathbf{X}^j, \mathcal{W}^j)$.

Definition 3.5 The Hypothesis Class $\mathbf{G}(\Theta)$: In the statistical pattern recognition context, the hypothesis class $\mathbf{G}(\Theta)$ is the set of all possible discriminators $\mathcal{G}(\mathbf{X}|\theta)$:

$$\begin{aligned}\mathbf{G}(\Theta) &\triangleq \{\mathcal{G}(\mathbf{X}|\theta) : \theta \in \Theta\} \\ &= \{\{g_1(\mathbf{X}|\theta), \dots, g_C(\mathbf{X}|\theta)\} : \theta \in \Theta\},\end{aligned}\tag{3.6}$$

where $\mathcal{G}(\mathbf{X}|\theta)$ and $g_i(\mathbf{X}|\theta)$ satisfy the conditions of (2.4) and (2.5) and represent functions in a particular basis or combination of bases $\mathbf{G}$ (e.g., polynomial basis, Gaussian radial basis, logistic functional basis, etc.). We denote the set of all hypothesis classes by $\mathbb{G}$, such that $\mathbf{G}(\Theta) \subset \mathbb{G}$.

Example 3.1 Consider the $\mathcal{C} = 3$ -class pattern recognition task involving the scalar feature x . The classifier with the discriminator $\mathcal{G}(x|\theta) = \{g_1(x|\theta), g_2(x|\theta), g_3(x|\theta)\}$ is used. Each discriminant function $g_i(x|\theta) \in \mathcal{G}(x|\theta)$ is a 10th-order real polynomial function of x :

$$g_i(x|\theta) = \sum_{k=0}^{10} \theta_{i,k} \cdot (x)^k; \quad i = 1, 2, 3\tag{3.7}$$

Thus, the classifier has a total of 33 parameters (a different set of 11 for each of the 3 discriminant functions).⁶ We denote all of these parameters by θ , and parameter space is the 33-dimensional space of real numbers (i.e., $\theta \in \Theta = \mathbb{R}^{33}$). The hypothesis class $\mathbf{G}(\Theta)$ in this example is therefore the set of all discriminators having 3 discriminant functions that are at most 10th-order real polynomials of x . $\mathcal{G}$ in this example is the set of all discriminators having 3 real-valued discriminant functions of the scalar x — an infinitely larger set of possibilities than $\mathbf{G}(\Theta)$.

In assessing how well the classifier/learning strategy will generalize, we must consider not just one error rate $P_e(\mathcal{G}|\theta)$ corresponding to one learning trial involving a single training sample of size n and a single initial parameterization θ_0 ; we must consider the expected error rate over *all* such trials. From this perspective and (3.5), the posterior parameterization θ depends on the classifier's initial parameterization θ_0 , the training sample S^n , the classifier's hypothesis class $\mathbf{G}(\Theta)$, and the learning strategy Λ . As a result, we can express the expected value of the classifier's error rate $P_e(\mathcal{G}|\theta)$ over all possible training samples of size n and all possible initial parameterizations θ_0 . We use the notation $E_z[\cdot]$ to denote the expectation operator taken over the space on which z is defined, and $E_{z, \dots, \zeta}[\cdot]$ to denote the expectation operator taken over the joint space on which $z, \dots, \zeta$ are defined. The expected value of the classifier's error rate raised to the v th power is therefore

$$E_{S^n, \theta_0} [(P_e(\mathcal{G}|\theta))^v] \triangleq \quad (3.8)$$

$$\int_{(\mathcal{X} \times \Omega)^n} \int_{\Theta} \underbrace{\left(P_e \left(\mathcal{G} \mid \underbrace{\theta = \Lambda(\theta_0 | S^n, \mathbf{G}(\Theta))}_{(3.2)} \right) \right)}_{(3.2)}^v \rho_{\theta}(\theta_0) d\theta_0 \rho_{\mathbf{x}^1, \mathbf{w}^1, \dots, \mathbf{x}^n, \mathbf{w}^n}(S^n) dS^n$$

if the training examples are not iid, or

$$\begin{aligned} E_{S^n, \theta_0} [(P_e(\mathcal{G}|\theta))^v] \\ \triangleq \int_{\mathcal{X} \times \Omega} \cdots \int_{\mathcal{X} \times \Omega} \int_{\Theta} \underbrace{\left(P_e \left(\mathcal{G} \mid \underbrace{\theta = \Lambda(\theta_0 | S^n, \mathbf{G}(\Theta))}_{(3.2)} \right) \right)}_{(3.2)}^v \end{aligned}$$

⁶We use the notation $g_i(\mathbf{X}|\theta)$ for the i th discriminant function of the general feature vector $\mathbf{X}$ throughout this text. It is somewhat misleading because it implies that each discriminant function makes use of *all* the classifier's parameters. Although this may be true, it is not necessarily so; in the case of the present example, none of the discriminant functions shares a parameter with any other discriminant function. In other cases (e.g., multi-layer perceptron classifiers) different discriminant functions do share common parameters. We leave these details implicit in the interest of simplified notation.

$$\rho_{\theta}(\theta_0) d\theta_0 \rho_{\mathbf{X}, \mathcal{W}}(\mathbf{X}^1, \mathcal{W}_p^1) d(\mathbf{X}^1, \mathcal{W}_p^1) \dots \rho_{\mathbf{X}, \mathcal{W}}(\mathbf{X}^n, \mathcal{W}_p^n) d(\mathbf{X}^n, \mathcal{W}_p^n) \quad (3.9)$$

if the training examples are iid.

3.2.2 Discriminant Error and the Efficient Classifier

Armed with the expressions in (3.8) — (3.9), we can assess the classifier's *discriminant error* — the degree to which its error rate exceeds the minimum Bayes error rate. We can characterize the expectation of this discriminant error over all learning trials in terms of the traditional notions of an estimator's bias and variance. These metrics allow us to assess how well and how consistently the classifier approximates the Bayes-optimal classifier.

Definition 3.6 Discriminant error: We define the discriminant error as the difference between the classifier's error rate and the Bayes error rate:

$$\text{DError}[\mathcal{G}|\theta] \triangleq P_e(\mathcal{G}|\theta) - P_e(\mathcal{F}_{\text{Bayes}}) \quad (3.10)$$

Remark: Since the Bayes error rate is the minimum achievable, the discriminant error is always non-negative:

$$0 \leq \text{DError}[\mathcal{G}|\theta] \leq 1 - P_e(\mathcal{F}_{\text{Bayes}}) \leq 1 \quad (3.11)$$

Definition 3.7 Discriminant bias: We define the discriminant bias as the expected value of the classifier's discriminant error, using the notation $\text{DBias}[\mathcal{G}|n, \mathbf{G}(\Theta), \Lambda]$ to signify that discriminant bias (as the expressions for discriminant variance and mean-squared discriminant error that follow) ultimately depends on the training sample size n , the hypothesis class $\mathbf{G}(\Theta)$, and the learning strategy⁷ Λ . This dependence is made clear by (3.5) — (3.9):

$$\begin{aligned} \text{DBias}[\mathcal{G}|n, \mathbf{G}(\Theta), \Lambda] &\triangleq E_{S^n, \theta_0}[\text{DError}[\mathcal{G}|\theta]] \\ &= E_{S^n, \theta_0}[P_e(\mathcal{G}|\theta)] - P_e(\mathcal{F}_{\text{Bayes}}) \end{aligned} \quad (3.12)$$

Remark: Since, by (3.11), the discriminant error is always non-negative, the expectation of this error (i.e., the discriminant bias) is always non-negative:

$$0 \leq \text{DBias}[\mathcal{G}|n, \mathbf{G}(\Theta), \Lambda] \leq 1 - P_e(\mathcal{F}_{\text{Bayes}}) \leq 1 \quad (3.13)$$

⁷The dependence on the probabilistic nature of $\mathbf{X}$ is left implicit in order to simplify notation.

Definition 3.8 Discriminant variance: We define the discriminant variance as the second central moment of the classifier's error rate:

$$\begin{aligned} \text{DVar} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda] &\triangleq E_{S^n, \theta_0} \left[\left(P_e(\mathcal{G} | \theta) - E_{S^n, \theta_0} [P_e(\mathcal{G} | \theta)] \right)^2 \right] \\ &= E_{S^n, \theta_0} [P_e(\mathcal{G} | \theta)^2] - \left(E_{S^n, \theta_0} [P_e(\mathcal{G} | \theta)] \right)^2 \end{aligned} \quad (3.14)$$

Definition 3.9 Mean-squared discriminant error (MSDE): We define the mean-squared discriminant error (MSDE) as the expected value of the squared discriminant error:⁸

$$\begin{aligned} \text{MSDE} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda] &\triangleq E_{S^n, \theta_0} \left[(\text{DError} [\mathcal{G} | \theta])^2 \right] \\ &= E_{S^n, \theta_0} [(P_e(\mathcal{G} | \theta) - P_e(\mathcal{F}_{\text{Bayes}}))^2] \\ &= (\text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda])^2 + \text{DVar} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda] \end{aligned} \quad (3.15)$$

Remark: We view the mean-squared discriminant error as a measure of the classifier's ability to generalize well: the lower the MSDE, the better the classifier generalizes. The quantity $(\text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda])^2$ measures how well, on average, the classifier discriminates in comparison to the Bayes-optimal classifier; $\text{DVar} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda]$ measures how consistently the classifier discriminates over multiple independent learning trials.

Definition 3.10 The asymptotically unbiased classifier: The classifier is an asymptotically unbiased estimator of the Bayes-optimal classifier if its discriminant bias is zero for asymptotically large training sample sizes:

$$\lim_{n \rightarrow \infty} \text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda] = 0 \quad (3.16)$$

Definition 3.11 The consistent classifier: The classifier is a consistent estimator of the Bayes-optimal classifier if its error rate $P_e(\mathcal{G} | \theta)$ converges in probability to the Bayes error rate plus some non-negative constant β as the training sample size grows large:

⁸Note that MSDE — the mean-squared difference between the classifier's error rate and the Bayes error rate — is *not* the same thing as MSE — the mean-squared functional error (described in chapter 2 and section 3.4) between the classifier's discriminant functions and the strictly probabilistic form of the BDF.

$$\lim_{n \rightarrow \infty} P \left\{ \underbrace{P_e(\mathcal{G}|\theta) - P_e(\mathcal{F}_{\text{Bayes}})}_{\text{DError}[\mathcal{G}|\theta]} - \beta < \varepsilon \right\} = 1; \quad \varepsilon \geq 0 \quad (3.17)$$

If $P_e(\mathcal{G}|\theta)$ converges in mean-square to the Bayes error rate plus some non-negative constant β

$$\lim_{n \rightarrow \infty} E_{S^n, \theta_0} \left[\left(\underbrace{P_e(\mathcal{G}|\theta) - P_e(\mathcal{F}_{\text{Bayes}})}_{\text{DError}[\mathcal{G}|\theta]} - \beta \right)^2 \right] = 0 \quad (3.18)$$

(3.17) is satisfied,⁹ and the classifier is consistent. Note that (3.18) holds if

$$\lim_{n \rightarrow \infty} \text{DError} \left[\mathcal{G} \mid \underbrace{\theta = \Lambda(\theta_0 | S^n, \mathbf{G}(\Theta))}_{(3.5)} \right] = \beta \quad \forall \{S^n, \theta_0\} \quad (3.19)$$

Definition 3.12 The efficient classifier:

Let $\mathbf{L}$ denote the set of all possible learning strategies and recall that $\mathcal{G}$ denotes the set of all hypothesis classes. The classifier $\mathcal{D}^*(\mathbf{X}) = \Gamma(\mathcal{G}^*(\mathbf{X}|\theta^* = \Lambda_*(\theta_0 | S^n, \mathbf{G}(\Theta)^*)))$ generated from the hypothesis class $\mathbf{G}(\Theta)^* \in \mathcal{G}$ by the learning strategy Λ_* is efficient for a given training sample size n if and only if, given a feature vector $\mathbf{X}$ with specific class-conditional pdfs $\{\rho_{\mathbf{x}|\mathbf{W}}(\mathbf{X}|\omega_1), \dots, \rho_{\mathbf{x}|\mathbf{W}}(\mathbf{X}|\omega_C)\}$ and class prior probabilities $\{P_{\mathbf{W}}(\omega_1), \dots, P_{\mathbf{W}}(\omega_C)\}$, there exists no other classifier in $\mathcal{G}$ that exhibits lower MSDE:

$$\begin{aligned} \mathcal{D}^*(\mathbf{X}) = \Gamma(\mathcal{G}^*(\mathbf{X}|\theta^* = \Lambda_*(\theta_0 | S^n, \mathbf{G}(\Theta)^*))) \text{ is efficient iff} \\ \text{MSDE}[\mathcal{G}^* | n, \mathbf{G}(\Theta)^*, \Lambda_*] \leq \text{MSDE}[\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda] \\ \forall \mathbf{G}(\Theta) \in \mathcal{G}, \forall \Lambda \in \mathbf{L}; \mathcal{G}^*(\mathbf{X}|\theta^*) \in \mathbf{G}(\Theta)^*, \mathcal{G}(\mathbf{X}|\theta) \in \mathbf{G}(\Theta) \end{aligned} \quad (3.20)$$

Remark: The efficient classifier $\mathcal{D}^*(\mathbf{X})$ generalizes best because it exhibits the minimum MSDE. By this definition, the efficient classifier always exists, since there is always some classifier that exhibits lower

⁹Convergence in mean-square guarantees convergence in probability; see, for example, [45, pp. 148-149].

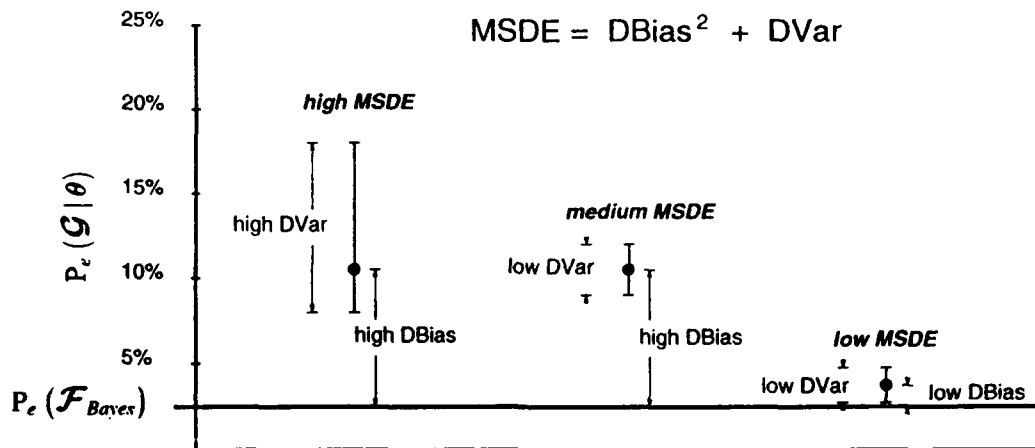


Figure 3.1: The *discriminant bias* and *discriminant variance* of three different classifier paradigms, as determined by an oracle over an infinite number of independent learning trials. The training sample size n is the same finite number for each trial, so each classifier's error rate varies across trials. **Left:** this classifier has high discriminant bias, so on average its error rate is significantly higher than the Bayes error rate $P_e(\mathcal{F}_{Bayes})$. Additionally, its high discriminant variance indicates that its error rate fluctuates substantially across independent trials. As a result, its *mean-squared discriminant error* (MSDE) is high. **Middle:** this classifier has high discriminant bias, so on average its error rate is significantly higher than the Bayes error rate. However, its low discriminant variance indicates that it is a more consistent classifier than the one on the left; as a result, its MSDE is lower and it is preferable to the classifier on the left. **Right:** this classifier has low discriminant bias and low discriminant variance. As a result it yields a consistently good approximation to the Bayes error rate. Its MSDE is therefore low.

MSDE than any other classifier. Readers familiar with the classical definition of the efficient estimator will note that our definition, like that of [39, ch. 5], is less restrictive than that of [107] [22, ch. 32].¹⁰

Example 3.2 For those not familiar with the notion of an efficient estimator, consider the error rates of three different classifiers that have learned to perform the same pattern recognition task. Each classifier therefore represents a different estimator of the Bayes-optimal classifier. This is a thought experiment in which we imagine that the classifiers learn repeatedly over an infinite number of trials. In each trial all three classifiers learn the same training sample of size n (n is finite), and are subsequently tested by the oracle. The training sample for each trial is drawn independent of all other training samples. At the end of each trial, the error rate for each classifier is determined by the oracle and recorded; the results for all trials are compiled. Figure 3.1 summarizes the results for each of the three classifiers. Because the training sample size n is the same finite number for each trial, each classifier's posterior parameterization (and, as a result, its error rate) varies from trial to trial. This variance is depicted by the bars of the whisker plots in figure 3.1. Specifically, the discriminant variance is proportional to the square of the distance between the upper and lower bounds in

¹⁰The classical definition of an efficient estimator requires that it be unbiased and that its variance match the Cramer-Rao bound; by this more rigorous definition, the efficient estimator does not always exist. While the Cramer-Rao bound is clearly defined for the parameter estimation context, it is unclear whether there is an analogous bound in the pattern recognition context. Please refer to section 3.6 for more on this subject.

each whisker plot. The discriminant bias of each classifier is equal to the distance between the mean value of its whisker plot (denoted by the dot) and the horizontal line denoting the Bayes error rate $P_e(\mathcal{F}_{\text{Bayes}})$. The classifier on the left is a poor estimator of the Bayes-optimal classifier because it exhibits both high discriminant bias and high discriminant variance. This means that 1) on average the classifier's error rate is much greater than the Bayes error rate (high discriminant bias), and 2) the classifier's error rate varies significantly across trials (high discriminant variance). As a result, the classifier exhibits high MSDE. The classifier in the middle is a somewhat better estimator of the Bayes-optimal classifier because, although it exhibits the same high discriminant bias as its counterpart on the left, its error rate is more consistent across trials. As a result, it exhibits lower MSDE. The classifier on the right is a good estimator of the Bayes-optimal classifier because it exhibits low discriminant bias and its error rate is consistent across trials. As a result it exhibits low MSDE. Whether or not this classifier is efficient depends on whether it satisfies the constraints of definition 3.12.

Measuring the goodness of an estimator by its mean-squared error has a long history in the estimation theory literature, dating back to Gauss.¹¹ R. A. Fisher, H. Cramer, and C. R. Rao played central roles in further defining the "efficient" estimator as one that exhibits minimum mean-squared error (see, for example, [107, 22, 134, 39]). In fact, it is this body of literature that motivates us to view the classifier's error rate as an estimator of the Bayes error rate. The preceding definitions of discriminant bias, discriminant variance, mean-squared discriminant error, and the efficient classifier follow immediately from such a view. Again, we remind the reader that these definitions are quite different from the definitions of *functional* bias, variance, and mean-squared error typically discussed in the connectionist, machine learning, and statistical pattern recognition literature.

3.2.3 Efficient Learning

Despite the similarities between an efficient estimator and an efficient classifier, there are notable differences. In the classical estimation context, we have a unique parametric model, and the efficient parameter estimator — if it exists — is uniquely specified. In the pattern recognition context, there are an infinite number of hypothesis classes with which to approximate the Bayes-optimal classifier of $\mathbf{X}$. Each hypothesis class constitutes a different parametric model, and some choices will be better than others in terms of the minimum MSDE they can attain for a given training sample size of the random feature vector. Recognizing this, we must acknowledge that our choice of hypothesis class $\mathbf{G}(\Theta)$ might not contain the efficient classifier of definition 3.12. In such a case, we would like the classifier that our learning strategy generates to exhibit the lowest MSDE allowed by the choice of $\mathbf{G}(\Theta)$. This implies a notion of *relative* efficiency (e.g., [91]), which depends in part on whether or not the hypothesis class constitutes a *proper parametric model* of the feature

¹¹C. R. Rao traces the notion of the minimum mean-squared error estimator to a paper that Gauss presented to the Royal Society of Göttingen in 1809 [108, pg. 123].

vector $\mathbf{X}$.

Definition 3.13 The proper parametric model $\mathbf{G}(\Theta, \mathbf{X})_{\text{proper}}$: If the C -class random vector $\mathbf{X}$ has a posteriori class probabilities that are described by the parametric equations

$$P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}) = f_i(\mathbf{X}|\theta^*), \quad i = 1, \dots, C \quad (3.21)$$

and if the hypothesis class $\mathbf{G}(\Theta)$ describes the set of all discriminators $\mathcal{G}(\mathbf{X}|\theta)$ for which

$$g_i(\mathbf{X}|\theta) = f_i(\mathbf{X}|\theta), \quad i = 1, \dots, C \quad (3.22)$$

such that

$$g_i(\mathbf{X}|\theta) = P_{\mathcal{W}|\mathbf{x}}(\omega_i|\mathbf{X}) \quad \forall i \quad \text{if } \theta = \theta^*, \quad (3.23)$$

then $\mathbf{G}(\Theta)$ is the proper parametric model of $\mathbf{X}$, which we denote by $\mathbf{G}(\Theta, \mathbf{X})_{\text{proper}}$. The proper parametric model of $\mathbf{X}$ exists if and only if the a posteriori class probabilities (and the class conditional pdfs) of $\mathbf{X}$ can be described in parametric form.

Remark: If $\mathbf{G}(\Theta, \mathbf{X})_{\text{proper}}$ exists, it is unique, as there is one and only one exact set of parametric expressions for the a posteriori class probabilities of $\mathbf{X}$. This uniqueness assertion rests on the difference between functional identity and functional equivalence. The proper parametric model's discriminant functions are, for the correct choice of parameters θ^* , identical to the a posteriori class probabilities of $\mathbf{X}$, by (3.23). Granted, Bayes rule allows the a posteriori class probabilities of $\mathbf{X}$ to be expressed in terms of its class-conditional pdfs (i.e., its likelihood functions) and class prior probabilities via (2.3). This representation is also unique, as there is one and only one exact set of expressions for the class-conditional pdfs and class prior probabilities of $\mathbf{X}$. Thus, if $\mathbf{G}(\Theta, \mathbf{X})_{\text{proper}}$ exists, it can be expressed in one of two forms: directly, as a "partially parametric" [91, pg. 255] form equating to the a posteriori class probabilities of $\mathbf{X}$, or indirectly, as a fully parametric form equating to the products of class-conditional pdfs and class prior probabilities. Many readers will recognize the fully parametric form of $\mathbf{G}(\Theta, \mathbf{X})_{\text{proper}}$ as the foundation of maximum-likelihood parameter estimation.

Definition 3.14 An improper parametric model: An hypothesis class that is not the proper parametric model of $\mathbf{X}$, as defined above, is an improper parametric model of $\mathbf{X}$ (i.e., $\mathbf{G}(\Theta) \neq \mathbf{G}(\Theta, \mathbf{X})_{\text{proper}}$).

Remark: Readers familiar with parametric discrimination (e.g., [29, 40, 91]) will note that proper parametric models are what White terms "correctly specified" parametric models [140]; improper parametric models are what White terms "misspecified" parametric models. So called "non-parametric" models for statistical pattern recognition are named thus because the models' discriminant functions are not exact expressions of

the feature vector's class-conditional pdfs or *a posteriori* class probabilities. The nomenclature is unfortunate, because it incorrectly implies that such models have no parameters. We take the view that *all* models are parametric, so all differentiable "non-parametric" models that are trained in supervised fashion are, by our definition, improper parametric models.

There is at most one proper parametric model (which can be expressed either fully or partially) versus an infinitely large set of improper parametric models for a given feature vector $\mathbf{X}$. Thus, we are infinitely more likely to choose an improper parametric model of $\mathbf{X}$ absent any information or analysis suggesting the proper parametric model. Kolmogorov's theorem [77] can be interpreted as proving that without strong *a priori* information, $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ — if it exists at all — can be identified only by exhaustive hypothesis testing of all models in $\mathcal{G}$. In practical terms, we might test a few hypothesis classes to see if any of them constitutes $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ with high likelihood. If one of the candidate $\mathbf{G}(\Theta)$ s does, then a specific form of probabilistic learning, tailored to $\mathbf{G}(\Theta, \mathbf{X})_{proper}$, might very well generate the efficient classifier $\mathcal{D}^*(\mathbf{X})$ of definition 3.12 (see section 3.6 and chapter 4). If none of the candidates prove likely to constitute $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ we will want to achieve the best generalization allowed by our choice of improper parametric model, whatever that choice ultimately is. This desire raises the issue of the relatively efficient classifier.

Definition 3.15 The relatively efficient classifier:

The classifier $\mathcal{D}^*(\mathbf{X}) = \Gamma(\mathcal{G}(\mathbf{X}|\theta^* = \Lambda_*(\theta_0|S^n, \mathbf{G}(\Theta))))$ generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ_* is relatively efficient for a given training sample size n if and only if, given a feature vector $\mathbf{X}$ with specific class-conditional pdfs $\{\rho_{\mathbf{x}|\omega_1}(\mathbf{X}|\omega_1), \dots, \rho_{\mathbf{x}|\omega_C}(\mathbf{X}|\omega_C)\}$ and class prior probabilities $\{P_{\mathbf{W}}(\omega_1), \dots, P_{\mathbf{W}}(\omega_C)\}$, there exists no other classifier in $\mathbf{G}(\Theta)$ that exhibits lower MSDE:

$$\begin{aligned} \mathcal{D}^*(\mathbf{X}) = \Gamma(\mathcal{G}(\mathbf{X}|\theta^* = \Lambda_*(\theta_0|S^n, \mathbf{G}(\Theta)))) \text{ is relatively efficient iff} \\ \mathbb{E}[\mathcal{G}|n, \mathbf{G}(\Theta), \Lambda_*] \leq \text{MSDE}[\mathcal{G}|n, \mathbf{G}(\Theta), \Lambda] \quad (3.24) \\ \forall \mathbf{G}(\Theta) \in \mathcal{G}, \forall \Lambda \in \mathcal{L}; \mathcal{G}(\mathbf{X}|\theta^*) \in \mathbf{G}(\Theta), \mathcal{G}(\mathbf{X}|\theta) \in \mathbf{G}(\Theta) \end{aligned}$$

Remark: The relatively efficient classifier $\mathcal{D}^*(\mathbf{X})$ exhibits the lowest MSDE allowed by the choice of hypothesis class. Whether or not it is the efficient classifier of definition 3.12 (i.e., whether or not $\mathcal{D}^*(\mathbf{X}) = \mathcal{D}^*(\mathbf{X})$) depends upon the choice of hypothesis class $\mathbf{G}(\Theta)$. If $\mathbf{G}(\Theta) = \mathbf{G}(\Theta)^*$ then $\mathcal{D}^*(\mathbf{X}) = \mathcal{D}^*(\mathbf{X})$; otherwise, $\mathcal{D}^*(\mathbf{X})$ is the closest approximation to $\mathcal{D}^*(\mathbf{X})$ (as measured by $\text{MSDE}[\mathcal{G}|n, \mathbf{G}(\Theta), \Lambda_*]$) allowed by $\mathbf{G}(\Theta)$.

Definition 3.16 The efficient learning strategy: *The learning strategy Λ_* is efficient if and only if, given a feature vector $\mathbf{X}$ with arbitrary class-conditional pdfs $\{\rho_{\mathbf{x}|\mathbf{w}}(\mathbf{X}|\omega_1), \dots, \rho_{\mathbf{x}|\mathbf{w}}(\mathbf{X}|\omega_c)\}$ and class prior probabilities $\{P_{\mathbf{w}}(\omega_1), \dots, P_{\mathbf{w}}(\omega_c)\}$, any training sample size n , and any proper or improper parametric model $\mathbf{G}(\Theta) \in \mathbb{G}$, Λ_* always generates the relatively efficient classifier of definition 3.15.*

Remark: The efficient learning strategy always guarantees the lowest MSDE allowed by the training sample size and the hypothesis class; the guarantee holds for any and all training sample sizes. Frankly, we doubt that this kind of universally efficient learning strategy exists, for the simple reason that we cannot conceive of one single learning strategy that can produce the relatively efficient classifier for both improper *and* proper parametric models of the feature vector (see section 3.6). Regardless of whether or not the efficient learning strategy exists, a less stringent form of *asymptotically* efficient learning certainly does exist.

Definition 3.17 The asymptotically efficient learning strategy: *The learning strategy $\Lambda_{\rightarrow*}$ is asymptotically efficient if and only if, given a feature vector $\mathbf{X}$ with arbitrary class-conditional pdfs $\{\rho_{\mathbf{x}|\mathbf{w}}(\mathbf{X}|\omega_1), \dots, \rho_{\mathbf{x}|\mathbf{w}}(\mathbf{X}|\omega_c)\}$ and class prior probabilities $\{P_{\mathbf{w}}(\omega_1), \dots, P_{\mathbf{w}}(\omega_c)\}$, an “asymptotically large” training sample size n , and any proper or improper parametric model $\mathbf{G}(\Theta) \in \mathbb{G}$, there exists no other learning strategy that produces a classifier from $\mathbf{G}(\Theta)$ with lower MSDE:*

$\Lambda_{\rightarrow*}$ is asymptotically efficient iff

$$\lim_{n \rightarrow \infty} \text{MSDE}[\mathcal{G}|n, \mathbf{G}(\Theta), \Lambda_{\rightarrow*}] \leq \text{MSDE}[\mathcal{G}|n, \mathbf{G}(\Theta), \Lambda] \quad (3.25)$$

$$\forall \left\langle \{\rho_{\mathbf{x}|\mathbf{w}}(\mathbf{X}|\omega_1), \dots, \rho_{\mathbf{x}|\mathbf{w}}(\mathbf{X}|\omega_c)\}, \{P_{\mathbf{w}}(\omega_1), \dots, P_{\mathbf{w}}(\omega_c)\} \right\rangle, \forall \mathbf{G}(\Theta) \in \mathbb{G},$$

$$\forall \Lambda \in \mathbb{L}; \mathcal{G}(\mathbf{X}|\theta^{\rightarrow*} = \Lambda_{\rightarrow*}(\theta_0|S^n, \mathbf{G}(\Theta))) \in \mathbf{G}(\Theta), \mathcal{G}(\mathbf{X}|\theta) \in \mathbf{G}(\Theta)$$

Remark: There is only one difference between the efficient learning strategy and the asymptotically efficient learning strategy: efficient learning is guaranteed to generate the relatively efficient classifier for large and small training sample sizes; asymptotically efficient learning is guaranteed to generate the relatively efficient classifier for large training sample sizes only.

Characterizing a learning strategy as asymptotically efficient is a strong statement for two reasons:

- First: Asymptotically efficient learning is guaranteed to generate the relatively efficient classifier, given *any* hypothesis class, as long as the training sample size is sufficiently large. Although this might not seem to be a strong statement in absolute terms, it does indicate that asymptotically efficient

learning is preferable to *inefficient* learning. That is, a learning strategy that always generates the relatively efficient classifier for large training sample sizes is preferable to any alternative strategy that usually fails to generate the relatively efficient classifier for *any* training sample size. The relevance of this obvious statement to current machine learning strategies for statistical pattern recognition becomes clear in section 3.4, wherein we prove that probabilistic learning is inefficient.

- Second: Asymptotically efficient learning generates the relatively efficient classifier for small as well as large training sample sizes, given *most* choices of hypothesis class. It fails to generate the relatively efficient classifier for small n only when the hypothesis class is a good approximation of the proper parametric model of $\mathbf{X}$ (i.e., when $\mathbf{G}(\Theta) \cong \mathbf{G}(\Theta, \mathbf{X})_{proper}$ — see section 3.6, chapter 4, and section 8.5).

3.3 Differential Learning is Asymptotically Efficient

Recall that S^n denotes the training sample of size n . By (2.86), the sample average value of the CFM objective function converges to its expected value over all feature vectors as the training sample size grows large:¹²

$$\begin{aligned} \lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{CFM}(S^n | \theta) &= E_{\mathbf{X}} [\text{CFM}(\mathbf{X} | \theta)] \\ &= \int_{\mathbf{X}} \left[\underbrace{\sum_{i=1}^c \sigma[\delta_i(\mathbf{X} | \theta), \psi]}_{\text{CFM}(\mathbf{X} | \theta)} \cdot P_{W|\mathbf{X}}(\omega_i | \mathbf{X}) \right] \rho_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \quad (3.26) \end{aligned}$$

Recall from definition 2.11 and (2.26) that the CFM objective function $\sigma[\delta_i(\mathbf{X} | \theta), \psi]$ becomes a step function as its confidence parameter ψ goes to zero:

$$\lim_{\psi \rightarrow 0^+} \sigma[\delta_i(\mathbf{X} | \theta), \psi] = \begin{cases} 0, & \delta_i(\mathbf{X} | \theta) \leq 0 \\ 1, & \delta_i(\mathbf{X} | \theta) > 0 \end{cases} \quad (3.27)$$

Finally, recall from (2.87) – (2.90) that a positive discriminant differential $\delta_i(\mathbf{X} | \theta)$ indicates that the corresponding i th discriminator output is greater than all other outputs ($\delta_i(\mathbf{X} | \theta) = \delta_{(i)}(\mathbf{X} | \theta) > 0$ iff $g_i(\mathbf{X} | \theta) > g_j(\mathbf{X} | \theta) \forall j \neq i$) such that the class label assigned to $\mathbf{X}$ is $\Gamma(\mathcal{G}(\mathbf{X} | \theta)) = \omega_i = \omega_{(i)}$. Given (3.27), $\text{CFM}(\mathbf{X} | \theta)$ in (3.26) has only one non-zero term, corresponding to class $\Gamma(\mathcal{G}(\mathbf{X} | \theta)) = \omega_{(i)}$.

¹²See appendix B.

Thus, $\text{CFM}(\mathbf{X} | \theta)$ converges to the *a posteriori* probability of class $\omega_{(1)}$ (i.e., the class corresponding to the discriminator's largest output):

$$\begin{aligned} \lim_{\substack{n \rightarrow \infty \\ \psi' \rightarrow 0^+}} \text{CFM}(\mathbf{X} | \theta) &= P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X}) \\ &= P_{\mathbf{W}|\mathbf{X}}(\Gamma(\mathcal{G}(\mathbf{X} | \theta) | \mathbf{X})) \\ &= 1 - \underbrace{P_e(\mathcal{G}(\mathbf{X} | \theta))}_{(3.1)} \end{aligned} \quad (3.28)$$

As a result, the expected value of $\text{CFM}(\mathbf{X} | \theta)$ converges to one minus classifier's error rate:

$$\lim_{\psi' \rightarrow 0^+} E_{\mathbf{X}} [\text{CFM}(\mathbf{X} | \theta)] = 1 - \underbrace{\int_{\mathcal{X}} P_e(\mathcal{G}(\mathbf{X} | \theta)) \rho_{\mathbf{X}}(\mathbf{X}) d\mathbf{X}}_{(3.2)} \quad (3.29)$$

By (3.2) and (3.10),

$$\begin{aligned} \lim_{\psi' \rightarrow 0^+} E_{\mathbf{X}} [\text{CFM}(\mathbf{X} | \theta)] &= 1 - P_e(\mathcal{G} | \theta) \\ &= \underbrace{1 - P_e(\mathcal{F}_{Baves})}_{\text{constant}} - \text{DError}[\mathcal{G} | \theta] \end{aligned} \quad (3.30)$$

Equations (3.26) – (3.30) prove that the CFM objective function (in its step functional limiting form) converges to a constant minus the classifier's discriminant error, given an asymptotically large training sample size. By this result, we state the following theorem:

Theorem 3.1 *In the limit that the CFM objective function becomes a step function, the associated differential learning strategy Λ_{Δ} described in chapter 2 is asymptotically efficient. The asymptotic efficiency of differential learning is independent of both the probabilistic nature of the feature vector and the hypothesis class from which the classifier is generated.*

Assumption 3.1 *We assume that Λ_{Δ} employs a search algorithm (i.e., a numerical optimization procedure) that is guaranteed to find the posterior parameters θ^{Δ} that maximize the sample-average value of the CFM objective function, given $\mathbf{X}$ and the hypothesis class $\mathbf{G}(\Theta)$, regardless of the discriminator's initial parameterization θ_0 . That is, we assume that the search algorithm will not halt in a local maximum.*

Proof : The differential learning strategy Λ_Δ maximizes the CFM objective function for the training sample S^n ; the maximum is found with respect to the classifier's parameters, as described by definitions 2.8 and 2.10. We use $\theta^\Delta = \Lambda_\Delta(\theta_0 | S^n, \mathbf{G}(\Theta))$ to denote the posterior parameterization of the classifier generated by the differential learning strategy, given the step functional form of CFM when $\psi \rightarrow 0^+$, the training sample S^n , the initial parameterization θ_0 , and the hypothesis class $\mathbf{G}(\Theta)$. By (3.26) – (3.30), maximizing the step form of CFM is equivalent to maximizing a constant minus the classifier's discriminant error as the training sample size n grows large. As a result,

$$\begin{aligned} \lim_{\substack{n \rightarrow \infty \\ \psi \rightarrow 0^+}} 1 - P_e(\mathcal{F}_{\text{Bayes}}) - \text{DError}[\mathcal{G} | \theta^\Delta = \Lambda_\Delta(\theta_0 | S^n, \mathbf{G}(\Theta))] \\ \geq 1 - P_e(\mathcal{F}_{\text{Bayes}}) - \text{DError}[\mathcal{G} | \theta] \quad \forall \theta \in \Theta \end{aligned} \quad (3.31)$$

By (3.11), we know that $\text{DError}[\mathcal{G} | \theta]$ is always non-negative, so (3.31) leads to¹³

$$\lim_{\substack{n \rightarrow \infty \\ \psi \rightarrow 0^+}} \left(\text{DError}[\mathcal{G} | \theta^\Delta = \Lambda_\Delta(\theta_0 | S^n, \mathbf{G}(\Theta))] \right)^v = \min_{\theta} (\text{DError}[\mathcal{G} | \theta])^v; \quad v > 0 \quad (3.32)$$

or

$$\lim_{\substack{n \rightarrow \infty \\ \psi \rightarrow 0^+}} \left(\text{DError}[\mathcal{G} | \theta^\Delta = \Lambda_\Delta(\theta_0 | S^n, \mathbf{G}(\Theta))] \right)^v \leq (\text{DError}[\mathcal{G} | \theta])^v \quad \forall \theta \in \Theta; \quad v > 0 \quad (3.33)$$

That is, maximizing the step form of CFM is equivalent to minimizing the classifier's discriminant error for asymptotically large training sample sizes. Clearly, if we minimize the classifier's discriminant error for each trial involving an independently drawn training sample S^n and an independently drawn initial parameterization θ_0 , then we minimize the *expected value* of the classifier's discriminant error over all such trials. By (3.8) — (3.9) and (3.33),

$$\begin{aligned} \lim_{\substack{n \rightarrow \infty \\ \psi \rightarrow 0^+}} E_{S^n, \theta_0} \left[\left(\text{DError}[\mathcal{G} | \theta^\Delta = \Lambda_\Delta(\theta_0 | S^n, \mathbf{G}(\Theta))] \right)^v \right] \\ \leq E_{S^n, \theta_0} \left[(\text{DError}[\mathcal{G} | \theta = \Lambda(\theta_0 | S^n, \mathbf{G}(\Theta))])^v \right] \quad \forall \Lambda \in \mathbf{L}; \quad v > 0 \end{aligned} \quad (3.34)$$

It follows immediately from definitions 3.7 and 3.9 that maximizing the step form of CFM minimizes the classifier's discriminant bias and MSDE for asymptotically large training sample sizes:

¹³The following equations hold for all training sample/initial parameterization combinations (i.e., $\forall \{S^n, \theta_0\}$) owing to the convergence properties of the training sample (see appendix B) and the assumption of theorem 3.1.

$$\lim_{\substack{n \rightarrow \infty \\ \eta^1 \rightarrow 0^+}} \text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda_\Delta] \leq \text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda] \quad \forall \Lambda \in \mathbf{L} \quad (3.35)$$

$$\lim_{\substack{n \rightarrow \infty \\ \eta^1 \rightarrow 0^+}} \text{MSDE} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda_\Delta] \leq \text{MSDE} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda] \quad \forall \Lambda \in \mathbf{L} \quad (3.36)$$

Since (3.36) holds regardless of the choice of hypothesis class $\mathbf{G}(\Theta)$ or the probabilistic nature of $\mathbf{X}$, it is equivalent to (3.25) in definition 3.17. Thus, differential learning via the CFM objective function ($\eta^1 \rightarrow 0^+$) is asymptotically efficient. ■

Remark: The preceding proof is significant because it guarantees that differential learning generates the relatively efficient classifier of definition 3.15 — regardless of $\mathbf{G}(\Theta)$ — as long as the training sample size is sufficiently large. At present, we know of no other learning strategy that provides this guarantee. We emphasize that establishing the asymptotic efficiency of differential learning does *not* refute the existence of some other asymptotically efficient learning strategy Λ_0 with *better* convergence properties:¹⁴ $\text{MSDE} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda_0] < \text{MSDE} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda_\Delta] \quad \forall n \ll \infty$. This is the problem with asymptotically efficient estimators in general: we know they do good things with large sample sizes, but we can't be sure that there isn't another asymptotically efficient estimator that does even better for small sample sizes.

3.3.1 Differential Learning Generates Consistent Classifiers

When $\nu = 1$ in (3.32)

$$\lim_{\substack{n \rightarrow \infty \\ \eta^1 \rightarrow 0^+}} \text{DError} [\mathcal{G} | \theta^\Delta = \Lambda_\Delta(\theta_0 | \mathcal{S}^n, \mathbf{G}(\Theta))] = \underbrace{\min_{\theta} \text{DError} [\mathcal{G} | \theta]}_{\theta}, \quad (3.37)$$

which, owing to the convergence properties of the training sample (see appendix B) and the assumption of theorem 3.1, holds for each and every combination of training sample $\mathcal{S}^n$ and initial parameterization θ_0 . As a result, (3.37) is equivalent to (3.19), and differential learning generates consistent classifiers, by definition 3.11 (page 59).

¹⁴We remind the reader that the asymptotically efficient learning strategy must generate the relatively efficient classifier for large training sample sizes, regardless of the choice of hypothesis class. To be sure, there are probabilistic learning strategies that exhibit better convergence properties than differential learning does when $\mathbf{G}(\Theta) \cong \mathbf{G}(\Theta)$, but these learning strategies provably fail to generate the relatively efficient classifier for the arbitrarily chosen $\mathbf{G}(\Theta)$. This issue is the subject of sections 3.4 and 3.6.

Equation (3.37) also guarantees that the discriminant variance of the differentially generated classifier converges to zero for asymptotically large training sample sizes. As mentioned in the preceding section, the guarantee does not set a bound on the rate of this convergence; it simply states that 1) the convergence takes place, and 2) beyond some lower-bound value of n , the differentially-generated classifier's MSDE is the lowest allowed by the choice of hypothesis class.

The value of this lower-bound on n is determined by the probabilistic nature of $\mathbf{X}$, over which the learning machine has no control, as well as the choice of hypothesis class $\mathbf{G}(\Theta)$, over which the learning machine (or the person controlling it) *does* have control. Generally we have access to training sample sizes that are orders of magnitude too small to ensure convergence of the training sample statistics to their underlying class-conditional probability densities and *a posteriori* class probabilities, so (3.30) fails to hold as an identity (rather it holds only as an approximation), and the classifier's ability to generalize (i.e., its MSDE) is quite sensitive to the complexity of $\mathbf{G}(\Theta)$.

In order to prove our claim that differential learning is efficient (not just asymptotically efficient) for most choices of hypothesis class, we prove its minimum-complexity requirements in section 3.5 and link these to Vapnik and Chervonenkis's seminal work relating model (i.e., hypothesis class) complexity to generalization [137] [136, ch. 6].

3.3.2 A Word Regarding "Agnostic" Learning

In concluding this section, we remind the reader that our definition of asymptotically efficient learning is unconditional, in that it places no restriction on the probabilistic nature of the feature vector or the hypothesis class from which the classifier is generated. Differential learning is an asymptotically efficient *agnostic learning strategy* because it generates the relatively efficient classifier for any and all feature vector/hypothesis class combinations, given a sufficiently large training sample size. The term, "efficient agnostic learning," has been coined recently by Kearns, Schapire, and Sellie [73, 72]. Although our definitions of efficient learning are more universal than theirs (in that definitions 3.16 and 3.17 pertain to not just one, but almost all and all (respectively) feature vector/hypothesis class combinations — cf. [73, sec. 2.4]),¹⁵ it is clear that the two are motivated by a similar philosophical perspective. Furthermore, we acknowledge and share their notion of an agnostic learning strategy as one that places the fewest constraints on the form of the classifier's discriminant functions. Since differential learning, in the limit that the CFM objective function becomes a step function, requires the least restrictive conditions necessary for Bayesian discrimination (stated in definition 2.2), we cannot conceive of a more agnostic learning strategy.

¹⁵Moreover, our definition of efficient learning has its basis in classical estimation theory, whereas Kearns and Shapire's definition has its basis in theoretical computer science. By their definition, learning is efficient if it exhibits polynomial time and sample complexity.

3.4 Discriminant Error Versus Functional Error, and the Inefficiency of Probabilistic Learning

Readers familiar with the connectionist and machine learning literature (e.g., [136, 10, 59, 41, 146]) will note that our definition of *discriminant error* is very different from the definitions of *functional error* typically discussed. Recall that section 2.3 describes the properties of many functional error measures. By the proofs of that section, all of these error measures are associated with the probabilistic learning strategy, which for asymptotically large training sample sizes seeks to minimize some measure of functional error between each discriminator output $g_i(\mathbf{X}|\theta)$ and its corresponding *a posteriori* probability $P_{W|\mathbf{X}}(\omega_i|\mathbf{X})$.

In the preceding section, taking the expectation of the step form of the CFM objective function over all feature vectors showed that CFM constitutes an asymptotically unbiased estimator of one minus the classifier's error rate. Thus, learning by maximizing CFM proved to be asymptotically efficient. We know of no error measure that constitutes an unbiased estimator of the arbitrary differentiable supervised classifier's error rate for the general C -class pattern recognition task; as a result, we know of no probabilistic learning strategy that is asymptotically efficient according to definition 3.17.

Consider the expected value of mean-squared error over all feature vectors, described by (2.70):

$$\lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{MSE}(\mathcal{S}^n|\theta) = E_{\mathbf{X}} [\text{MSE}(\mathbf{X}|\theta)] \quad (3.38)$$

$$\begin{aligned} &= \frac{1}{2} \sum_{i=1}^c \int_{\mathcal{X}} [(g_i(\mathbf{X}|\theta) - 1)^2 \cdot P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) \\ &\quad g_i(\mathbf{X}|\theta)^2 \cdot (1 - P_{W|\mathbf{X}}(\omega_i|\mathbf{X}))] \rho_{\mathbf{X}}(\mathbf{X}) d\mathbf{X} \\ &= \frac{1}{2} \sum_{i=1}^c \left[E_{\mathbf{X}} \left[\underbrace{(g_i(\mathbf{X}|\theta) - P_{W|\mathbf{X}}(\omega_i|\mathbf{X}))^2}_{\text{functional error}} - P_{W|\mathbf{X}}(\omega_i|\mathbf{X})^2 + P_{W|\mathbf{X}}(\omega_i|\mathbf{X}) \right] \right] \\ &= \frac{1}{2} \sum_{i=1}^c \left[E_{\mathbf{X}} \left[\underbrace{(g_i(\mathbf{X}|\theta) - P_{W|\mathbf{X}}(\omega_i|\mathbf{X}))^2}_{\text{squared functional error}} \right] \right. \\ &\quad \left. - \underbrace{E_{\mathbf{X}} [P_{W|\mathbf{X}}(\omega_i|\mathbf{X})^2] + P_W(\omega_i)}_{\text{constant}} \right] \quad (3.39) \\ &\neq \text{DError}[\mathcal{G}|\theta] + \text{some constant} \end{aligned}$$

Expressions for functional bias and variance can be derived from this mean-squared functional error

expression (see, for example, [41, sections 2 & 3]). Learning probabilistically by minimizing the MSE objective function ultimately minimizes the squared *functional* error between the discriminator outputs and their corresponding *a posteriori* class probabilities, as shown in (3.39). Since this measure of functional error is *not* a measure of discriminant error, minimizing it does not guarantee that the classifier's error rate will be minimized.¹⁶ By the same token, functional bias and variance expressions, while very useful in the function approximation context, are not necessarily relevant to the pattern recognition context. This is because they bear no direct relationship to discriminant bias and discriminant variance. In chapter 4 we shall see that classifiers can exhibit very high functional bias and variance, while exhibiting very low discriminant bias and variance. In chapter 5 we shall see that decreasing a classifier's functional error can have the undesirable effect of increasing its discriminant error.

In fact, all error measures we have encountered have the same flaws that mean-squared error has; this is indicated by the expected value of the general error measure over all feature vectors, described by (2.37):

$$\begin{aligned}
 \lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} \text{EM}(S^n | \theta) &= E_X [\text{EM}(X | \theta)] \\
 &= \sum_{i=1}^c \int_{\mathcal{X}} [f(D - g_i(X | \theta)) \cdot P_{W|X}(\omega_i | X) \\
 &\quad + f(g_i(X | \theta) - \neg D) \cdot (1 - P_{W|X}(\omega_i | X))] \rho_X(X) dX \\
 &\neq \text{DError}[\mathcal{G} | \theta] + \text{some constant}
 \end{aligned} \tag{3.40}$$

In short, no error measure we know of is monotonically related to discriminant error for the general C -class pattern recognition task employing the arbitrary differentiable supervised classifier (see section 5.3). Thus, probabilistic learning cannot be asymptotically efficient by the (universal) definition 3.17. It can, however, produce the efficient classifier of X for the special case in which $G(\theta) = G(\theta, X)_{\text{proper}}$: this is the subject of section 3.6.

3.5 Differential Learning Requires the Minimum-Complexity Classifier

By choosing a particular hypothesis class $G(\theta)$ with which to classify X , we reduce our possible choices for modeling the BDF of X to subsets of the four forms of the BDF defined in section 2.2.2:

¹⁶No doubt, some readers will discern a philosophical parallel between our argument against using the MSE objective function for probabilistic learning and earlier arguments against using MSE as a parameter estimation criterion. These earlier arguments are published in the estimation theory literature of the past three decades (see, for example, [71, 108, 74]). Of course, our argument is not against the MSE objective function *per se*, but against error measure objective functions in general, since they engender inefficient learning. Again, we return to the assertion that minimizing a classifier's functional error is not the same as minimizing its discriminant error.

$$\begin{aligned}
\mathbf{G}(\Theta)_{\text{Bayes-Strictly Probabilistic}} &\triangleq \{ \mathcal{G}(X|\theta) : \mathcal{G}(X|\theta) \in \mathbf{F}_{\text{Bayes-Strictly Probabilistic}} \} \\
\mathbf{G}(\Theta)_{\text{Bayes-Probabilistic}} &\triangleq \{ \mathcal{G}(X|\theta) : \mathcal{G}(X|\theta) \in \mathbf{F}_{\text{Bayes-Probabilistic}} \} \\
\mathbf{G}(\Theta)_{\text{Bayes-Strictly Differential}} &\triangleq \{ \mathcal{G}(X|\theta) : \mathcal{G}(X|\theta) \in \mathbf{F}_{\text{Bayes-Strictly Differential}} \} \\
\mathbf{G}(\Theta)_{\text{Bayes-Differential}} &\triangleq \{ \mathcal{G}(X|\theta) : \mathcal{G}(X|\theta) \in \mathbf{F}_{\text{Bayes-Differential}} \} \\
\text{i.e., } \mathbf{G}(\Theta)_{\text{Bayes-Differential}} &= \mathbf{G}(\Theta)_{\text{Bayes}} \subset \mathbf{F}_{\text{Bayes-Differential}} = \mathbf{F}_{\text{Bayes}}
\end{aligned} \tag{3.41}$$

By the definitions of section 2.2.2, these sub-sets of the hypothesis class are related as follows:

$$\begin{aligned}
\mathbf{G}(\Theta)_{\text{Bayes-Strictly Probabilistic}} &\subset \mathbf{G}(\Theta)_{\text{Bayes-Strictly Differential}} \\
&\subset \mathbf{G}(\Theta)_{\text{Bayes-Probabilistic}} \subset \mathbf{G}(\Theta)_{\text{Bayes-Differential}} = \mathbf{G}(\Theta)_{\text{Bayes}} \subset \mathbf{F}_{\text{Bayes}}
\end{aligned} \tag{3.42}$$

All of the subsets of $\mathbf{G}(\Theta)$ in (3.41) and (3.42) may be empty, in which case $\mathbf{G}(\Theta)$ does not contain a Bayes-optimal discriminator of $\mathbf{X}$ (i.e., $\{ \mathcal{G}(X|\theta) : \mathcal{G}(X|\theta) \in \mathbf{F}_{\text{Bayes}} \} = \emptyset$). Regardless of the specific nature of $\mathbf{G}(\Theta)$ and whether or not it contains a Bayes-optimal discriminator of $\mathbf{X}$, it certainly *does* contain a discriminator $\mathcal{G}(X|\theta^{* \rightarrow \infty})$ that exhibits the minimum discriminant error of *any* discriminator in $\mathbf{G}(\Theta)$:

$$\text{DError} [\mathcal{G} | \theta^{* \rightarrow \infty}] \triangleq \min_{\theta} \text{DError} [\mathcal{G} | \theta] \tag{3.43}$$

Recall from definitions 3.11 and 3.15 that if the relatively efficient classifier is consistent, its discriminator will converge to $\mathcal{G}(X|\theta^{* \rightarrow \infty})$ as the training sample size grows asymptotically large. Under this condition, its asymptotic discriminant variance will be zero and its asymptotic MSDE will be given by

$$\lim_{n \rightarrow \infty} \text{MSDE} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda] = \lim_{n \rightarrow \infty} (\text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda])^2 = \underbrace{\left(\text{DError} [\mathcal{G} | \theta^{* \rightarrow \infty}] \right)^2}_{(3.43)} \tag{3.44}$$

Since the differentially-generated classifier is both relatively efficient and consistent, its asymptotic discriminant bias is indeed $\text{DError} [\mathcal{G} | \theta^{* \rightarrow \infty}]$ — the minimum allowed by $\mathbf{G}(\Theta)$.

If differential learning is guaranteed to produce the least biased approximation to the Bayes-optimal classifier allowed by $\mathbf{G}(\Theta)$ for asymptotically large training sample sizes, then we might consider the set of all hypothesis classes in $\mathbf{G}$ for which $\text{DError} [\mathcal{G} | \theta^{* \rightarrow \infty}]$ is equal to some specified value $\text{DBias}_{\text{spec}}$:

$$\mathcal{G}_{spec} \triangleq \left\{ \mathbf{G}(\Theta) : \mathbf{G}(\Theta) \in \mathcal{G} \cap \underbrace{\text{DError}[\mathcal{G} | \theta \rightarrow \infty]}_{(3.43)} = \text{DBias}_{spec} \right\}; \quad (3.45)$$

$$0 \leq \text{DBias}_{spec} \leq 1$$

From $\mathcal{G}_{spec}$ we might wish to choose the hypothesis class with the least functional complexity; this choice would therefore have the least functional complexity necessary to classify $\mathbf{X}$ with the DBias_{spec} level of asymptotic discriminant bias. By making this minimum-complexity choice, we would follow the maxim of Occam's razor: "the simplest model is the best one," [130].

This, of course, presumes that we have defined an acceptable measure of functional complexity. At present, a universally accepted definition remains the subject of debate. For this reason, we employ a general non-specific measure of functional complexity.

The general functional complexity measure $\Upsilon[\cdot]$ The general functional complexity measure $\Upsilon[\cdot]$ is a well-defined real measure $\Upsilon[\mathcal{G}(\mathbf{X}|\theta)] \in \mathbb{R}$ (the larger the measure, the higher the complexity). The notation $\Upsilon_{max}[\mathbf{G}(\Theta)] \in \mathbb{R}$ denotes the upper bound on the complexity of the hypothesis class $\mathbf{G}(\Theta)$:

$$\Upsilon_{max}[\mathbf{G}(\Theta)] \triangleq \max_{\theta} \Upsilon[\mathcal{G}(\mathbf{X}|\theta)] \quad (3.46)$$

Remark: Three well-known complexity measures that satisfy this notion of the general complexity measure are

- The Vapnik-Chervonenkis (VC) dimension $\text{dim}_{VC}(\cdot)$ [137, 136].
- The number of parameters in the hypothesis class (i.e., the dimensionality of parameter space) $\text{dim}(\Theta)$ (e.g., [113, 135]).
- The *effective* number of parameters in the hypothesis class [96, 97].

Example 3.3 The hypothesis class $\mathbf{G}(\Theta)$ described in example 3.1 (page 56) has the following complexity:

- If $\Upsilon_{max}[\mathbf{G}(\Theta)] \triangleq \sum_{i=1}^C \text{dim}_{VC}(g_i(\mathbf{X}|\theta))$, then the complexity measure is equal to the sum of the VC dimension $\text{dim}_{VC}(g_i(\mathbf{X}|\theta))$ for each of the discriminator's C discriminant functions.¹⁷ For

¹⁷See [137], [136, ch. 6], and [100, ch. 2.3] for detailed descriptions of the VC dimension and its computation. To the uninitiated, we recommend the lovely summary by Abu-Mostafa [1]. Strictly speaking, the VC dimension pertains to a 2-class pattern recognition task. The more general C -class task is viewed as C 2-class tasks in which each of the C discriminant functions $\mathcal{G}(\mathbf{X}|\theta) = \{g_1(\mathbf{X}|\theta), \dots, g_C(\mathbf{X}|\theta)\}$ maps the feature vector to the binary classification ω_i or $-\omega_i$ (read, "not class i "). The VC dimension is computed for each of the C discriminant functions separately; the complexity measure does *not* sum across discriminant functions, at least not for the purpose of estimating training sample sizes necessary for good generalization [136, ch. 6]. In this sense, our using the sum of each discriminant function's VC dimension ($\sum_{i=1}^C \text{dim}_{VC}(g_i(\mathbf{X}|\theta))$) as the over-all complexity measure for the hypothesis class is not consistent with the original intent of the measure. We simply use this sum as a convenient way to express the overall complexity of the hypothesis class with a single number.

the real polynomial discriminant functions described by (3.7), the VC dimension of each discriminant function is one plus the maximum number of real roots in the function's polynomial. Thus,

$$\Upsilon_{\max} [\mathbf{G}(\Theta)] = 3 \cdot (10 + 1) = 33.$$

- If $\Upsilon_{\max} [\mathbf{G}(\Theta)] \triangleq \dim(\Theta)$, then the complexity measure is simply the total number of parameters in $\mathbf{G}(\Theta)$, which in this particular case is equal to the sum of the VC dimension for each discriminant function: $\dim(\Theta) = \sum_{i=1}^C \dim_{\text{VC}} (g_i(\mathbf{X}|\theta)) = 3 \cdot (10 + 1) = 33$.

Given one's preferred measure of functional complexity, there is some minimum-complexity hypothesis class $\mathbf{G}(\Theta \downarrow) \in \mathcal{G}_{\text{spec}}$ that contains a discriminator with minimum discriminant error $\min_{\theta \downarrow} \text{DError} [\mathcal{G}|\theta \downarrow] = \text{DBias}_{\text{spec}}$. Since the differentially-generated classifier, given the hypothesis class $\mathbf{G}(\Theta \downarrow)$, exhibits $\min_{\theta \downarrow} \text{DError} [\mathcal{G}|\theta \downarrow]$ for asymptotically large training sample sizes, the following corollary to theorem 3.1 holds:

Corollary 3.1 *Differential learning requires the hypothesis class with the least functional complexity necessary to approximate the Bayes-optimal classifier with specified precision. The precision of the approximation is measured in terms of asymptotic discriminant bias (i.e., the discriminant bias of definition 3.7, given an asymptotically large training sample size $n \rightarrow \infty$).*

Proof : The proof follows from theorem 3.1 (page 67) and the preceding set-theoretic argument. ■

Remark: The corollary simply states that differential learning requires the least complex model necessary for the data. This is generally not the case for probabilistic learning, owing to the relationships of (3.42). As we shall see empirically in chapter 4 and theoretically in chapter 6, it generally requires substantially more functional complexity to ensure that $\mathbf{G}(\Theta)_{\text{Bayes-Strictly Probabilistic}}$ and $\mathbf{G}(\Theta)_{\text{Bayes-Probabilistic}}$ are not empty sets than it does to ensure that $\mathbf{G}(\Theta)_{\text{Bayes-Differential}}$ is not an empty set.

Corollary 3.1 says *nothing* about differential learning's role in *determining* the minimum-complexity hypothesis class $\mathbf{G}(\Theta \downarrow)$ sufficient for Bayesian discrimination. By Kolmogorov's theorem [77] there is no algorithm short of exhaustive search for determining it *a priori*. Instead we must restrict ourselves to a particular *a priori* choice of hypothesis class $\mathbf{G}(\Theta)$ — an educated guess, which we *believe* contains a Bayes-optimal classifier of $\mathbf{X}$. In making this choice of $\mathbf{G}(\Theta)$, we must weigh its discriminant bias for large sample sizes against its discriminant variance for small sample sizes — the ubiquitous bias-variance tradeoff that every estimator exhibits. A high-complexity hypothesis class will ensure low asymptotic discriminant bias at the cost of high discriminant variance for small training sample sizes. A low-complexity hypothesis class will ensure low discriminant variance for small training sample sizes, but will it exhibit low discriminant bias (i.e., a low error rate)? Differential learning guarantees that it will exhibit the lowest

possible error rate for large training sample sizes (theorem 3.1); this in turn guarantees that we can attain a specific error rate (greater than or equal to the Bayes error rate) with the least complex hypothesis class necessary (corollary 3.1). Since discriminant variance increases with the complexity of the hypothesis class under VC analysis, theorem 3.1 and corollary 3.1 assure us that we can achieve both low discriminant bias and low discriminant variance by pairing a low-complexity hypothesis class with differential learning.

Because the VC dimension is a complexity measure that satisfies the general characteristics described above, corollary 3.1 asserts that differential learning requires the hypothesis class $G(\Theta)$ with the smallest VC dimension necessary to approximate the Bayes-optimal classifier with a specified level of precision. Consequently, differential learning allows us, under VC analysis [137] [136, ch. 6], to minimize the probability that the classifier's worst-case failure to generalize (i.e., the worst case deviation of the classifier's empirical training sample error rate from its true error rate) will exceed an unacceptable level. This level is specified by ϵ in [137, Theorem 2]. It is worth noting that this minimal worst-case bound does not stem from the efficiency of differential learning; it stems solely from the minimum complexity requirements of differential learning. Thus, there are two ostensibly independent mechanisms by which the differentially generated classifier generalizes well: the efficiency of the learning strategy itself and its minimum-complexity requirements.

The only question that remains is whether there is ever a learning strategy with better convergence properties than differential learning (Λ_Δ), given a *particular* low-complexity hypothesis class. That is, under what conditions might there be a specific hypothesis class $G(\Theta)$, learning strategy Λ , and feature vector X combination for which

$$\begin{aligned} \text{MSDE} [G | n, G(\Theta), \Lambda] &< \text{MSDE} [G | n, G(\Theta), \Lambda_\Delta], \quad n \ll \infty \\ &\& \\ \lim_{n \rightarrow \infty} \text{MSDE} [G | n, G(\Theta), \Lambda] &= \text{MSDE} [G | n, G(\Theta), \Lambda_\Delta] \end{aligned} \quad (3.47)$$

We know of only one such case: if $G(\Theta)$ is a proper parametric model of X and Λ is a maximum-likelihood probabilistic learning strategy, (3.47) can hold.

3.6 The Case for Probabilistic Learning

At the end of chapter 2 the reader might have wondered why differential learning would be desirable. By this point, the reader might wonder just the opposite: why use probabilistic learning if it is inefficient? Although we argue in favor of differential learning in most cases, we believe there are at least three scenarios under which probabilistic learning can be preferable:

- Probabilistic learning is preferable when one specifically wants to estimate the *a posteriori* class probabilities of X . By using probabilistic learning, we are obliged to choose a hypothesis class with

sufficient functional complexity to approximate the *a posteriori* class probabilities of $\mathbf{X}$ with high precision. This, in turn, generally dictates very large training sample sizes for robust probabilistic estimates (see chapter 6).

- Probabilistic learning might be desirable when one class is *always* more likely than any other, regardless of $\mathbf{X}$. As an example, there might be no combination of clinical factors for which the probability of death exceeds the probability of surviving coronary bypass surgery, yet there will be combinations of clinical factors for which the probability of death is *relatively* high. A physician counselling bypass surgery candidates might therefore want a robust probability-of-mortality estimate for each patient. Again, if probabilities must be estimated, we are obliged to satisfy the complexity and training sample size requirements of probabilistic learning.
- Probabilistic learning might be preferable if the hypothesis class $\mathbf{G}(\Theta)$ is a proper parametric model of $\mathbf{X}$ (see definition 3.13).

The first two scenarios involve a subjective choice; the third scenario does not. Rather it stems from the existence of the proper parametric model.

The proper parametric model $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ employing a consistent form of probabilistic learning clearly satisfies the condition of (3.47) — namely that its asymptotic MSDE when generated probabilistically is equal to its asymptotic MSDE when generated differentially:

$$\lim_{n \rightarrow \infty} \text{MSDE} [\mathcal{G} | n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_P] = \text{MSDE} [\mathcal{G} | n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_\Delta] \quad (3.48)$$

As we mentioned earlier in this chapter, a rigorous proof that (3.47) holds when $\mathbf{G}(\Theta) = \mathbf{G}(\Theta, \mathbf{X})_{proper}$ is beyond both our interest and stamina. Instead, we merely hypothesize why (3.47) holds, sketch a proof, and describe one particular case (subsequently illustrated in chapter 4) for which the proof holds.

Hypothesis 3.1 *If*

1. the proper parametric model $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ of $\mathbf{X}$ exists, and
2. maximum-likelihood estimators of its parameters exist (which we obtain via the maximum-likelihood probabilistic learning strategy Λ_{P-ML}), and
3. the variance of each maximum-likelihood parameter estimator matches the Cramer-Rao bound [107] [22, ch's. 32-33],

then the classifier $\Gamma(\mathcal{G}(\mathbf{X}|\theta_{ML}))$ generated from $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ by the maximum-likelihood probabilistic learning procedure Λ_{P-ML} is the relatively efficient classifier of $\mathbf{X}$ (definition 3.15) for all training sample sizes (i.e., $\forall n$). If, in addition, $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ is the minimum-complexity hypothesis class containing a

Bayes-optimal classifier of $\mathbf{X}$, then $\Gamma(\mathcal{G}(\mathbf{X}|\theta_{ML}))$ constitutes the efficient classifier of $\mathbf{X}$ (definition 3.12) for all training sample sizes.

Sketch of Proof : By [107] [22, ch's. 32-33], the maximum-likelihood estimate θ_{ML} of the parameter vector θ^* in (3.21) exhibits the lowest possible variance of any estimator of θ^* for any training sample size n . Thus θ_{ML} converges to θ^* with probability one faster than any other estimator of θ^* . This fastest convergence implies that $\text{DBias}[\mathcal{G}|n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_{P-ML}]$ converges to zero with probability one faster than it does for any other estimator of the Bayes-optimal classifier obtained from $\mathbf{G}(\Theta, \mathbf{X})_{proper}$. By the definitions of section 3.2, the maximum-likelihood probabilistic learning strategy Λ_{P-ML} therefore generates the relatively efficient classifier of $\mathbf{X}$ from $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ for all training sample sizes:

$$\begin{aligned} \text{MSDE}[\mathcal{G}|n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_{P-ML}] &\leq \text{MSDE}[\mathcal{G}|n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda] \\ \forall n, \forall \Lambda \in \mathbf{L} \end{aligned} \quad (3.49)$$

If, in addition, $\Upsilon_{\max}[\mathbf{G}(\Theta, \mathbf{X})_{proper}]$ is the lowest complexity of any hypothesis class containing a Bayes-optimal classifier of $\mathbf{X}$, then Λ_{P-ML} generates the efficient classifier of $\mathbf{X}$ from $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ for all training sample sizes:

$$\begin{aligned} \text{MSDE}[\mathcal{G}|n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_{P-ML}] &\leq \text{MSDE}[\mathcal{G}|n, \mathbf{G}(\Theta)' , \Lambda] \\ \forall n, \forall \Lambda \in \mathbf{L}, \forall \mathbf{G}(\Theta)' \in \mathcal{G} \end{aligned} \quad (3.50)$$

□

Remark: Hypothesis 3.1 is not a theorem, and the preceding argument is not a proof, owing to the lack of rigor on two points:

- First, it is not clear (although it may seem intuitively sensible) that fastest convergence in the discriminator's parameters necessarily guarantees fastest convergence in the classifier's error rate. It is possible to make this linkage for *specific* proper parametric models (see section 3.6.1), but it remains unclear whether there exists a proof of it for the *general* proper parametric model satisfying the requirements of hypothesis 3.1.
- Second, it is easy to conceive of a feature vector $\mathbf{X}$ for which the proper parametric model exist, but for which the proper parametric model is *not* the minimum-complexity hypothesis class containing a Bayes-optimal discriminator of $\mathbf{X}$. In such a case, it might be possible to prove that the classifier generated differentially from the minimum-complexity improper parametric model containing a Bayes-optimal discriminator of $\mathbf{X}$ is more efficient than the one generated from the proper parametric model

(i.e., that the improper parametric model's MSDE converges faster than that of the more complex proper parametric model).

Example 3.4 The normal-based linear discriminant (i.e., Gaussian maximum-likelihood) and logistic regression models are probably the best-known example of a hypothesis class/learning strategy combination (fully-parametric and partially parametric, respectively) that generates the efficient classifier of $\mathbf{X}$ when $\mathbf{X}$ has homoscedastic Gaussian class-conditional pdfs with the additional nice properties described in appendix F.¹⁸ Readers familiar with both traditional logistic regression and neural network models will recognize that the C -output multi-layer perceptron with logistic nonlinearities and no hidden layer units is the C -class logistic regression model; this partially-parametric model and its associated fully-parametric Normal-based linear discriminant model (e.g., [91, ch. 3]) are described in section 4.2 and appendix F. Maximum-likelihood probabilistic learning for the partially-parametric model takes the form of minimizing a Kullback-Leibler information distance expression; for the fully-parametric model it takes the form of minimizing a mean-squared error expression (see appendix F). The fully-parametric variant is somewhat more efficient than the partially-parametric variant [30] and constitutes the efficient classifier of $\mathbf{X}$ when $\mathbf{X}$ is the Gaussian feature vector described above (see section 4.2).

3.6.1 Assessing the Asymptotic Relative Efficiency (ARE) of a non-Differential Learning Strategy

Equation (3.47) and hypothesis 3.1 acknowledge the possibility that differential learning (Λ_Δ) is not the only asymptotically efficient learning strategy for hypothesis classes that are proper parametric models of the feature vector (or close approximations thereto).

Definition 3.18 *Asymptotic relative efficiency:* Given the hypothesis class $\mathbf{G}(\Theta)$ and two learning strategies Λ and Λ' , we define the asymptotic relative efficiency (ARE) of Λ' with respect to Λ as the ratio of the MSDE expressions

$$\text{ARE}_{n \rightarrow \infty}[\Lambda', \Lambda | \mathbf{G}(\Theta)] \triangleq \frac{\text{MSDE}[\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda']}{\text{MSDE}[\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda]} \quad (3.51)$$

Remark: This ratio is a generalization of the efficiency ratios for 1) the general estimator [22, sec. 32.3], and 2) fully-parametric and partially-parametric classifier's of the Gaussian feature vector [30] (see sections 4.2 and F.3). The definition focuses on classifiers that differ only in the learning strategy employed; a

¹⁸Homoscedastic pdfs all have the same covariance matrix. Under a simple linear transformation, a feature vector with homoscedastic Gaussian class-conditional pdfs has class-conditional covariance matrices that are all of the form $\sigma \cdot \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix.

simple generalization of (3.51) allows a comparison of classifiers that differ in terms of their hypothesis classes as well. If the ARE can be expressed in closed form for the proper parametric model $\mathbf{G}(\Theta, \mathbf{X})_{proper}$, the differential learning strategy Λ_{Δ} , and the maximum-likelihood learning strategy Λ_{P-ML} , it tells us which strategy learns the Bayes-optimal classifier faster (i.e., with fewer training examples). The answer lies in the ratio of MSDE for the two learning strategies, expressed in terms of n as n grows large. If $\text{ARE}_{n \rightarrow \infty}[\Lambda_{P-ML}, \Lambda_{\Delta} | \mathbf{G}(\Theta, \mathbf{X})_{proper}] < 1$ the probabilistically-generated (i.e., maximum-likelihood) proper parametric model is more efficient than its differentially-generated counterpart for finite training sample sizes, and hypothesis 3.1 is substantiated for the given proper parametric model.

The notions of definition 3.18 and (3.51) require some explanation. By theorem 3.1 it seems that $\text{ARE}_{n \rightarrow \infty}[\Lambda_{P-ML}, \Lambda_{\Delta} | \mathbf{G}(\Theta, \mathbf{X})_{proper}]$ should never be less than unity, since we have proven that differential learning generates the relatively efficient classifier for asymptotically large training sample sizes (recall (3.36)). The existence of cases in which $\text{ARE}_{n \rightarrow \infty}[\Lambda_{P-ML}, \Lambda_{\Delta} | \mathbf{G}(\Theta, \mathbf{X})_{proper}] < 1$ would seem to refute theorem 3.1 by contradiction of (3.36), but it does not. The explanation lies in the difference between a limit and the rate at which an expression converges to that limit; it is best expressed by a simple example.

Example 3.5 Let us assume that there is a feature vector $\mathbf{X}$ for which the proper parametric model $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ exists. Furthermore, let us assume that the following MSDE expressions hold:

$$\begin{aligned} \text{MSDE}[\mathcal{G} | n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_{\Delta}] &= n^{-3.1} \\ \text{MSDE}[\mathcal{G} | n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_{P-ML}] &= n^{-3.3} \end{aligned} \quad (3.52)$$

Equation (3.52) guarantees that the asymptotic MSDE exhibited by both learning strategies is

$$\lim_{n \rightarrow \infty} \text{MSDE}[\mathcal{G} | n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_{P-ML}] = \lim_{n \rightarrow \infty} \text{MSDE}[\mathcal{G} | n, \mathbf{G}(\Theta, \mathbf{X})_{proper}, \Lambda_{\Delta}] = 0, \quad (3.53)$$

even though the ARE of the maximum-likelihood learning strategy Λ_{P-ML} is much less than unity:

$$\text{ARE}_{n \rightarrow \infty}[\Lambda_{P-ML}, \Lambda_{\Delta} | \mathbf{G}(\Theta, \mathbf{X})_{proper}] = n^{-\frac{1}{3}} \ll 1 \quad (3.54)$$

That is, both learning strategies generate the Bayes-optimal classifier of $\mathbf{X}$, but the probabilistically-generated classifier converges to the Bayes-optimal $\mathcal{O}[n^{-\frac{1}{3}}]$ faster than its differentially-generated counterpart.

We remind the reader that even if the hypothesis class is a proper parametric model, the differentially generated classifier will generalize as well as the probabilistically-generated (maximum-likelihood) classifier, as long as the training sample size is very large. If (3.51) can be evaluated for a particular $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ and Λ_{P-ML} , then we have a means of evaluating whether or not probabilistic learning is preferable (i.e., whether or not hypothesis 3.1 is substantiated for $\mathbf{G}(\Theta, \mathbf{X})_{proper}$ and Λ_{P-ML}) when the training sample size is small.

3.6.2 A Word Regarding “Proper Models”

The term “proper model” probably originates with Dawes [26]. From the statistical pattern recognition perspective, he uses the term to describe a parametric model of the feature vector, the parameters of which are learned (i.e., estimated) by a “proper” — by which he means probabilistically motivated — method. As mentioned above, logistic regression is an example of a proper model when the feature vector has homoscedastic Gaussian class-conditional pdfs with the additional nice properties described in appendix F. The model parameters are learned by the “proper” method of maximum-likelihood. Dawes characterizes an improper model as one for which the parameters are learned by some “improper” method (i.e., one that is not probabilistically defensible). He argues both eloquently and persuasively that improper models often yield better pattern classifiers than proper ones.

We agree. Indeed, we submit that the proofs of this and the preceding chapter explain the phenomenon: so-called “proper” probabilistic learning strategies are *inefficient*, yielding Bayesian discrimination only if the parametric model with which they are paired is a proper one for the feature vector. Dawes’ “improper” parameter estimation strategies are superior when the parametric model is improper by our definition because they are more efficient, generating classifiers that exhibit lower MSDE than their probabilistically-generated counterparts. The irony is that probabilistic learning strategies aren’t always the best ones to employ. Indeed, it motivates us to restrict the adjectives “proper” and “improper” to the parametric model (i.e., hypothesis class) *alone*. In our view there are no proper or improper learning strategies, only efficient and inefficient ones.

3.7 Summary

In this chapter we have defined the efficient classifier, the relatively efficient classifier, the efficient learning strategy, and the asymptotically efficient learning strategy. We have defined the mean-squared *discriminant* error of a classifier, and motivated its use as a measure of generalization — how closely and how consistently the classifier approximates the Bayes-optimal classifier’s minimum error rate. We have proven that the differential learning strategy is asymptotically efficient, guaranteeing the best approximation to the Bayes-optimal classifier allowed by one’s *a priori* choice of hypothesis class when the training sample size is asymptotically large. Moreover, we have proven that differential learning requires the minimum classifier complexity necessary for Bayesian discrimination. We have proven that probabilistic learning strategies are usually inefficient, failing to generate the best approximation to the Bayes-optimal classifier for all but possibly one choice of hypothesis class, regardless of the training sample size. Furthermore, probabilistic learning generally requires more than the minimum classifier complexity necessary for Bayesian discrimination.

Viewing all differentiable supervised classifiers as parametric models of the feature vector, we have distinguished between proper and improper parametric models. We have shown that if the proper parametric

model of the feature vector exists, it is possible for probabilistic learning to generate a better approximation to the Bayes-optimal classifier than differential learning can when the training sample size is small. Given the explicit desire to estimate the *a posteriori* probabilities of the feature vector or a reasonable likelihood that the hypothesis class constitutes a proper parametric model — as determined by traditional hypothesis testing procedures (e.g., see [140]) — probabilistic learning is the preferred strategy. Absent these, differential learning is the best strategic choice, provably requiring the simplest model of the data and the smallest training sample size necessary for good generalization.

Chapter 4

The Robust Beauty of Differentially-Generated Improper Parametric Models¹

Outline

We analyze two “toy” problems in order to make the theoretical arguments of chapters 2 and 3 more tangible. We begin with a familiar 2-class Gaussian pattern recognition task; we illustrate that the probabilistically-generated classifier can be more efficient than its differentially-generated counterpart for small training sample sizes if the hypothesis class is the proper parametric model of the feature vector. We contrast that task with a simple 3-class pattern recognition task in order to illustrate that differential learning is asymptotically efficient and requires the minimum-complexity classifier necessary for Bayesian discrimination. The analysis confirms that differential learning generates the relatively efficient classifier for small as well as large training sample sizes when the hypothesis class is not a proper parametric model of the feature vector. Probabilistic learning fails to generate the relatively efficient classifier — regardless of the training sample size — when the hypothesis class is an improper parametric model.

4.1 Introduction

The purpose of this chapter is to illustrate the theoretical points of chapters 2 and 3 so that the reader will gain an intuitive appreciation of differential learning — a more tangible understanding of the arguments we have made so far. We analyze two “toy” problems in order to illustrate

- the asymptotically efficient, minimum-complexity nature of differential learning,
- the generally inefficient, high-complexity nature of probabilistic learning, and

¹The title of this chapter is inspired by R. M. Dawes’ paper *The Robust Beauty of Improper Linear Models* [26]. This chapter is a revised and extended version of work first published in [52].

- the special circumstances under which probabilistic learning can be efficient.

We illustrate these characteristics by contrasting a pattern recognition task for which the chosen parametric model is proper with a task for which the chosen parametric model is improper.

We begin with a familiar 2-class pattern recognition task for which the single feature is a homoscedastic, Gaussian-distributed random variable; we learn to classify this random feature with its proper parametric model in both full and partial forms, showing that probabilistically-generated variants are more efficient than the differentially-generated variant when the training sample size is small. All learning strategies generate equally efficient classifiers from the proper parametric hypothesis class as the training sample size grows large. We contrast this result with a simple 3-class pattern recognition task for which the single feature is a heteroscedastic, uniformly-distributed random variable; we learn to classify this random feature with a polynomial classifier, which is an improper parametric model, showing that differentially-generated variants are always more efficient than their probabilistically-generated counterparts for both small and large training sample sizes. The 3-class task also illustrates the minimal complexity requirements of differential learning.

Both of these illustrative tasks lend themselves to closed-form analysis, so the classifiers' learning/classification characteristics can be derived for asymptotically large training sample sizes. We analyze the classifiers' characteristics for small training sample sizes via simulations. In effect, we play the role of an oracle like the one described in section 3.2. Each classifier/learning strategy that we analyze learns repeatedly over ten independent trials. In each trial all the different classifier/learning strategies learn the same training sample of size n ; the training sample for each trial is drawn independent of all other training samples. We compute the *true* error rate for each classifier at the end of each trial, using the exact expressions for the feature's *a posteriori* class probabilities and class prior probabilities; we compile the results for all trials. Each classifier's posterior parameterization (and, as a result, its error rate) varies from trial to trial when the training sample size n is finite. When n is infinite, each classifier's error rate is a constant (i.e., its discriminant variance is zero), owing to the convergence properties of the random feature (see appendix B) and the consistency of each classifier.² We obtain approximate values of each classifier's discriminant bias, discriminant variance, and mean-squared-discriminant error (MSDE) by computing the sample mean and sample variance of the classifier's error rate across the ten trials for each training sample size. Thus, the only difference between the experimental protocols of this chapter and the protocol for the oracle in example 3.2 (page 61) is that we run a finite — as opposed to an infinite — number of independent learning/test trials for each classifier.

We emphasize the differences between differential and probabilistic learning strategies by contrasting the results of these two experiments. In the process of evaluating the 2-class proper parametric model, we demonstrate experimental results that are consistent with Efron's theoretical comparison of the logistic and

²All of the learning strategies we employ lead to consistent classifiers (see definition 3.11). The differentially-generated classifier is proven to be consistent in section 3.3.1; the probabilistic consistency proofs proceed along the lines of the differential proof; we leave the details to the interested reader.

normal-based linear discriminant analysis paradigms [30]. More importantly, our experiments reflect the theoretical findings of chapters 2 and 3: differential learning *always* generates the most efficient classifier allowed by the hypothesis class, given a sufficiently large training sample; probabilistic learning, in contrast, fails to generate the most efficient classifier allowed — regardless of the training sample size — unless the hypothesis class is a proper parametric model of the data.

4.2 Analysis of a Proper Parametric Model

Figure 4.1 illustrates a two-class scalar x with homoscedastic³ Gaussian class-conditional pdfs for classes ω_1 and ω_2 . There is one class boundary ($\mathcal{B}_{1,2 \text{ Bayes}} = 0$) for the Bayes-optimal classifier of x . The class-conditional pdfs of x are given by

$$\begin{aligned} p_{x|W}(x|\omega_1) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x - \mu_1)^2\right] \\ p_{x|W}(x|\omega_2) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x - \mu_2)^2\right]; \\ \mu_1 &= -1.65, \mu_2 = 1.65, \sigma^2 = 1 \end{aligned} \quad (4.1)$$

The class prior probabilities are $P_W(\omega_1) = P_W(\omega_2) = \frac{1}{2}$. The *a posteriori* class probabilities of x are

$$\begin{aligned} P_{W|x}(\omega_1|x) &= [1 + \exp\left[-\frac{1}{2\sigma^2}(2x(\mu_1 - \mu_2) - \mu_1^2 + \mu_2^2)\right]]^{-1} \\ P_{W|x}(\omega_2|x) &= 1 - P_{W|x}(\omega_1|x) \\ &= [1 + \exp\left[-\frac{1}{2\sigma^2}(2x(\mu_2 - \mu_1) + \mu_1^2 - \mu_2^2)\right]]^{-1}; \\ \mu_1 &= -1.65, \mu_2 = 1.65, \sigma^2 = 1 \end{aligned} \quad (4.2)$$

Given the values of μ_1 , μ_2 , and σ^2 , the Bayes error rate is 4.9% when the following classification strategy is employed:

$$\begin{aligned} x &\leq \mathcal{B}_{1,2 \text{ Bayes}}, & \text{choose } \omega_1 \\ x &> \mathcal{B}_{1,2 \text{ Bayes}}, & \text{choose } \omega_2 \end{aligned} ; \quad \mathcal{B}_{1,2 \text{ Bayes}} = 0 \quad (4.3)$$

4.2.1 The Proper Parametric Model

We employ two forms of the same “logistic linear” classifier⁴ to learn the Bayes-optimal classifier of x . The first form is the fully-parametric proper model; the second is the partially-parametric proper model.

³Recall that homoscedastic pdfs all have the same variance parameter (or covariance matrix).

⁴See section 7.2.2.

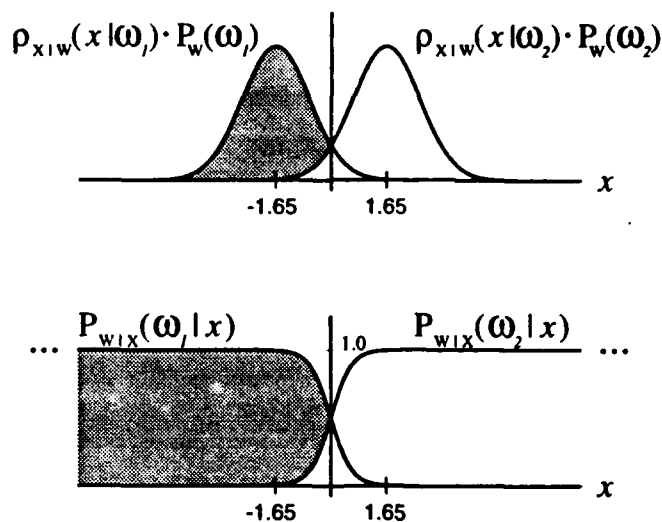


Figure 4.1: A two-class scalar feature discrimination task. The single feature is a homoscedastic, Gaussian-distributed random variable. **Top:** class-conditional density - class prior products $\rho_{x|w}(x|\omega_i) \cdot P_w(\omega_i)$; those for class ω_1 are shown in dark gray; those for class ω_2 are shown in white; the region of overlap is shown in light gray. **Bottom:** the *a posteriori* class probabilities $P_{w|x}(\omega_1|x)$ and $P_{w|x}(\omega_2|x)$.

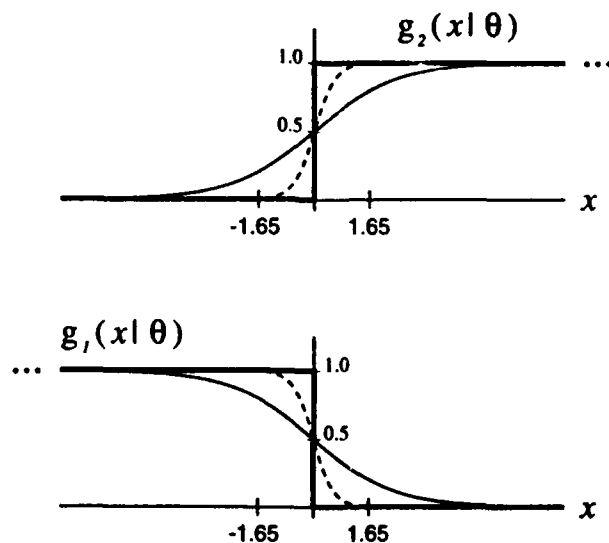


Figure 4.2: The proper parametric model of x . The logistic linear hypothesis class follows from both the partially-parametric and fully-parametric proper models of x . Discriminant functions are shown for three parameterizations that yield Bayes-optimal discrimination of x . The dashed line denotes the parameterization by which the discriminant functions are identically the *a posteriori* class probabilities of x . The solid lines denote two different parameterizations by which the discriminant functions partition feature space in the Bayes-optimal fashion; note that neither of these parameterizations yields discriminant functions that are identically the *a posteriori* class probabilities of x .

The **fully-parametric proper model** describes x in terms of its class-conditional pdfs; its parameters model the unknown class-conditional means and the unknown variance parameter of x :

$$\begin{aligned}
 g_1(x|\theta) &= \frac{\rho_{x|W_1}(x|\omega_1, \tilde{\mu}_1, \tilde{\sigma}^2)}{\rho_{x|W_1}(x|\omega_1, \tilde{\mu}_1, \tilde{\sigma}^2) + \rho_{x|W_2}(x|\omega_2, \tilde{\mu}_2, \tilde{\sigma}^2)} \\
 &= [1 + \exp[\frac{1}{\tilde{\sigma}^2}(\mu_2 - \mu_1)x + \frac{1}{2\tilde{\sigma}^2}(\mu_1^2 - \mu_2^2)]]^{-1} \\
 g_2(x|\theta) &= 1 - g_1(x|\theta) \\
 &= \frac{\rho_{x|W_2}(x|\omega_2, \tilde{\mu}_2, \tilde{\sigma}^2)}{\rho_{x|W_1}(x|\omega_1, \tilde{\mu}_1, \tilde{\sigma}^2) + \rho_{x|W_2}(x|\omega_2, \tilde{\mu}_2, \tilde{\sigma}^2)} \\
 &= [1 + \exp[-\frac{1}{\tilde{\sigma}^2}(\mu_2 - \mu_1)x - \frac{1}{2\tilde{\sigma}^2}(\mu_1^2 - \mu_2^2)]]^{-1}
 \end{aligned} \tag{4.4}$$

where

$$\rho_{x|W_i}(x|\omega_i, \tilde{\mu}_i, \tilde{\sigma}^2) = \frac{1}{\sqrt{2\pi\tilde{\sigma}^2}} \exp\left[-\frac{1}{2\tilde{\sigma}^2}(x - \tilde{\mu}_i)^2\right] \tag{4.5}$$

The parameter vector θ in $g_i(x|\theta)$ above denotes the three parameters $\{\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\sigma}^2\}$.

The classifier described by (4.4) is depicted in figure 4.2. It is, by definitions 3.13 and 3.14, a proper parametric model of x because the discriminant functions of (4.4) are identically equal to the *a posteriori* class probabilities of (4.2) when $\tilde{\mu}_1 = \mu_1$, $\tilde{\mu}_2 = \mu_2$, and $\tilde{\sigma}^2 = \sigma^2$. More specifically, (4.4) describes the *fully-parametric* proper model of x , generally called the *normal-based linear discriminant analysis* paradigm in the statistical pattern recognition literature (e.g., [91]). We denote this model with the initials "ML" since the model learns by the method of maximum-likelihood, described in detail for the general homoscedastic Gaussian feature vector in section F.1. The resulting maximum-likelihood parameters are

$$\begin{aligned}
 \tilde{\mu}_i &= \frac{1}{n_i} \sum_{j=1}^n \tau_i(\langle x^j, W^j \rangle) \cdot x^j \\
 \tilde{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^c \sum_{j=1}^n \tau_i(\langle x^j, W^j \rangle) \cdot (x^j - \tilde{\mu}_i)^2,
 \end{aligned} \tag{4.6}$$

where x^j denotes the j th example of x (as opposed to the j th power of x , which we denote by $(x)^j$). Note that

$$\tau_i(\langle x^j, \mathcal{W}^j \rangle) = \begin{cases} 1, & \mathcal{W}^j = \omega_i \\ 0, & \text{otherwise} \end{cases}, \quad (4.7)$$

and n_i denotes the number of examples having the class label ω_i in the training sample of size n .

The class boundary $\mathcal{B}_{1,2 ML}$ formed by the fully-parametric (ML) model is the value for which $g_1(\mathcal{B}_{1,2 ML} | \theta) = g_2(\mathcal{B}_{1,2 ML} | \theta) = \frac{1}{2}$; this occurs at

$$\mathcal{B}_{1,2 \text{ Fully-Parametric}} = \frac{\tilde{\mu}_1 + \tilde{\mu}_2}{2} \quad (4.8)$$

The partially-parametric proper model describes x in terms of its *a posteriori* class probabilities; its parameters model the unknown parameters of these probabilities. Given the following definitions

$$\begin{aligned} \alpha &\triangleq \frac{\mu_2 - \mu_1}{\sigma^2} \\ \beta &\triangleq \frac{\mu_1^2 - \mu_2^2}{2\sigma^2}, \end{aligned} \quad (4.9)$$

the *a posteriori* class probabilities in (4.2) can be written as follows:

$$\begin{aligned} P_{\mathcal{W}|x}(\omega_1 | x) &= [1 + \exp[\alpha x + \beta]]^{-1} \\ P_{\mathcal{W}|x}(\omega_2 | x) &= [1 + \exp[-\alpha x - \beta]]^{-1} \end{aligned} \quad (4.10)$$

The partially-parametric proper model of x is given by

$$\begin{aligned} g_1(x | \theta) &= [1 + \exp[\theta_{1,1} x + \theta_{1,0}]]^{-1} \\ g_2(x | \theta) &= [1 + \exp[-\theta_{1,1} x - \theta_{1,0}]]^{-1}, \end{aligned} \quad (4.11)$$

where $\theta = \{\theta_{1,0}, \theta_{1,1}\}$. It is, by definitions 3.13 and 3.14, a proper parametric model of x because the discriminant functions of (4.11) are identically equal to the *a posteriori* class probabilities of (4.2) and (4.10) when $\theta_{1,1} = \alpha = \frac{\mu_2 - \mu_1}{\sigma^2}$ and $\theta_{1,0} = \beta = \frac{\mu_1^2 - \mu_2^2}{2\sigma^2}$. More specifically, (4.4) describes the *partially-parametric* proper model of x , generally called the *logistic discriminant analysis* (or *logistic regression*) paradigm in the statistical pattern recognition literature (e.g., [91]). The model learns by the method of maximum-likelihood, described in detail for the general homoscedastic Gaussian feature vector in section F.2. The resulting maximum-likelihood parameters cannot be expressed in closed form. However, section F.2 proves that these parameters are obtained by minimizing the Kullback-Leibler information distance (CE [82, 81], see section 2.3.2) between the discriminant functions and their corresponding empirical *a posteriori*

class probabilities over the domain of x . The proof has been worked previously by Akaike and White [2, 140, 141] and by Hjort [65]. We denote the partially-parametric model generated probabilistically via the maximum-likelihood/Kullback-Leibler information distance learning procedure with the initials "CE".

The class boundary $\mathcal{B}_{1,2 CE}$ formed by the partially-parametric model is the value for which $g_1(\mathcal{B}_{1,2 CE} | \theta) = g_2(\mathcal{B}_{1,2 CE} | \theta) = \frac{1}{2}$; this occurs at

$$\mathcal{B}_{1,2 \text{ Partially-Parametric}} = \frac{-\theta_{1,0}}{\theta_{1,1}} \quad (4.12)$$

4.2.2 Probabilistic Learning for the Asymptotically Large Training Sample

For the asymptotically large training sample size (i.e., $n \rightarrow \infty$), the maximum-likelihood parameters of the fully-parametric (ML) proper model are

$$\begin{aligned} \widetilde{\mu}_1 &= \mu_1 \\ \lim_{n \rightarrow \infty} \widetilde{\mu}_2 &= \mu_2 \\ \widetilde{\sigma}^2 &= \sigma^2 \end{aligned} \quad (4.13)$$

(see section F.1). By (4.2) and (4.8), $\lim_{n \rightarrow \infty} \mathcal{B}_{1,2 ML} = \mathcal{B}_{1,2 \text{ Bayes}} = 0$, and the ML classifier exhibits Bayesian discrimination.

For asymptotically large training sample sizes, the maximum-likelihood parameters of the partially-parametric model are given by

$$\begin{aligned} \theta_{1,1} &= \alpha = \frac{\mu_2 - \mu_1}{\sigma^2} \\ \lim_{n \rightarrow \infty} \theta_{1,0} &= \beta = \frac{\mu_1^2 - \mu_2^2}{2\sigma^2} \end{aligned} \quad (4.14)$$

(see section F.2). By (4.2) and (4.12), $\lim_{n \rightarrow \infty} \mathcal{B}_{1,2 CE} = \mathcal{B}_{1,2 \text{ Bayes}} = 0$, and the CE-generated partially-parametric proper model exhibits Bayesian discrimination.

Since the partially-parametric proper model constitutes a differentiable supervised classifier (definition 2.8, page 25), it can be generated with *any* error measure, not just CE. We denote the general error measure by "EM". We denote the training sample of size n by S^n , and we denote a particular unique value (or *pattern*) of x by x_p . If there are P unique patterns in S^n , and for each of these patterns there are $n_{p,i}$ examples belonging to class ω_i , the sample EM is given by⁵

⁵Please see section 2.3.1 for specific constraints on the forms of $f(D - g_i(x|\theta))$ and $f(g_i(x|\theta) - \neg D)$.

$$\text{EM}(\mathcal{S}^n | \theta) = \sum_{i=1}^{C=2} \sum_{p=1}^P \frac{n_p}{n} \left[\frac{n_{p,i}}{n_p} \cdot f(D - g_i(\mathbf{X}_p | \theta)) + \frac{(n_p - n_{p,i})}{n_p} \cdot f(g_i(\mathbf{X}_p | \theta) - \neg D) \right];$$

$$\sum_{p=1}^P n_{p,i} = n$$
(4.15)

From section 2.3.1, we know that the classifier's EM can be expressed by the following expectation as the training sample size grows asymptotically large:

$$\mathbb{E}_x [\text{EM}(x | \theta)] = \int_{\mathcal{X}} \left[\sum_{i=1}^{C=2} [f(D - g_i(x | \theta)) \cdot P_{\mathcal{W}|x}(\omega_i | x) + f(g_i(x | \theta) - \neg D) \cdot (1 - P_{\mathcal{W}|x}(\omega_i | x))] \right] \rho_{\mathcal{X}}(x) dx \quad (4.16)$$

The parameterization θ^* that minimizes the classifier's EM for the asymptotically large training sample size can be found by substituting the discriminant function expressions of (4.11) into (4.16), deriving the expression for the gradient $\nabla_{\theta} (\mathbb{E}_x [\text{EM}(x | \theta^*)])$, setting this gradient equal to the zero vector, and solving the resulting normal equations (see section 2.3.1). Since the partially-parametric model is proper, and since the general error measure is, by definition (see section 2.3.1), minimal when $g_i(x | \theta) = P_{\mathcal{W}|x}(\omega_i | x) \quad \forall i$, the general error measure generates the parameters of (4.14) for asymptotically large training sample sizes. By (4.12), the general error measure therefore generates the Bayes-optimal classifier from the partially-parametric model, given an asymptotically large training sample; that is, $\lim_{n \rightarrow \infty} \mathcal{B}_{1,2 \text{ EM}} = \mathcal{B}_{1,2 \text{ Bayes}} = 0$, and the EM-generated partially-parametric proper model exhibits Bayesian discrimination.

4.2.3 Differential Learning via CFM for the Asymptotically Large Training Sample

The partially-parametric proper model can also learn differentially. Differential learning is implemented by maximizing the classification figure-of-merit (CFM) objective function described in section 2.2.4, chapter 5, and appendix D. The procedure is virtually the same as probabilistic learning, except for the change in objective function. From section 2.4, the sample CFM is given by

$$\text{CFM}(\mathcal{S}^n | \theta) = \sum_{i=1}^{C=2} \sum_{p=1}^P \frac{n_p}{n} \left[\sigma[\delta_i(x_p | \theta), \psi] \cdot \frac{n_{p,i}}{n_p} \right]; \quad \sum_{p=1}^P n_{p,i} = n \quad (4.17)$$

In this $C = 2$ -class case, the discriminant differentials⁶ are given by

$$\begin{aligned}\delta_1(x_p | \theta) &= 2g_1(x_p | \theta) - 1 \\ \delta_2(x_p | \theta) &= -\delta_1(x_p | \theta) \\ &= 1 - 2g_1(x_p | \theta),\end{aligned}\tag{4.18}$$

and $g_1(x | \theta)$ is given by (4.11). Details of the CFM confidence parameter ψ' are given in appendix D. The classifier's CFM can be expressed by the following expectation as the training sample size grows asymptotically large (section 2.4):

$$E_x [\text{CFM}(x | \theta)] = \sum_{i=1}^{C=2} \int_{-\infty}^{\infty} \sigma[\delta_i(x | \theta), \psi'] P_{W|x}(\omega_i | x) \rho_X(x) dx \tag{4.19}$$

Section 2.4 proves that when the classifier is differentially-generated, the CFM objective function ($\lim_{\psi' \rightarrow 0^+}$) is maximized when the top-ranked discriminant differential (i.e., $\delta_{(1)}(x | \theta)$) corresponds to the most likely class of x over the feature's domain:⁷ mathematically,

$$\begin{aligned}\lim_{\substack{n \rightarrow \infty \\ \psi' \rightarrow 0^+}} E_x [\text{CFM}(x | \theta^*)] \text{ in (4.19) is maximized if } \theta^* \text{ is such that} \\ \delta_{(1)}(x | \theta^*) = \delta_*(x | \theta^*); \quad P_{W|x}(\omega_* | x) \geq \max_{k \neq *} P_{W|x}(\omega_k | x)\end{aligned}\tag{4.20}$$

Of course (4.20) holds only if the classifier has sufficient functional complexity to yield Bayesian discrimination. The logistic linear classifier does, so it maximizes CFM, satisfies (4.20), and constitutes the Bayes-optimal classifier when its parameters satisfy the following constraints:

$$\begin{aligned}\theta_{1,0 \text{ CFM}} &= 0 \\ \theta_{1,1 \text{ CFM}} &< 0\end{aligned}\tag{4.21}$$

When these conditions are satisfied, the resulting class boundary equals the Bayes-optimal boundary by (4.12):

$$\lim_{\substack{n \rightarrow \infty \\ \psi' \rightarrow 0^+}} \mathcal{B}_{1,2 \text{ CFM}} = \mathcal{B}_{1,2 \text{ Bayes}} = 0 \tag{4.22}$$

⁶Recall definition 2.7 and section 2.4.

⁷We remind the reader that the logistic linear classifier for the $C = 2$ -class pattern recognition task has only one discriminant function $g_1(x | \theta)$; we create a phantom second discriminant function $g_2(x | \theta) = 1 - g_1(x | \theta)$ for the purpose of computing and using the discriminant differentials $\delta_1(x | \theta)$ and $\delta_2(x | \theta)$. This is an artifice by which differential learning is extended to the single-output, differentiable supervised classifier.

4.2.4 Results of Differential and Probabilistic Learning for Asymptotically Large and Small Training Samples

A word regarding error rates: Recall from definition 3.1 (page 55) that the **true** error rate $P_e(\mathcal{G}|\theta)$ for the classifier of x is given by

$$P_e(\mathcal{G}|\theta) \triangleq E_x[P_e(\mathcal{G}(x|\theta))] = \int_{\mathcal{X}} P_e(\mathcal{G}(x|\theta)) \rho_x(x) dx \quad (4.23)$$

where

$$\begin{aligned} P_e(\mathcal{G}(X|\theta)) &\triangleq 1 - P_{W|x}(\mathcal{D}(x|\theta) | x) \\ &= 1 - P_{W|x}(\Gamma(\mathcal{G}(x|\theta)) | x) \end{aligned} \quad (4.24)$$

The error rates that we quote in this chapter — for both the proper and improper parametric models — are computed according to (4.23), since we play the role of an oracle, we know the probabilistic nature of the feature x , and the associated integrals are tractable. Specific details of the error rate computations for this section and section 4.3.4 are given in appendix G.

Figure 4.3 displays the empirical distribution of the error rates for four logistic linear classifiers of x , based on ten independent learning/testing trials. Statistics are shown for the fully-parametric model and the partially-parametric model; the latter employs two forms of probabilistic learning (via the MSE and CE error measure objective functions) as well as differential learning (via the CFM objective function). Results for differential learning via CFM are shown in white; results for probabilistic learning via ML (fully-parametric model) and CE and MSE (partially-parametric model) are shown in gray. The results are shown in box-plot [131, ch. 2] statistical summaries. In brief, the box of each plot has vertical extrema that match the first and third quartiles of the sample data; the horizontal line dividing the box delineates the median of the sample data; the inner and (if shown) outer “T”-shaped “fences” of each plot depict the nominal lower bound of the first quartile and nominal upper bound of the fourth quartile. Extreme values in the first and fourth quartiles falling beyond the outer fence(s) are plotted as dots (see appendix C for details of the box plot statistical summary). All results for finite training sample sizes are based on 10 independent trials for the specified training sample size (all classifiers learn the same training sample in a given trial, for a given sample size). Learning for the fully-parametric proper model is a simple maximum-likelihood parameter estimation procedure, the computations of which are specified by (4.6). For the partially-parametric model, learning takes the form of a steepest descent (MSE, CE, etc.) or steepest ascent (CFM) search over parameter space, using a modified form of the backpropagation algorithm (e.g., [119, 120]); learning begins from a tabula

rasa state in which all parameters are initialized randomly according to a uniform distribution on the closed interval $[-.3, .3]$. All trials are completely automated, so learning is done without any human intervention. All experimental conditions (except, of course, the objective function used) are identical for differential and probabilistic learning. Classifier parameterizations for the asymptotically large training sample size are derived as described in sections 4.2.2 and 4.2.3.

The box plots of figure 4.3 are empirical analogs to the whisker plots of figure 3.1. That is, they give us approximations of the discriminant bias, discriminant variance, and mean-squared discriminant error (MSDE) for each classifier/learning strategy. They show that the fully-parametric proper model is the most efficient estimator of the Bayes-optimal classifier for small (10) and medium (100) training sample sizes. The CFM-generated partially-parametric model is the least efficient for small training sample sizes (note the one trial for which the CFM-generated classifier's error rate is 27%, depicted as a dot in the figure). As the training sample size goes to 100 examples, all the partially-parametric models are roughly comparable, although significantly less efficient than the fully-parametric model. When the training sample size increases to 1000, all the classifiers appear comparable. These box plots are noteworthy for two reasons: first, they demonstrate that the differentially-generated classifier is indeed asymptotically efficient (as the training sample size increases beyond 1000 examples, the differentially-generated model is as good as any of the others); second, they demonstrate that the probabilistically-generated proper parametric models are the most efficient classifiers for small training sample sizes.

This second finding is consistent with our theoretical description of the circumstances under which probabilistic learning is more efficient than differential learning for small training sample sizes (section 3.6). An analysis of a single 10-example learning trial lends a qualitative dimension to the theoretical description of the phenomenon. Figure 4.4 shows the empirical class-conditional pdfs — empirical class prior probability products of a 10-example random sample of x ; they are shown in histogram form, superimposed on the true class-conditional pdfs — class prior probability products of x . There are five examples of each class; all the examples of ω_1 fall to left of $x = -.75$; all the examples of ω_2 fall to right of $x = 1.4$. As a result, there is an interval $[-.75, 1.4]$ inside which there are no training examples. Nevertheless, the partially-parametric proper model generated probabilistically from these ten training examples (using the CE objective function) forms a class boundary $B_{1,2 CE}$ very close to the Bayes-optimal boundary $B_{1,2 Bayes} = 0$. The model's logistic linear discriminant functions and the partitioning of feature space they produce are shown in figure 4.5; they exhibit a 5.1% error rate — a good approximation to the Bayes error rate of 4.9%, despite the small training sample size and the lack of any examples on the interval $[-.75, 1.4]$. Because it reduces to estimating the parameters of a model that is known to be proper in this particular case, probabilistic learning via CE is efficient even for small training sample sizes. The lack of training examples in the vicinity of $B_{1,2 Bayes}$ is of little consequence because *all* training examples contain information about the class-conditional means μ_1 and μ_2 and the variance parameter σ^2 of the partially-parametric proper model;

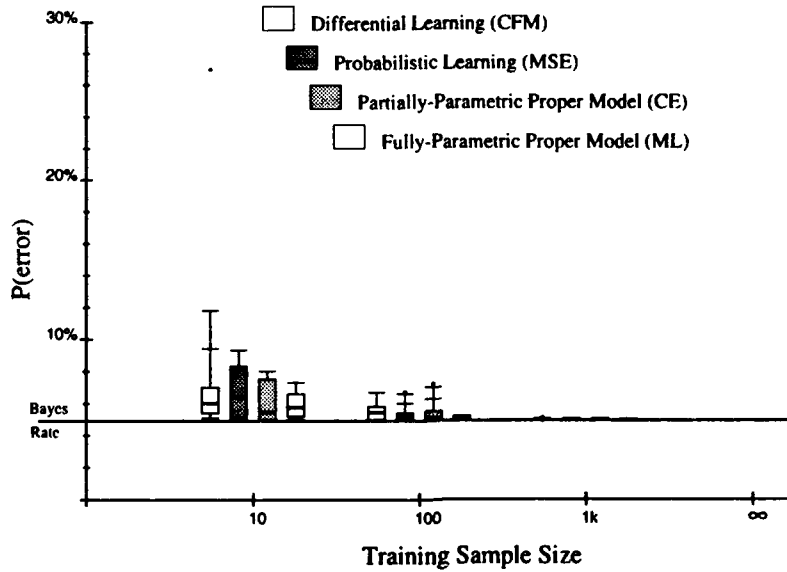


Figure 4.3: A comparison of error rates for differentially (CFM) and probabilistically (MSE, CE, and ML) generated logistic linear classifiers. Results for the differentially generated classifier are shown in white; those for the probabilistically generated classifiers are shown in gray (MSE = dark gray, CE = medium gray, ML = light gray). Note that the ML-generated logistic linear classifier is the fully-parametric proper model of x ; the CE-generated logistic linear classifier is the partially-parametric proper model of x .

that is, good estimates of these parameters are possible even when the training sample contains no examples near $\mathcal{B}_{1,2 \text{ Bayes}}$. Differential learning, on the other hand, does *not* exploit the proper nature of the logistic linear discriminant functions; it views the discriminant functions in a completely “agnostic” way, employing them in *any* manner that classifies the training sample correctly. Since any partially-parametric proper model with the parameters

$$\begin{aligned} -0.75 &\leq \theta_{1,0 \text{ CFM}} \leq 1.4 \\ \theta_{1,1 \text{ CFM}} &< 0 \end{aligned} \quad (4.25)$$

will classify the training sample without error, there is a wide choice of maximum-CFM parameterizations for the classifier. The discriminant functions of the CFM-generated partially-parametric proper model, given these 10 training examples, are shown in figure 4.6. Note that their partitioning of feature space deviates significantly from the Bayes-optimal partitioning; they exhibit a 10.1% error rate — a poor approximation to the Bayes error rate of 4.9%.

These results illustrate a simple fact: if there is a proper model for the data, probabilistic learning will generate the efficient classifier from it by exploiting the proper nature of the model. Indeed, it can be argued (via the Cramer-Rao notion of efficient parameter estimation [107] [22, ch’s. 32-33]) that

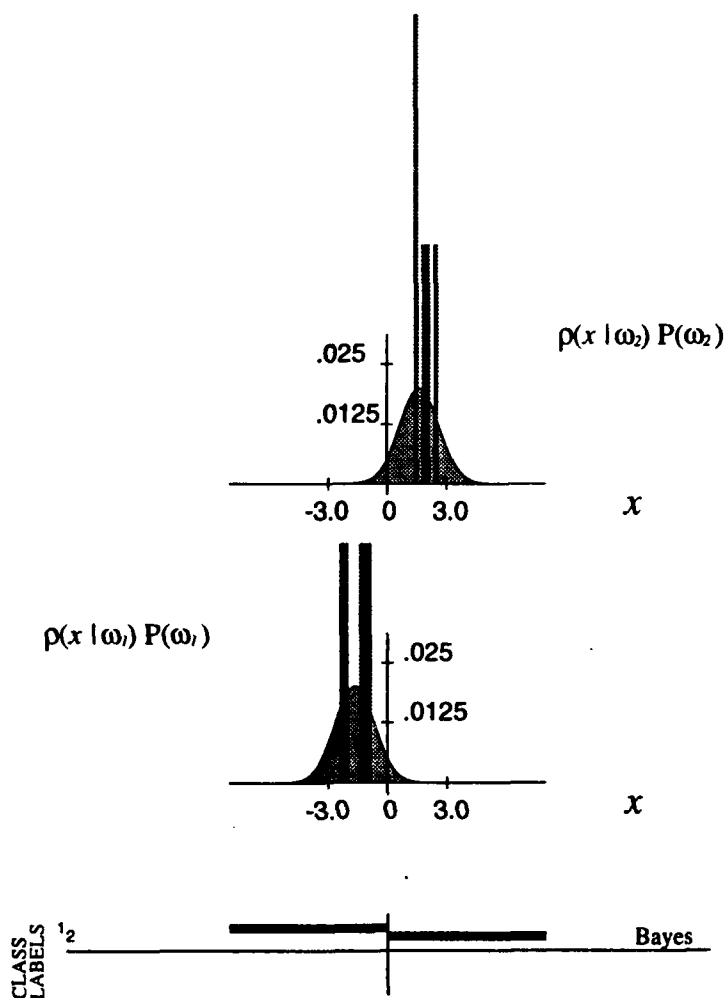


Figure 4.4: The empirical class-conditional pdfs of x multiplied by their empirical class prior probabilities for a training sample size of 10 examples; they are shown in histogram form (dark gray), superimposed on the true class-conditional pdf — class prior probability products (lighter gray). There are five examples for each of the two classes. The bar-graph below the class-conditional pdfs depicts the Bayes-optimal partitioning of feature space; the class boundary occurs where the bar-graph shifts from class ω_1 to class ω_2 .

the probabilistically-generated proper parametric model exploits *all* of the Fisher information (e.g., [2])⁸ contained in the training sample, whereas differential learning by its very nature does not. We stress that the Fisher information content of the training sample pertains to the unknown parameters of the model; it does *not* pertain specifically to the Bayes-optimal class boundaries on feature space. Thus, the information is useful (i.e., it is valid information) for pattern recognition purposes only if the model is indeed proper.

Efron has proven that the fully-parametric proper model is the most efficient classifier of the 2-class feature vector with homoscedastic Gaussian pdfs; the partially-parametric model is somewhat less efficient

⁸See [28, sec. 7.8] for an concise, readable discussion of Fisher information and its relationship to the Cramer-Rao bound.

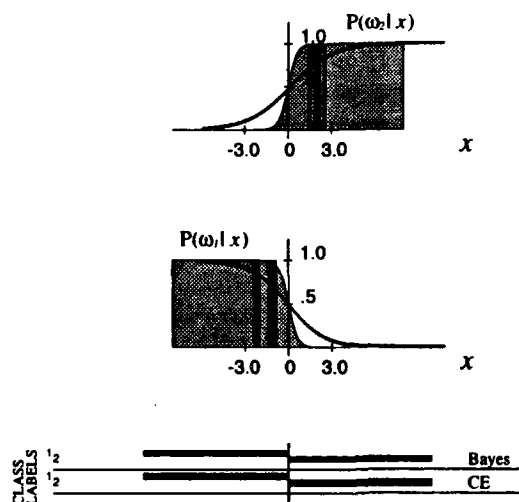


Figure 4.5: The empirical *a posteriori* class probabilities of x for the 10 examples of figure 4.4; they are shown in histogram form (dark gray), superimposed on the true probabilities (lighter gray), which are shown only for the finite interval $-10 \leq x \leq 10$. Histograms take a default value of zero for regions on the domain of x where no training samples occur. The discriminant functions of the CE-generated logistic linear classifier (i.e., the partially-parametric model of x) are superimposed in black. Note that the CE-generated classifier's partitioning of feature space is a close approximation to the Bayes-optimal partitioning.

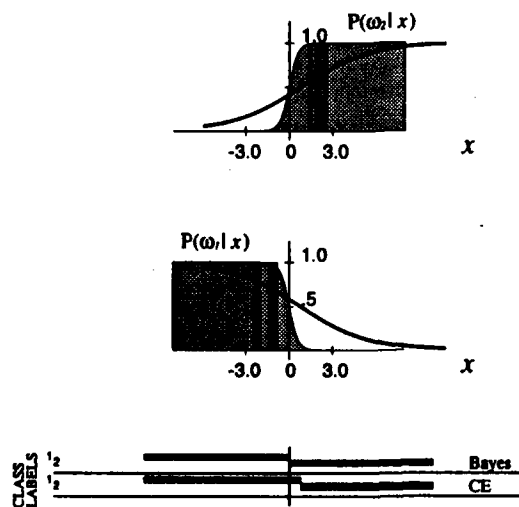


Figure 4.6: The same empirical *a posteriori* class probabilities shown in figure 4.5. The discriminant functions of the CFM-generated logistic linear classifier are superimposed in black. Note the large gap between the examples of ω_1 and ω_2 on the domain of x : since differential learning via CFM is discriminative, any set of discriminant functions that forms a class boundary in this gap is "optimal". As a result, the CFM-generated classifier's partitioning of feature space is a poor approximation to the Bayes-optimal partitioning, given this small training sample size ($n = 10$).

[30]. As we describe in section F.3, Efron's notion of asymptotic relative efficiency differs from ours: he defines ARE as the ratio of one model's discriminant bias to another model's; we define ARE as the ratio of one model's MSDE (*squared discriminant bias plus discriminant variance*) to another model's. This difference notwithstanding, the philosophical motivation for both definitions is similar. Since our feature x has class-conditional means of $\mu_1 = -1.65$ and $\mu_2 = 1.65$ and a variance parameter of $\sigma^2 = 1$, the Mahalanobis distance (e.g., [29, pg. 24]) between the class-conditional means is 3.3. Given this and the equal class prior probabilities of $\frac{1}{2}$, Efron predicts that the asymptotic discriminant bias of the fully-parametric proper model will be $\sim .55$ that of the partially-parametric model (see [30, (1.12)]).

Figure 4.7 displays the approximated MSDE ($\sim$ MSDE) for the four classifiers in figure 4.3; figure 4.8 displays their approximated discriminant bias ($\sim$ DBias) for the experiments. The box plots in figure 4.3 display the results in quartiles, whereas $\sim$ MSDE and $\sim$ DBias are based on sample averages; this accounts for slight differences among the figures. Again, the fully-parametric proper model (ML) and the partially-parametric proper models (CE, MSE) are all probabilistically-generated; one partially-parametric proper model (CFM) is differentially-generated. The ML model's $\sim$ MSDE is consistently lowest for all finite training sample sizes; the other probabilistically-generated models' $\sim$ MSDE is appreciably lower than the CFM model's for a training sample size of 10. For sample sizes greater than 100 all the partially-parametric models are roughly equivalent. For asymptotically large training sample sizes, all four classifiers exhibit zero MSDE. The gray-shaded region in figure 4.7 denotes values of $\sim$ MSDE less than 10^{-6} . We consider all classifiers that exhibit $\sim$ MSDE below this threshold to be equally good approximations of the Bayes-optimal classifier. To put this in perspective, a classifier with 0.1% discriminant bias and no discriminant variance exhibits a MSDE of 10^{-6} , as does a classifier with no discriminant bias and a discriminant variance of 10^{-6} . Thus, this MSDE threshold constitutes a rigorous standard of good approximation. The gray-shaded region in figure 4.8 denotes values of $\sim$ DBias less than 10^{-3} , a threshold that is identical to the $\sim$ MSDE threshold if the classifier has no discriminant variance. We consider all classifiers that exhibit $\sim$ DBias below this threshold to be equally unbiased approximations of the Bayes-optimal classifier. Both figures show that probabilistic learning generates more efficient, less biased classifiers for small training sample sizes than differential learning does. As the training sample size grows large (i.e., as it exceeds 10^3), the differentially-generated classifier becomes as good an approximation to the Bayes-optimal classifier as any of the probabilistic models — a phenomenon consistent with the asymptotic efficiency of differential learning. It is not surprising that the ML model is consistently more efficient than all the others. Indeed, based on our 10-trial experiments, the $\sim$ DBias of the ML model is 3×10^{-4} , whereas it is 7×10^{-4} for the CE model. Since the ML model is the fully-parametric maximum-likelihood paradigm and the CE model is the partially-parametric maximum-likelihood paradigm, Efron's prediction applies. We denote the logistic linear hypothesis class (ultimately employed by both the ML and CE models) by $\mathbf{G}(\Theta)$; we denote the ML model's maximum-likelihood probabilistic learning scheme by $\Lambda_{P,ML}$, and we denote the CE model's

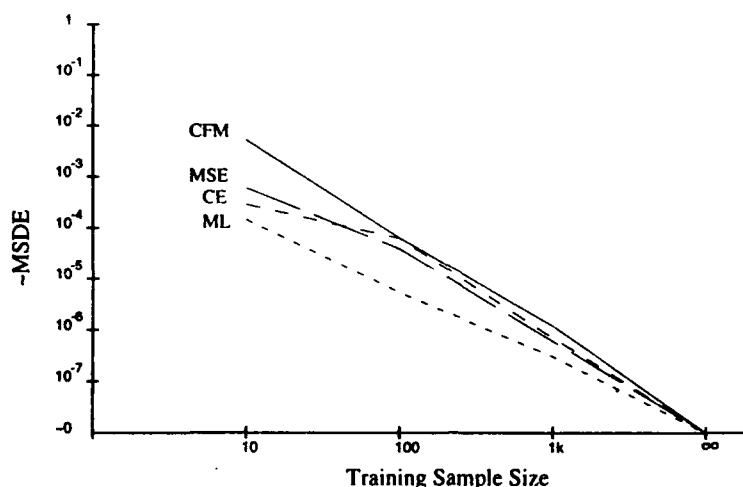


Figure 4.7: A comparison of the approximated mean-squared discriminant error ($\sim\text{MSDE}$) for the differentially (CFM) and probabilistically (MSE, CE, and ML) generated classifiers. Results for the differentially generated classifier are shown by the solid line; those for the probabilistically generated classifiers are shown by dashed lines. The gray background depicts the value of $\sim\text{MSDE}$ below which we consider all classifiers equally good approximations to the Bayes-optimal classifier. The CFM-generated classifier is not as efficient as its probabilistically-generated counterparts when the training sample size is small ($\mathcal{O}[10]$); however, owing to the asymptotic efficiency of differential learning, the difference between the CFM-generated classifier and its probabilistically-generated counterparts is negligible for sample sizes greater than $\mathcal{O}[10^3]$ (cf. figure 4.3).

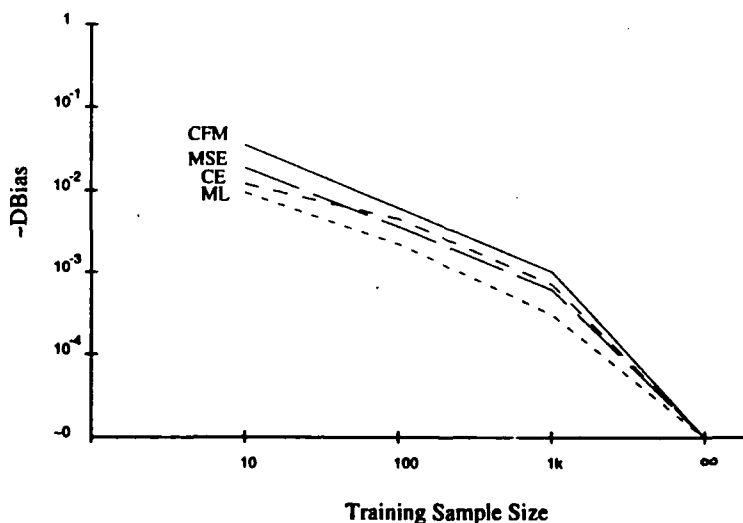


Figure 4.8: A comparison of the approximated discriminant bias ($\sim\text{DBias}$) for the differentially (CFM) and probabilistically (MSE, CE, and ML) generated classifiers. Results for the differentially generated classifier are shown by the solid line; those for the probabilistically generated classifiers are shown by dashed lines. The gray background depicts the value of $\sim\text{DBias}$ below which we consider all classifiers equally good approximations to the Bayes-optimal classifier. The CFM-generated classifier exhibits higher discriminant bias than its probabilistically-generated counterparts when the training sample size is small ($\mathcal{O}[10]$); however, owing to the asymptotic efficiency of differential learning, the difference between the CFM-generated classifier and its probabilistically-generated counterparts is negligible for sample sizes greater than $\mathcal{O}[10^3]$ (cf. figure 4.3).

maximum-likelihood probabilistic learning scheme by $\Lambda_{P,CE}$. Based on the probabilistic nature of x Efron's prediction is

$$\lim_{n \rightarrow \infty} \frac{\text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda_{P,ML}]}{\text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda_{P,CE}]} \cong 0.55 \quad (4.26)$$

Our experiments, depicted in figure 4.8, yield

$$\frac{\sim \text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda_{P,ML}]}{\sim \text{DBias} [\mathcal{G} | n, \mathbf{G}(\Theta), \Lambda_{P,CE}]} \cong 0.43; \quad n = 10^3, \quad (4.27)$$

which is a good approximation to Efron's prediction, considering the small number of trials and the small training sample size used.

Efron poses (and answers) the rhetorical question, "why use the partially-parametric model if the fully-parametric model is more efficient." The reason is that the fully-parametric model is proper if and only if x has homoscedastic Gaussian class-conditional pdfs; the partially-parametric model remains proper for a broader set of exponentially-distributed feature vectors (e.g., [83]). We extend the rhetorical question one more level: why use the differentially-generated parametric model if the probabilistically-generated models are more efficient? The reason is that the fully- and partially-parametric models are proper for x only so long as it has a specific probabilistic form; if the probabilistic nature of x deviates from this form, the parametric models are no longer proper. Under these circumstances, differential learning will still generate the most efficient classifier allowed by the parametric model for asymptotically large training sample sizes, whereas the probabilistic learning strategies will generate decidedly *inefficient* classifiers from the model for both small and large training sample sizes. We analyze an improper scenario in the following section in order to illustrate this point.

4.3 Analysis of an Improper Parametric Model

Figure 4.9 illustrates a three-class scalar x with heteroscedastic⁹ uniform class-conditional pdfs for the three classes $(\omega_1, \omega_2, \omega_3)$. There are two class boundaries ($\mathcal{B}_{1,2 \text{ Bayes}} = -4.0$, $\mathcal{B}_{2,3 \text{ Bayes}} = 4.0$) for the Bayes-optimal classifier of x . The class-conditional pdfs of x are given by

$$\begin{aligned} \rho_{x|W}(x|\omega_1) &= \frac{1}{2} [u(x + 5.8) - u(x + 3.8)] \\ \rho_{x|W}(x|\omega_2) &= \frac{1}{8} [u(x + 4) - u(x - 4)] \\ \rho_{x|W}(x|\omega_3) &= \frac{1}{2} [u(x - 3.8) - u(x - 5.8)] , \end{aligned} \quad (4.28)$$

where $u(\cdot)$ denotes the Heaviside step function. The class prior probabilities are

⁹Heteroscedastic pdfs have different variance parameters (or covariance matrices).

$$\begin{aligned} P_W(\omega_1) &= P_W(\omega_3) = 0.1 \\ P_W(\omega_2) &= 0.8, \end{aligned} \quad (4.29)$$

and the *a posteriori* class probabilities of x are given by

$$\begin{aligned} P_{W|x}(\omega_1|x) &= u(x + 5.8) - \frac{2}{3}u(x + 4) - \frac{1}{3}u(x + 3.8) \\ P_{W|x}(\omega_2|x) &= \frac{2}{3}u(x + 4) + \frac{1}{3}u(x + 3.8) - \frac{1}{3}u(x - 3.8) - \frac{2}{3}u(x - 4) \\ P_{W|x}(\omega_3|x) &= \frac{1}{3}u(x - 3.8) + \frac{2}{3}u(x - 4) - u(x - 5.8) \end{aligned} \quad (4.30)$$

Thus, the Bayes error rate is 2.0%, given the following classification strategy:

$$\begin{aligned} B_{1,2 \text{ Bayes}} \leq \begin{cases} x < B_{1,2 \text{ Bayes}}, & \text{choose } \omega_1 \\ x \leq B_{2,3 \text{ Bayes}}, & \text{choose } \omega_2 \\ x > B_{2,3 \text{ Bayes}}, & \text{choose } \omega_3 \end{cases} \end{aligned} \quad (4.31)$$

4.3.1 The Improper Parametric Model

We learn to classify x with a 3-output discriminator that has polynomial discriminant functions of the form

$$\mathcal{G}(x|\theta) = \left\{ y_i = g_i(x|\theta) = \sum_{k=0}^{K_i} \theta_{i,k} \cdot (x)^k; \quad i = 1, \dots, C = 3 \right\}, \quad (4.32)$$

where K_i represents the order of the polynomial expression for the i th discriminant function (again, we use the notation $(x)^k$ to denote the k th power of x , as opposed to x^k , which denotes the k th example of x). As described in section 2.2.1, we interpret the discriminator output with the largest value as the classifier's vote for the class of its scalar input x . This polynomial classifier is depicted in figure 4.10 and it is generated with a modified form of the backpropagation algorithm (e.g., [119, 120]). It is, by definitions 3.13 and 3.14, an improper parametric model of x because the discriminant functions of (4.32) are not under any circumstances identically equal to the *a posteriori* class probabilities of (4.30).

Given our interpretation of the classifier's outputs in section 2.2.1, it is clear that if y_1 and y_3 are linear functions of x and y_2 is a constant, the resulting classifier depicted by the white nodes and black connections in figure 4.10 has the minimum functional complexity necessary to learn the Bayes-optimal classifier of x (the gray nodes and connections depict more complex hypothesis classes — equating to higher order polynomial expressions in (4.32) — for this task).¹⁰

¹⁰Reference [52] incorrectly states that the minimum-complexity polynomial classifier has two linear discriminant functions and one quadratic discriminant function. As described herein, the third discriminant function need only be a *constant* for the classifier to yield Bayesian discrimination.

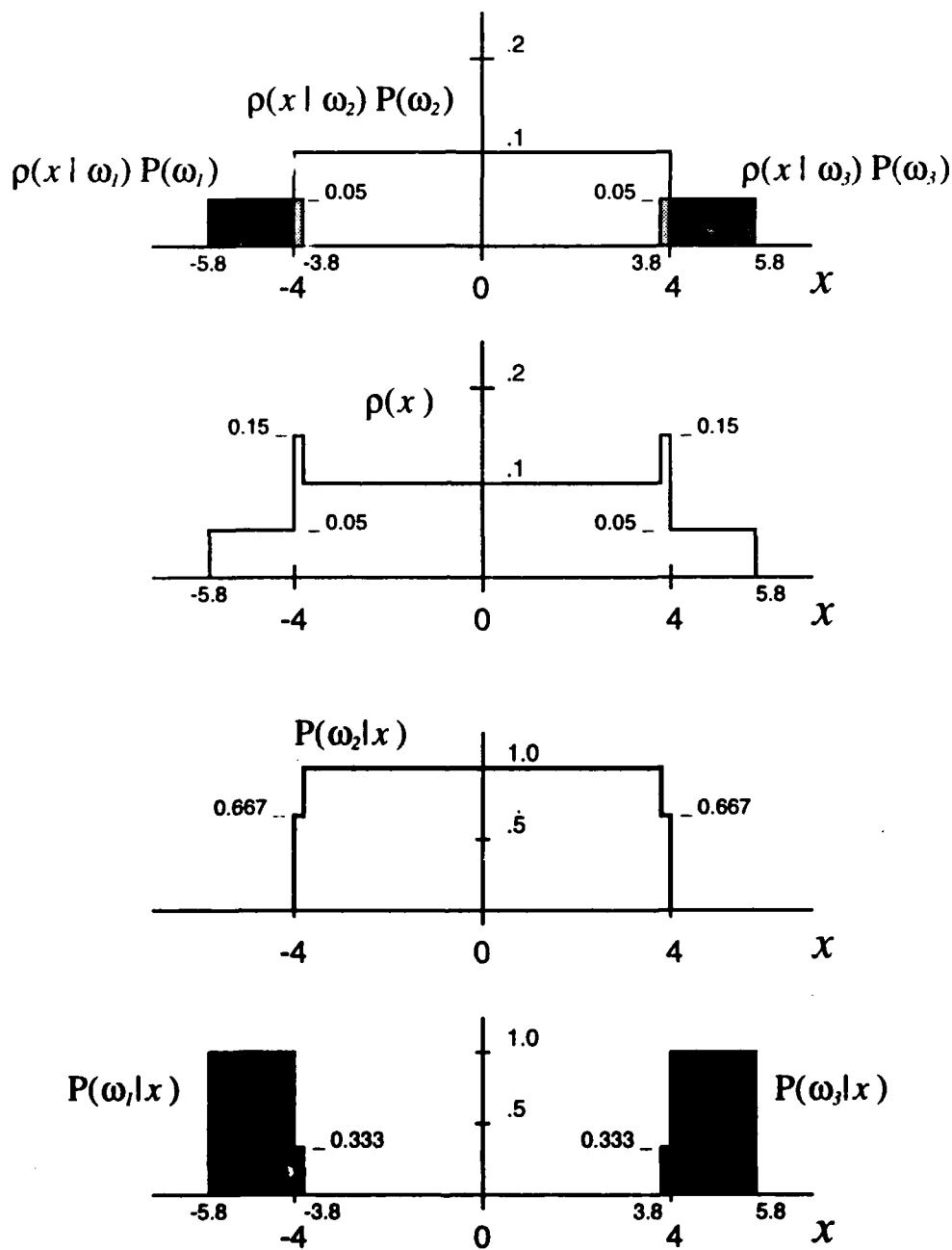


Figure 4.9: A three-class scalar feature discrimination task. The single feature is a heteroscedastic, uniformly-distributed random variable. From top to bottom: the class-conditional density - class prior products $\rho_{x|\mathcal{W}}(x|\omega_i) \cdot P_{\mathcal{W}}(\omega_i)$; the pdf of x $\rho_X(x)$; the *a posteriori* probability of class ω_2 $P_{\mathcal{W}|x}(\omega_2|x)$; the *a posteriori* probabilities of classes ω_1 and ω_3 $P_{\mathcal{W}|x}(\omega_1|x)$, $P_{\mathcal{W}|x}(\omega_3|x)$.

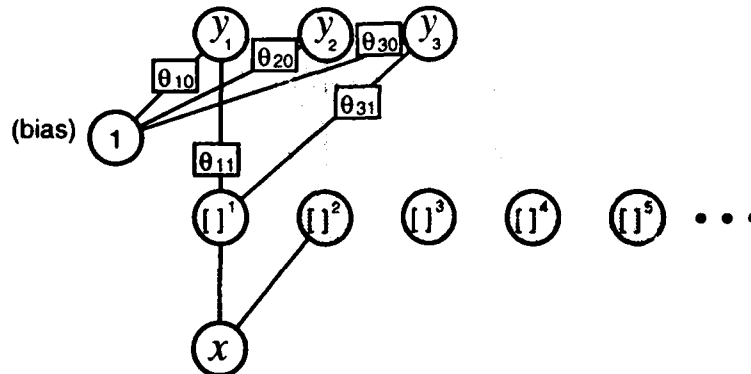


Figure 4.10: The polynomial classifier of x depicted as a neural network paradigm. Hidden layer nodes compute powers of x ; output nodes are linear combinations of these powers — cf. (4.32). The polynomial classifier with the minimum complexity necessary for Bayesian discrimination of x is indicated by the white nodes and black “connections” (i.e., parameters); the minimum-complexity parameters are labeled.

Minimum-, low-, and high-complexity classifiers: For the purpose of this illustration K_i — the order of the i th polynomial in (4.32) — may be taken as the complexity measure for the i th discriminant function $g_i(x|\theta)$. We generate classifiers from three polynomial hypothesis classes of increasing complexity. As described above, the minimum-complexity hypothesis class has discriminant functions of order $K_1 = 1$, $K_2 = 0$, and $K_3 = 1$; we often use the notation “1-0-1” to denote this hypothesis class. Our choice of low-complexity hypothesis class has discriminant functions of order $K_1 = 1$, $K_2 = 2$, and $K_3 = 1$; we often use the notation “1-2-1” to denote this hypothesis class. Our choice of high-complexity hypothesis class has discriminant functions of order $K_1 = 10$, $K_2 = 10$, and $K_3 = 10$; we often use the notation “10-10-10” to denote this hypothesis class.

4.3.2 Probabilistic Learning via MSE for the Asymptotically Large Training Sample

Probabilistic learning is implemented by minimizing a measure of the difference between the discriminator output vector $\mathbf{Y}$ and a corresponding target vector denoting the class of the training example (see section 2.3); the minimization is done for all examples in the training sample, and generally takes the form of an iterative search procedure. We employ backpropagation, a well-known probabilistic learning paradigm; its iterative search procedure is gradient descent, and the gradient of the classifier’s MSE with respect to the parameter vector θ is computed by the chain-rule [119, 120].¹¹

¹¹Backpropagation generally employs MSE, although other objective functions can be used. We employ only the MSE objective function for probabilistic learning. The CE objective function, for example, cannot be used because the polynomial classifier’s outputs are unbounded; this violates the conditions necessary for using CE (see section 2.3.2). When paired with the CFM objective function and a gradient *ascent* search, backpropagation constitutes a differential learning strategy.

Again, we denote the training sample of size n by S^n , and we denote a particular unique value (or pattern) of x by x_p . If there are P unique patterns in S^n , and for each of these patterns there are $n_{p,i}$ examples belonging to class ω_i , the sample MSE is given by

$$\text{MSE}(S^n | \theta) = \sum_{i=1}^{C=3} \sum_{p=1}^P \frac{n_p}{n} \left[(g_i(x_p | \theta) - 1)^2 \cdot \frac{n_{p,i}}{n_p} + (g_i(x_p | \theta))^2 \cdot \frac{n_p - n_{p,i}}{n_p} \right]; \quad (4.33)$$

$$\sum_{p=1}^P n_{p,i} = n$$

From section 2.3.2 we know that the classifier's MSE can be expressed by the following expectation as the training sample size grows asymptotically large:

$$E_x [\text{MSE}(x | \theta)] = \frac{1}{2} \sum_{i=1}^{C=3} \int_{-\infty}^{\infty} \left[(g_i(x | \theta) - 1)^2 \cdot P_{W|x}(\omega_i | x) + (g_i(x | \theta))^2 \cdot P_{W|x}(-\omega_i | x) \right] \rho_x(x) dx \quad (4.34)$$

where

$$P_{W|x}(-\omega_i | x) \triangleq 1 - P_{W|x}(\omega_i | x) \quad (4.35)$$

The parameterization θ^* that minimizes the classifier's MSE for the asymptotically large training sample size can be found by substituting the discriminant function expressions of (4.32) into (4.35), deriving the expression for the gradient $\nabla_{\theta} (E_x [\text{MSE}(x | \theta^*)])$, setting this gradient equal to the zero vector, and solving the resulting normal equations (see section 2.3.2).

Barnard and Casasent use this technique for deriving the minimum-MSE parameterization of a linear classifier, given a 2-class Gaussian feature [6]. Appendix H derives distribution-independent expressions for the asymptotic minimum-MSE parameterization of the i th discriminant function $g_i(x | \theta)$ in (4.32); expressions are given for constant, linear, and quadratic discriminant functions (i.e., for $K_i = 0, 1, 2$). Distribution-independent expressions for the minimum-MSE parameterizations of higher-order polynomial discriminant functions become cumbersome, so we derive the minimum-MSE parameterization of the high-complexity classifier (i.e., the MSE-generated "10-10-10" model) in distribution-dependent form using (4.28), (4.30), (4.32), and (4.35). Table 4.1 summarizes the results of appendix H; it lists the minimum-MSE parameterizations of the minimum-, low-, and high-complexity classifiers, given an asymptotically large training sample size.

Asymptotic Minimum-MSE Parameterizations $n \rightarrow \infty$		
Minimum-Complexity Classifier "1-0-1"		
$g_1(x \theta)$	$g_2(x \theta)$	$g_3(x \theta)$
$\theta_{1,0} = 0.1$ $\theta_{1,1} = -.0536833$	$\theta_{2,0} = 0.8$ $\theta_{2,1} = 0$	$\theta_{3,0} = 0.1$ $\theta_{3,1} = 0.0536833$
Low-Complexity Classifier "1-2-1"		
$g_1(x \theta)$	$g_2(x \theta)$	$g_3(x \theta)$
$\theta_{1,0} = 0.1$ $\theta_{1,1} = -.0536833$	$\theta_{2,0} = 1.13764$ $\theta_{2,1} = 0$	$\theta_{3,0} = 0.1$ $\theta_{3,1} = 0.0536833$
$\theta_{1,2} = 0$	$\theta_{2,2} = -0.0377619$	$\theta_{3,2} = 0$
High-Complexity Classifier "10-10-10"		
$g_1(x \theta)$	$g_2(x \theta)$	$g_3(x \theta)$
$\theta_{1,0} = -0.0222177$ $\theta_{1,1} = -0.0535861$ $\theta_{1,2} = 0.0513984$ $\theta_{1,3} = 0.0273086$ $\theta_{1,4} = -0.0172172$ $\theta_{1,5} = -0.00319705$ $\theta_{1,6} = 0.00179515$ $\theta_{1,7} = 0.000110813$ $\theta_{1,8} = -0.0000664743$ $\theta_{1,9} = -0.00000120399$ $\theta_{1,10} = .000000815606$	$\theta_{2,0} = 1.04444$ $\theta_{2,1} = 0$ $\theta_{2,2} = -0.102797$ $\theta_{2,3} = 0$ $\theta_{2,4} = 0.0344344$ $\theta_{2,5} = 0$ $\theta_{2,6} = -0.00359029$ $\theta_{2,7} = 0$ $\theta_{2,8} = 0.000132949$ $\theta_{2,9} = 0$ $\theta_{2,10} = -0.00000163121$	$\theta_{3,0} = -0.0222177$ $\theta_{3,1} = 0.0535861$ $\theta_{3,2} = 0.0513984$ $\theta_{3,3} = -0.0273086$ $\theta_{3,4} = -0.0172172$ $\theta_{3,5} = 0.00319705$ $\theta_{3,6} = 0.00179515$ $\theta_{3,7} = -0.000110813$ $\theta_{3,8} = -0.0000664743$ $\theta_{3,9} = 0.00000120399$ $\theta_{3,10} = .000000815606$

Table 4.1: The minimum-MSE parameterizations for the minimum-, low-, and high-complexity polynomial classifiers of x when the training sample size n is asymptotically large (i.e., $n \rightarrow \infty$).

4.3.3 Differential Learning via CFM for the Asymptotically Large Training Sample

Differential learning is implemented for the improper parametric model in the same way it is implemented for the partially-parametric proper model of section 4.2.3. The minimum-complexity polynomial classifier maximizes CFM, satisfies (4.20), and constitutes the Bayes-optimal classifier when its parameters satisfy the following constraints:

$$\begin{aligned}
 \theta_{1,1 \text{ CFM}} &< 0 \\
 \theta_{3,1 \text{ CFM}} &> 0 \\
 \theta_{1,1 \text{ CFM}} \cdot \mathcal{B}_{1,2 \text{ Bayes}} + \theta_{1,0 \text{ CFM}} - \theta_{2,0 \text{ CFM}} &= 0 \\
 \theta_{3,1 \text{ CFM}} \cdot \mathcal{B}_{2,3 \text{ Bayes}} + \theta_{3,0 \text{ CFM}} - \theta_{2,0 \text{ CFM}} &= 0
 \end{aligned} \tag{4.36}$$

When these conditions are satisfied, the resulting class boundaries equal the Bayes-optimal boundaries:

$$\begin{aligned} \mathcal{B}_{1,2 \text{ CFM}} &= \mathcal{B}_{1,2 \text{ Bayes}} \\ \mathcal{B}_{2,3 \text{ CFM}} &= \mathcal{B}_{2,3 \text{ Bayes}} \end{aligned} \quad (4.37)$$

4.3.4 Results of Differential and Probabilistic Learning for Asymptotically Large and Small Training Samples

Regardless of the strategy used to determine the classifier's parameters, the resulting class boundaries occur at all x for which more than one discriminant function is maximal. For the minimum-complexity polynomial classifier, this occurs at

$$\begin{aligned} \mathcal{B}_{1,2} &= \frac{\theta_{2,0} - \theta_{1,0}}{\theta_{1,1}} \\ \mathcal{B}_{2,3} &= \frac{\theta_{2,0} - \theta_{3,0}}{\theta_{3,1}} \end{aligned} \quad (4.38)$$

Figure 4.11 illustrates the discriminant functions of several polynomial classifiers that have learned to recognize the three classes that x represents, given an asymptotically large training sample. Both the top and bottom figures show the discriminant functions superimposed in color on the gray *a posteriori* class probabilities $P_{W|x}(\omega_1|x)$, $P_{W|x}(\omega_2|x)$, $P_{W|x}(\omega_3|x)$. There is a bar-graph display associated with each classifier underneath the discriminant functions. The bar-graph shows how its associated classifier partitions feature space.¹² The Bayes-optimal classifier's partitioning is always shown in gray for reference. The top figure shows two minimum-complexity classifiers: one generated probabilistically via the MSE objective function (red, short-dashed lines), and one generated differentially via the CFM objective function (solid green lines). Because the probabilistically generated classifier is attempting to approximate $P_{W|x}(\omega_1|x)$ and $P_{W|x}(\omega_3|x)$ with linear functions of x and $P_{W|x}(\omega_2|x)$ with a constant, the minimum-MSE discriminant functions are as shown (their parameter values are given in table 4.1, top), and the resulting classifier labels all examples of x as ω_2 . As a result, the classifier misclassifies all examples of ω_1 and ω_3 ; its error rate is therefore 20%.¹³ The differentially-generated classifier shown is one set of an infinite number of possible maximum-CFM discriminant functions. Its parameterization is such that $g_i(x|\theta)$ is always maximum on the

¹²Note that the legend "CFM 1-0-1", for example, denotes the differentially generated, minimum-complexity polynomial classifier.

¹³The MSE-generated minimum-complexity classifier would exhibit an error rate much closer to the Bayes-optimal rate of 2% for large training sample sizes if the linear discriminant functions for class ω_1 and ω_3 were replaced by logistic discriminant functions. This is because the resulting hypothesis class would be a substantially better approximation to the proper parametric model of x . This extends to a general argument for hypothesis classes with a logistic functional basis: many real-world feature vectors have unimodal class-conditional pdfs, so their *a posteriori* class probabilities are reasonably well modeled by a logistic functional basis. This accounts for the success and wide-spread use of probabilistically-generated logistic regression models and multi-layer perceptrons... a subject we address further in chapter 11. Of course, our choice of the functional basis is intentionally malicious here: we wish to illustrate the disadvantages of probabilistic learning when the parametric model is *improper*.

sub-domain of x for which ω_i is the most likely class. As a result, the classifier satisfies (4.20) and exhibits the Bayes error rate of 2%.

It is clear from figure 4.11 (top) that the minimum-complexity classifier has insufficient functional complexity to learn the Bayes-optimal classifier probabilistically (i.e., to approximate the *a posteriori* class probabilities of x). Figure 4.11 (bottom) illustrates what is required to do this. If we increase the complexity of $g_2(x|\theta)$ by making it a quadratic function of x , the resulting minimum-MSE discriminant functions are shown in short-dashed red lines (their parameter values are given in table 4.1, middle). This low-complexity classifier has enough functional complexity to classify *some* examples of ω_1 and ω_3 correctly, although it still lacks sufficient complexity for Bayesian discrimination. Its error rate is 7.8%. The differentially generated low-complexity classifier (solid and shaded green lines) — like its minimum-complexity counterpart — yields the Bayes error rate of 2%. Again, there are innumerable maximum-CFM parameterizations for the differentially-generated classifier; the green shaded lines in the figure depict several of these. Finally, we increase the complexity of the probabilistically generated classifier so that all three discriminant functions are 10th-order polynomials in x . These discriminant functions are shown by the blue dashed lines in the lower figure; only the MSE-generated classifier is shown (its parameter values are given in table 4.1, bottom). This high-complexity classifier has sufficient functional complexity to approximate the *a posteriori* class probabilities of x reasonably well when generated via the MSE objective function; it exhibits a 2.2% error rate — nominally the Bayes error rate.

Figure 4.11 illustrates that differential learning requires the minimum-complexity polynomial classifier necessary for Bayesian discrimination. The minimum-complexity requirements of differential learning hold for any and all choices of hypothesis class, as proven in section 3.5. Probabilistic learning, in contrast, requires a high-complexity polynomial classifier in order to approximate the Bayes-optimal classifier — a result that is representative of the generally excessive complexity requirements of probabilistic learning, the single notable exception being when the hypothesis class is a proper parametric model.

So far we have considered the asymptotic case in which we have an unlimited number of training examples. It is more realistic to consider the case in which we have a limited amount of training data. Figure 4.12 depicts the same classifiers shown in figure 4.11 with one difference: the classifiers in figure 4.12 have been generated with a single training sample containing only $n = 100$ examples of x (the different classifiers all learn the same 100 examples). The minimum-complexity classifiers (top) behave in much the same way as they do for the asymptotically large training sample. The probabilistically generated classifier (red, short-dashed lines) misclassifies all examples of ω_1 and ω_3 . Owing to the small sample size, the empirical *a posteriori* class probabilities of x are crude approximations to the true probabilities. As a result, the differentially-generated classifier's partitioning of feature space (solid green lines) deviates slightly from the Bayes-optimal partitioning, and its error rate is 3.4%. The low-complexity classifiers (bottom) also behave in much the same way as they do for the asymptotically large training sample. The probabilistically

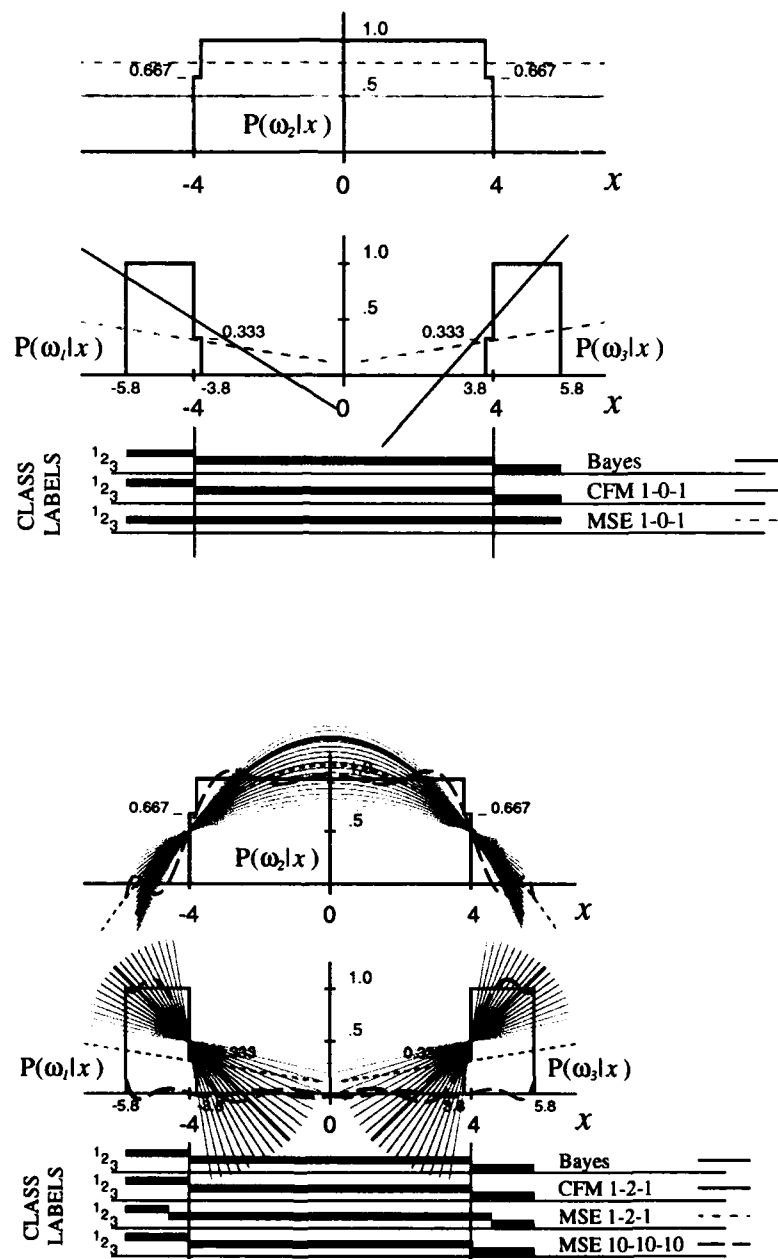


Figure 4.11: Discriminant functions of probabilistically (MSE) and differentially (CFM) generated polynomial classifiers of x for an asymptotically large training sample size (i.e., $n \rightarrow \infty$). The functions are shown superimposed on their associated *a posteriori* class probabilities (shown in gray). Each of the bar-graphs underneath the discriminant functions depicts how its associated polynomial classifier partitions feature space. **Top:** the minimum-complexity classifier ("1-0-1") having one constant and two linear discriminant functions. **Bottom:** a low complexity classifier ("1-2-1") having one quadratic and two linear discriminant functions, and a high-complexity classifier ("10-10-10") having three 10th-order polynomial discriminant functions (MSE-generated only). Numerous low-complexity CFM-maximizing classifiers are shown (green shaded lines) in order to emphasize that there are innumerable optimal solutions when differential learning is employed.

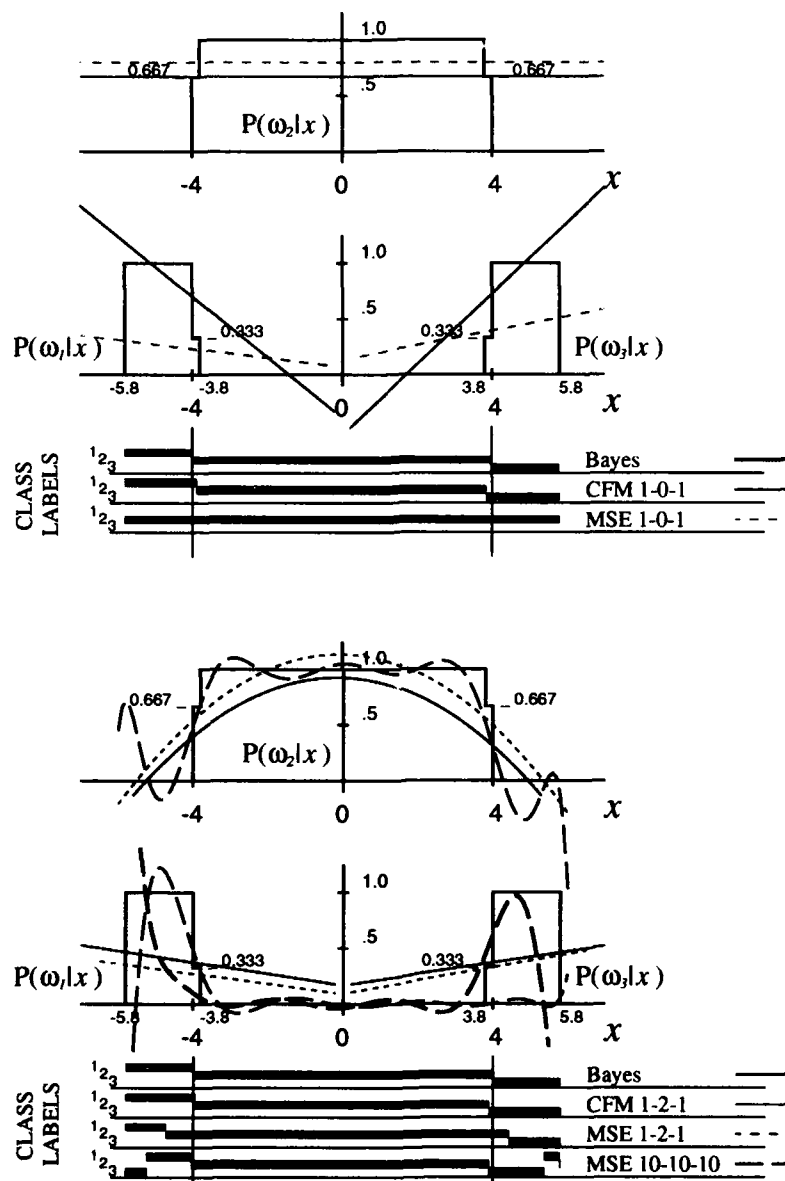


Figure 4.12: Discriminant functions of probabilistically (MSE) and differentially (CFM) generated polynomial classifiers of x for a typical training sample of size $n = 100$. Again, the functions are shown superimposed on their associated *a posteriori* class probabilities (shown in gray), and each of the bar-graphs underneath the discriminant functions depicts how its associated polynomial classifier partitions feature space. **Top:** the minimum-complexity classifier ("1-0-1") having one constant and two linear discriminant functions. **Bottom:** a low complexity classifier ("1-2-1") having one quadratic and two linear discriminant functions, and a high-complexity classifier ("10-10-10") having three 10th-order polynomial discriminant functions (MSE-generated only).

generated classifier (red, short-dashed lines) exhibits an error rate of 7.8%, and the differentially-generated classifier (solid green lines) exhibits an error rate of 3.3%.

Recall that the high-complexity classifier exhibits a 2.2% error rate for the asymptotically large training sample. Since the empirical *a posteriori* class probabilities of x are crude approximations to the true probabilities when $n = 100$, there are regions on the domain of x where no training examples occur. The classifier's discriminant functions are unconstrained in these regions during learning. Figure 4.12 (bottom) illustrates what happens as a result. The high-complexity classifier's discriminant functions (blue dashed lines) are unconstrained for values of x above ~ 5.2 and below ~ -5.0 because the training sample contains no examples beyond these limits. As a result, the discriminant function for ω_3 is maximal for $x < \sim -5.0$, the discriminant function for ω_1 is maximal for $\sim 5.2 < x < \sim 5.7$, and the discriminant function for ω_2 is maximal for $x > \sim 5.7$. The resulting partitioning of feature space (bottom blue bar-graph) is poor, and the classifier exhibits a 7.8% error rate. This is a classic expression of Occam's razor [130, 21], in which the classifier has so much functional complexity it fails to generalize well for small training sample sizes.

Figure 4.13 displays the empirical distribution of the error rates for minimum-, low-, and high-complexity polynomial classifiers of x . Results for differential learning via CFM are shown in white box plots; results for probabilistic learning via MSE are shown in gray box plots. As with the proper parametric model experiments, all results for finite training sample sizes are based on 10 independent trials for the specified training sample size (all classifiers learn the same training sample in a given trial, for a given sample size). Learning takes the form of a steepest descent (MSE) or steepest ascent (CFM) search over parameter space, using a modified form of the backpropagation algorithm (e.g., [119, 120]). Learning begins from a tabula rasa state in which all parameters are initialized randomly according to a uniform distribution on the closed interval $[-.3, .3]$. All trials are completely automated, so learning is done without any human intervention. All experimental conditions (except, of course, for the objective function used) are identical for differential and probabilistic learning. Classifier parameterizations for the asymptotically large training sample size are derived as described in section 4.3.2, appendix H, and section 4.3.3.

Figure 4.14 plots sample-average approximations of each classifier's mean-squared discriminant error ($\sim$ MSDE): these values correspond to the box plot statistics in figure 4.13 (the box plots display the results in quartiles, whereas $\sim$ MSDE is based on sample averages; this accounts for slight differences between the two figures). The minimum-complexity differentially generated classifier is the most efficient, exhibiting consistently low error rates for small training sample sizes. Based on chapter 6, we predict that 1121 samples of x are necessary to guarantee (with 95% confidence) an error rate of no more than 4.0% using differential learning. Note that the empirical upper bound on the differentially generated minimum-complexity classifier's error rate is 3.3% when the sample size is 1000. Increasing the differentially generated classifier's complexity increases its empirical discriminant variance, according to Occam's razor (i.e., excessively complex models are anathema).

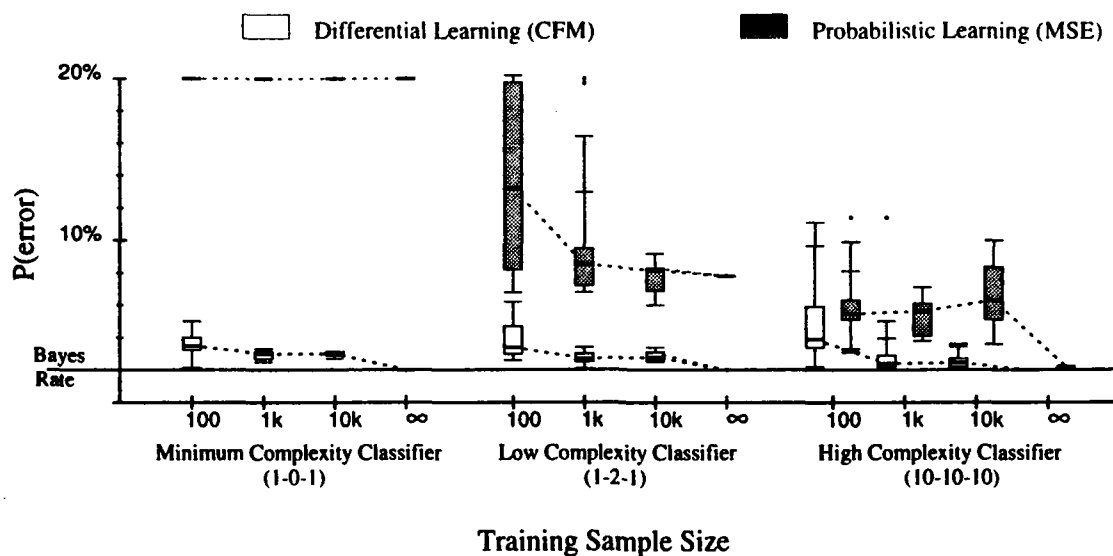


Figure 4.13: A comparison of error rates for differentially (CFM) and probabilistically (MSE) generated polynomial classifiers. Results for the differentially generated classifiers are shown in white; those for the probabilistically generated classifier are shown in gray. **Left:** the minimum-complexity classifier having one constant and two linear discriminant functions; **Middle:** a low complexity classifier having one quadratic and two linear discriminant functions; **Right:** a high-complexity classifier having three 10th-order polynomial discriminant functions.

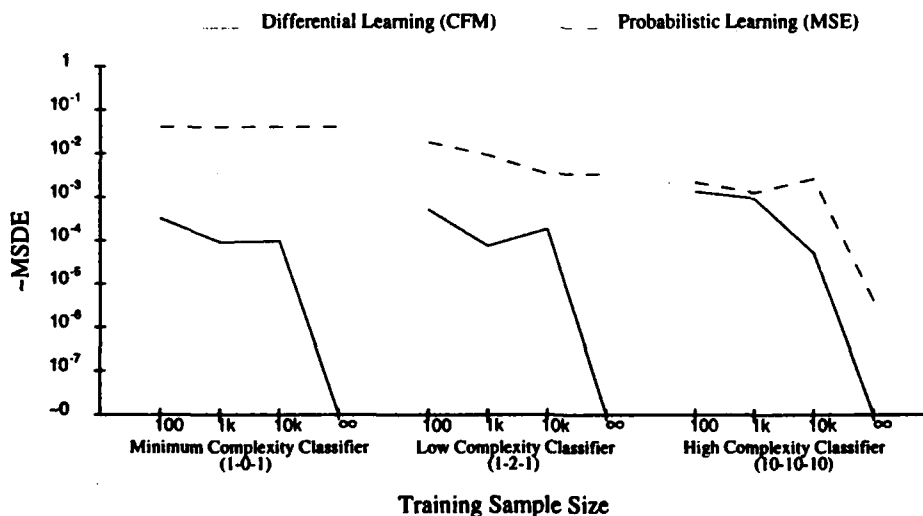


Figure 4.14: A comparison of the approximated mean-squared discriminant error ($\sim$ MSDE) for differentially (CFM) and probabilistically (MSE) generated polynomial classifiers. These statistics are based on the same data used to generate the box plots in figure 4.13. Results for the differentially generated classifiers are shown in solid lines; those for the probabilistically generated classifier are shown in dashed lines. The gray background depicts values of $\sim$ MSDE for which we consider the classifier to be a good approximation to the Bayes-optimal classifier. **Left:** the minimum-complexity classifier having one constant and two linear discriminant functions; **Middle:** a low complexity classifier having one quadratic and two linear discriminant functions; **Right:** a high-complexity classifier having three 10th-order polynomial discriminant functions. Owing to the inefficiency of probabilistic learning, none of the MSE-generated classifiers is relatively efficient for *any* training sample size.

The inefficiency of probabilistic learning is clear in figures 4.13 and 4.14. The minimum-complexity classifier has no discriminant variance, but its approximate discriminant bias is $20\% - 2\% = 18\%$. The low-complexity classifier has substantially lower approximate discriminant bias ($8.3\% - 2\% = 6.3\%$ for $n = 1000$), but its discriminant variance is high (2.2×10^{-3}). The high-complexity classifier has moderate discriminant bias ($5.4\% - 2\% = 3.4\%$ for $n = 1000$), and moderate discriminant variance (1.3×10^{-4}) — substantially better than the probabilistically generated low-complexity classifier, but substantially worse than the differentially generated minimum-complexity classifier.

The gray background of figure 4.14 denotes values of $\sim\text{MSDE}$ for which we consider a classifier to be a good approximation to the Bayes-optimal classifier. Specifically, if the classifier's $\sim\text{MSDE}$ is less than 10^{-6} , we consider it a good approximation.¹⁴ All of the differentially-generated polynomial classifiers are asymptotically good approximations to the Bayes-optimal classifier, whereas none of the probabilistically-generated classifiers are. Moreover, the minimum- and low-complexity differentially-generated classifiers are between one and two orders of magnitude more efficient than their probabilistically-generated counterparts. As the model complexity becomes high, both differentially and probabilistically generated classifiers are inefficient models of the data for small training sample sizes — a clear expression of Occam's razor.

4.4 Summary

The "toy" experiments of this chapter illustrate the theoretical proofs of chapters 2 and 3. Differential learning is asymptotically efficient, regardless of the hypothesis class (i.e., parametric model) employed, whereas probabilistic learning is efficient (for both large and small training sample sizes) *only if* the hypothesis class is a proper parametric model of the data. This implies a kind of robust beauty in the differentially-generated classifier: it is guaranteed to be the best approximation of the Bayes-optimal classifier allowed by the model of the data, so long as the training sample size is sufficiently large. As we stated in chapter 3, we know of no other learning strategy that can make this guarantee.

There is no doubt that probabilistic learning in the form of maximum-likelihood parameter estimation¹⁵ is the most efficient learning strategy if the parametric model is indeed a good approximation to the proper one. Our contention is that this is not always the case. If the parametric model is simple, traditional statistical hypothesis testing procedures (e.g., see [140]) can verify whether or not it is proper. If the model is complex, complexity theory argues against its being proper, particularly when the training sample size is small. This leads us to conclude that differential learning is the best choice of learning strategy if the model is likely to be improper. The experiments of part II consistently show this to be a valid conclusion.

¹⁴Again, a classifier with a discriminant bias of 0.1% and a discriminant variance of zero exhibits an $\sim\text{MSDE}$ of 10^{-6} , so this is a rigorous standard for good approximation.

¹⁵We remind the reader that maximum-likelihood parameter estimation generally equates to probabilistic learning via an error measure objective function.

Chapter 5

Properties of the CFM Objective Function¹

Outline

We examine the relationship between the objective function's monotonicity and the efficiency of the learning strategy it implements: the objective function *must* be monotonic for the learning strategy to be efficient.² This chapter's proofs that the CFM objective function is monotonic for the general C -class pattern recognition task parallel the chapter 3 proofs that differential learning is asymptotically efficient. Likewise, the proofs that error measures are non-monotonic for the general C -class pattern recognition task parallel the chapter 3 proofs that probabilistic learning is inefficient. Moreover, probabilistic learning becomes increasingly inefficient as the number of classes C increases, owing to the increasingly non-monotonic nature of error measures. We develop a simple taxonomy of training examples in order to show that differential learning via CFM focuses on un-learned examples. Among these, there are easy and hard examples. We explain why easy examples can be learned with high confidence, whereas hard examples must be learned with low confidence. We conclude by examining the specific functional characteristics of CFM in order to motivate the synthetic form we employ. We prove that differential learning via the synthetic form of CFM remains both efficient and reasonably fast as learning confidence is reduced. In contrast, differential learning via the original functional forms of CFM [55] is unreasonably slow and/or inefficient.

5.1 Introduction

Differentiable supervised classifiers that learn iteratively employ an objective function (or empirical risk measure) that evaluates how well the classifier has learned to classify all the examples of the training sample. A monotonic objective function is one that is *always* a strictly decreasing (or increasing) function of the

¹ Portions of this chapter were first published in [55].

² An objective function is monotonic if and only if it is either a strictly increasing or a strictly decreasing function of the classifier's empirical training sample error rate (see definition 5.10).

classifier's empirical training sample error rate. Chapter 2 establishes a link between families of objective functions and the learning strategies they implement: error measures engender probabilistic learning, whereas the CFM objective function engenders differential learning. Section 3.3 indirectly proves that maximizing the CFM objective function minimizes the classifier's empirical training sample error rate (the proof is part of the larger proof that differential learning via CFM is asymptotically efficient). In other words, CFM is a monotonic objective function. Section 3.4 proves that minimizing the general error measure does *not* minimize the classifier's empirical training sample error rate. In other words, error measures are non-monotonic objective functions; they engender inefficient learning (unless the hypothesis class with which they are paired is the proper parametric model of the feature vector).

In this chapter, we take a geometric view of the discriminator's output state in order to illustrate the monotonic nature of the CFM objective function and the non-monotonic nature of error measures. We demonstrate that probabilistic learning strategies become increasingly inefficient as the number of classes C in the pattern recognition task increases, owing to the non-monotonic nature of error measures. In the process, we develop a simple taxonomy of training examples. Fundamentally, each training example falls into one of two categories: learned and (as yet) un-learned. The un-learned examples are either easy to learn or hard to learn (terms we define in section 5.4). We show that differential learning via the synthetic CFM objective function focuses on the un-learned examples; we explain why easy examples can be learned with high confidence, whereas hard examples must be learned with low confidence.

We conclude by analyzing the functional characteristics of CFM in order to motivate the synthetic form we employ. The analysis focuses on the process of learning hard examples with necessarily low confidence. Since the differentiable supervised classifier learns by searching over parameter space, the speed of the learning procedure (i.e., its convergence rate) is proportional to the step size of the search procedure. We prove that differential learning via the synthetic form of CFM is reasonably fast for both easy and hard examples: convergence to the CFM-maximizing parameters proceeds at a rate that decreases polynomially with respect to the synthetic CFM confidence parameter ψ . In contrast, we prove that differential learning via the original forms of CFM [55] is inefficient and/or unreasonably slow. In the latter case, convergence to the CFM-maximizing parameters proceeds at a rate that decreases exponentially with respect to the CFM confidence parameter.

5.2 Discriminator Output Space

We precede our discussion of monotonicity with a number of definitions that follow from a geometric view of discriminator output space $\mathcal{Y}$. Recall from section 2.2.1, we generally assume that discriminator output space is infinite and uncountable for the C -class discriminator (i.e., $\mathcal{Y} = \mathbb{R}^C$). In this section we will assume that each discriminator output is uncountable on a closed interval with lower and upper bounds l and h :

$$\mathbf{Y} \in \mathcal{Y} = [l, h]^C \quad (5.1)$$

When $l \rightarrow -\infty$ and $h \rightarrow \infty$, (5.1) is equivalent to $\mathcal{Y} = \mathbb{R}^C$. We therefore use the bounds l and h without loss of generality.

Some of the following definitions might seem rather abstract and tedious at first reading. We encourage the reader to persevere, reading the definitions first without dwelling on the associated mathematical details, which are simply more formal expressions of what we say in words. Those who want to work through the mathematical details can go back through the definitions a second time. The concepts are all geometric and rather simple, which should become apparent as one proceeds through the first reading of the definitions. Examples and figures help to clarify the concepts.

Definition 5.1 The discriminant continuum: Consider the classifier with the discriminator output space $\mathcal{Y}$ defined by (5.1), given the j th example $\mathbf{X}^j$ as its input. The example's class label is $\mathcal{W}^j$. The classifier's discriminant continuum, given $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, is an imaginary line drawn between two particular, opposite vertices of $\mathcal{Y}$. The "incorrect" vertex of $\mathcal{Y}$ is the point $\mathbf{Y}_{\text{incorrect}}$ at which the discriminator output associated with the class $\mathcal{W}^j$ has a minimum value; all other discriminator outputs have a maximum value.³

$$\mathbf{Y}_{\text{incorrect}} \triangleq \langle y_1, \dots, y_C \rangle : y_i = \begin{cases} l, & \mathcal{W}^j = \omega_i \\ h, & \text{otherwise} \end{cases} \quad (5.2)$$

The opposite "correct" vertex of $\mathcal{Y}$ is the point $\mathbf{Y}_{\text{correct}}$ at which the discriminator output associated with the class $\mathcal{W}^j$ has a maximum value; all other discriminator outputs have a minimum value:

$$\mathbf{Y}_{\text{correct}} \triangleq \langle y_1, \dots, y_C \rangle : y_i = \begin{cases} h, & \mathcal{W}^j = \omega_i \\ l, & \text{otherwise} \end{cases} \quad (5.3)$$

The discriminant continuum is the line between $\mathbf{Y}_{\text{incorrect}}$ and $\mathbf{Y}_{\text{correct}}$:

$$\{\mathbf{Y} : \mathbf{Y} = \alpha \cdot \mathbf{Y}_{\text{correct}} + (1 - \alpha) \cdot \mathbf{Y}_{\text{incorrect}}\}; \quad 0 \leq \alpha \leq 1 \quad (5.4)$$

Remark: Note that the discriminant continuum is a notion that is tied to specific examples of the random feature vector: each of these examples has an associated class label $\mathcal{W}^j$, and this class label determines the specific mathematical expression for the discriminant continuum via (5.2) – (5.4). Definitions 5.2 – 5.9 are tied to specific examples of the random feature vector in precisely the same manner.

³Recall from section 2.2.4 that we use the notation y_r to denote the discriminator output associated with the class $\mathcal{W}^j = \omega_r$; we use the notation $\bar{y}_r$ to denote the largest *other* discriminator output. We remind the reader that we rely on these notational conventions throughout the text.

Definition 5.2 Reduced discriminator output space: Consider the classifier with the discriminator output space $\mathcal{Y}$ defined by (5.1), given the j th example $\mathbf{X}^j$ as its input. The example's class label is $\mathcal{W}^j$. If we re-express discriminator output space thus

$$\mathcal{Y} = \mathcal{Y}_1 \times \dots \times \mathcal{Y}_C; \quad \mathcal{Y}_i = [l, h], \quad (5.5)$$

reduced discriminator output space, given $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, is the 2-dimensional sub-space of $\mathcal{Y}$ comprising the domain $\mathcal{Y}_\tau$ of the discriminator output y_τ (corresponding to $\mathcal{W}^j = \omega_\tau$) and the domain $\overline{\mathcal{Y}_\tau}$ of the discriminator's largest other output $\overline{y}_\tau$. Mathematically, reduced discriminator output space for the example/class label pair $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$ is given by

$$\begin{aligned} & \mathcal{Y}_\tau \times \overline{\mathcal{Y}_\tau}; \\ & \mathcal{W}^j = \omega_\tau, \quad y_\tau \in \mathcal{Y}_\tau, \quad \overline{y}_\tau \in \overline{\mathcal{Y}_\tau}, \quad \overline{y}_\tau = \max_{k \neq \tau} y_k \end{aligned} \quad (5.6)$$

Example 5.1 Figure 5.1 illustrates reduced discriminator output space for a hypothetical classifier with C discriminator outputs that take on values between zero and one (i.e., $\mathcal{Y} = [l = 0, h = 1]^C$). Three training examples are projected onto the space as gray dots. Each example $\mathbf{X}^j$ elicits an output state $\mathcal{G}(\mathbf{X}^j | \theta) = \{g_1(\mathbf{X}^j | \theta), \dots, g_C(\mathbf{X}^j | \theta)\} = \{y_1, \dots, y_C\}$ in the discriminator. Given the j th training example, the position of the dot along the horizontal axis denotes the value of the discriminator output $y_\tau = g_\tau(\mathbf{X}^j | \theta)$, which corresponds to the example's class label $\mathcal{W}^j = \omega_\tau$; the position of the dot along the vertical axis denotes the value of the largest other discriminator output $\overline{y}_\tau = \max_{k \neq \tau} g_k(\mathbf{X}^j | \theta)$.

Definition 5.3 The reduced discriminant continuum: The reduced discriminant continuum is the projection of the discriminant continuum (definition 5.1) onto reduced discriminator output space (definition 5.2). Consider the classifier with the discriminator output space $\mathcal{Y}$ defined by (5.1). Given the j th example/class label pair $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, the discriminator outputs y_τ and $\overline{y}_\tau$ and their corresponding domains $\mathcal{Y}_\tau$ and $\overline{\mathcal{Y}_\tau}$ are determined by the example's class label $\mathcal{W}^j = \omega_\tau$ and the discriminator output state $\mathcal{G}(\mathbf{X}^j | \theta)$. The reduced discriminant continuum is the line between 1) the point in reduced discriminator output space for which y_τ takes on its minimum value and $\overline{y}_\tau$ takes on its maximum value, and 2) the point in reduced discriminator output space for which y_τ takes on its maximum value and $\overline{y}_\tau$ takes on its minimum value. In vector notation, the reduced discriminant continuum is given by

$$\begin{aligned} & \left\{ \begin{bmatrix} y_\tau \\ \overline{y}_\tau \end{bmatrix} : \begin{bmatrix} y_\tau \\ \overline{y}_\tau \end{bmatrix} = \alpha \cdot \begin{bmatrix} h \\ l \end{bmatrix} + (1 - \alpha) \cdot \begin{bmatrix} l \\ h \end{bmatrix} \right\}; \\ & \mathcal{W}^j = \omega_\tau, \quad 0 \leq \alpha \leq 1 \end{aligned} \quad (5.7)$$

Remark: We often abuse terminology by using the terms (reduced) discriminant continuum and (reduced) discriminator output space synonymously. We assume that the reader understands that the two concepts are inextricably linked, so each term implies the other.

Example 5.2 Figure 5.1 illustrates the reduced discriminant continuum on the reduced discriminator output space of our hypothetical classifier. It is the line between the point $\langle y_\tau = 0, \bar{y}_\tau = 1 \rangle$ and the point $\langle y_\tau = 1, \bar{y}_\tau = 0 \rangle$. The line is offset by dashed lines for clarity, and it is labelled “Discriminant Continuum” rather than “Reduced Discriminant Continuum” for the sake of simplicity.

Remark: Intuitively, the discriminant continuum and its reduced counterpart represent a line between the worst possible incorrect classification and the best possible correct classification of an example. Our use of the terms “worst” and “best” are quantitative in the following sense: the worst possible incorrect classification occurs when $\mathbf{Y} = \mathbf{Y}_{\text{incorrect}}$ — the discriminator output corresponding to the example’s class label is minimum and all the other outputs (corresponding to incorrect classifications of the example) are maximum; the best possible correct classification occurs when $\mathbf{Y} = \mathbf{Y}_{\text{correct}}$ — the discriminator output corresponding to the example’s class label is maximum and all the other outputs (corresponding to incorrect classifications of the example) are minimum.

Definition 5.4 The discriminant boundary: Consider the classifier with the discriminator output space $\mathcal{Y}$ defined by (5.1), given the j th example $\mathbf{X}^j$ as its input. The example’s class label is $\mathcal{W}^j$. The discriminant boundary is the set of all discriminator output states, given $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, for which the output y_τ (corresponding to the example’s class label $\mathcal{W}^j = \omega_\tau$) is equal to the largest other output $\bar{y}_\tau$ and greater than or equal to all other discriminator outputs:

$$\{\mathbf{Y} : y_\tau = \bar{y}_\tau \cap y_\tau \geq y_k \forall k \neq \tau\}; \mathcal{W}^j = \omega_\tau \quad (5.8)$$

Definition 5.5 The reduced discriminant boundary: The reduced discriminant boundary is the projection of the discriminant boundary onto reduced discriminator output space:

$$\{\langle y_\tau, \bar{y}_\tau \rangle : y_\tau = \bar{y}_\tau\}; \mathcal{W}^j = \omega_\tau \quad (5.9)$$

Example 5.3 Figure 5.1 illustrates the reduced discriminant boundary on the reduced discriminator output space of our hypothetical classifier. It is the line between the point $\langle y_\tau = 0, \bar{y}_\tau = 0 \rangle$ and the point $\langle y_\tau = 1, \bar{y}_\tau = 1 \rangle$. The line is labelled “Discriminant Boundary” rather than “Reduced Discriminant Boundary” for the sake of simplicity.

Definition 5.6 The “incorrect” side of discriminator output space $\mathcal{Y}_{incorrect}$: Consider the classifier with the discriminator output space $\mathcal{Y}$ defined by (5.1), given the j th example $\mathbf{X}^j$ as its input. The example’s class label is $\mathcal{W}^j$. The “incorrect” side of discriminator output space, given $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, is the set of all discriminator output states for which the output y_τ (corresponding to the example’s class label $\mathcal{W}^j = \omega_\tau$) is not maximal. Mathematically, the incorrect side of discriminator output space is given by

$$\mathcal{Y}_{incorrect} \triangleq \{Y : y_\tau \leq y_k \text{ for some } k \neq \tau\}; \mathcal{W}^j = \omega_\tau \quad (5.10)$$

Definition 5.7 The “incorrect” side of reduced discriminator output space: The incorrect side of reduced discriminator output space is the projection of the incorrect side of discriminator output space onto reduced discriminator output space:

$$\{(y_\tau, \bar{y}_\tau) : y_\tau \leq \bar{y}_\tau\}; \mathcal{W}^j = \omega_\tau \quad (5.11)$$

Incorrect space: We sometimes use the term “incorrect space” to denote the incorrect side of discriminator (and reduced discriminator) output space.

Example 5.4 Figure 5.1 illustrates the incorrect side of reduced discriminator output space for our hypothetical classifier. It is the region above and to the left of the reduced discriminant boundary.

Definition 5.8 The “correct” side of discriminator output space $\mathcal{Y}_{correct}$: Consider the classifier with the discriminator output space $\mathcal{Y}$ defined by (5.1), given the j th example $\mathbf{X}^j$ as its input. The example’s class label is $\mathcal{W}^j$. The “correct” side of discriminator output space, given $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, is the set of all discriminator output states for which the output y_τ (corresponding to the example’s class label $\mathcal{W}^j = \omega_\tau$) is greater than all other outputs. Mathematically, the correct side of discriminator output space is given by

$$\mathcal{Y}_{correct} \triangleq \{Y : y_\tau > y_k \forall k \neq \tau\}; \mathcal{W}^j = \omega_\tau \quad (5.12)$$

Definition 5.9 The “correct” side of reduced discriminator output space: The correct side of reduced discriminator output space is the projection of the correct side of discriminator output space onto reduced discriminator output space:

$$\{(y_\tau, \bar{y}_\tau) : y_\tau > \bar{y}_\tau\}; \mathcal{W}^j = \omega_\tau \quad (5.13)$$

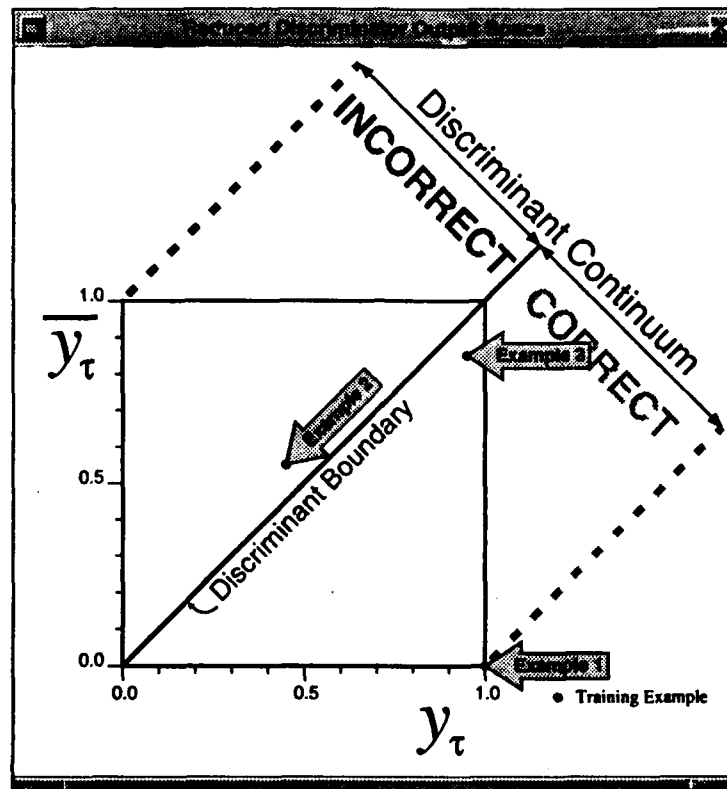


Figure 5.1: Reduced discriminator output space for a hypothetical classifier with C outputs that take on values between zero and one. The reduced space is 2-dimensional: the abscissa corresponds to y_τ , the discriminator output corresponding to the class label of the example that the classifier is processing; the ordinate corresponds to $\bar{y}_\tau$, the largest *other* discriminator output. The discriminator output states generated by three different hypothetical examples are projected onto this space. Examples 1 and 3 are correctly classified since they generate a discriminator output state in which the output associated with the example's class is larger than all other outputs (i.e., $y_\tau > \bar{y}_\tau$). Example 2 is incorrectly classified since it generates a discriminator output state in which the output associated with the example's class is smaller than at least one other output (i.e., $y_\tau < \bar{y}_\tau$).

Correct space: We sometimes use the term "correct space" to denote the correct side of discriminator (and reduced discriminator) output space.

Example 5.5 Figure 5.1 illustrates the correct side of reduced discriminator output space for our hypothetical classifier. It is the region below and to the right of the reduced discriminant boundary.

5.2.1 The Discriminant Differential δ_τ , the Reduced Discriminant Continuum, and the Reduced Discriminant Boundary

Recall from (2.22) that the discriminant differential δ_τ for the example/class label pair $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$ is given by

$$\underbrace{\delta_\tau(\mathbf{X}^j | \theta)}_{\delta_\tau} \triangleq \underbrace{g_\tau(\mathbf{X}^j | \theta)}_{y_\tau} - \underbrace{\max_{k \neq \tau} g_k(\mathbf{X}^j | \theta)}_{\bar{y}_\tau}; \quad \mathcal{W}^j = \omega_\tau \quad (5.14)$$

Note that the set of all reduced discriminator output states corresponding to a specific value of δ_τ is given by

$$\{ \langle y_\tau, \bar{y}_\tau \rangle : y_\tau = \bar{y}_\tau + \delta_\tau \}; \quad \mathcal{W}^j = \omega_\tau, \quad (5.15)$$

which, but for a constant, is identical to the expression for the reduced discriminant boundary in (5.9). Thus, all examples that generate the same discriminant differential lie on a line that is parallel to the reduced discriminant continuum in reduced discriminator output space.

Indeed, δ_τ is the Euclidean distance between the classifier's reduced discriminator output state and the reduced discriminant boundary; equivalently, it is the projection of the classifier's reduced discriminator output state onto the reduced discriminant continuum. Figure 5.2 illustrates these relationships for the reduced discriminator output space and examples shown in figure 5.1. The lower left part of the figure shows the domain of δ_τ : the diagonal gray lines at intervals of 0.2 project up onto reduced discriminator output space and the reduced discriminant continuum. Note that these lines are parallel to the reduced discriminant boundary.

Positive discriminant differentials δ_τ indicate correct classifications: *Positive values of δ_τ correspond to correct classifications (i.e., classifier output states that lie in correct space). Non-positive values of δ_τ correspond to incorrect classifications (i.e., classifier output states that lie in incorrect space).*

Since the classifier represented in figure 5.2 has discriminator outputs that are bounded on $[0,1]$, δ_τ is bounded on $[-1,1]$.

The lower left of figure 5.2 also shows $\sigma[\delta_\tau, \psi]$, the CFM of δ_τ , given a confidence parameter value of 0.4. Since CFM is a strictly non-decreasing function of δ_τ and all reduced discriminator output states generating a specific value of δ_τ lie on a line that is parallel to the reduced discriminant boundary, the CFM objective function has contours of constant value that are parallel to the reduced discriminant boundary.

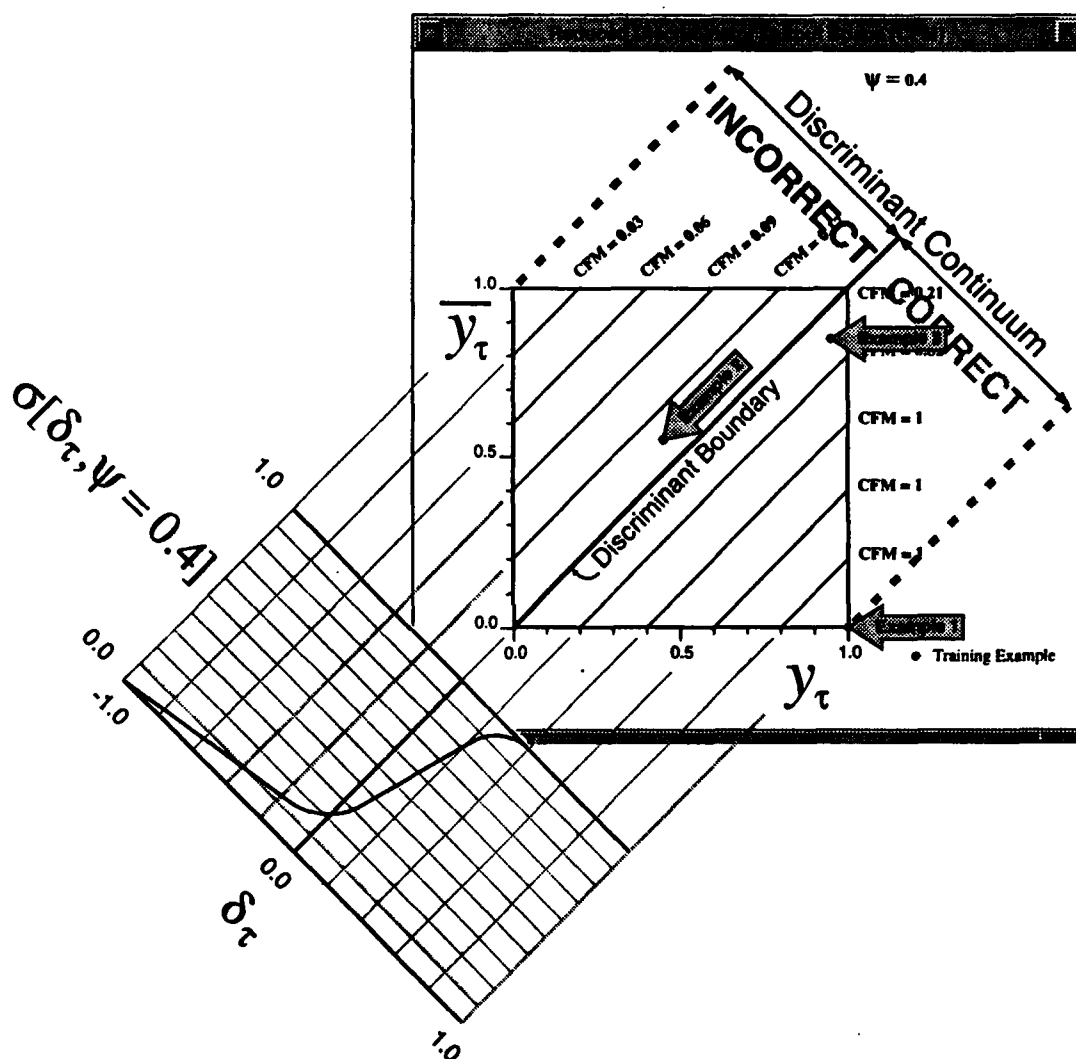


Figure 5.2: An illustration of the discriminant differential δ_τ and its relationship to reduced discriminator output space. Examples that generate the same discriminant differential lie on a line that is parallel to the reduced discriminant boundary. Since CFM (i.e., $\sigma[\delta_\tau, \psi]$) is a strictly non-decreasing function of δ_τ , the contours of constant CFM lie parallel to the reduced discriminant boundary — a necessary characteristic of the monotonic objective function.

5.3 Objective Function Monotonicity and Learning Efficiency

The objective function *must* be *monotonic* if it is to engender efficient learning regardless of the choice of hypothesis class.

Definition 5.10 A monotonic objective function: A monotonic objective function is **always** a strictly decreasing (or increasing) function of the classifier's empirical training sample error rate.

Remark: If the monotonic objective function is a strictly decreasing function of the classifier's empirical training sample error rate (e.g., CFM), then learning is accomplished by maximizing the objective function with respect to the discriminator parameters, given the training sample. If the monotonic objective function is a strictly increasing function of the classifier's empirical training sample error rate (e.g., one minus CFM), then learning is accomplished by minimizing the objective function with respect to the discriminator parameters, given the training sample.

A necessary condition for monotonicity: We use the notation $\Phi(Y)$ to denote the value of the objective function, given the discriminator output state Y . In order for Φ to be monotonic on $\mathcal{Y}$, it must exhibit a more optimal value for every discriminator output state in correct space than it exhibits for any value in incorrect space. Mathematically, Φ must satisfy

$$\Phi(Y) \text{ is more optimal than } \Phi(Y') \quad \forall (Y, Y') : Y \in \mathcal{Y}_{\text{correct}}, Y' \in \mathcal{Y}_{\text{incorrect}} \quad (5.16)$$

in order to be monotonic on $\mathcal{Y}$. We stress that (5.16) is a necessary condition for monotonicity, but it is not sufficient. The sufficient conditions for true monotonicity are discussed in section 5.3.6.

Clearly, (5.16) must hold if the objective function is monotonic, otherwise some discriminator output states in correct space will generate less optimal values of the objective function than other output states in incorrect space. If Φ fails to satisfy (5.16) it is surely non-monotonic.

Definition 5.11 A non-monotonic objective function: A non-monotonic objective function is **not** always a strictly decreasing (or increasing) function of the classifier's empirical training sample error rate.

Error measures are non-monotonic because their contours of constant value are not parallel to the discriminant boundary. They become increasingly non-monotonic — and probabilistic learning becomes increasingly inefficient — as the number of classes C increases. In order to prove this, we need to define some sub-sets of discriminator output space, define some associated measures, and present a few lemmas. We provide examples in support of the definitions. These examples are associated with a classifier having $C = 2$ discriminant functions on the space $\mathcal{Y} = [l = 0, h = 1]^2$; the classifier learns probabilistically

via the MSE objective function, and is illustrated in figure 5.3. Contours of constant MSE are projected onto discriminator output space, forming the concentric circular arcs in the figure.

Cardinality: We denote the cardinality of a set (i.e., the number of elements in the set) by $|\cdot|$. When the set is an uncountable space — as $\mathcal{Y}$ is in (5.1) — we express cardinalities as volumes. Note that

$$|\mathcal{Y}| = (h - l)^C, \quad (5.17)$$

given $\mathcal{Y}$ in (5.1).

Definition 5.12 **Correct fraction of discriminator output space $\mathcal{CF}$:** We denote the correct fraction of discriminator output space by $\mathcal{CF}$. If we think of discriminator output space and its associated sub-spaces as sets of points, $\mathcal{CF}$ is the ratio of two cardinalities associated with two sets: the numerator is the cardinality of the set of all discriminator output states in correct space $\mathcal{Y}_{\text{correct}}$; the denominator is the cardinality of discriminator output space $\mathcal{Y}$.

$$\mathcal{CF} \triangleq \frac{|\mathcal{Y}_{\text{correct}}|}{|\mathcal{Y}|} \quad (5.18)$$

Lemma 5.1 The correct fraction of discriminator output space $\mathcal{Y}_{\text{correct}}$ decreases as C^{-1} for all $C \geq 2$.

Proof : Given the expression for correct space in (5.12), its cardinality (i.e., volume) is given by

$$\begin{aligned} |\mathcal{Y}_{\text{correct}}| &= \int_l^h \underbrace{\int_l^{y_\tau} d\alpha_1 \cdots \int_l^{y_\tau} d\alpha_{C-1}}_{C-1 \text{ integral terms}} dy_\tau \\ &= \frac{1}{C} \cdot (h - l)^C \end{aligned} \quad (5.19)$$

where $\alpha_1 \dots \alpha_{C-1}$ are simply the $C - 1$ dummy variables of integration for the discriminator outputs *not* associated with y_τ . By (5.1) and (5.17) – (5.19), $\mathcal{CF}$ is therefore

$$\mathcal{CF} = \frac{1}{C} \quad (5.20)$$

■

Example 5.6 For the $C = 2$ -class task depicted in figure 5.3, correct space comprises one half of discriminator output space (i.e., $\mathcal{CF} = \frac{1}{C} = \frac{1}{2}$).

Definition 5.13 *Incorrect fraction of discriminator output space $\mathcal{IF}$:* We denote the incorrect fraction of discriminator output space by $\mathcal{IF}$. It is the ratio of two cardinalities: the numerator is the cardinality of the set of all discriminator output states in incorrect space $\mathcal{Y}_{\text{incorrect}}$; the denominator is the cardinality of discriminator output space $\mathcal{Y}$.

$$\mathcal{IF} \triangleq \frac{|\mathcal{Y}_{\text{incorrect}}|}{|\mathcal{Y}|} \quad (5.21)$$

or, equivalently,

$$\mathcal{IF} = 1 - \mathcal{CF} \quad (5.22)$$

Lemma 5.2 *The incorrect fraction of discriminator output space $\mathcal{Y}_{\text{incorrect}}$ increases as $\frac{C-1}{C}$ for all $C \geq 2$.*

Proof : The proof follows immediately from (5.20) and (5.22). ■

Example 5.7 For the $C = 2$ -class task depicted in figure 5.3, incorrect space comprises one half of discriminator output space (i.e., $\mathcal{IF} = \frac{C-1}{C} = \frac{1}{2}$).

Definition 5.14 *Non-monotonic correct fraction of discriminator output space $\mathcal{CF}_{\text{non-mono}}$:* We denote the non-monotonic correct fraction of discriminator output space by $\mathcal{CF}_{\text{non-mono}}$. It is the ratio of two cardinalities: the numerator is the cardinality of the set of all discriminator output states in correct space $\mathcal{Y}_{\text{correct}}$ for which there is at least one discriminator output state in incorrect space $\mathcal{Y}_{\text{incorrect}}$ that generates a more optimal objective function value; the denominator is the cardinality of discriminator output space.

$$\mathcal{CF}_{\text{non-mono}} \triangleq \frac{|\{Y : Y \in \mathcal{Y}_{\text{correct}} \cap \exists Y' \in \mathcal{Y}_{\text{incorrect}} \text{ s.t. } \Phi(Y) \text{ is less optimal than } \Phi(Y')\}|}{|\mathcal{Y}|} \quad (5.23)$$

Example 5.8 For the $C = 2$ -class task depicted in figure 5.3, the non-monotonic region of correct space is the light gray shaded region below and to the right of the discriminant boundary. The fraction of discriminator output space that this region encompasses is $\mathcal{CF}_{\text{non-mono}}$; in this particular case, $\mathcal{CF}_{\text{non-mono}} \cong 0.107$.

Definition 5.15 *Monotonic correct fraction of discriminator output space $\mathcal{CF}_{\text{mono}}$:* We denote the monotonic correct fraction of discriminator output space by $\mathcal{CF}_{\text{mono}}$. It is the ratio of two cardinalities: the numerator is the cardinality of the set of all discriminator output states in correct space $\mathcal{Y}_{\text{correct}}$ for which all discriminator output states in incorrect space $\mathcal{Y}_{\text{incorrect}}$ generate less optimal objective function values; the denominator is the cardinality of discriminator output space.

$$CF_{mono} \triangleq \frac{|\{Y : Y \in \mathcal{Y}_{correct} \cap \Phi(Y) \text{ is more optimal than } \Phi(Y'') \forall Y'' \in \mathcal{Y}_{incorrect}\}|}{|\mathcal{Y}|} \quad (5.24)$$

or, equivalently,

$$CF_{mono} = CF - CF_{\neg mono} \quad (5.25)$$

Example 5.9 For the $C = 2$ -class task depicted in figure 5.3, the monotonic region of correct space is the unshaded region in correct space (i.e., the unshaded region below and to the right of the discriminant boundary). The fraction of discriminator output space that this region encompasses is CF_{mono} : in this particular case, $CF_{mono} = CF - CF_{\neg mono} \cong 0.393$.

Definition 5.16 **Non-monotonic incorrect fraction of discriminator output space $IF_{\neg mono}$:** We denote the non-monotonic incorrect fraction of discriminator output space by $IF_{\neg mono}$. It is the ratio of two cardinalities: the numerator is the cardinality of the set of all discriminator output states in incorrect space $\mathcal{Y}_{incorrect}$ for which there is at least one discriminator output state in correct space $\mathcal{Y}_{correct}$ that generates a less optimal objective function value; the denominator is the cardinality of discriminator output space.

$$IF_{\neg mono} \triangleq \frac{|\{Y : Y \in \mathcal{Y}_{incorrect} \cap \exists Y''' \in \mathcal{Y}_{correct} \text{ s.t. } \Phi(Y) \text{ is more optimal than } \Phi(Y''')\}|}{|\mathcal{Y}|} \quad (5.26)$$

Example 5.10 For the $C = 2$ -class task depicted in figure 5.3, the non-monotonic region of incorrect space is the dark gray shaded region above and to the left of the discriminant boundary. The fraction of discriminator output space that this region encompasses is $IF_{\neg mono}$: in this particular case, $IF_{\neg mono} \cong 0.285$.

Definition 5.17 **Monotonic incorrect fraction of discriminator output space IF_{mono} :** We denote the monotonic incorrect fraction of discriminator output space by IF_{mono} . It is the ratio of two cardinalities: the numerator is the cardinality of the set of all discriminator output states in incorrect space $\mathcal{Y}_{incorrect}$ for which all discriminator output states in correct space $\mathcal{Y}_{correct}$ generate more optimal objective function values; the denominator is the cardinality of discriminator output space.

$$IF_{mono} \triangleq \frac{|\{Y : Y \in \mathcal{Y}_{incorrect} \cap \Phi(Y''') \text{ is more optimal than } \Phi(Y) \forall Y''' \in \mathcal{Y}_{correct}\}|}{|\mathcal{Y}|} \quad (5.27)$$

or, equivalently,

$$IF_{mono} = IF - IF_{\neg mono} \quad (5.28)$$

Example 5.11 For the $C = 2$ -class task depicted in figure 5.3, the monotonic region of incorrect space is the unshaded region in incorrect space (i.e., the unshaded region above and to the left of the discriminant boundary). The fraction of discriminator output space that this region encompasses is IF_{mono} : in this particular case, $IF_{mono} = IF - IF_{\neg mono} \cong 0.215$.

It should be clear from definitions 5.14 – 5.17 that the monotonic and non-monotonic fractions of discriminator output space sum to one as follows:

$$\underbrace{IF_{\neg mono} + IF_{mono}}_{IF} + \underbrace{CF_{\neg mono} + CF_{mono}}_{CF} = 1 \quad (5.29)$$

Definition 5.18 **Monotonic fraction of discriminator output space $\mathcal{MF}$:** We denote the monotonic fraction of discriminator output space by $\mathcal{MF}$. It is the sum of the monotonic correct and incorrect fractions:

$$\begin{aligned} \mathcal{MF} &\triangleq IF_{mono} + CF_{mono} \\ \text{s.t. } 0 &\leq \mathcal{MF} \leq 1 \end{aligned} \quad (5.30)$$

Example 5.12 For the $C = 2$ -class task depicted in figure 5.3, the monotonic region of discriminator output space is the unshaded region (i.e., the combined unshaded regions on both sides of the discriminant boundary). The fraction of discriminator output space that this region encompasses is $\mathcal{MF}$: in this particular case, $\mathcal{MF} \cong 0.608$.

We measure an objective function's monotonicity — or lack thereof — by its monotonic fraction $\mathcal{MF}$, given discriminator output space $\mathcal{Y}$. If the objective function is monotonic, the monotonic fraction $\mathcal{MF}$ is unity; likewise, the monotonic fractions of incorrect (IF_{mono}) and correct (CF_{mono}) spaces are equal to IF and CF , respectively. If the objective function is non-monotonic, $\mathcal{MF}$ is less than unity; likewise, IF_{mono} and/or CF_{mono} are less than IF and/or CF , respectively. Simply put, objective functions with lower values of $\mathcal{MF}$, IF_{mono} , and CF_{mono} are increasingly non-monotonic.

Intuitively, one can view the monotonic fraction $\mathcal{MF}$ as a kind of correlation coefficient between the act of optimizing the objective function and the act of minimizing the classifier's empirical training sample error rate, absent any specific knowledge regarding whether or not the classifier's hypothesis class is a proper parametric model of the feature vector. If $\mathcal{MF} = 1$, every discriminator output state in correct space generates a more optimal value of the objective function than any output state in incorrect space; as a result, optimizing the objective function is monotonically related to minimizing the training sample error

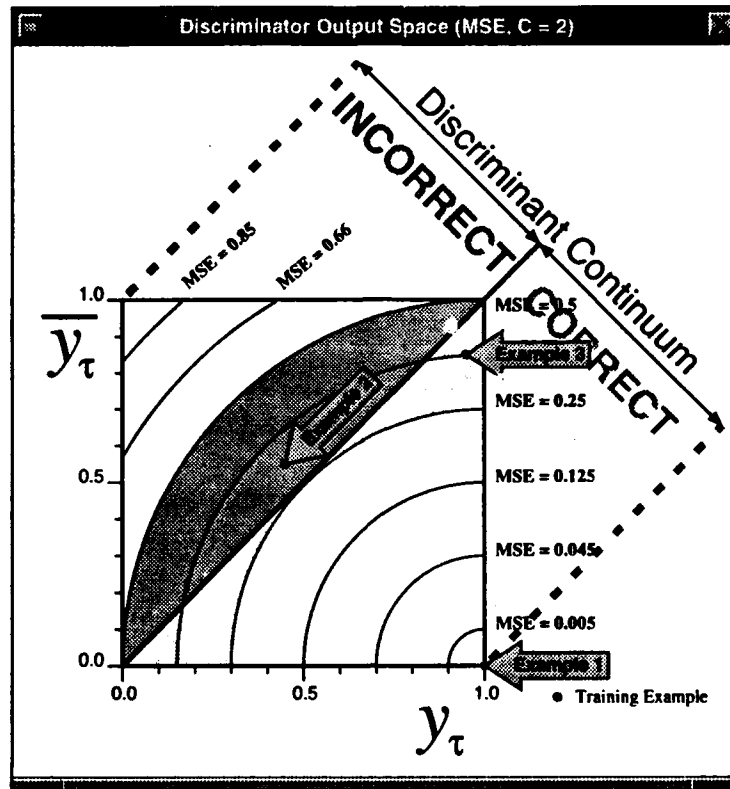


Figure 5.3: The MSE objective function is non-monotonic. Example 3 generates higher (less optimal) MSE than example 2, even though example 3 is correctly classified, whereas example 2 is not. For every discriminator output state in the light gray shaded region of correct space there is at least one output state in the dark gray shaded region of incorrect space with lower MSE. The figure depicts discriminator output space for a hypothetical $C = 2$ -class task in which the discriminator's outputs are bounded on $\mathcal{Y} = [l = 0, h = 1]^2$. Since the classifier has two discriminant functions, discriminator output space and reduced discriminator output space are one and the same.

rate, regardless of the choice of hypothesis class. If, on the other hand, $\mathcal{MF} < 1$, some discriminator output states in correct space generate less optimal values of the objective function than other output states in incorrect space; as a result, optimizing the objective function does not necessarily minimize the training sample error rate — a phenomenon we discuss further in section 5.3.5.

5.3.1 MAE is Non-Monotonic

The mean absolute error (MAE) measure⁴ discussed in section 2.3.3 is the sum of the absolute difference between each discriminator output y_i and its target value τ_i :

⁴Recall that the mean absolute error measure is known by other names such as least absolute error (LAE) and least absolute deviation (LAD, e.g., [9]).

$$\text{MAE} = \sum_{i=1}^c \xi[y_i, \tau_i] \quad (5.31)$$

Given the training example/class label pair $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, the target value for the output y_τ (corresponding to the class label $\mathcal{W}^j$) is D , and the target values for all the other outputs are $\neg D$:

$$\xi[y_i, \tau_i] = \begin{cases} \underbrace{h}_{D} - y_i, & \mathcal{W}^j = \omega_\tau = \omega_i \text{ s.t. } \tau_i = D = h \\ y_i - \underbrace{l}_{\neg D}, & \mathcal{W}^j \neq \omega_\tau \text{ s.t. } \tau_i = \neg D = l \end{cases} \quad (5.32)$$

In the following arguments, we assume that y_τ is always y_1 in order to simplify notation. Under this condition, (5.31) and (5.32) reduce to

$$\begin{aligned} \text{MAE} &= h - y_1 + \sum_{j=2}^c (y_j - l) \\ &= h - y_1 - (C - 1) \cdot l + \sum_{j=2}^c y_j \end{aligned} \quad (5.33)$$

The maximum MAE generated by a correctly classified example of ω_1 occurs at the vertex of correct space farthest from $\mathbf{Y}_{\text{correct}}$. This is the output state in which $y_1 = h$ and all the other discriminator outputs are smaller by an infinitesimally small positive value ϵ :

$$\max \text{MAE correct} = \lim_{\epsilon \rightarrow 0^+} h - \underbrace{h}_{y_1} + \sum_{j=2}^c \left(\underbrace{(h - \epsilon)}_{y_j} - l \right); \quad \epsilon > 0 \quad (5.34)$$

$$< (C - 1)(h - l) \quad (5.35)$$

Thus, (5.35) defines the "inner" boundary value of MAE in monotonic incorrect space. If $\mathbf{X}^j$ is an example of ω_1 , it is *surely* misclassified if $\text{MAE} \geq (C - 1)(h - l)$: by (5.33) and (5.35), the example is surely misclassified if

$$y_1 \leq \sum_{j=2}^c y_j - (C - 2) \cdot h \quad (5.36)$$

The minimum MAE generated by an incorrectly classified example of ω_1 occurs when y_τ (i.e., y_1) is infinitesimally smaller than $\bar{y}_\tau$ and all the other discriminator outputs are minimal. By (5.33),

$$\begin{aligned}
 \min \text{MAE incorrect} &= \lim_{\epsilon \rightarrow 0} h - \underbrace{y_1}_{y_r} + \underbrace{y_1 + \epsilon}_{\bar{y}_r} - l; \quad \epsilon \geq 0 \\
 &= h - l
 \end{aligned} \tag{5.37}$$

Thus, the minimum value of MAE for an incorrectly classified example occurs when y_r (i.e., y_1) equals $\bar{y}_r$, and (5.37) defines the “outer” boundary value of MAE in monotonic correct space. If $\mathbf{X}^j$ is an example of ω_1 , it is *surely* correctly classified if $\text{MAE} < (h - l)$ such that

$$y_1 > \sum_{j=2}^C y_j - (C - 2) \cdot l \tag{5.38}$$

Note that when $C = 2$, (5.33) reduces to

$$\begin{aligned}
 \text{MAE}_{C=2} &= h - \underbrace{y_1}_{y_r} + \underbrace{y_2}_{\bar{y}_r} - l \\
 &= \underbrace{h - l}_{\text{constant}} - \delta_r,
 \end{aligned} \tag{5.39}$$

and, by (5.38), a training example is *surely* correctly classified if $\delta_r > 0$. $\text{MAE}_{C=2}$ is therefore a strictly decreasing function of δ_r . In this sense, the MAE objective function is a special (and so far as we know, unique) error measure: when the discriminator has two discriminant functions associated with a $C = 2$ -class learning/classification task, the contours of constant MAE lie parallel to the discriminant boundary, as shown in figure 5.4. Since there is no discriminator output state in correct space that generates greater MAE than any discriminator output state in incorrect space, (5.16) is satisfied, and the monotonic fraction — which we denote by $\text{MAE} \mathcal{MF}(C = 2)$ — is unity.

However, the MAE objective function is, by (5.16), (5.35), and (5.37), non-monotonic for $C > 2$. Indeed, section I.1 proves that the monotonic fractions of the MAE objective function decrease with increasing values of C according to the following formulae, in which $\Gamma(\cdot)$ denotes the gamma function (e.g., [80, pp. A76-A77]):

$$\text{MAE} \mathcal{IF}_{\text{mono}}(C) = \frac{1}{\Gamma(C + 1)} \tag{5.40}$$

$$\text{MAE} \mathcal{CF}_{\text{mono}}(C) = \frac{1}{\Gamma(C + 1)} \tag{5.41}$$

$$\therefore \text{MAE} \mathcal{MF}(C) = \frac{2}{\Gamma(C + 1)} \tag{5.42}$$

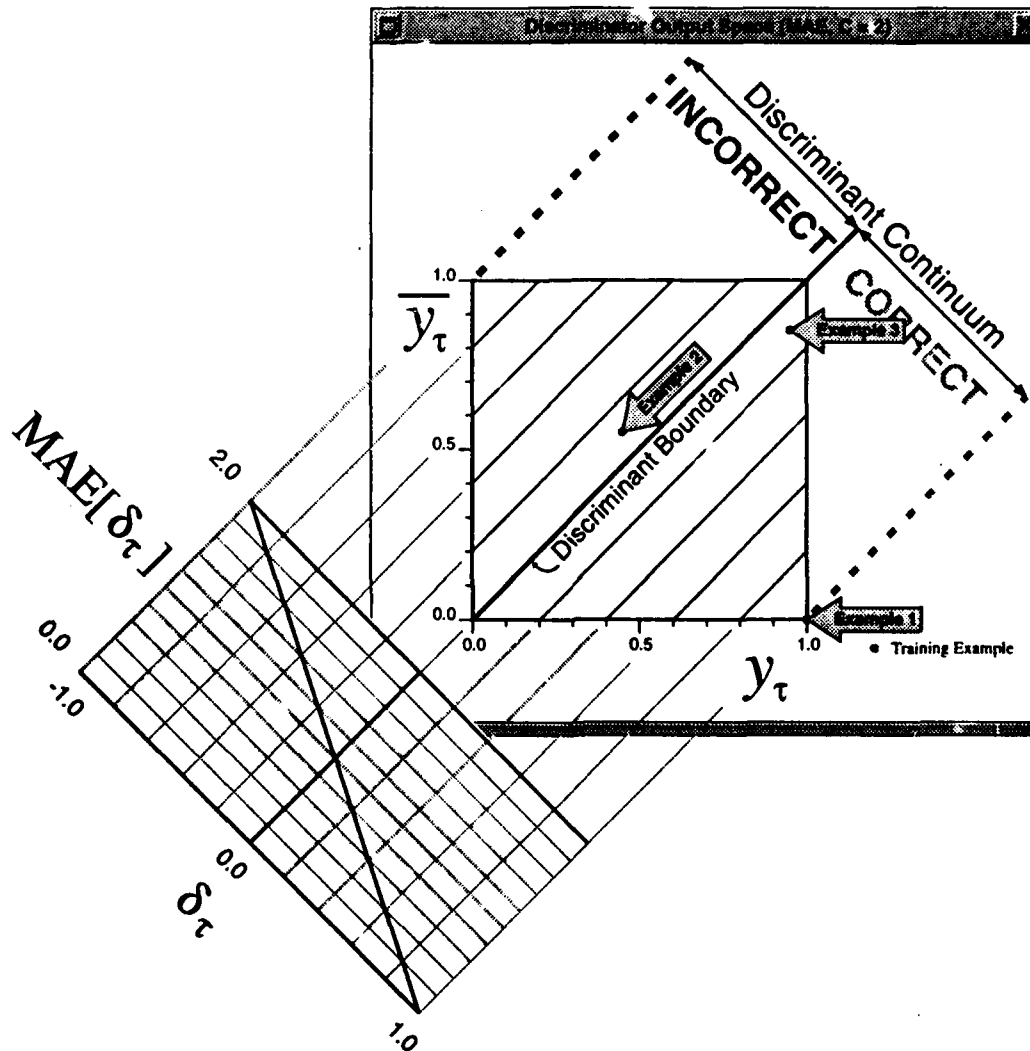


Figure 5.4: The MAE objective function can be monotonic if and only if the number of classes is $C = 2$; this is because MAE can be expressed as a strictly decreasing function of $\delta_\tau = y_\tau - \bar{y}_\tau$. As a result, the contours of constant MAE are parallel to the discriminant boundary, a necessary condition for monotonicity. The figure depicts discriminator output space for a hypothetical $C = 2$ -class task in which the discriminator's outputs are bounded on $\mathcal{Y} = [l = 0, h = 1]^2$. Since the classifier has two discriminant functions, discriminator output space and reduced discriminator output space are one and the same.

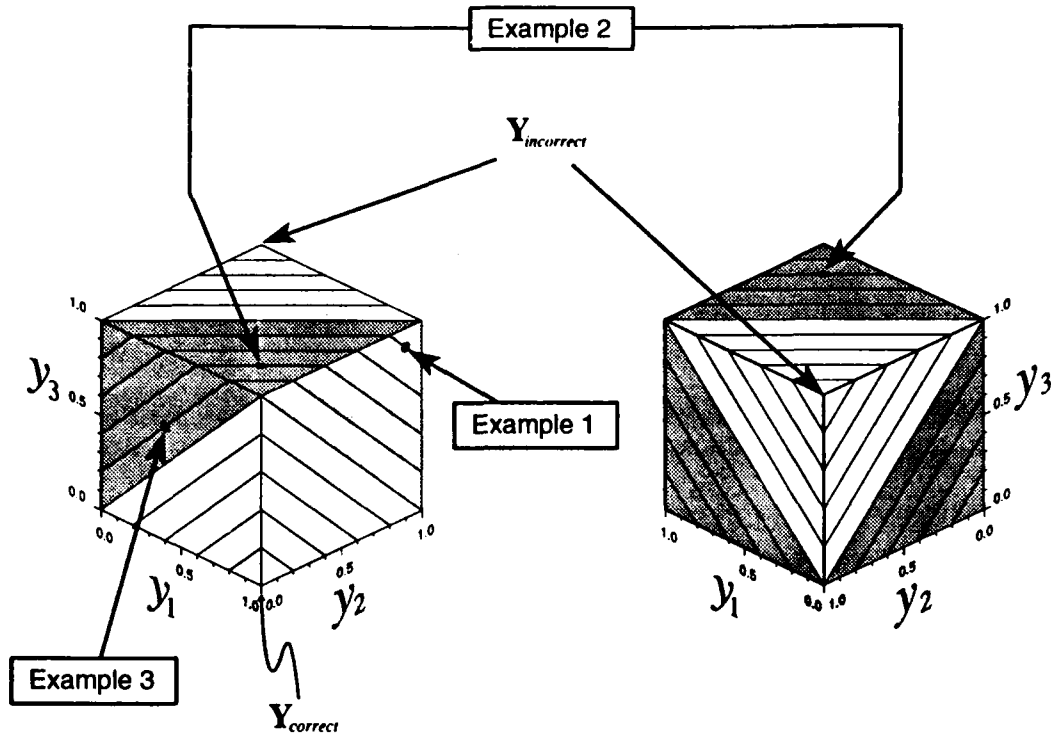


Figure 5.5: The MAE objective function is increasingly non-monotonic as C increases. Output y_1 is y_τ , the output corresponding to the class label for the three training examples shown. Contours of constant MAE are projected onto the bounding faces of discriminator output space. White denotes the monotonic regions of discriminator output space, light gray denotes the non-monotonic region of correct space, and dark gray denotes the non-monotonic region of incorrect space.^a Example 1 is correctly classified, yet it generates a higher (less optimal) value of MAE than examples 2 and 3, which are both incorrectly classified. **Left:** The discriminator output space from the perspective of $\mathbf{Y}_{correct}$. **Right:** The discriminator output space from the perspective of $\mathbf{Y}_{incorrect}$. The monotonic fractions of incorrect and correct spaces are $\frac{1}{6}$, so the monotonic fraction of discriminator output space is $\frac{1}{3}$.

^aThe light gray shading underneath the cubic form of the discriminator output space is an imaginary shadow; it helps to clarify the cube's orientation.

Example 5.13 Figure 5.5 illustrates the non-monotonic nature of MAE when $C = 3$. The figure shows two views of discriminator output space (which is $\mathcal{Y} = [l = 0, h = 1]^3$ for the purpose of illustration). As in previous figures, white denotes the monotonic regions of discriminator output space, light gray denotes the non-monotonic region of correct space, and dark gray denotes the non-monotonic region of incorrect space. The left-hand figure shows discriminator output space from the perspective of $\mathbf{Y}_{correct} = \langle y_1 = 1, y_2 = 0, y_3 = 0 \rangle$; the right-hand figure shows discriminator output space from the perspective of $\mathbf{Y}_{incorrect} = \langle y_1 = 0, y_2 = 1, y_3 = 1 \rangle$. By (5.40) – (5.42), $\text{MAE} \mathcal{IF}_{mono}(3) = \text{MAE} \mathcal{CF}_{mono}(3) = \frac{1}{6}$, and $\text{MAE} \mathcal{MF}(3) = \frac{1}{3}$. Example 1 (i.e., $\mathbf{X}^1$ generates a discriminator output state of $\mathcal{G}(\mathbf{X}^1 | \theta) = \mathbf{Y}^1 = \langle 1, .9, .9 \rangle$, so it is correctly classified; $\text{MAE}(\mathbf{Y}^1) = 1.8$. Example 2 generates a

discriminator output state of $Y^2 = (.8, .2, 1)$, so it is incorrectly classified; $MAE(Y^2) = 1.4$. Example 3 generates a discriminator output state of $Y^3 = (.4, 0, .6)$, so it is incorrectly classified; $MAE(Y^3) = 1.2$. Thus, we have the particularly undesirable situation in which the correctly classified example generates a less optimal value of MAE than the two incorrectly classified examples generate — a clear manifestation of the MAE objective function's non-monotonic nature ($C \geq 3$).

By (5.40) – (5.42), all the MAE monotonic fractions go to zero as the number of classes C grows large:

$$\begin{aligned} \lim_{C \rightarrow \infty} MAE \mathcal{IF}_{mono}(C) &= 0 \\ \lim_{C \rightarrow \infty} MAE \mathcal{CF}_{mono}(C) &= 0 \\ \therefore \lim_{C \rightarrow \infty} MAE \mathcal{MF}(C) &= 0 \end{aligned} \quad (5.43)$$

Moreover, $MAE \mathcal{MF}(C)$, $MAE \mathcal{CF}_{mono}(C)$, and $MAE \mathcal{IF}_{mono}(C)$ decrease super-exponentially as C increases. For example, when the number of classes is ten, the monotonic fractions are quite small:

$$\begin{aligned} MAE \mathcal{MF}(10) &= 5.51 \times 10^{-7} \\ MAE \mathcal{IF}_{mono}(10) &= MAE \mathcal{CF}_{mono}(10) = 2.75 \times 10^{-7} \end{aligned} \quad (5.44)$$

5.3.2 MSE is Non-Monotonic

The mean squared error (MSE) measure discussed in section 2.3.2 is given by

$$MSE = \sum_{i=1}^C \xi[y_i, \tau_i] \quad (5.45)$$

where

$$\xi[y_i, \tau_i] = \begin{cases} \frac{1}{2} \left(\underbrace{h}_D - y_i \right)^2, & \mathcal{W}^j = \omega_\tau = \omega_i \text{ s.t. } \tau_i = D = h \\ \frac{1}{2} \left(y_i - \underbrace{l}_{-D} \right)^2, & \mathcal{W}^j \neq \omega_\tau \text{ s.t. } \tau_i = -D = l \end{cases} \quad (5.46)$$

As in the preceding section, we assume that y_τ is always y_1 in order to simplify notation. Under this condition, (5.45) and (5.46) reduce to

$$MSE = \frac{1}{2} \left[(h - y_1)^2 + \sum_{j=2}^C (y_j - l)^2 \right] \quad (5.47)$$

The maximum MSE generated by a correctly classified example of ω_1 occurs at the vertex of correct space farthest from $\mathbf{Y}_{correct}$. This is the output state in which $y_1 = h$ and all the other discriminator outputs are smaller by an infinitesimally small positive value ϵ :

$$\max \text{MSE correct} = \lim_{\epsilon \rightarrow 0^+} \frac{1}{2} \left[\left(h - \underbrace{h}_{y_1} \right)^2 + \sum_{j=2}^c \left(\underbrace{(h - \epsilon)}_{y_j} - l \right)^2 \right]; \quad \epsilon > 0 \quad (5.48)$$

$$< \frac{(c-1)}{2} (h-l)^2 \quad (5.49)$$

Thus, (5.49) defines the "inner" boundary value of MSE in monotonic incorrect space. If $\mathbf{X}^j$ is an example of ω_1 , it is *surely* misclassified if $\text{MSE} \geq \frac{(c-1)}{2} (h-l)^2$: by (5.47) and (5.49), the example is surely misclassified if

$$y_1 \leq h - \left[(c-1)(h-l)^2 - \sum_{j=2}^c (y_j - l)^2 \right]^{\frac{1}{2}} \quad (5.50)$$

The minimum MSE generated by an incorrectly classified example of ω_1 occurs when y_τ (i.e., y_1) is infinitesimally smaller than $\bar{y}_\tau$ and all the other discriminator outputs are minimal. By (5.47),

$$\begin{aligned} \min \text{MSE incorrect} &= \lim_{\epsilon \rightarrow 0} \frac{1}{2} \left[\left(h - \underbrace{y_1}_{y_\tau} \right)^2 + \left(\underbrace{y_1 + \epsilon}_{\bar{y}_\tau} - l \right)^2 \right]; \quad \epsilon \geq 0 \\ &= \frac{1}{2} [(h - y_1)^2 + (y_1 - l)^2], \end{aligned} \quad (5.51)$$

which is minimal when

$$\underbrace{y_\tau}_{y_1} = \bar{y}_\tau = \frac{h+l}{2} \quad (5.52)$$

Thus, (5.51) reduces to

$$\min \text{MSE incorrect} = \frac{1}{4} (h-l)^2 \quad (5.53)$$

Equation (5.53) defines the "outer" boundary value of MSE in monotonic correct space. If $\mathbf{X}^j$ is an example of ω_1 , it is *surely* correctly classified if $\text{MSE} < \frac{1}{4} (h-l)^2$ such that

$$y_1 > h - \left[\frac{1}{2} (h - l)^2 - \sum_{j=2}^C (y_j - l)^2 \right]^{\frac{1}{2}} \quad (5.54)$$

The MSE objective function is, by (5.16), (5.49), and (5.53), non-monotonic for $C \geq 2$. Indeed, section I.2 proves that the monotonic fractions of the MSE objective function decrease with increasing values of C according to the following formulae:

$$\text{MSEIF}_{\text{mono}}(C) < \frac{1}{\Gamma(C+1)} \quad (5.55)$$

$$\text{MSECF}_{\text{mono}}(C) = \frac{\left(\frac{\pi}{8}\right)^{\frac{C}{2}}}{\Gamma\left(\frac{C}{2} + 1\right)} \quad (5.56)$$

$$\therefore \text{MSEMF}(C) < \frac{\left(\frac{\pi}{8}\right)^{\frac{C}{2}}}{\Gamma\left(\frac{C}{2} + 1\right)} + \frac{1}{\Gamma(C+1)} \quad (5.57)$$

Note that (5.57) is loosely bounded from above by $\frac{2}{\Gamma(\frac{C}{2} + 1)}$ for both small and large C .

Example 5.14 Figure 5.3 illustrates the non-monotonic nature of MSE when $C = 2$ ($\mathcal{Y} = [l = 0, h = 1]^2$ for the purpose of illustration). By (5.56), $\text{MSECF}_{\text{mono}}(2) = .393$. The fraction $\text{MSEIF}_{\text{mono}}(2)$ can be computed exactly, obviating the need to use the less precise bound of (5.55): it is simply one fourth the area of a circle of unit radius, minus one half (i.e., $\text{MSEIF}_{\text{mono}}(2) = .285$). Thus, by (5.30), $\text{MSEMF}(2) = .678$. Example 1 generates a discriminator output state of $\mathbf{Y}^1 = \langle 1, 0 \rangle$, so it is correctly classified; $\text{MSE}(\mathbf{Y}^1) = 0$. Example 2 generates a discriminator output state of $\mathbf{Y}^2 = \langle .45, .55 \rangle$, so it is incorrectly classified; $\text{MSE}(\mathbf{Y}^2) = .30$. Example 3 generates a discriminator output state of $\mathbf{Y}^3 = \langle .93, .85 \rangle$, so it is correctly classified; $\text{MSE}(\mathbf{Y}^3) = .36$. The MSE generated by example 1 is minimal. However, the incorrectly classified example 2 generates a more optimal value of MSE than the correctly classified example 3 generates...

Example 5.15 Figure 5.6 illustrates the non-monotonic nature of MSE when $C = 3$. The figure shows two views of discriminator output space (which is $\mathcal{Y} = [l = 0, h = 1]^3$ for the purpose of illustration). The left-hand figure shows discriminator output space from the perspective of $\mathbf{Y}_{\text{correct}}$; the right-hand figure shows discriminator output space from the perspective of $\mathbf{Y}_{\text{incorrect}}$. By (5.55) – (5.57), $\text{MSEIF}_{\text{mono}}(3) < \frac{1}{6}$, $\text{MSECF}_{\text{mono}}(3) = .185$, and $\text{MSEMF}(3) < .352$. Example 1 generates a discriminator output state of $\mathbf{Y}^1 = \langle 1, .922, .922 \rangle$, so it is correctly classified; $\text{MSE}(\mathbf{Y}^1) = .85$. Example 2 generates a discriminator

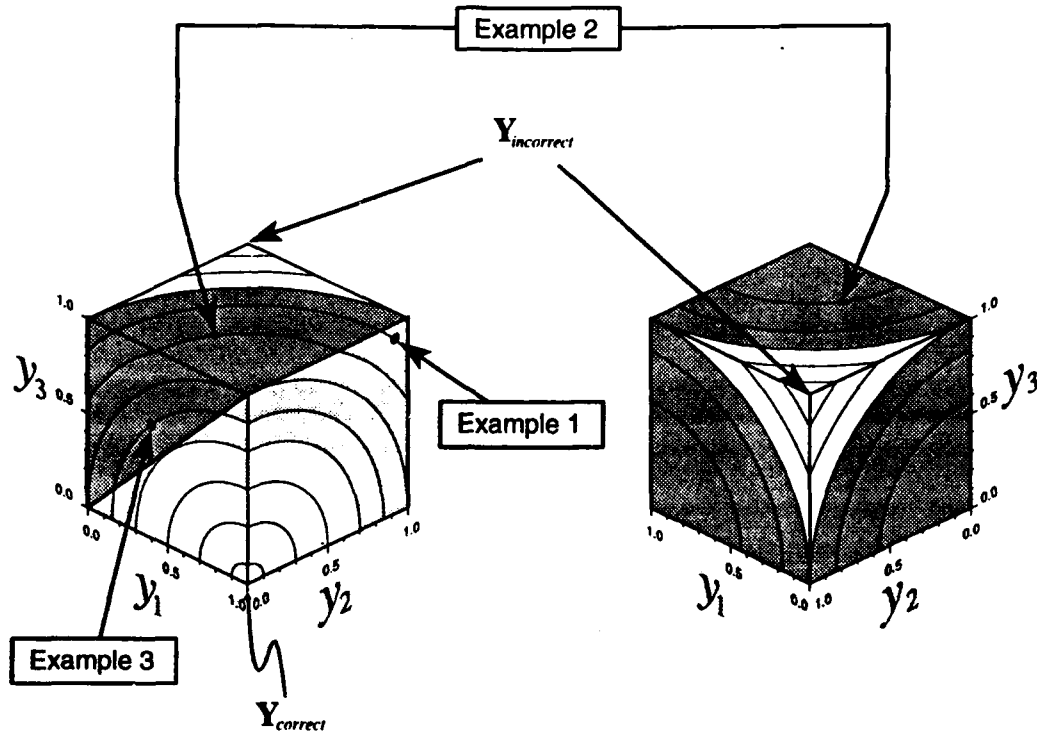


Figure 5.6: The MSE objective function is increasingly non-monotonic as C increases. Output y_1 is y_r , the output corresponding to the class label for the three training examples shown. Contours of constant MSE are projected onto the bounding faces of discriminator output space. White denotes the monotonic regions of discriminator output space, light gray denotes the non-monotonic region of correct space, and dark gray denotes the non-monotonic region of incorrect space.^a Example 1 is correctly classified, yet it generates a higher (less optimal) value of MSE than examples 2 and 3, which are both incorrectly classified. **Left:** The discriminator output space from the perspective of $Y_{correct}$. **Right:** The discriminator output space from the perspective of $Y_{incorrect}$. The monotonic fraction of incorrect space is less than $\frac{1}{6}$; the monotonic fraction of correct space is .185; the monotonic fraction of discriminator output space is therefore less than .352.

^aThe light gray shading underneath the cubic form of the discriminator output space is an imaginary shadow; it helps to clarify the cube's orientation.

output state of $Y^2 = \langle .508, .284, 1 \rangle$, so it is incorrectly classified; $MSE(Y^2) = .66$. Example 3 generates a discriminator output state of $Y^3 = \langle .399, 0, .601 \rangle$, so it is incorrectly classified; $MSE(Y^3) = .36$. Thus, the correctly classified example generates a less optimal value of MSE than the two incorrectly classified examples generate; like MAE, the MSE objective function is non-monotonic ($C \geq 2$).

By (5.55) – (5.57), all the MSE monotonic fractions go to zero as the number of classes C grows large:

$$\begin{aligned} \lim_{C \rightarrow \infty} MSEIF_{mono}(C) &= 0 \\ \lim_{C \rightarrow \infty} MSECF_{mono}(C) &= 0 \\ \therefore \lim_{C \rightarrow \infty} MSEMF(C) &= 0 \end{aligned} \tag{5.58}$$

Like their MAE counterparts, $MSEMF(C)$, $MSECF_{mono}(C)$, and $MSEIF_{mono}(C)$ decrease super-

exponentially as C increases, although $\text{MSEMF}(C)$ is $\mathcal{O} \left[\left(\Gamma \left(\frac{C}{2} + 1 \right) \right)^{-1} \right]$ whereas $\text{MAEMF}(C)$ is $\mathcal{O} \left[\left(\Gamma(C + 1) \right)^{-1} \right]$. Nevertheless, when the number of classes is ten, the monotonic fractions are still quite small:

$$\begin{aligned} \text{MSEIF}_{\text{mono}}(10) &< 2.75 \times 10^{-7} \\ \text{MSECF}_{\text{mono}}(10) &= 7.78 \times 10^{-5} \\ \text{MSEMF}(10) &< 7.81 \times 10^{-5} \end{aligned} \quad (5.59)$$

5.3.3 The Kullback-Leibler Information Distance is Non-Monotonic

The Kullback-Leibler information distance (a.k.a. cross-entropy — CE) discussed in section 2.3.2 is given by

$$\text{CE} = \sum_{i=1}^c \xi[y_i, \tau_i] \quad (5.60)$$

where

$$\xi[y_i, \tau_i] = \begin{cases} -\log \left(y_i - \underbrace{l}_{\neg D} \right), & \mathcal{W}^j = \omega_\tau = \omega_i \text{ s.t. } \tau_i = D = h \\ -\log \left(\underbrace{h}_D - y_i \right), & \mathcal{W}^j \neq \omega_\tau \text{ s.t. } \tau_i = \neg D = l \end{cases} \quad (5.61)$$

As in the preceding two sections, we assume that y_τ is always y_1 in order to simplify notation. Under this condition, (5.60) and (5.61) reduce to

$$\text{CE} = -\log(y_1 - l) - \sum_{j=2}^c \log(h - y_j) \quad (5.62)$$

The maximum CE generated by a correctly classified example of ω_1 occurs at the vertex of correct space farthest from $\mathbf{Y}_{\text{correct}}$. This is the output state in which $y_1 = h$ and all the other discriminator outputs are smaller by an infinitesimally small positive value ϵ :

$$\max \text{CE correct} = \lim_{\epsilon \rightarrow 0^+} -\log \left(\underbrace{h}_{y_1} - l \right) - \sum_{j=2}^c \log \left(h - \underbrace{(h - \epsilon)}_{y_j} \right); \quad \epsilon > 0 \quad (5.63)$$

$$< \infty \quad (5.64)$$

Thus, the "inner" boundary value of CE in monotonic incorrect space is infinite. That is, an example is never *surely* misclassified, since a correctly classified example can generate an infinite value of CE. As a result, the monotonic fraction of incorrect space is zero for all $C \geq 2$:

$$\text{CEIF}_{\text{mono}} C = 0 \quad \forall C \geq 2 \quad (5.65)$$

The minimum CE generated by an incorrectly classified example of ω_1 occurs when y_τ (i.e., y_1) is infinitesimally smaller than $\bar{y}_\tau$ and all the other discriminator outputs are minimal. By (5.62),

min CE incorrect =

$$\begin{aligned} & \lim_{\epsilon \rightarrow 0} -\log \left(\underbrace{y_1}_{y_\tau} - l \right) - \log \left(h - \underbrace{y_1}_{\bar{y}_\tau} + \epsilon \right) - (C - 2) \cdot \log(h - l); \quad \epsilon \geq 0 \\ & = -\log(y_1 - l) - \log(h - y_1) - (C - 2) \cdot \log(h - l), \end{aligned} \quad (5.66)$$

which is minimal when

$$\underbrace{y_\tau}_{y_1} = \bar{y}_\tau = \frac{h + l}{2} \quad (5.67)$$

Thus, (5.66) reduces to

$$\text{min CE incorrect} = -C \cdot \log(h - l) + \log(4) \quad (5.68)$$

Equation (5.68) defines the "outer" boundary value of CE in monotonic correct space. If $\mathbf{X}^j$ is an example of ω_1 , it is *surely* correctly classified if $\text{CE} < -C \cdot \log(h - l) + \log(4)$. Assuming a logarithmic basis of b (i.e., the notation $\log$ actually denotes $\log_b$), an example is surely correctly classified when

$$y_1 > l + b \left[C \log(h - l) - \sum_{j=2}^C \log(h - y_j) - \log(4) \right] \quad (5.69)$$

The CE objective function is, by (5.16), (5.64), and (5.68), non-monotonic for $C \geq 2$. Indeed, $\text{CEIF}_{\text{mono}}(C) = 0$ for all $C \geq 2$, by (5.65), and section I.3 proves that the two other monotonic fractions of the CE objective function decrease with increasing values of C according to the following formulae when $\mathcal{Y} = [l = 0, h = 1]^C$:

$$\text{CEMF}(C) = \text{CECF}_{\text{mono}}(C) = 1 - \lambda \cdot \underbrace{\sum_{j=0}^{C-1} \frac{(-1)^j}{j!} \cdot [\ln(\lambda)]^j}_{\zeta(C)}; \quad (5.70)$$

$$\mathcal{Y} = [l = 0, h = 1]^C, \quad \lambda = \left(\frac{h+l}{2} \right)^2 = \frac{1}{4}$$

It is straightforward to prove the following relationship

$$\lim_{C \rightarrow \infty} \zeta(C) = \exp[-\ln(\lambda)] = \frac{1}{\lambda}, \quad (5.71)$$

so all the CE monotonic fractions go to zero as the number of classes C grows large:

$$\begin{aligned} \lim_{C \rightarrow \infty} \text{CEIF}_{mono}(C) &= 0 \\ \text{CECF}_{mono}(C) &= 0 \quad \forall C \geq 2 \\ \therefore \lim_{C \rightarrow \infty} \text{CEMF}(C) &= 0 \end{aligned} \quad (5.72)$$

Expressions for $\text{CEMF}(C)$ and $\text{CECF}_{mono}(C)$ are considerably more cumbersome for the more general discriminator output space $\mathcal{Y} = [l, h]^C$ (see section 1.3). Although the specific values of $\text{CEMF}(C)$ and $\text{CECF}_{mono}(C)$ change with l and h , the general dependence on C is well-described by (5.70) as long as l and h are finite — a constraint that is consistent with (2.60). For these reasons, we omit the more general expressions.

Example 5.16 Figure 5.7 illustrates the non-monotonic nature of CE when $C = 2$ ($\mathcal{Y} = [l = 0, h = 1]^2$ for the purpose of illustration). The logarithmic basis of (5.61) is 10 (i.e., $\log(\cdot)$ denotes $\log_{10}(\cdot)$ in (5.61)). By (5.70), the monotonic fractions are $\text{CEIF}_{mono}(2) = 0$ and $\text{CEMF}(2) = \text{CECF}_{mono}(2) = .403$. Example 1 generates a discriminator output state of $\mathbf{Y}^1 = \langle 1, 0 \rangle$, so it is correctly classified; $\text{CE}(\mathbf{Y}^1) = 0$. Example 2 generates a discriminator output state of $\mathbf{Y}^2 = \langle .45, .55 \rangle$, so it is incorrectly classified; $\text{CE}(\mathbf{Y}^2) = .69$. Example 3 generates a discriminator output state of $\mathbf{Y}^3 = \langle .93, .85 \rangle$, so it is correctly classified; $\text{CE}(\mathbf{Y}^3) = .86$. The CE generated by example 1 is minimal. However, the incorrectly classified example 2 generates a more optimal value of CE than the correctly classified example 3 generates...

Example 5.17 Figure 5.8 illustrates the non-monotonic nature of CE when $C = 3$. The figure shows two views of discriminator output space (which is $\mathcal{Y} = [l = 0, h = 1]^3$ for the purpose of illustration). Again, $\log(\cdot)$ denotes $\log_{10}(\cdot)$ in (5.61). The left-hand figure shows discriminator output space from the perspective of $\mathbf{Y}_{correct}$; the right-hand figure shows discriminator output space from the perspective of $\mathbf{Y}_{incorrect}$. By (5.70), $\text{CEIF}_{mono}(3) = 0$ and $\text{CEMF}(3) = \text{CECF}_{mono}(3) = .163$. Example 1 generates a discriminator output state of $\mathbf{Y}^1 = \langle 1, .776, .776 \rangle$, so it is correctly classified; $\text{CE}(\mathbf{Y}^1) = 1.3$. Example 2 generates a discriminator output state of $\mathbf{Y}^2 = \langle .387, 0, .613 \rangle$, so it is incorrectly classified; $\text{CE}(\mathbf{Y}^2) = .83$. Thus, the correctly classified example 1 generates a less optimal value of CE than the incorrectly classified example 2 generates; like MAE and MSE, the CE objective function is non-monotonic ($C \geq 2$).

Given (5.71), (5.70) can also be expressed as

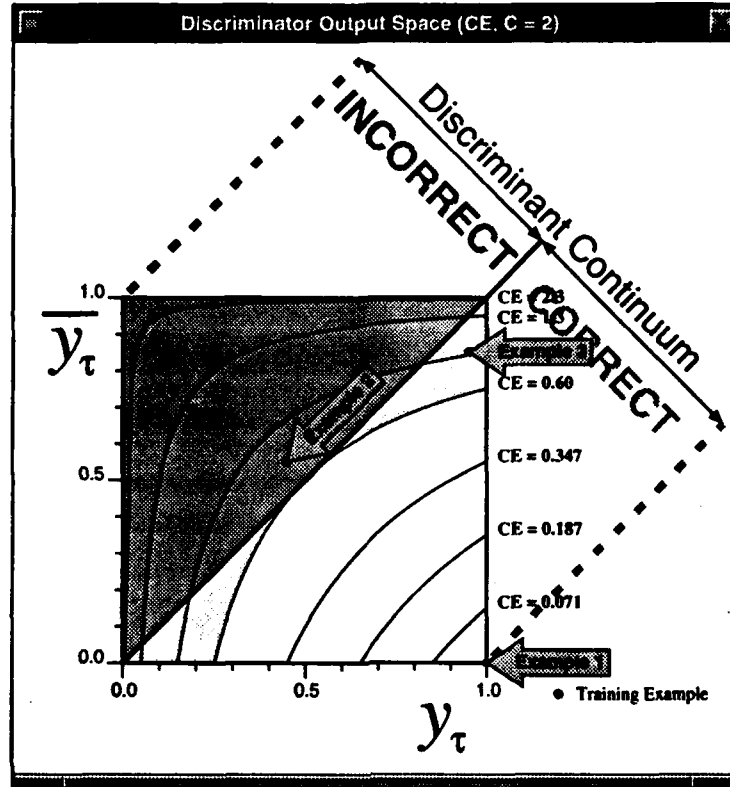


Figure 5.7: The Kullback-Leibler information distance (CE objective function) is non-monotonic. Example 3 generates higher (less optimal) CE than example 2, even though example 3 is correctly classified, whereas example 2 is not. For every point in the light gray shaded region of correct space there is at least one point in the dark gray shaded region of incorrect space with lower CE. The figure depicts discriminator output space for a hypothetical $C = 2$ -class task in which the discriminator's outputs are bounded on $\mathcal{Y} = [l = 0, h = 1]^2$. Since the classifier has two discriminant functions, discriminator output space and reduced discriminator output space are one and the same.

$$\begin{aligned}
 \text{CEMF}(C) = \text{CECF}_{\text{mono}}(C) &= \lambda \cdot \sum_{j=C}^{\infty} \frac{(-1)^j}{j!} \cdot [\ln(\lambda)]^j \\
 &< \frac{[-\ln(\lambda)]^C}{\Gamma(C+1)}; \\
 \mathcal{Y} &= [l = 0, h = 1]^C, \quad \lambda = \left(\frac{h+l}{2}\right)^2 = \frac{1}{4}
 \end{aligned} \tag{5.73}$$

The upper bound of (5.73) is tight for small C and loose for large C . Thus, the CE objective function is like its MAE and MSE counterparts: $\text{CEMF}(C)$ and $\text{CECF}_{\text{mono}}(C)$ decrease super-exponentially as C increases, although $\text{CEMF}(C)$ is $\mathcal{O} \left[[-\ln(\lambda)]^C \cdot [\Gamma(C+1)]^{-1} \right]$ whereas $\text{MAEMF}(C)$ is $\mathcal{O} \left[[\Gamma(C+1)]^{-1} \right]$ and

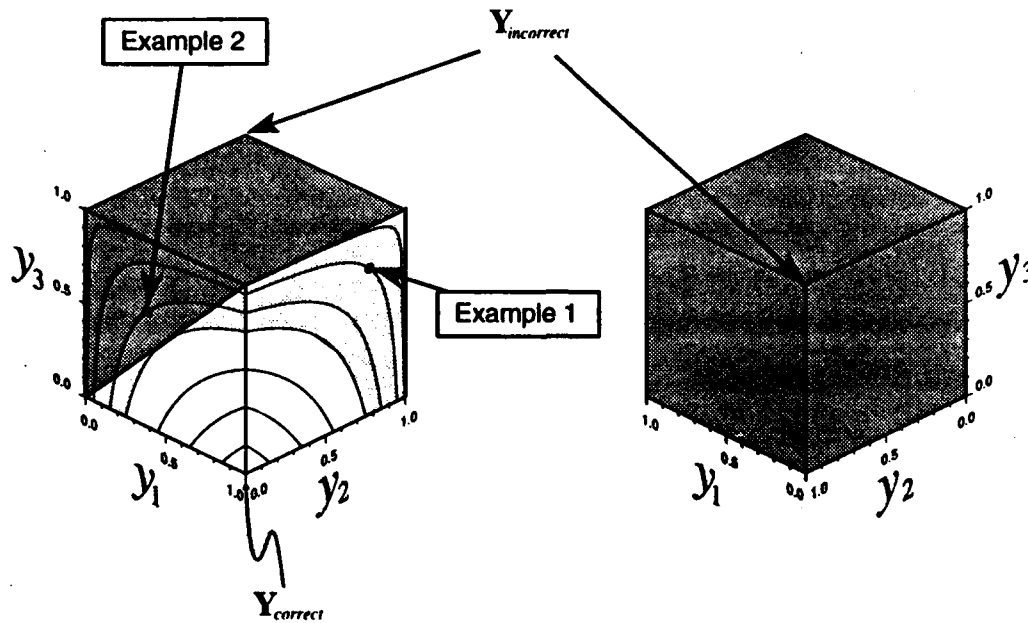


Figure 5.8: The CE objective function is increasingly non-monotonic as C increases. Output y_1 is y_r , the output corresponding to the class label for the two training examples shown. Contours of constant CE are projected onto the bounding faces of discriminator output space. White denotes the monotonic regions of discriminator output space, light gray denotes the non-monotonic region of correct space,^a and dark gray denotes the non-monotonic region of incorrect space. Example 1 is correctly classified, yet it generates a higher (less optimal) value of CE than example 2, which is incorrectly classified. Left: The discriminator output space from the perspective of $Y_{correct}$. Right: The discriminator output space from the perspective of $Y_{incorrect}$. The monotonic fraction of incorrect space is zero; the monotonic fraction of correct space is .163; the monotonic fraction of discriminator output space is therefore .163.

^aThe light gray shading underneath the cubic form of the discriminator output space is an imaginary shadow; it helps to clarify the cube's orientation.

$MSEMF(C)$ is $O\left[\left(\Gamma\left(\frac{C}{2} + 1\right)\right)^{-1}\right]$. When the number of classes is ten, the monotonic fractions are quite small:

$$\begin{aligned} CE MF(10) &= CE CF_{mono}(10) = 2.06 \times 10^{-6} \\ CE IF_{mono}(10) &= 0; \\ \mathcal{V} &= [l = 0, h = 1]^{10} \end{aligned} \tag{5.74}$$

5.3.4 The General Error Measure is Non-Monotonic

The findings of the preceding three sections are directly linked with the section 3.4 proofs that error measures engender inefficient learning. Specifically, (3.40) proves that minimizing the general error measure does not minimize the classifier's error rate. Thus, (3.40) proves that error measures fail to satisfy (5.16) and definition 5.10: they are non-monotonic.

The section 3.6 exception to the rule that probabilistic learning is inefficient does not contradict our assertion herein that error measures are non-monotonic. When the hypothesis class is a proper parametric model of the feature vector, probabilistic learning via the error measure that generates maximum-likelihood estimates of the discriminator's parameters is efficient. Under these circumstances, the functional characteristics of the discriminator's hypothesis class compensate for the non-monotonic nature of the error measure (see below); the error measure itself remains non-monotonic, independent of the choice of hypothesis class.

5.3.5 The Link Between Objective Function Monotonicity and Learning Efficiency

The monotonic objective function engenders asymptotically efficient learning, regardless of the choice of hypothesis class (section 3.3). The non-monotonic objective function engenders inefficient learning if the hypothesis class is improper (section 3.4); if the hypothesis class is proper, there may be an error measure that induces efficient learning, despite its non-monotonic nature (section 3.6). That is, under proper conditions the non-monotonic error measure induces efficient learning because the discriminator's proper nature compensates for the small value of $\mathcal{MF}$. In some cases (see below) the "proper" discriminator's correct output states are constrained to lie in the monotonic region of correct space. It remains an open question whether this is *always* the case when the hypothesis class constitutes a proper parametric model of the feature vector.

Under improper conditions the discriminator's functional properties fail to constrain the discriminator's correct output states to lie in the monotonic region of correct space, and the small value of $\mathcal{MF}$ results in inefficient learning. Indeed, as $\mathcal{MF} \rightarrow 0$, the objective function's value during learning may tell us absolutely *nothing* about the classifier's empirical training sample error rate if the hypothesis class is not the proper parametric model of the feature vector and the error measure is not the one associated with the maximum-likelihood probabilistic learning procedure described in hypothesis 3.1 (page 77).

Chapter 4 clearly illustrates the three fundamental scenarios we address in chapter 3 and this section:

A non-monotonic objective function paired with a proper parametric model can induce efficient learning — The discriminant functions of the partially parametric proper model in section 4.2 are given by (4.4). These equations guarantee that the discriminator output state lies on the discriminant continuum (definition 5.1), *regardless* of the discriminator's parameterization. This constraint compensates for the non-monotonic nature of the error measures used for probabilistic learning because the discriminator simply cannot produce output states that lie in the non-monotonic region of correct space. As a result, the hypothesis class's functional properties in (4.4) ensure that (5.16) is never violated. This ensures that probabilistic learning via the general error measure is asymptotically efficient. Moreover, if the error measure associated with the maximum-likelihood learning procedure defined in hypothesis 3.1 exists, it will induce efficient learning for small sample sizes as well. This is precisely the scenario we find in section 4.2.

A non-monotonic objective function paired with an improper parametric model always induces inefficient learning — The discriminant functions of the improper parametric model in section 4.3 are given by (4.32). These equations do not guarantee that the discriminator's correct output states will always lie in the monotonic region of correct space. As a result, the hypothesis class's functional properties fail to ensure that (5.16) is never violated. This ensures that probabilistic learning via the general error measure is inefficient. Indeed, by (5.57), $\text{MSE } \mathcal{IF}_{\text{mono}}(3) < \frac{1}{6}$, $\text{MSE } \mathcal{CF}_{\text{mono}}(3) = 0.185$, and $\text{MSE } \mathcal{MF}(3) < 0.352$, and the MSE-generated minimum-complexity polynomial classifier of section 4.3 exhibits an 18% discriminant bias for both small and large training sample sizes.

A monotonic objective function paired with any hypothesis class always induces asymptotically efficient learning — Since the CFM objective function is monotonic (see section 5.3.6 below), definition 5.10 and (5.16) are always satisfied, regardless of the hypothesis class's functional properties. As a result, differential learning is always asymptotically efficient — a fact that is demonstrated in the experiments of chapter 4 and part II.

5.3.6 CFM is Monotonic

As we stated earlier, if the objective function is to be monotonic, (5.16) must hold. The CFM objective function *always* satisfies (5.16), since it is a function of only two discriminator outputs (y_τ and $\bar{y}_\tau$) *regardless* of the number of classes C . Thus, the CFM generated by a correctly classified example is always greater than the CFM generated by an incorrectly classified example (e.g., see figure 5.2.) This statement is true for any and all choices of the CFM confidence parameter ψ . As a result,

$$\begin{aligned} \text{CFM } \mathcal{IF}_{\text{mono}}(C) &= \mathcal{IF} = \frac{C-1}{C} & \forall C \geq 2 \\ \text{CFM } \mathcal{CF}_{\text{mono}}(C) &= \mathcal{CF} = \frac{1}{C} & \forall C \geq 2 \\ \therefore \text{CFM } \mathcal{MF}(C) &= 1 & \forall C \geq 2 \end{aligned} \tag{5.75}$$

So far we have focused on whether or not the objective function's contours of constant value are parallel to the discriminant boundary. Although this condition — expressed mathematically in (5.16) — is a *necessary* one for monotonicity, it is not *sufficient* to satisfy definition 5.10. The reason for this lies in the difference between an objective function's being monotonic for a *single* example, versus its being monotonic for *all* examples in the training sample. This latter kind of monotonicity is *true* monotonicity.

CFM is Truly Monotonic

In the case of CFM, (2.102) and (2.104) must also be satisfied in order for CFM to be truly monotonic, thereby inducing asymptotically efficient learning. We clarify this notion of true monotonicity with a hypothetical

example, showing that CFM is non-monotonic when (2.104) is violated, but monotonic when (2.104) is satisfied. A similar illustration can be made regarding the constraint of (2.102).

Consider a two-class pattern recognition task for which the feature is a random scalar x . The class prior probabilities are

$$P_W(\omega_1) = P_W(\omega_2) = \frac{1}{2}, \quad (5.76)$$

and the class-conditional pdfs of x are given by

$$\begin{aligned} \rho_{x|W}(x|\omega_1) &= \delta(x + 1) \\ \rho_{x|W}(x|\omega_2) &= .1 \cdot \delta(x + .9) + .9 \cdot \delta(x - 1), \end{aligned} \quad (5.77)$$

where $\delta(\cdot)$ denotes the Dirac delta function (e.g., [80, pg. 266]). The *a posteriori* class probabilities are therefore

$$\begin{aligned} P_{W|x}(\omega_1|x) &= \delta(x + 1) \\ P_{W|x}(\omega_2|x) &= \delta(x + .9) + \delta(x - 1) \end{aligned} \quad (5.78)$$

Figure 5.9 depicts the class-conditional pdf — class prior probability products as well as the *a posteriori* class probabilities of x . The light blue arrows depict the Dirac delta functions associated with class ω_1 , and the red arrows depict those associated with class ω_2 . The colored circles in the *a posteriori* class probability plots indicate the values of x for which $P_{W|x}(\omega_1|x)$ (light blue) and $P_{W|x}(\omega_2|x)$ (red) are zero. Obviously, the two classes that x can represent are linearly separable: a linear classifier need only form a boundary on the open interval $(-1, -.9)$, which constitutes the Bayes-optimal class boundary

$$-1 < \mathcal{B}_{1,2 \text{ Bayes}} < -.9 \quad (5.79)$$

This open interval is depicted by the swath of gray shading in figure 5.9.

We employ a minimum-complexity linear classifier⁵ with the discriminator $\mathcal{G}(x|\theta) = \{g_1(x|\theta), g_2(x|\theta)\}$; the discriminant functions of $\mathcal{G}(x|\theta)$ are given by

$$\begin{aligned} g_1(x|\theta) &= -\frac{1}{2}x + \theta_0 \\ g_2(x|\theta) &= -g_1(x|\theta) = \frac{1}{2}x - \theta_0, \end{aligned} \quad (5.80)$$

so the discriminator's parameter vector reduces to the single parameter θ_0 . By convention, we sometimes use the more general vector notation θ , assuming that the reader understands the equivalence between θ and θ_0 in this particular example. We denote the discriminator parameterization that maximizes CFM by θ_0^* . Given the discriminator in (5.80) and the parameter space $\Theta = \mathcal{R}$, discriminator output space is $\mathcal{Y} = \mathcal{R}^2$.

⁵The complexity measure is the classifier's number of parameters, which is one.

Learning is simply a search for θ_0^* over Θ (i.e., the domain of θ_0), given a specific value of the CFM confidence parameter ψ . Figure 5.10 illustrates this search space: CFM is plotted as a function of the discriminator's single parameter θ_0 and the CFM confidence parameter ψ . Specifically, the plot shows the CFM generated by an asymptotically large training sample, given the discriminator $\mathcal{G}(x|\theta)$ and ψ : $\lim_{n \rightarrow \infty} \text{CFM}(\mathcal{S}^n|\theta)$. The figure's color coding emphasizes the value of ψ : magenta indicates $\psi = 1$ and blue indicates $\psi \rightarrow 0^+$; intermediate values of ψ correspond to a natural progression from magenta to blue via the intermediate color yellow. The black contours on the figure denote the value of CFM as a function of ψ for fixed increments of θ_0 .

Given $\mathcal{G}(x|\theta)$ in (5.80), the class boundary it forms is

$$B_{1,2} = 2\theta_0 \quad (5.81)$$

It should be clear from the probabilistic nature of x illustrated in figure 5.9 that the classifier with the discriminator described by (5.80) will correctly classify all examples of x if

$$g_1(x|\theta) = g_2(x|\theta) = 0 \text{ for some } x \in (-1, -.9) \quad (5.82)$$

When (5.82) is satisfied, the differentially generated class boundary is Bayes-optimal:

$$\begin{aligned} &-.5 < \theta_0^* < -.45 \\ \text{s.t. } &B_{1,2 \text{ CFM}} = B_{1,2 \text{ Bayes}} \in (-1, -.9) \end{aligned} \quad (5.83)$$

The key question is, for what values of ψ is CFM monotonic: that is, what values of ψ guarantee that maximizing $\lim_{n \rightarrow \infty} \text{CFM}(\mathcal{S}^n|\theta)$ minimizes the classifier's error rate $P_e(\mathcal{G}|\theta)$ by generating the parameterization θ_0^* of (5.83)? The answer: those values of ψ for which (2.102) and (2.104) are satisfied.⁶ Maximizing CFM subject to the constraints of (2.102) and (2.104) minimizes the classifier's empirical training sample error rate and, by the proof of section 3.3, induces asymptotically efficient learning. When (2.102) and/or (2.104) are not satisfied, maximizing CFM is not guaranteed to minimize the classifier's empirical training sample error rate: CFM is non-monotonic and may induce inefficient learning.

Example 5.18 The largest positive discriminant differentials that $\mathcal{G}(x|\theta)$ in (5.80) can generate for examples of $x = -1$ and $x = -.9$ are

⁶Section 7.4 describes a practical approach to differential learning via a scheduled reduction of the CFM confidence parameter; this satisfies the upper bound constraints on ψ imposed by (2.102) and (2.104).

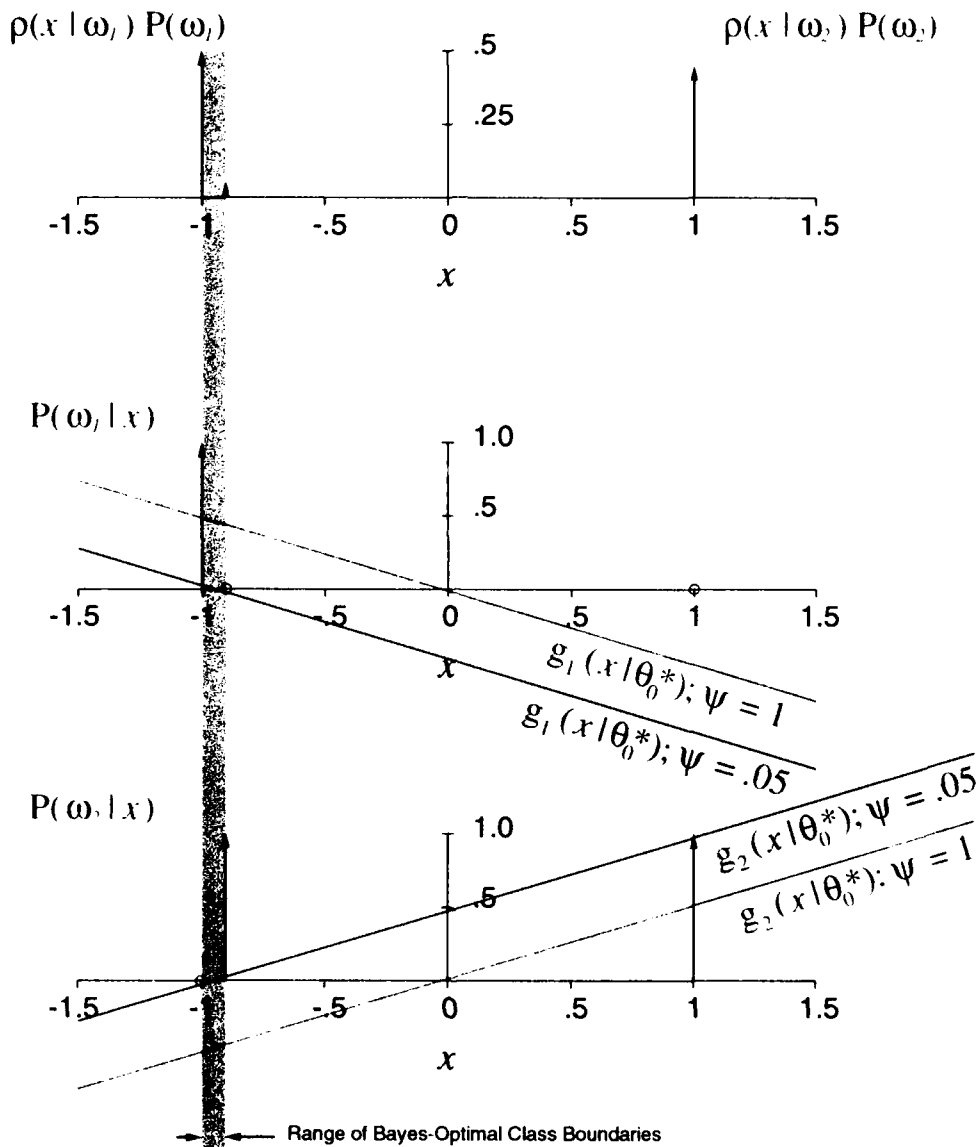


Figure 5.9: The simple two-class scalar feature discrimination task for which CFM is monotonic if and only if $\psi \lesssim .11$. From top to bottom: the class-conditional density - class prior products $\rho_{x|W}(x|\omega_1) \cdot P_W(\omega_1)$ and $\rho_{x|W}(x|\omega_2) \cdot P_W(\omega_2)$; $P_{W|x}(\omega_1|x)$, the *a posteriori* probability of class ω_1 ; $P_{W|x}(\omega_2|x)$, the *a posteriori* probability of class ω_2 . Two sets of linear discriminant functions are shown superimposed on the *a posteriori* class probabilities of x : the magenta-colored functions are generated by maximizing CFM, given the confidence parameter $\psi = 1$; the blue-colored functions are generated by maximizing CFM, given the confidence parameter $\psi = .05$. Note that classifier generated with $\psi = 1$ incorrectly classifies all examples of $x = -.9$, so CFM is non-monotonic for $\psi = 1$, given x described by (5.76) – (5.78) and the discriminator $\mathcal{G}(x|\theta)$ described by (5.80). However, when ψ is reduced to a value of .05, the classifier generated by CFM correctly classifies all examples of x . Indeed, CFM is *always* monotonic, given a sufficiently small value of ψ — a relationship formalized by the constraints of (2.102) and (2.104).

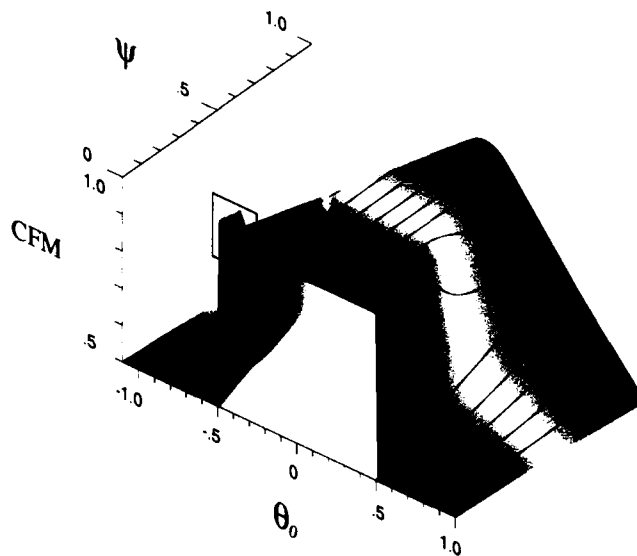


Figure 5.10: The CFM (i.e., $\lim_{n \rightarrow \infty} \text{CFM}(S^n | \theta)$) generated by an asymptotically large training sample of the two-class random feature x described by (5.76) – (5.78): CFM is plotted as a function of the single discriminator parameter θ_0 and the CFM confidence parameter ψ . Given x and the simple linear discriminator $\mathcal{G}(x | \theta)$ described by (5.80), the plot shows that CFM is monotonic if and only if ψ is small (boxed region). That is, the parameterization θ_0^* that maximizes CFM also minimizes the classifier's training sample error rate if and only if ψ is sufficiently small.

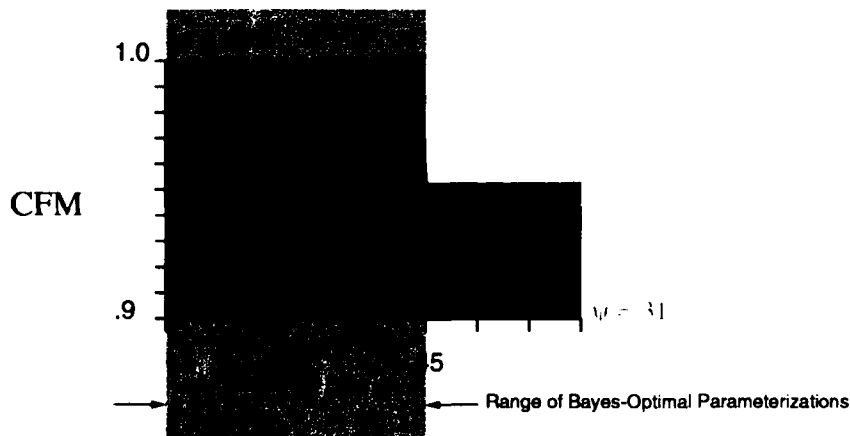


Figure 5.11: Details of the maximum CFM (i.e., $\max_{\theta} \lim_{n \rightarrow \infty} \text{CFM}(S^n | \theta)$) for small ψ generated by an asymptotically large training sample of the two-class random feature x . The figure shows the boxed region of figure 5.10 as it would appear viewed along an axis that is parallel to the ψ axis (i.e., the image plane of this figure is parallel to the $\text{CFM} - \theta_0$ plane in figure 5.10). CFM is plotted as a function of the single discriminator parameter θ_0 : different contours represent different CFM functions of θ_0 , given different values of the confidence parameter ψ . The color coding of the contours denotes the value of ψ and corresponds to the scheme used in figure 5.10. The contours are in increments of .01, from .31 (green) to .01 (blue); the lower-bound contour of the figure is $\psi = .002$. The CFM objective function is, by (2.104), guaranteed to be monotonic if $\psi \leq .05$ — an upper bound denoted by the white highlighted contour of the figure. Note that CFM is maximized, given this value of ψ , for $-.478 \lesssim \theta_0^* \lesssim .467$. Thus, the differentially generated classifier ($\psi \leq .05$) yields Bayesian discrimination, by (5.83), since CFM is maximal for θ_0 values that fall within the gray-shaded open interval $(-.5, .45)$.

$$\begin{aligned}
\delta_1(x = -1 | \theta_0 = -.475) &= \underbrace{g_1(x = -1 | \theta_0 = -.475)}_{.025} - \underbrace{g_2(x = -1 | \theta_0 = -.475)}_{-.025} = .05 \\
\delta_2(x = -.9 | \theta_0 = -.475) &= \underbrace{g_2(x = -.9 | \theta_0 = -.475)}_{.025} - \underbrace{g_1(x = -.9 | \theta_0 = -.475)}_{-.025} = .05
\end{aligned}
\tag{5.84}$$

By (2.104), ψ must therefore be no greater than .05 in order to guarantee that CFM is truly monotonic. If we maximize $\lim_{n \rightarrow \infty} \text{CFM}(S^n | \theta)$ using the synthetic CFM objective function described in appendix D with $\psi = 1$, $\theta_0^* = -.0125$. One can see that that CFM peaks at this value of θ_0 in figure 5.10: the back edge of the CFM function's magenta region corresponds to $\psi = 1$, and it peaks at $\theta_0 = -.0125$. This parameterization generates the magenta discriminant functions of figure 5.10, which form the class boundary $B_{1,2 \text{ CFM}} = -.025 \neq B_{1,2 \text{ Bayes}}$. Thus, CFM is non-monotonic for $\psi = 1$, given x and $\mathcal{G}(x | \theta)$.

If we reduce ψ to a small value, CFM becomes monotonic. Figures 5.10 and 5.11 illustrate this transformation. Again, the color-coding of figure 5.10 reflects the value of ψ : large values of ψ correspond to the magenta part of the image, intermediate values of ψ correspond to the yellow part of the image, and small values of ψ correspond to the blue part of the image. CFM takes on a maximum value of $\sim .95$ for all but small values of ψ , reflecting the linear classifier's inability to learn that all examples of $x = -.9$ are examples of class ω_2 . As ψ becomes small (i.e., $\lesssim .11$), CFM takes on a maximum value of ~ 1 for values of θ_0 corresponding to the boxed portion of the figure. Figure 5.11 depicts this region in detail. CFM is plotted as a function of the single discriminator parameter θ_0 , given small values of ψ . Different contours in the figure depict different CFM functions associated with different values of ψ from .31 (green) to .01 (blue) in increments of .01: the bounding (i.e., outer-most dark blue) contour is for $\psi = .002$. One can see from the figure that CFM is non-monotonic for $\psi > .11$ because it peaks at $\theta_0 > -.45$; therefore, $B_{1,2 \text{ CFM}} \neq B_{1,2 \text{ Bayes}}$. However, for $\psi \leq .05$ (the contour associated with $\psi = .05$ is highlighted in white), CFM is monotonic because it peaks between $-.478 \lesssim \theta_0^* \lesssim -.467$; therefore, $B_{1,2 \text{ CFM}} = B_{1,2 \text{ Bayes}}$. Given the parameterization $\theta_0^* = -.475$ for $\psi = .05$, the blue discriminant functions of figure 5.9 result; they form the class boundary $B_{1,2 \text{ CFM}} = -.95 = B_{1,2 \text{ Bayes}}$. Thus, CFM is truly monotonic for $\psi \leq .05$, given x and $\mathcal{G}(x | \theta)$.

MAE is not truly monotonic for any C — It is interesting to note that minimizing the MAE of $\mathcal{G}(x | \theta)$ (i.e., minimizing $\lim_{n \rightarrow \infty} \text{MAE}(S^n | \theta)$ for $D = 1$ and $\neg D = 0$) generates the "optimal" parameterization $-.5 < \theta_0^* < .5$ (we omit the details of the analysis in the interest of brevity). All parameterizations in this

interval exhibit the same (minimum) value of $\text{MAE} = .5$, but the classifier's error rate is minimized only if $-.5 < \theta_0^* < -.45$. Thus, despite its satisfying (5.16) for $C = 2$, the MAE objective function is not truly monotonic for the arbitrary combination of feature vector and hypothesis class.

5.4 Training Example Types

There is a taxonomy of training examples that follows naturally from the preceding illustrative example and the more general differential learning scenario. Figure 5.12 illustrates three categories of training examples: un-learned, learned, and transition examples (see definitions D.1 – D.3). Each type is characterized by the value of its associated discriminant differential⁷ δ . Un-learned training examples are misclassified; as a result, they exhibit negative discriminant differentials. Learned examples are ones that generate the maximum value of CFM, so they must have positive discriminant differentials. The minimum value of δ for which an example is, "learned," which is given by μ_{xp} in figure D.1, depends on the value of the confidence parameter ψ . As $\psi \rightarrow 0^+$ and the synthetic CFM sigmoid grows steep, this minimum value of δ decreases to zero. Transition examples exhibit discriminant differentials that correspond to the transition region of the synthetic CFM sigmoid (i.e., $x_{mn} < \delta < \mu_{xp}$, where x_{mn} and μ_{xp} are shown in figure D.1). Simply put, un-learned examples exhibit negative discriminant differentials, learned examples exhibit relatively large positive discriminant differentials, and transition examples exhibit small (positive or negative) discriminant differentials. By definitions D.1 – D.3, some un-learned examples are also transition examples.

The confidence parameter: Recall that $P_{W|X}(\omega_* | X)$ denotes the largest a posteriori class probability for X , ω_* denotes the Bayes-optimal (i.e., most likely) class, and $\delta_*(X | \theta^*)$ denotes the associated discriminant differential $g_*(X | \theta^*) - \max_{k \neq *} g_k(X | \theta^*)$. The notation θ^* indicates that the classifier's parameterization is the one generated by maximizing CFM. Throughout the present discussion, we assume that the discriminator possesses sufficient functional complexity to learn the Bayes-optimal classifier of X . As a result, we assume that $\delta_*(X | \theta^*)$ is positive, as long as ψ is sufficiently small.

The relationship between small values of $P_{W|X}(\omega_* | X)$ and/or $\delta_*(X | \theta^*)$ and small values of ψ , codified by (2.102) and (2.104), accounts for our use of the term "confidence parameter" for ψ . If $P_{W|X}(\omega_* | X)$ and/or $\delta_*(X | \theta^*)$ are small for a given $X \in \mathcal{X}$, ψ must be small — we should literally have low confidence that the classifier will learn ω_* , the Bayes-optimal class label of X . Our necessary lack of confidence reflects the small a posteriori probability of class ω_* and/or the functional limitations of the hypothesis class at X .

This notion of confidence is related to the difference between learning easy examples and learning hard

⁷ Again, absent a subscript, the notation δ implies the discriminant differential δ_* of (2.22) and (5.14).

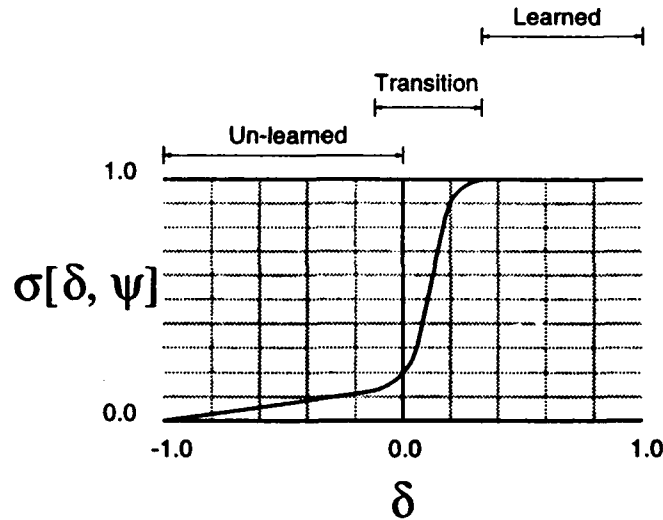


Figure 5.12: Three types of training examples: un-learned examples exhibit negative discriminant differentials; transition examples exhibit discriminant differentials that correspond to the transition region of the synthetic CFM sigmoid (therefore, some un-learned examples are also transition examples); learned examples have positive differentials that correspond to the maximum CFM value of unity.

examples.

Definition 5.19 *The easy training example: Probabilistically, the easy example $\mathbf{X}^j$ of class $\mathcal{W}^j = \omega_i$ is found far from the Bayes-optimal class boundaries on $\mathcal{X}$, near a mode of its class-conditional pdf (i.e., $\rho_{\mathbf{x}|\mathcal{W}}(\mathbf{X}^j|\omega_i) \approx \max_{\mathbf{X}} \rho_{\mathbf{x}|\mathcal{W}}(\mathbf{X}|\omega_i)$). Assuming nominally equal prior probabilities for all classes, the easy example's a posteriori class probability $P_{\mathcal{W}|\mathbf{x}}(\omega_*|\mathbf{X}^j)$ and the associated discriminant differential $\delta_*(\mathbf{X}^j|\theta^*)$ are therefore large (i.e., $\mathcal{O}[1]$), allowing learning to be accomplished with high confidence (i.e., $\psi \approx 1$) by (2.102) and (2.104).*

Example 5.19 All examples of $x = 1$ and $x = -1$ — where x is described by (5.76) — (5.78) — are easy. Note in figure 5.9 that these examples have a posteriori class probabilities of one, and $\mathcal{G}(x|\theta)$, described in (5.80), generates relatively large positive discriminant differentials for them when CFM is maximized for $\psi = 1$:

$$\begin{aligned} \delta_1(x = -1|\theta_0^* = -.0125) &= .975 \\ \delta_2(x = 1|\theta_0^* = -.0125) &= 1.025 \end{aligned} ; \quad \psi = 1 \quad (5.85)$$

Recall that the magenta discriminant functions of figure 5.9 are those that maximize CFM for $\psi = 1$.

Definition 5.20 *The hard training example: Probabilistically, the hard example $\mathbf{X}^j$ of class $\mathcal{W}^j = \omega_i$ is found in the vicinity of the class boundaries on $\mathcal{X}$, in a "tail" of its class-conditional pdf (i.e.,*

$\rho_{x|w}(X^j|\omega_i) \ll \max_x \rho_{x|w}(X|\omega_i)$. In these tails, $P_{w|x}(\omega_*|X^j)$ and/or $\delta_*(X^j|\theta^*)$ are relatively small (i.e., $\ll 1$), so the hard example must, by (2.102) and (2.104), be learned with low confidence (i.e., $\psi \rightarrow 0^+$) if it can be learned at all.⁸

Example 5.20 All examples of $x = -.9$ — where x is described by (5.76) — (5.78) — are hard. Note in figure 5.9 that these examples have a large *a posteriori* class probability of $P_{w|x}(\omega_2|X = -.9) = 1$, but their class-conditional pdf is small ($\rho_{x|w}(X = -.9|\omega_2) = .1 \ll \rho_{x|w}(X = 1|\omega_2) = .9$). The discriminator $\mathcal{G}(x|\theta^*)$, described in (5.80), at best generates relatively small positive discriminant differentials (i.e., $\leq .05$) for these examples, and does so only when CFM is maximized for $\psi \leq .05$ (recall (5.84) and note that the blue discriminant functions of figure 5.9 are those that maximize CFM for $\psi = .05$).

5.5 The Convergence Properties of Differential Learning via CFM

Differential learning via synthetic CFM, by design, ignores learned examples: these examples have no effect on learning because the synthetic CFM objective function is maximum and all its derivatives are zero for learned examples.⁹ Only unlearned and transition examples generate non-zero first and higher order derivatives of the synthetic objective function (see (D.7) — (D.9) in section D.1). From the preceding section and chapter 2, we know that CFM must approach a modified Heaviside step function (i.e., ψ must approach a value of zero) if hard examples are to be learned. Intuitively, then, this limiting form of CFM simply counts correct classifications. Since the Heaviside step is non-differentiable, there is no way to use it with differentiable supervised classifiers. Instead, we employ CFM, which remains differentiable *and* approximates the non-differentiable counting objective function with high precision as its confidence parameter ψ goes to zero.

The functional properties of CFM raise the issue of the *learning rate* — that is, the rate at which the search for the classifier's CFM-maximizing parameterization takes place. Given our definition of the differentiable supervised classifier and the means by which it learns, learning speed depends on the first order (and possibly higher order — depending on the specific search algorithm employed) derivative of the CFM objective function.

Formally, we are searching for a parameterization θ^* by which CFM is maximized, given the training sample S^n :

$$\text{CFM}(S^n|\theta^*) = \max_{\theta} \text{CFM}(S^n|\theta) \quad (5.86)$$

⁸For stochastic feature vectors with overlapping class-conditional pdfs, the Bayes error rate is non-zero: some example/class label pairs are inevitably un-learnable.

⁹This characteristic results in a substantial computational savings when differential learning is actually implemented on the computer (see section 7.5).

Knowing the value of CFM, given the parameterization $\theta[k]$ (where k is simply an iteration index), allows us to approximate its value for the parameterization $\theta[k + 1]$ via the Taylor series

$$\begin{aligned} \text{CFM}(S^n | \theta[k + 1]) = & \\ & \text{CFM}(S^n | \theta[k]) + (\theta[k + 1] - \theta[k])^T \nabla_{\theta} (\text{CFM}(S^n | \theta[k])) \\ & + \frac{1}{2} (\theta[k + 1] - \theta[k])^T \mathbf{H}_{\theta} (\text{CFM}(S^n | \theta[k])) (\theta[k + 1] - \theta[k]) \\ & + \dots \end{aligned} \quad (5.87)$$

where z^T denotes the transpose of vector z , $\nabla_{\theta} (\text{CFM}(S^n | \theta[k]))$ denotes the gradient of CFM with respect to the parameter vector θ , evaluated at $\theta[k]$, and $\mathbf{H}_{\theta} (\text{CFM}(S^n | \theta[k]))$ denotes the hessian of CFM with respect to the parameter vector θ , evaluated at $\theta[k]$.

As a first step towards iteratively finding the parameterization θ^* that maximizes CFM, we take the derivative of (5.87), set it equal to zero, and solve for $\theta[k + 1]$.

$$\begin{aligned} \nabla_{\theta} (\text{CFM}(S^n | \theta[k + 1])) = & \\ & \nabla_{\theta} (\text{CFM}(S^n | \theta[k])) + (\theta[k + 1] - \theta[k])^T \mathbf{H}_{\theta} (\text{CFM}(S^n | \theta[k])) \\ & + \text{higher-order terms} \\ = & 0 \end{aligned} \quad (5.88)$$

Dropping the higher order terms and rearranging the low-order terms yields the familiar quadratic approximation upon which all first and second order iterative search (i.e., optimization) algorithms are based (see, for example, [106, ch. 10]):

$$\theta[k + 1] \cong \theta[k] + \underbrace{\nabla_{\theta} (\text{CFM}(S^n | \theta[k])) \left[-\mathbf{H}_{\theta} (\text{CFM}(S^n | \theta[k])) \right]^{-1}}_{\Delta \theta[k]} \quad (5.89)$$

Second-order search algorithms compute (or approximate) the inverse hessian; first-order algorithms assume that the hessian is diagonal

$$\left[-\mathbf{H}_{\theta} (\text{CFM}(S^n | \theta[k])) \right]^{-1} = \epsilon \cdot \mathbf{I}, \quad (5.90)$$

such that (5.88) reduces to¹⁰

$$\theta[k + 1] \cong \theta[k] + \underbrace{\epsilon \nabla_{\theta} (\text{CFM}(S^n | \theta[k]))}_{\Delta \theta[k]} \quad (5.91)$$

¹⁰We use a modified form of the backpropagation algorithm to implement differential learning via synthetic CFM. Please see section D.5 for important additional details regarding the implementation.

The step size scaling factor ϵ has a positive sign, since CFM is being maximized.¹¹

The parameterization θ^* is taken to be $\theta[k]$ after k iterations, where k is large. We are concerned with the learning rate: how large does k have to be to ensure that $\theta[k]$ is a good approximation to θ^* , assuming that CFM ($S^n | \theta$) is a unimodal function on Θ . Given (5.89) and (5.91), the answer to this question depends on the search step size $\Delta\theta[k]$, which in turn depends on the gradient — and in the case of (5.89), the hessian — of CFM ($S^n | \theta[k]$). The gradient and hessian, in turn, depend on the training sample S^n , the functional properties of the discriminator $\mathcal{G}(\mathbf{X} | \theta[k]) = \{g_1(\mathbf{X} | \theta[k]), \dots, g_C(\mathbf{X} | \theta[k])\}$, and the functional properties of the CFM objective function itself. Dropping the iteration index k for notational simplicity, we recall from (2.81) that the sample CFM is

$$\text{CFM}(S^n | \theta) = \frac{1}{n} \sum_{j=1}^n (\sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] : \mathcal{W}^j = \omega_\tau) \quad (5.92)$$

Thus, the i th element of ∇_θ (CFM ($S^n | \theta$)) is given by

$$\frac{\partial}{\partial \theta_i} \text{CFM}(S^n | \theta) = \frac{1}{n} \sum_{j=1}^n \left(\frac{\partial}{\partial \theta_i} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] : \mathcal{W}^j = \omega_\tau \right) \quad (5.93)$$

Likewise, the i, l th element of $\mathbf{H}_\theta$ (CFM ($S^n | \theta$)) is given by

$$\frac{\partial^2}{\partial \theta_i \partial \theta_l} \text{CFM}(S^n | \theta) = \frac{1}{n} \sum_{j=1}^n \left(\frac{\partial^2}{\partial \theta_i \partial \theta_l} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] : \mathcal{W}^j = \omega_\tau \right) \quad (5.94)$$

The gradient ∇_θ (CFM ($S^n | \theta$)) and hessian $\mathbf{H}_\theta$ (CFM ($S^n | \theta$)) ultimately depend on the partial derivatives $\frac{\partial}{\partial \theta_i} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi]$ and $\frac{\partial^2}{\partial \theta_i \partial \theta_l} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi]$, which are given by

$$\frac{\partial}{\partial \theta_i} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] = \frac{\partial}{\partial \delta} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] \cdot \frac{\partial}{\partial \theta_i} \delta_\tau(\mathbf{X}^j | \theta) \quad (5.95)$$

and

$$\begin{aligned} \frac{\partial^2}{\partial \theta_i \partial \theta_l} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] &= \frac{\partial^2}{\partial \delta^2} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] \cdot \frac{\partial}{\partial \theta_i} \delta_\tau(\mathbf{X}^j | \theta) \cdot \frac{\partial}{\partial \theta_l} \delta_\tau(\mathbf{X}^j | \theta) \\ &\quad + \frac{\partial}{\partial \delta} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] \cdot \frac{\partial^2}{\partial \theta_i \partial \theta_l} \delta_\tau(\mathbf{X}^j | \theta) \end{aligned} \quad (5.96)$$

¹¹ When the objective function is an error measure to be minimized, ϵ has a negative sign, and (5.90) is a familiar part of the gradient descent equation used in the backpropagation algorithm [119, 120] (see section D.5).

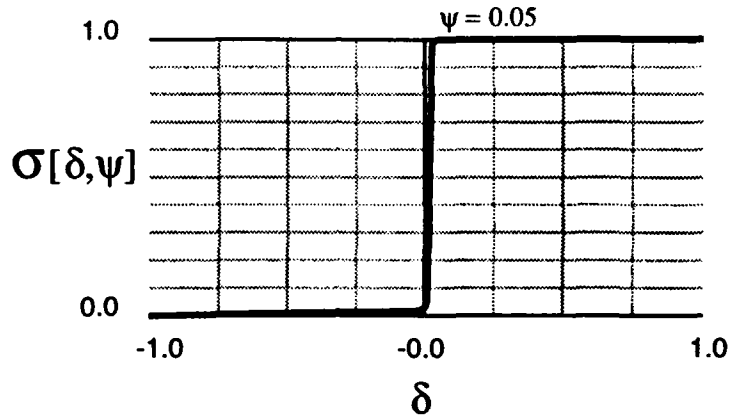


Figure 5.13: The synthetic CFM objective function, given a confidence parameter of $\psi = .05$.

Clearly, then, the step size $\Delta\theta[k]$ in (5.89) and (5.91) is proportional to the first derivative of the CFM objective function

$$\Delta\theta[k] \propto \frac{1}{n} \sum_{j=1}^n \left(\frac{\partial}{\partial \delta} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] : \mathbf{w}^j = \mathbf{w}_\tau \right) \quad (5.97)$$

Consequently, the learning rate is proportional to the first derivative of the CFM objective function.

5.5.1 Differential Learning via the Synthetic Form of CFM is Reasonably Fast

As we mentioned earlier, all derivatives of the synthetic CFM objective function are zero for learned examples (see (D.7) – (D.9) in section D.1). Thus, learning focuses on the un-learned and transition examples. Since hard examples require learning with low confidence, we are especially concerned with the rate at which these hard examples are learned, versus the rate at which transition examples are learned. Figure 5.13 illustrates the synthetic CFM objective function, given the confidence parameter $\psi = .05$ (recall that this is the level of confidence required to learn the hard examples of the two-class random feature described in section 5.3.6). Clearly, the first derivative of this function is quite large in the transition region (i.e., $\frac{\partial}{\partial \delta} \sigma[\delta \approx 0^+, \psi = .05] \cong 61$). However, it is considerably smaller for un-learned examples (i.e., $\frac{\partial}{\partial \delta} \sigma[\delta < 0, \psi = .05] \cong .02$). As a result, transition examples dominate the learning process when ψ is small, so un-learned examples are learned slowly.

This phenomenon is a natural and unavoidable consequence of the *necessary* functional properties of the CFM objective function. Thus, we are motivated to maximize the learning rate for un-learned examples as ψ becomes small, subject to the condition that CFM must simultaneously converge to a modified Heaviside

step function of δ . Section D.3 discusses this objective at length, rigorously defining *reasonably fast* and *unreasonably slow* learning in the process. Therein we denote the ratio of $\frac{d}{d\delta}\sigma[\delta, \psi]$ for transition examples to $\frac{d}{d\delta}\sigma[\delta, \psi]$ for unlearned examples by $\phi(\psi)$. If $\phi(\psi)$ increases exponentially with decreasing ψ , learning becomes dominated by the transition examples for small ψ : the classifier's parameters are updated to transform the transition examples into learned examples, while the un-learned examples are ignored (because the derivatives they elicit are so small in comparison to those of the transition examples). Under these circumstances, it takes an unreasonably long time (e.g., [58, pp. 155-158]) to learn the yet un-learned training examples, and we characterize the (differential) learning strategy as unreasonably slow. If, on the other hand, $\phi(\psi)$ increases polynomially with decreasing ψ , the learning strategy is reasonably fast.

Section D.3.2 proves that differential learning with the synthetic CFM objective function described in section D.1 is reasonably fast: $\phi(\psi)$ is $\mathcal{O}[\psi^{-2}]$. This characteristic allows hard examples to be learned in reasonable time — an assertion that is implicitly illustrated throughout the experiments of part II. Synthetic CFM has the additional property of being synthesized from three linear functions of δ connected by two circular arcs (see section D.1). Consequently, all derivatives of order ≥ 2 are zero for most values¹² of δ . As ψ becomes small, the synthetic function becomes approximately piece-wise linear in δ (see figure 5.13, for example). Thus, if $\delta_\tau(\mathbf{X}|\theta)$ is linear in θ — as it is, given a linear hypothesis class — learning is very fast, even for hard examples. This is because (5.91) is a good approximation to the ideal equation for $\theta[k+1]$ implied by (5.87). Specifically, let the matrix $\underline{\mathbf{A}}$ satisfy

$$[\nabla_\theta (\text{CFM}(S^n|\theta[k]))]^T \underline{\mathbf{A}} = \underline{\mathbf{I}}, \quad (5.98)$$

where $\underline{\mathbf{I}}$ denotes the identity vector. Owing to the approximately piece-wise linear nature of $\sigma[\delta|\psi \ll 1]$ and the choice of a linear hypothesis class (such that $\delta_\tau(\mathbf{X}|\theta)$ is linear in θ), (5.87) reduces to

$$\text{CFM}(S^n|\theta[k+1]) \cong \text{CFM}(S^n|\theta[k]) + [\nabla_\theta (\text{CFM}(S^n|\theta[k]))]^T (\theta[k+1] - \theta[k]) \quad (5.99)$$

Thus,

¹²That is, all higher-order derivatives are zero for values of δ corresponding to the linear segments of the synthetic function; higher-order derivatives corresponding to the arc segments of the synthetic function are non-zero. As ψ becomes small, the arc segments of the synthetic function also become small; synthetic CFM becomes approximately piece-wise linear, and all its higher-order derivatives are zero.

$$\begin{aligned} \underline{\underline{\mathbf{A}}} [\text{CFM}(S^n | \theta[k+1]) - \text{CFM}(S^n | \theta[k])] &\cong \underline{\underline{\mathbf{A}}} [\nabla_{\theta} (\text{CFM}(S^n | \theta[k]))]^T (\theta[k+1] - \theta[k]) \\ \text{s.t. } \theta[k+1] &\cong \theta[k] + \underbrace{\underline{\underline{\mathbf{A}}} [\text{CFM}(S^n | \theta[k+1]) - \text{CFM}(S^n | \theta[k])]}_{\Delta \theta[k]} \end{aligned} \quad (5.100)$$

Under these conditions, for which both synthetic CFM and the discriminator are linear functions of θ , differentially generated linear classifiers exhibit very fast learning. Sections 7.7 and 8.4.2 illustrate this phenomenon.

5.5.2 Differential Learning via the Original Forms of CFM is Unreasonably Slow and/or Inefficient

We were motivated to develop the synthetic form of CFM because the original functional forms described in [55] induce either unreasonably slow or inefficient learning. Figure 5.14 illustrates these functional forms. The original functional form of CFM was the logistic sigmoidal form on the left, given by

$$\sigma[\delta] = \alpha [1 + \exp(-\beta \cdot \delta + \zeta)]^{-1}, \quad (5.101)$$

where α is a superfluous linear scaling factor, ζ is a parameter that shifts the sigmoid along the δ axis, and $\frac{1}{\beta}$ is roughly equivalent to the synthetic form's confidence parameter ψ . Section D.3.1 proves that $\phi(\beta)$, the ratio of the function's derivative for transition examples to its derivative for un-learned examples, is

$$\phi(\beta) = \mathcal{O}[\exp(|\delta|\beta)], \quad \beta \gg 1, \quad \delta < 0 \quad (5.102)$$

Thus, $\phi(\beta)$ increases exponentially as β is increased (i.e., as the function's equivalent of the confidence parameter goes to zero). Under these circumstances, transition examples dominate the learning process, and un-learned examples are effectively ignored. The irony here is that β must be increased in order for the hard examples to be learned, but increasing β also ensures that it will take an unreasonably long time to learn the hard examples. In short, differential learning via the original logistic sigmoidal form of CFM is unreasonably slow, failing to learn hard examples in *reasonable time* (e.g., [58, pp. 155-158]).

The "maximally flat" form on the right of figure 5.14 is given by

$$\sigma[\delta] = -\alpha \log [1 + (\zeta - \delta)^{2\beta}] \quad (5.103)$$

where α , ζ , and β have the same interpretations as they do for the logistic sigmoidal form. It was developed in order to improve the learning rate for hard examples. Unfortunately, this form of CFM does not converge to a modified Heaviside step function as $\beta \rightarrow \infty$ (i.e., as its equivalent of the confidence

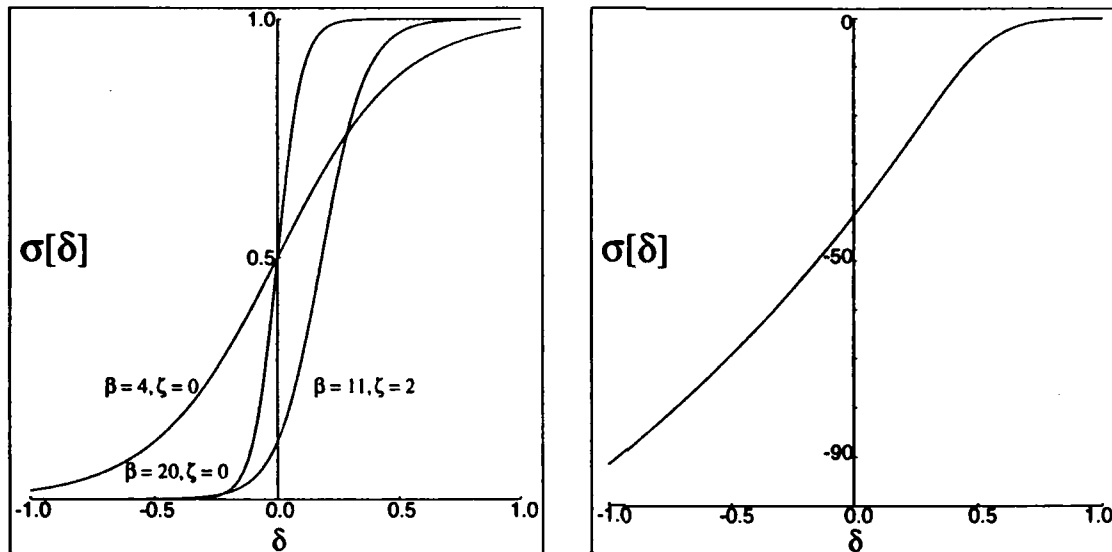


Figure 5.14: Old forms of the CFM objective function, described in [55]. **Left:** The logistic sigmoidal form satisfies the conditions for monotonicity, but leads to unreasonably slow learning. **Right:** The “maximally flat” form induces reasonably fast learning, but fails to satisfy the conditions for monotonicity.

parameter goes to zero) — a necessary characteristic if the objective function is to be monotonic and if it is to induce asymptotically efficient learning. Thus, differential learning via the maximally flat form of CFM is reasonably fast, but provably inefficient.

5.6 Summary

The link between an objective function’s monotonicity and the efficiency of the learning strategy it implements is intuitively appealing. The CFM objective function induces asymptotically efficient learning — regardless of the choice of hypothesis class — precisely because it is monotonic. Error measure objective functions induce inefficient learning¹³ because they are non-monotonic. In fact, the monotonic fraction of discriminator output space is a good indicator of how efficient the resulting learning strategy will be, given a particular objective function, the number of classes C , and a hypothesis class that is assumed to be an improper parametric model of the feature vector. The experiments of part II bear this out. Although we do not quote the monotonic fractions explicitly, it is straightforward to show that the less monotonic objective functions induce less efficient learning when the hypothesis class is improper. When the hypothesis class is a proper parametric model of the feature vector, the non-monotonic nature of some error measures (i.e., those associated with maximum-likelihood learning) is offset by the parametric model’s functional properties. These properties

¹³...except, of course, for the case in which the hypothesis class is the proper parametric model of the feature vector and the error measure is associated with maximum-likelihood learning.

constrain the discriminator's correct output states to lie in the monotonic region of "correct space".

In order to be truly monotonic, the objective function must under all circumstance be a strictly increasing or strictly decreasing function of the classifier's empirical training sample error rate. When the learning task includes hard examples, the CFM objective function must approach a limiting step functional form in order to be monotonic (and induce efficient learning). This requirement, in turn, raises the issue of the learning rate. The synthetic form of CFM detailed in appendix D allows hard examples to be learned in reasonable time. Consequently, the synthetic form is superior to the original functional forms described in [55], which induce unreasonably slow and/or inefficient learning. The difference between reasonably fast learning and unreasonably slow learning is not to be underestimated in the case of hard examples. Many of the results in part II would be worse by a statistically significant margin if hard examples were not learnable in reasonable time via synthetic CFM.

Chapter 6

An Information-Theoretic View of Stochastic Concept Learning¹

Outline

We make a distinction between the probabilistic information content and the discriminant information content of a randomly-drawn training sample, the former being associated with probabilistic learning, and the latter being associated with differential learning. We show that a simple unfair (or "rigged") game of dice forms the basis of all learning/statistical pattern recognition tasks. We analyze this game in order to prove that the discriminant information contained in a training sample is always at least as great as the probabilistic information contained therein. The information-theoretic argument relies on Rissanen's notion of stochastic complexity (e.g., [115]) and can be viewed as an extension of the chapter 3 proofs that differential learning is 1) asymptotically efficient, and 2) requires the least functional complexity necessary to generate a Bayes-optimal classifier. We derive tight, distribution-dependent lower bounds on the functional complexity and training sample size necessary for "winning" the dice game via the differential and probabilistic learning strategies. The differential learning strategy's functional complexity and sample size requirements are usually less (and never more) than the probabilistic learning strategy's. We show how simple extensions of the single die paradigm can lead to analogous lower bounds on the hypothesis class's functional complexity and the training sample size necessary for good generalization in learning/pattern recognition tasks. We conclude by discussing the limitations of this generalization to the uncountable feature vector space.

6.1 Introduction

The essence of this chapter lies in the analysis of a rigged (or unfair) game of dice that, in turn, lies at the heart of all stochastic concept learning/statistical pattern recognition tasks. Specifically, we have a C sided

¹This chapter is a revised version of work first published in [51]

die. Each face of the die has some probability of turning up when the die is cast (i.e., tossed). One face is always more likely to turn up than any of the others; thus, all the face probabilities may be different, but at most $C - 1$ of them — the lesser probabilities — can be identical. The objective of the game is to identify the most likely face with specified high confidence by observing a sequence of independent casts of the die.

The relationship between this rigged die paradigm and learning stochastic concepts for statistical pattern recognition becomes clear if one realizes that a single unfair die is analogous to a specific point on the domain of the random feature vector being classified. Just as there are specific *a posteriori* class probabilities associated with a *point* in feature vector space, a *die* has specific probabilities associated with each of its faces. The number of faces on the die (C) equals the number of classes associated with the analogous point in feature vector space. Identifying the most likely die face is equivalent to identifying the largest *a posteriori* class probability for the analogous point in feature vector space — the requirement for Bayesian discrimination, as described in chapter 2.

We begin by defining two measures of functional complexity, based on Rissanen's definitions of *stochastic complexity* and minimum description length [112, 113, 114, 115]. *Probabilistic complexity* (definition 6.1) is the stochastic complexity measure associated with probabilistic learning, whereas *differential complexity* (definition 6.2) is the stochastic complexity measure associated with differential learning. The relationship between probabilistic and differential complexity parallels the relationship between the strictly probabilistic and differential forms of the Bayesian discriminant function described in section 2.2.1.

We analyze the rigged game of dice, proving that it requires only one bit of differential complexity to learn the identity of the most likely die face differentially from a sequence of independent die casts; in contrast, it typically requires more (and *never* requires less) differential complexity to learn the identity of the most likely die face probabilistically via the same sequence of independent die casts. Moreover, the identity of the most likely die face becomes empirically evident with far fewer casts of the die than are required to estimate the die's face probabilities with specified precision. In more formal terms associated with learning for statistical pattern recognition, the *discriminant information content* of a sequence of independent die casts is usually higher (and never less) than its *probabilistic information content*. These information-theoretic proofs are analogous to the estimation-theoretic proofs of chapter 3 that differential learning is asymptotically efficient (Theorem 3.1), requiring the hypothesis class with the least functional complexity necessary for Bayesian discrimination (Corollary 3.1).

Following these arguments, we formulate tight, distribution-dependent lower bounds on the functional complexity and the number of casts necessary to identify the most likely die face empirically with high confidence. We show how simple extensions of the single die paradigm can lead to analogous lower bounds on the hypothesis class's functional complexity and the training sample size requirements for good generalization in learning/pattern recognition tasks. We conclude by describing the limitations of the generalization to the uncountable feature vector space: since the rigged die paradigm is fundamentally discrete, it over-estimates

— perhaps by orders of magnitude — the training sample size requirements it predicts are necessary for generalizing well with uncountable feature vectors.

6.2 Probabilistic versus Differential Complexity

Axiom 6.1 We view the number of bits M_q in the finite-precision approximation $q_M[x]$ to the real number $x \in (-1, 1]$ as a measure of the approximation's functional complexity. That is, the functional complexity of the approximation is the number of bits with which it represents the real number.

We compute the M_q -bit approximation $q_M[z]$ to the real number $z \in (-1, 1]$ using the following fixed-point representation:

$$\begin{aligned} \text{MSB (most significant bit)} &= \text{sign}[z] \\ \text{MSB} - 1 &= 2^{-1} \\ &\vdots \\ \text{LSB (least significant bit)} &= 2^{-(M_q-1)} \end{aligned} \tag{6.1}$$

The specific value of $q_M[z]$ is the mid-point of the $2^{-(M_q-1)}$ -wide half-open interval on which z is located:²

$$q_M[z] \triangleq \begin{cases} \text{sign}[z] \cdot (\lfloor |z| \cdot 2^{(M_q-1)} \rfloor \cdot 2^{-(M_q-1)} + 2^{-M_q}), & |z| < 1 \\ \text{sign}[z] \cdot (1 - 2^{-M_q}), & |z| \geq 1 \end{cases} \tag{6.2}$$

The lower and upper bounds on the quantization interval are $L_{M_q}[z]$ and $U_{M_q}[z]$, such that

$$L_{M_q}[z] < z \leq U_{M_q}[z] \tag{6.3}$$

$$L_{M_q}[z] = q_M[z] - 2^{-M_q} \tag{6.4}$$

$$U_{M_q}[z] = q_M[z] + 2^{-M_q} \tag{6.5}$$

The fixed-point representation described by (6.1) – (6.5) differs from standard fixed-point representations in its choice of quantization interval. The choice of (6.2) – (6.5) represents zero as a *non-positive*, finite precision number (i.e., by (6.2), $q_M[0] = -2^{-M_q}$).

Note that if

²The notation $\lfloor z \rfloor$ denotes the largest decimal integer not greater than z , and the notation $\lceil z \rceil$ denotes the smallest decimal integer not less than z (e.g., [75, pg. 37]).

$$z_1 = z_0 + 2^{-(M_q-1)} \quad (6.6)$$

(i.e., if z_0 and z_1 are in adjacent quantization intervals for M_q -bit quantization) then

$$L_{M_q}[z_1] = U_{M_q}[z_0] \quad (6.7)$$

Given this information-theoretic measure of functional complexity, we define two functional complexity measures: one is for the *a posteriori* class probabilities $\{P_{W|X}(\omega_1|X), \dots, P_{W|X}(\omega_C|X)\}$ of the C -class random feature vector X ; the other is for the *a posteriori* class differentials $\{\Delta_{W|X}(\omega_1|X), \dots, \Delta_{W|X}(\omega_C|X)\}$ of X . Both definitions are based on Rissanen's notion of *stochastic complexity* and minimum description length [112, 113, 114, 115]

Definition 6.1 Probabilistic Complexity M_{qp} : The probabilistic complexity of an approximation to the *a posteriori* class probability $P_{W|X}(\omega_i|X)$, lying on the half-open interval $(l, u]$, is the minimum number of bits M_{qp} necessary to ensure that $q_{M_{qp}}[P_{W|X}(\omega_i|X)]$ lies between the lower and upper bounds l and u :

$$\begin{aligned} l \leq L_{M_{qp}}[P_{W|X}(\omega_i|X)] < P_{W|X}(\omega_i|X) \leq U_{M_{qp}}[P_{W|X}(\omega_i|X)] \leq u \\ l < u, \quad 0 \leq l \leq 1, \quad 0 \leq u \leq 1 \end{aligned} \quad (6.8)$$

Remark: Thus, our definition of *probabilistic complexity* is identical to Rissanen's definition of *stochastic complexity*, applied to approximating the feature vector's *a posteriori* class probabilities.

Definition 6.2 Differential Complexity $M_{q\Delta}$: The differential complexity of an approximation to the *a posteriori* class differential $\Delta_{W|X}(\omega_i|X)$, lying on the half-open interval $(l', u']$, is the minimum number of bits $M_{q\Delta}$ necessary to ensure that $q_{M_{q\Delta}}[\Delta_{W|X}(\omega_i|X)]$ lies between the lower and upper bounds l' and u' :

$$\begin{aligned} l' \leq L_{M_{q\Delta}}[\Delta_{W|X}(\omega_i|X)] < \Delta_{W|X}(\omega_i|X) \leq U_{M_{q\Delta}}[\Delta_{W|X}(\omega_i|X)] \leq u' \\ l' < u', \quad -1 \leq l' \leq 1, \quad -1 \leq u' \leq 1 \end{aligned} \quad (6.9)$$

Remark: Our definition of *differential complexity* is consistent with Rissanen's definition of *stochastic complexity*, but it focuses on approximating the feature vector's *a posteriori* class differentials rather than its *a posteriori* class probabilities.

The relationship between probabilistic and differential complexity is formalized in section 6.3.3.

6.3 Exploring the Curious Relationship Between Winning a Rigged Game of Dice and Building an Efficient Classifier

Consider the C -sided die X with face probabilities $\{P_{W|X}(\omega_1 | X), \dots, P_{W|X}(\omega_C | X)\}$, which sum to one: $P_{W|X}(\omega_i | X)$ is the probability that the i th face ω_i will turn up on any given cast of the die. We assume that one die face is more likely than all the others. Using the notational conventions of section 2.4, we denote the probability of this most likely face by

$$P_{W|X}(\omega_{(1)} | X) \triangleq \max_k P_{W|X}(\omega_k | X), \quad (6.10)$$

More generally, $P_{W|X}(\omega_{(j)} | X)$ denotes the probability of the j th most likely face $\omega_{(j)}$, whereas $P_{W|X}(\omega_j | X)$ merely denotes the probability of the face with the randomly-assigned identifying index j (i.e., ω_j).

Notational Conventions for Empirical Estimates of Probabilities: Given n casts of the die in which the i th face turns up k_i times, we employ the following notational conventions:

The integer k_i denotes the number of occurrences of face ω_i , such that

$$\hat{P}_{W|X}(\omega_i | X) = \frac{k_i}{n}$$

denotes the empirical estimate of $P_{W|X}(\omega_i | X)$. The integer $k_{(i)}$ denotes the number of occurrences of the i th most likely face $\omega_{(i)}$, such that

$$\hat{P}_{W|X}(\omega_{(i)} | X) = \frac{k_{(i)}}{n}$$

denotes the empirical estimate of $P_{W|X}(\omega_{(i)} | X)$. The integer $k_{\sim(i)}$ denotes the number of occurrences of the i th empirically most likely face $\omega_{\sim(i)}$, such that

$$\hat{P}_{W|X}(\omega_{\sim(i)} | X) = \frac{k_{\sim(i)}}{n}$$

denotes the j th empirically-ranked face probability estimate. Note that the j th empirically most likely die face $\omega_{\sim(i)}$ is not necessarily the same face as the true j th most likely die face $\omega_{(i)}$ for a finite number of die casts n .

The empirical estimates of the face probabilities are multinomially distributed [33], such that for n casts of the die, the probability that there will be exactly k_i occurrences of the i th face is

$$P(n \cdot \hat{P}_{W|X}(\omega_1 | X) = k_1, n \cdot \hat{P}_{W|X}(\omega_2 | X) = k_2, \dots, n \cdot \hat{P}_{W|X}(\omega_C | X) = k_C) = \frac{n! \cdot \prod_{i=1}^C \frac{P_{W|X}(\omega_i | X)^{k_i}}{k_i!}}{n! \cdot \prod_{i=1}^C \frac{P_{W|X}(\omega_i | X)^{k_i}}{k_i!}} \quad (6.11)$$

where

$$\sum_{i=1}^C k_i = n \quad (6.12)$$

The question we would like to answer is, "How many casts of the die must occur before the most likely die face $\omega_{(1)}$ becomes empirically evident with probability α ." Given n tosses, we can identify the most likely die face by first estimating the probability of each face

$$\hat{P}_{W|X}(\omega_i | X) = \frac{k_i}{n}; \quad (6.13)$$

we then rank these estimates and choose the empirically most likely face $\omega_{\sim(1)}$ (i.e., the one with the largest estimated probability $\hat{P}_{W|X}(\omega_{\sim(1)} | X)$) as our estimate for $\omega_{(1)}$. We refer to this strategy as the *probabilistic strategy* for learning the most likely die face through empirical observation. Another way to identify the most likely die face is to estimate the discriminant differential

$$\delta_i(X) = \hat{P}_{W|X}(\omega_i | X) - \max_{j \neq i} \hat{P}_{W|X}(\omega_j | X) = \frac{k_i - \max_{j \neq i} k_j}{n} \quad (6.14)$$

for each of the C die faces. Only one of these (i.e., $\delta_{\sim(i)}(X)$) will be positive, and it will be associated with the empirically most likely die face $\omega_{\sim(1)}$ — our estimate for $\omega_{(1)}$. We refer to this strategy as the *differential strategy* for learning the most likely die face through empirical observation. We analyze the differential strategy first and follow with an analysis and comparison of the probabilistic strategy.

6.3.1 The Differential Mechanism by which the Most Likely Die Face Becomes Empirically Evident

Consider the C *a posteriori* class differentials, originally defined in (2.13):

$$\Delta_{W|X}(\omega_i | X) \triangleq P_{W|X}(\omega_i | X) - \max_{j \neq i} P_{W|X}(\omega_j | X) \quad i = 1, \dots, C \quad (6.15)$$

Note that when $i = (1)$

$$\Delta_{W|X}(\omega_{(1)} | X) = P_{W|X}(\omega_{(1)} | X) - P_{W|X}(\omega_{(2)} | X), \quad (6.16)$$

and when $i \neq (1)$

$$\Delta_{W|X}(\omega_{(i)} | X) = P_{W|X}(\omega_{(i)} | X) - P_{W|X}(\omega_{(1)} | X) \quad \forall i \geq 2 \quad (6.17)$$

Note also, by (6.15),

$$\Delta_{W|X}(\omega_{(2)} | X) = -\Delta_{W|X}(\omega_{(1)} | X) \quad (6.18)$$

Thus,

$$\sum_{i=1}^C \Delta_{W|X}(\omega_i | X) = \sum_{i=(3)}^{(C)} P_{W|X}(\omega_i | X) - (C-2) \cdot P_{W|X}(\omega_{(1)} | X), \quad (6.19)$$

and we can use the relationship $\sum_{i=1}^C P_{W|X}(\omega_i | X) = 1$ to show that the C *a posteriori* class differentials of (6.15) yield the C die face probabilities as follows:

$$P_{W|X}(\omega_{(1)} | X) = \frac{1}{C} \left[1 - \sum_{i=(2)}^{(C)} \Delta_{W|X}(\omega_i | X) \right] \quad (6.20)$$

$$P_{W|X}(\omega_{(j)} | X) = \Delta_{W|X}(\omega_{(j)} | X) + P_{W|X}(\omega_{(1)} | X) \quad \forall j \geq 2$$

From this perspective, estimating the C *a posteriori* class differentials with high precision is equivalent to estimating the C die face probabilities with high precision, and vice-versa. However, since we need only know the *signs* of the *a posteriori* class differentials to identify the most likely die face

$$\begin{aligned} \Delta_{W|X}(\omega_{(1)} | X) &> 0 \\ \Delta_{W|X}(\omega_{(j)} | X) &< 0 \quad \forall j \geq 2, \end{aligned} \quad (6.21)$$

we need only estimate each *a posteriori* class differential to one (sign) bit precision in order to identify the most likely die face. Specifically, in order to correctly identify the most likely die face $\omega_{(1)}$ with probability at least α , using the discriminant differentials $\{\delta_{\sim(1)}(X), \dots, \delta_{\sim(C)}(X)\}$, we need only ensure that the die is cast enough times so that the identity of the *empirically* most likely die face is that of the true most likely die face (i.e., $\omega_{\sim(1)} \rightarrow \omega_{(1)}$) with probability at least α . If this is the case,³ the signs of the discriminant differentials $\{\delta_{(1)}(X), \dots, \delta_{(C)}(X)\}$ match the signs of their corresponding *a posteriori* class differentials $\{\Delta_{W|X}(\omega_{(1)} | X), \dots, \Delta_{W|X}(\omega_{(C)} | X)\}$ — recall that this is the condition of (2.17) necessary for Bayesian discrimination:

³The notational conventions for the discriminant differentials follow those for the face probabilities described in the note on page 163. Thus, $\delta_j(X)$ is associated with ω_j , $\delta_{(j)}(X)$ is associated with $\omega_{(j)}$, and $\delta_{\sim(j)}(X)$ is associated with $\omega_{\sim(j)}$.

$$P(\text{sign}[\delta_{(i)}(\mathbf{X})] = \text{sign}[\Delta_{\mathcal{W}|\mathbf{X}}(\omega_{(i)}|\mathbf{X})] \quad \forall i) =$$

$$P\left(\begin{array}{l} \delta_{\sim(1)}(\mathbf{X}) = \delta_{(1)}(\mathbf{X}) > 0 \\ \delta_{(j)}(\mathbf{X}) < 0 \quad \forall j \geq 2 \end{array}\right) \geq \alpha \quad (6.22)$$

When (6.22) holds, the largest empirical face probability reflects the identity of the true most likely die face:

$$P\left(\hat{P}_{\mathcal{W}|\mathbf{X}}(\omega_{\sim(1)}|\mathbf{X}) = \hat{P}_{\mathcal{W}|\mathbf{X}}(\omega_{(1)}|\mathbf{X}) > \hat{P}_{\mathcal{W}|\mathbf{X}}(\omega_{(j)}|\mathbf{X}) \quad \forall j \geq 2\right) \geq \alpha \quad (6.23)$$

The computation to determine the smallest number of die casts n for which (6.22) and (6.23) are satisfied is intractable for all but small values of C and n . For this reason, we make the following assumption.

Assumption 6.1 *We assume that the empirical estimate of the most likely die face's probability is greater than the empirical estimates of all other die face probabilities if it is greater than the empirical estimate of the second most likely die face's probability. Mathematically,*

$$\begin{aligned} \text{We assume that if } \hat{P}_{\mathcal{W}|\mathbf{X}}(\omega_{(1)}|\mathbf{X}) > \hat{P}_{\mathcal{W}|\mathbf{X}}(\omega_{(2)}|\mathbf{X}) \\ \text{then } \hat{P}_{\mathcal{W}|\mathbf{X}}(\omega_{(1)}|\mathbf{X}) > \hat{P}_{\mathcal{W}|\mathbf{X}}(\omega_{(j)}|\mathbf{X}) \quad \forall j > 2 \end{aligned} \quad (6.24)$$

Under this assumption, the multinomial expansion implied by (6.22) and (6.23) can be simplified as follows. **Note:** we use the short-hand notation $\hat{P}(\omega_j)$ to denote $\hat{P}_{\mathcal{W}|\mathbf{X}}(\omega_j|\mathbf{X})$ and $P(\omega_j)$ to denote $P_{\mathcal{W}|\mathbf{X}}(\omega_j|\mathbf{X})$; this makes the following formula more readable.

$$P\left(\hat{P}(\omega_{(1)}) > \hat{P}(\omega_{(j)}) \quad \forall j \geq 2\right) =$$

$$P(\delta_{(1)}(\mathbf{X}) > 0, \delta_{(j)}(\mathbf{X}) < 0 \quad \forall j \geq 2) \quad (6.25)$$

$$\cong P\left(\hat{P}(\omega_{(1)}) > \hat{P}(\omega_{(2)})\right) \quad (6.26)$$

$$= \sum_{k_{(1)}=\lambda_1}^n P\left(n \cdot \hat{P}(\omega_{(1)}) = k_{(1)}, n \cdot \hat{P}(\omega_{(2)}) < k_{(1)}\right)$$

$$\cong n! \sum_{k_{(1)}=\lambda_1}^{v_1} \frac{P(\omega_{(1)})^{k_{(1)}}}{k_{(1)}!} \cdot \left[\sum_{k_{(2)}=\lambda_2}^{v_2} \frac{P(\omega_{(2)})^{k_{(2)}} (1 - P(\omega_{(1)}) - P(\omega_{(2)}))^{(n-k_{(1)}-k_{(2)})}}{k_{(2)}! (n - k_{(1)} - k_{(2)})!} \right] \quad (6.27)$$

$$\geq \alpha$$

where⁴

$$\lambda_1 = \begin{cases} 1, & C = 2 \\ \max\left(1, \frac{n-k_{(2)}}{C-1} + 1\right), & C > 2 \end{cases} \quad (6.28)$$

$$v_1 = n \quad (6.29)$$

$$\lambda_2 = 0 \quad (6.30)$$

$$v_2 = \min(k_{(1)} - 1, n - k_{(1)}) \quad (6.31)$$

The sufficiency of (6.25) – (6.27) as an indicator that we have correctly identified the most likely die face rests on the validity of assumption 6.1, which allows the reduction of (6.25) to (6.26). The upper bound on $k_{(2)}$ in (6.27), given by v_2 in (6.31), simply enforces the constraint of (6.12) — namely, that all the k s sum to n . The lower bound on $k_{(1)}$ in (6.27), given by λ_1 in (6.28), is related to the reduction of (6.25) to (6.27). If λ_1 were 1, then (6.27) would be an exact expression of (6.26). However, we really want (6.27) to be a good approximation to (6.25). A necessary and sufficient condition for $\hat{P}_{W|X}(\omega_{(1)} | X) > \hat{P}_{W|X}(\omega_{(j)} | X) \quad \forall j \geq 2$ is that the most likely die face occur more frequently than any other. Thus, a necessary (albeit insufficient) condition for $\hat{P}_{W|X}(\omega_{(1)} | X) > \hat{P}_{W|X}(\omega_{(j)} | X) \quad \forall j \geq 2$ is

$$\sum_{j=3}^C k_{(j)} = n - k_{(1)} - k_{(2)} < (C - 2)k_{(1)} \quad \forall C > 2 \quad (6.32)$$

such that

$$k_{(1)} > \frac{n - k_{(2)}}{C - 1} \quad \forall C > 2 \quad (6.33)$$

— the constraint enforced by (6.28). Clearly there will be cases for which (6.33) holds, but our assumption that $\hat{P}_{W|X}(\omega_{(1)} | X) > \hat{P}_{W|X}(\omega_{(j)} | X) \quad j \geq 2$ does not (i.e., (6.31) is a necessary but *insufficient* condition for assumption 6.1 to hold in all cases). Thus it is important to qualify assumption 6.1 with the scenarios under which it might fail, yielding a higher-than-warranted estimate of the probability in (6.25). We envisage two such scenarios: 1) the scenario in which n is small, and 2) the scenario in which $P_{W|X}(\omega_{(2)} | X) \rightarrow P_{W|X}(\omega_{(C)} | X)$ are nominally equal. Both scenarios can give rise to cases in which the empirical estimates of $\{\hat{P}_{W|X}(\omega_{(2)} | X), \dots, \hat{P}_{W|X}(\omega_{(C)} | X)\}$ do not reflect the rankings of the true probabilities $\{P_{W|X}(\omega_{(2)} | X), \dots, P_{W|X}(\omega_{(C)} | X)\}$.

The significance of the first scenario is diminished by the requirement that $P_{W|X}(\omega_{(1)} | X)$ be considerably greater than $P_{W|X}(\omega_{(2)} | X)$ in order for their corresponding estimates to be ranked appropriately

⁴The operator $\max(a, b)$ returns the greater of a and b . Likewise, the operator $\min(a, b)$ returns the lesser of a and b .

with high probability for small n . This would suggest that our estimate of the probability in (6.25) is not significantly affected by assumption 6.1.

In the second scenario, if $P_{W|X}(\omega_{(1)} | X)$ is only marginally larger than $P_{W|X}(\omega_{(2)} | X)$, then n will have to be so large to assure with high confidence that the rankings of $\hat{P}_{W|X}(\omega_{(1)} | X)$ and $\hat{P}_{W|X}(\omega_{(2)} | X)$ are valid, the effect of the lesser-ranked probabilities on the approximation of (6.27) will be minimal. If on the other hand $P_{W|X}(\omega_{(1)} | X)$ is significantly larger than $P_{W|X}(\omega_{(2)} | X)$, then this scenario becomes a special case of the first scenario, and again we are reasonably safe in our assumption.

These mitigating factors notwithstanding, there are surely cases in which assumption 6.1 does not hold. In these cases, (6.27) over-estimates the probability that the most likely die face has become empirically evident after n casts of the die.

It is enlightening to recognize that the necessary condition for a reliable 1-bit approximation of $\Delta_{W|X}(\omega_{(1)} | X)$ given in (6.25) is equivalent to the condition

$$P(0 < \delta_{(1)}(X) \leq 1) \geq \alpha \quad (6.34)$$

This condition is, in turn, a specific example of the more general necessary condition for a reliable M_q -bit approximation $\delta_{(1)}(X)$ of $\Delta_{W|X}(\omega_{(1)} | X)$:

$$P(L_{M_q\Delta}[\Delta_{W|X}(\omega_{(1)} | X)] < \delta_{(1)}(X) \leq U_{M_q\Delta}[\Delta_{W|X}(\omega_{(1)} | X)]) \geq \alpha \quad (6.35)$$

where $L_{M_q\Delta}$ and $U_{M_q\Delta}$ are the lower and upper bounds on the M_q -bit quantization interval for $q_{M_q\Delta}[\Delta_{W|X}(\omega_{(1)} | X)]$ (recall definition 6.2). Given a positive integer n and a real number $z \in (-1, 1]$, let $k_{L_{M_q\Delta}}[z]$ be the smallest value of k for which $\frac{k}{n} > L_{M_q\Delta}[z]$, where $L_{M_q\Delta}[z]$ is given by (6.5); likewise, let $k_{U_{M_q\Delta}}[z]$ be the largest value of k for which $\frac{k}{n} \leq U_{M_q\Delta}[z]$, where $U_{M_q\Delta}[z]$ is given by (6.5). More formally,

$$\begin{aligned} k_{L_{M_q\Delta}}[z] &\triangleq \lfloor n \cdot L_{M_q\Delta}[z] \rfloor + 1 \\ k_{U_{M_q\Delta}}[z] &\triangleq \lfloor n \cdot U_{M_q\Delta}[z] \rfloor \end{aligned} \quad (6.36)$$

Then if z_0 and z_1 are in adjacent $M_q\Delta$ -bit quantization intervals, with z_1 in the "upper" interval (as formally specified by (6.6) – (6.7)), the following relationship holds:

$$k_{L_{M_q\Delta}}[z_1] = k_{U_{M_q\Delta}}[z_0] + 1 \quad (6.37)$$

With these relationships, (6.25) – (6.31) can be generalized as follows. Note: again, we use the short-hand notation $\hat{P}(\omega_j)$ to denote $\hat{P}_{W|X}(\omega_j | X)$ and $P(\omega_j)$ to denote $P_{W|X}(\omega_j | X)$. Likewise, we use $\Delta(\omega_j)$ to denote $\Delta_{W|X}(\omega_j | X)$ in the interest of notational simplicity.

$$P(L_{M_{q\Delta}}[\Delta(\omega_{(1)})] < \delta_{(1)}(\mathbf{X}) \leq U_{M_{q\Delta}}[\Delta(\omega_{(1)})], \delta_{(j)}(\mathbf{X}) < 0 \quad \forall j \geq 2) =$$

$$P(L_{M_{q\Delta}}[\Delta(\omega_{(1)})] < \delta_{(1)}(\mathbf{X}) \leq U_{M_{q\Delta}}[\Delta(\omega_{(1)})]) \quad (6.38)$$

$$= P(0 < \delta_{(1)}(\mathbf{X}) - L_{M_{q\Delta}}[\Delta(\omega_{(1)})] \leq U_{M_{q\Delta}}[\Delta(\omega_{(1)})] - L_{M_{q\Delta}}[\Delta(\omega_{(1)})])$$

$$= P(\hat{P}(\omega_{(2)}) < \hat{P}(\omega_{(1)}) - L_{M_{q\Delta}}[\Delta(\omega_{(1)})] \leq \hat{P}(\omega_{(2)}) + U_{M_{q\Delta}}[\Delta(\omega_{(1)})] - L_{M_{q\Delta}}[\Delta(\omega_{(1)})])$$

$$= P(\hat{P}(\omega_{(2)}) < \hat{P}(\omega_{(1)}) - L_{M_{q\Delta}}[\Delta(\omega_{(1)})], \hat{P}(\omega_{(2)}) \geq \hat{P}(\omega_{(1)}) - U_{M_{q\Delta}}[\Delta(\omega_{(1)})])$$

$$\cong n! \sum_{k_{(1)}=\lambda_1}^{v_1} \frac{P(\omega_{(1)})^{k_{(1)}}}{k_{(1)}!} \left[\sum_{k_{(2)}=\lambda_2}^{v_2} \frac{P(\omega_{(2)})^{k_{(2)}} (1 - P(\omega_{(1)}) - P(\omega_{(2)}))^{(n-k_{(1)}-k_{(2)})}}{k_{(2)}! (n - k_{(1)} - k_{(2)})!} \right] \quad (6.39)$$

$$\geq \alpha$$

where

$$\lambda_1 = \begin{cases} k_{L_{M_{q\Delta}}}[\Delta(\omega_{(1)})], & C = 2 \\ \max(k_{L_{M_{q\Delta}}}[\Delta(\omega_{(1)})], \frac{n-k_{(2)}}{C-1} + 1), & C > 2 \end{cases} \quad (6.40)$$

$$v_1 = n \quad (6.41)$$

$$\lambda_2 = \max(0, k_{(1)} - k_{U_{M_{q\Delta}}}[\Delta(\omega_{(1)})]) \quad (6.42)$$

$$v_2 = \min(k_{(1)} - k_{L_{M_{q\Delta}}}[\Delta(\omega_{(1)})], n - k_{(1)}) \quad (6.43)$$

The limits of (6.40), (6.41), and (6.43) correspond to those in (6.28), (6.29), and (6.31); the limit of (6.42) is an additional one imposed by our desire for an $M_q > 1$ bit approximation, which places both lower and upper bounds on the discriminant differential approximation $\delta_{(1)}(\mathbf{X})$.

Theorem 6.1 *The differential learning strategy attempts to approximate the a posteriori class differentials $\{\Delta_{W|X}(\omega_{(1)} | \mathbf{X}), \dots, \Delta_{W|X}(\omega_{(C)} | \mathbf{X})\}$ to one (sign) bit precision. The probability that differential learning will achieve this goal, given a sample size n , is higher than the probability of success for any other learning strategy by which the most likely die face $\omega_{(1)}$ might be ascertained.*

Proof : As mentioned earlier, the necessary and sufficient condition for the most likely die face to be empirically evident after n casts of the die is simply that the number of occurrences of the most likely face be greater than the number of occurrences of any other face. Equivalently, the discriminant differentials $\{\delta_{(1)}(\mathbf{X}), \dots, \delta_{(C)}(\mathbf{X})\}$ must approximate the a posteriori class differentials

$\{\Delta_{W|X}(\omega_{(1)} | \mathbf{X}), \dots, \Delta_{W|X}(\omega_{(C)} | \mathbf{X})\}$ to at least one (sign) bit precision. Thus, if we employ the differential strategy of approximating the *a posteriori* class differentials, we end up choosing the empirically most likely face $\omega_{\sim(1)}$ as our estimate for the true most likely face $\omega_{(1)}$. Our probability of success using this strategy (i.e., successfully approximating the *a posteriori* class differentials to one bit precision, thereby identifying $\omega_{(1)}$ correctly) is equal to the probability of all possible cast sequences of length n for which $k_{(1)}$ is maximal. This, in turn, is equal to the sum of all legal expressions (that is, all expressions in which the k s sum to n) of the multinomial probability mass function (pmf) given by (6.11) and (6.12). We re-express the multinomial pmf here, using rank indices:

$$P(k_{(1)}, \dots, k_{(C)}) = n! \cdot \prod_{i=1}^C \frac{P_{W|X}(\omega_{(i)} | \mathbf{X})^{k_{(i)}}}{k_{(i)}!}; \quad \sum_{i=1}^C k_{(i)} = n \quad (6.44)$$

Since the multinomial pmf in (6.44) is non-negative, and since the sum of all the pmf terms satisfying the constraints

$$k_{(1)} > k_{(j)} \quad \forall j \geq 2; \quad \sum_{i=1}^C k_{(i)} = n \quad (6.45)$$

represents the cumulative probability of *all* cast sequences of length n for which $\omega_{(1)}$ is empirically evident, the cumulative probability that differential learning will meet its goal is the largest possible for any strategy by which the most likely die face might be ascertained. That is, the differential strategy described in theorem 6.1 has the greatest probability of success in achieving its goal. ■

Remark: Unfortunately, there is no compact way to express the cumulative multinomial probability described in the preceding proof. The only way to compute the sum exactly is to evaluate each and every possible combination of k s to determine if it satisfies the constraints of (6.45); if it does, then the corresponding multinomial expression is added to the cumulative sum. As mentioned earlier, this computation is intractable for all but very small values of C and n . We resort to the approximations of (6.27) and (6.39) in order to estimate the probabilities of (6.25) and (6.38) via a tractable computation (see section 6.4). It is interesting to note that the logic of the preceding proof holds for the approximations as well. Specifically, it should be clear from a comparison of (6.27) – (6.31) and (6.39) – (6.43) that the probability $P(L_{M_{q\Delta}}[\Delta(\omega_{(1)})] < \delta_{(1)}(\mathbf{X}) \leq U_{M_{q\Delta}}[\Delta(\omega_{(1)})])$ is largest for a given sample size n when $M_{q\Delta} = 1$ such that $L_{M_{q\Delta}}[\Delta(\omega_{(1)})] = 0$ and $U_{M_{q\Delta}}[\Delta(\omega_{(1)})] = 1$ (i.e., $k_{L_{M_{q\Delta}}}[\Delta(\omega_{(1)})] = 1$ and $k_{U_{M_{q\Delta}}}[\Delta(\omega_{(1)})] = n$). In simple terms, attempting to estimate $\Delta(\omega_{(1)})$ — recall that this notation is short-hand for the top-ranked *a posteriori* class differential $\Delta_{W|X}(\omega_{(1)} | \mathbf{X})$ — with more than one bit precision constitutes a learning strategy with a lower probability of success than the differential learning

strategy. This is because the goal of estimating the *a posteriori* class differentials with higher precision places tighter constraints on the values of $\{k_{(1)}, \dots, k_{(C)}\}$ that satisfy the more rigorous learning goal. Because there are some cast sequences of length n that satisfy (6.45) but do not satisfy the more stringent goal, there will be fewer terms in the multinomial sum for the higher-precision strategy. Thus, the higher-precision strategy will have a lower probability of success.

Corollary 6.1 *The differential strategy for learning the most likely die face requires the minimum differential complexity necessary for the task.*

Proof : The discriminant differential $\delta_{(1)}(\mathbf{X})$ need only approximate the *a posteriori* class differential $\Delta_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X})$ in sign in order for the most likely die face to be evident. That is, the necessary and sufficient condition on $\delta_{(1)}(\mathbf{X})$ for correctly identifying the most likely die face is

$$\text{sign}[\delta_{(1)}(\mathbf{X})] = \text{sign}[\Delta_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X})], \quad (6.46)$$

which follows directly from the constraint $k_{(1)} > k_{(j)} \quad \forall j \geq 2$ in (6.45). Equivalently,

$$q_{M_{q\Delta \min}}[\delta_{(1)}(\mathbf{X})] = q_{M_{q\Delta \min}}[\Delta_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X})]; \quad M_{q\Delta \min} = 1 \quad (6.47)$$

■

Remark: Note that the condition of (6.46) is precisely that of (2.17).

Corollary 6.2 *The differential learning goal of approximating $\Delta_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X})$ to at least one (sign) bit precision with confidence not less than α requires the smallest sample size n_{Δ} of any learning strategy by which the most likely die face $\omega_{(1)}$ might be ascertained.*

Proof : The proof follows immediately from the proof of theorem 6.1. ■

6.3.2 The Probabilistic Mechanism by which the Most Likely Die Face Becomes Empirically Evident

As described earlier, the probabilistic strategy for identifying the most likely die face involves estimating the C face probabilities $\{P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X}), \dots, P_{\mathbf{W}|\mathbf{X}}(\omega_{(C)} | \mathbf{X})\}$. In order for us to identify the most likely face $\omega_{(1)}$ with probability not less than α , we must distinguish $\hat{P}_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X})$ from $\hat{P}_{\mathbf{W}|\mathbf{X}}(\omega_{(2)} | \mathbf{X})$ with probability not less than α . This implies that we choose a quantization level M_{qp} such that

$$P \left(\begin{array}{l} L_{M_{qp}}[P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X})] < \hat{P}_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X}) \leq U_{M_{qp}}[P_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X})], \\ q_{M_{qp}}[\hat{P}_{\mathbf{W}|\mathbf{X}}(\omega_{(1)} | \mathbf{X})] > q_{M_{qp}}[\hat{P}_{\mathbf{W}|\mathbf{X}}(\omega_{(2)} | \mathbf{X})] \end{array} \right) \geq \alpha \quad (6.48)$$

The minimum value of M_{qP} that satisfies (6.48) for asymptotically large n (by making the quantized difference $q_{M_{qP}}[\hat{P}_{W|X}(\omega_{(1)} | X)] - q_{M_{qP}}[\hat{P}_{W|X}(\omega_{(2)} | X)]$ implied by (6.48) exceed one least significant bit) is

$$M_{qP \min} = \begin{cases} \underbrace{1}_{\text{sign bit}} + \underbrace{\lceil -\log_2 [\Delta_{W|X}(\omega_{(1)} | X)] \rceil}_{\text{magnitude bits}}, & -\log_2 [P_{W|X}(\omega_{(j)} | X)] \notin \mathcal{Z}^+, \\ & j \in \{1, 2\} \\ \underbrace{1}_{\text{sign bit}} + \underbrace{\lceil -\log_2 [\Delta_{W|X}(\omega_{(1)} | X)] \rceil + 1}_{\text{magnitude bits}}, & \text{otherwise} \end{cases} \quad (6.49)$$

Recall that $\mathcal{Z}^+$ represents the set of all positive integers. Note also that the conditional nature of $M_{qP \min}$ in (6.49) prevents the case in which $\lim_{\epsilon \rightarrow 0} P_{W|X}(\omega_{(1)} | X) - \epsilon = L_{M_q}[P_{W|X}(\omega_{(1)} | X)]$ or $P_{W|X}(\omega_{(2)} | X) = U_{M_q}[P_{W|X}(\omega_{(2)} | X)]$; either case would require an infinitely large sample size before the variance of the corresponding estimate became small enough to distinguish $q_M[\hat{P}_{W|X}(\omega_{(1)} | X)]$ from $q_M[\hat{P}_{W|X}(\omega_{(2)} | X)]$. The sign bit in (6.49) is not required to estimate the probabilities in (6.48), since all probabilities are positive; it merely follows the conventions of section 6.2. Thus, the probabilistic complexity of the M_{qP} -bit approximation is actually $M_{qP} - 1$ rather than M_{qP} .

Using $M_{qP} = M_{qP \min}$ from (6.49), (6.48) is given by

$$\begin{aligned} P \left(\begin{array}{l} L_{M_{qP}}[P_{W|X}(\omega_{(i)} | X)] < \hat{P}_{W|X}(\omega_{(i)} | X) \leq U_{M_{qP}}[P_{W|X}(\omega_{(i)} | X)], \\ q_{M_{qP}}[\hat{P}_{W|X}(\omega_{(1)} | X)] > q_{M_{qP}}[\hat{P}_{W|X}(\omega_{(2)} | X)] \end{array} \right) = \\ n! \sum_{k_{(1)}=\lambda_{(1)}}^{v_{(1)}} \cdots \sum_{k_{(c)}=\lambda_{(c)}}^{v_{(c)}} \prod_{i=1}^c \frac{P_{W|X}(\omega_{(i)} | X)^{k_{(i)}}}{k_{(i)}!}; \quad \sum_{i=1}^c k_{(i)} = n \quad (6.50) \\ \geq \alpha, \end{aligned}$$

where

$$\lambda_i = k_{L_{M_{qP \min}}}[P_{W|X}(\omega_{(i)} | X)] \quad (6.51)$$

$$v_i = k_{U_{M_{qP \min}}}[P_{W|X}(\omega_{(i)} | X)] \quad (6.52)$$

$$\lambda_1 > v_j \quad \forall j \leq 2 \quad (6.53)$$

when we require that each of the approximations of $\hat{P}_{W|X}(\omega_{(i)} | X)$ fall within a single quantization interval with probability not less than α . This is the case if our goal is to estimate $\{P_{W|X}(\omega_{(1)} | X), \dots, P_{W|X}(\omega_{(c)} | X)\}$ with M_{qP} -bit precision.

Of course, if our goal is merely to distinguish between the quantized approximations $q_M[\hat{P}_{W|X}(\omega_{(1)} | X)]$ and $q_M[\hat{P}_{W|X}(\omega_{(2)} | X)]$ — the weakest form of probabilistic learning — we need only enforce a lower bound on the approximation of $P_{W|X}(\omega_{(1)} | X)$ and an upper bound on the approximation of $P_{W|X}(\omega_{(2)} | X)$. This leads to the following approximate formula (in short-hand notation), analogous to those of (6.25) and (6.38):

$$\begin{aligned}
 & P\left(q_M[\hat{P}(\omega_{(1)})] > q_M[\hat{P}(\omega_{(2)})], \forall j > 1\right) \\
 & \cong n! \sum_{k_{(1)}=\lambda_1}^{v_1} \frac{P(\omega_{(1)})^{k_{(1)}}}{k_{(1)}!} \left[\sum_{k_{(2)}=\lambda_2}^{v_2} \frac{P(\omega_{(2)})^{k_{(2)}} (1 - P(\omega_{(1)}) - P(\omega_{(2)}))^{(n-k_{(1)}-k_{(2)})}}{k_{(2)}! (n - k_{(1)} - k_{(2)})!} \right] \quad (6.54) \\
 & \geq \alpha
 \end{aligned}$$

where

$$\lambda_1 = \max\left(B + 1, \frac{n - k_{(2)}}{C - 1} + 1\right) \quad \forall C > 2 \quad (6.55)$$

$$v_1 = n \quad (6.56)$$

$$\lambda_2 = 0 \quad (6.57)$$

$$v_2 = \min(B, n - k_{(1)}) \quad (6.58)$$

$$B = \{B_{M_{q \min}}\} = k_{U_{M_{q \min}}} [P(\omega_{(2)})] = k_{L_{M_{q \min}}} [P(\omega_{(1)})] - 1 \quad (6.59)$$

The restriction of (6.55) stems from (6.33), since this condition is necessary (although, again, insufficient) to ensure the validity of the approximation in (6.54).

Equation (6.59) illustrates that there is one and only one boundary B separating our quantized estimates of $P_{W|X}(\omega_{(1)} | X)$ and $P_{W|X}(\omega_{(2)} | X)$ for $M_{qP \min}$ -bit quantization. If, however, we use $(M_{qP} > M_{qP \min})$ -bit quantization along with equations (6.55) – (6.58), there are many boundaries that can be used in (6.54), via (6.55) and (6.58). Specifically, there is a simple recursion by which every possible boundary B for M_{qP} -bit quantization leads to itself and two additional boundaries for $(M_{qP} + 1)$ -bit quantization:

<u>M_{qp}-bit quantization</u>	<u>$(M_{qp} + 1)$-bit quantization</u>	
$B \rightarrow$	$\left\{ \begin{array}{ll} \left\{ \begin{array}{ll} B + 2^{-M_{qp}}, & B + 2^{-M_{qp}} \leq L_{M_q+1}[P_{W X}(\omega_{(1)} X)] \\ \text{no boundary,} & \text{otherwise} \end{array} \right. \\ B \\ \left\{ \begin{array}{ll} B - 2^{-M_{qp}}, & B - 2^{-M_{qp}} \geq U_{M_q+1}[P_{W X}(\omega_{(2)} X)] \\ \text{no boundary,} & \text{otherwise} \end{array} \right. \end{array} \right.$	(6.60)

Indeed, one can show that there are

$$|B_{M_q}| = \begin{cases} [L_{M_q}[P_{W|X}(\omega_{(1)} | X)] - U_{M_q}[P_{W|X}(\omega_{(2)} | X)] \cdot 2^{(M_{qp}-1)} + 1, & M_{qp} \geq M_{qp \min} \\ 0, & \text{otherwise} \end{cases} \quad (6.61)$$

members⁵ in the set of possible boundaries between $q_M[\hat{P}_{W|X}(\omega_{(1)} | X)]$ and $q_M[\hat{P}_{W|X}(\omega_{(2)} | X)]$ that can be used for B in equations (6.55) and (6.58).

Corollary 6.3 *The probabilistic learning strategy attempts to approximate the a posteriori class probabilities $\{P_{W|X}(\omega_{(1)} | X), \dots, P_{W|X}(\omega_{(C)} | X)\}$ to a specified level of precision, measured by the probabilistic complexity $M_{qp} - 1$ of the approximations. As a result, the strategy also attempts to approximate the a posteriori class differentials $\{\Delta_{W|X}(\omega_{(1)} | X), \dots, \Delta_{W|X}(\omega_{(C)} | X)\}$ to a specified level of precision, measured by the differential complexity $M_{q\Delta} = M_{qp}$ of the approximations. Probabilistic learning therefore requires higher functional complexity and has a lower probability of success than differential learning.*

Proof : Since differential learning attempts to approximate the a posteriori class differentials to one sign bit precision, the increased complexity requirements of probabilistic learning (as measured by its differential complexity) are self evident. Whether we use the exact expression of (6.50) or the approximation of (6.54), it is also clear that probabilistic learning places tighter constraints on the summation bounds for the cumulative multinomial expression. Thus, by the arguments of the proof to theorem 6.1, the probability that probabilistic learning will achieve its more precise goal of estimating the die face probabilities with M_{qp} bits of precision is less than the probability that differential learning will achieve its minimum-complexity goal of approximating the a posteriori class differentials to at least one sign bit precision. ■

⁵We use the notation $|B_{M_q}|$ to denote the cardinality of (i.e., the number of elements or members in) the set B_{M_q} .

We emphasize that the probability of success we discuss in theorem 6.1 and corollary 6.2 pertains to the goal of approximating $\{\Delta_{W|X}(\omega_{(1)} | X), \dots, \Delta_{W|X}(\omega_{(c)} | X)\}$ to $M_{q\Delta} = 1$ sign bit precision. This is the same as the goal of identifying the most likely die face $\omega_{(1)}$. The probability of success we discuss in corollary 6.3 pertains to a *different*, more complex goal: that of approximating $\{\Delta_{W|X}(\omega_{(1)} | X), \dots, \Delta_{W|X}(\omega_{(c)} | X)\}$ to $M_{q\Delta} > 1$ bit precision. Corollary 6.3 therefore does *not* assert that probabilistic learning is less likely than differential learning to identify the most likely die face *unless* the probabilistic complexity allocated for the learning process is inadequate. That is, if $M_{qp} < M_{qp \min}$ in (6.49), there will be insufficient functional complexity to distinguish $P_{W|X}(\omega_{(1)} | X)$ from $P_{W|X}(\omega_{(2)} | X)$, and probabilistic learning will fail to identify the most likely die face $\omega_{(1)}$ for *any* number of die casts. If, on the other hand, $M_{qp} \gg M_{qp \min}$, there will be sufficient functional complexity to distinguish the *a posteriori* class probabilities and to approximate the *a posteriori* class differentials with high precision. Thus, if the most likely die face is evident, probabilistic learning will identify it — with the same probability of success that differential learning has — given sufficient functional complexity.

This fact makes sense intuitively: the n casts of the die *alone* determine whether or not $\omega_{(1)}$ is empirically evident. The learning strategy by which we estimate the identity of $\omega_{(1)}$ has no effect on the observable statistics of the game; it affects only what we can infer from those statistics. The advantage of differential learning therefore lies in its ability to identify the most likely die face as soon as it becomes evident in the sequence of casts, while requiring the least differential complexity necessary to achieve this goal. *Indeed, theorem 6.1 and corollaries 6.1 and 6.2 are the information-theoretic equivalents of theorem 3.1 and corollary 3.1.* The minimum-complexity requirement of differential learning is an advantage from the standpoint of generalization since, under VC analysis [137, 136], excessive complexity is anathema.

6.3.3 Discriminant Information versus Probabilistic Information

If we approximate each of the die face probabilities $\{P_{W|X}(\omega_{(1)} | X), \dots, P_{W|X}(\omega_{(c)} | X)\}$ with $M_{qp} = M - 1$ bits of probabilistic complexity,⁶ then we approximate each of the *a posteriori* class differentials $\{\Delta_{W|X}(\omega_{(1)} | X), \dots, \Delta_{W|X}(\omega_{(c)} | X)\}$ with $M_{q\Delta} = M$ bits of differential complexity. This follows immediately from (6.15) and (6.20), which allow us to express the differentials in terms of the probabilities and vice-versa. The relationship between probabilistic and differential complexity allows us to make a direct comparison between the functional complexity requirements of differential learning and probabilistic learning.

⁶Recall that the sign bit is superfluous when the fixed-point binary representation described in section 6.2 is being used to approximate a (non-negative) probability.

The Information-Theoretic Argument for Differential Learning

Occam's razor [130, 21] stipulates that the complexity of an estimate's representation need not and should not exceed the information contained in the empirical data sample used to compute the estimate. This is born out by the cumulative probability expressions of sections 6.3.1 and 6.3.2: there is always more probably $M_{q\Delta \min} = 1$ bit of discriminant information in n casts of the die than there are $M_{q\Delta} > 1$ bits of discriminant information. When n is large, such that the *a posteriori* class differentials can probably be estimated with $M_{q\Delta} \gg 1$ -bit precision, the information content of the cast sequence justifies our doing this. Equivalently, we are justified in estimating all the die face probabilities with high precision. However, when n is small — as it almost always is in real-world learning/pattern recognition tasks — the information content of the cast sequence justifies no more than our estimating $\Delta_{W|X}(\omega_{(1)} | X)$ with one bit of precision. Equivalently, we are justified only in estimating the identity of the most likely die face $\omega_{(1)}$.

6.4 Bounds on the Training Sample Size Requirements of the Differential and Probabilistic Learning Strategies

Equation (6.27) is an approximate expression of the probability that the discriminant differential $\delta_{\omega_{(1)}}(X)$ associated with the most likely die face $\omega_{(1)}$ will be positive following n casts of the die X . Equation (6.54) is an approximate expression of the probability that the estimate $\hat{P}_{W|X}(\omega_{(1)} | X)$ will be greater than the threshold value $\frac{B}{n}$ and the estimate $\hat{P}_{W|X}(\omega_{(2)} | X)$ will be less than $\frac{B}{n}$ following n casts of the die. Thus (6.27) states the approximate probability that the goal of differential learning will be reached in n casts of the die; likewise, (6.54) states the approximate probability that the weakest goal of probabilistic learning will be reached in the same n casts.

These two equations can be evaluated numerically in order to estimate via iterative search the minimum values of n at which the differential (n_{Δ}) and probabilistic (n_P) learning goals will be reached with specified probability, given particular values of $P_{W|X}(\omega_{(1)} | X)$, $P_{W|X}(\omega_{(2)} | X)$, and C . The numerically estimated values of n_{Δ} and n_P are generally quite close to the empirical estimates obtained via Monte Carlo simulations, the one exception being when C is large and $P_{W|X}(\omega_{(2)} | X)$ is small. As mentioned in section 6.3.1, assumption 6.1 — on which the approximations of (6.27) and (6.54) are predicated — fails to hold under these circumstances, so n_{Δ} and n_P tend to be under-estimated.

Since we are looking for a greatest lower bound on n , above which each learning strategy is *guaranteed* to achieve its goal, we would prefer estimators of n_{Δ} and n_P that are positively biased, rather than negatively biased. Moreover, the iterative search required to estimate n_{Δ} and n_P numerically is computationally intensive. This motivates us to derive greatest lower bounds on n using classical techniques.

6.4.1 A Greatest Lower Bound on n_Δ

For the differential learning strategy, we want to know the value of n_Δ above which the discriminant differential $\delta_{\omega_{(1)}}(\mathbf{X})$ associated with the most likely die face $\omega_{(1)}$ is non-positive with probability less than $d = 1 - \alpha$. This is a "one-sided tail" probability, which can be bounded from above by a two-sided tail probability. Using short-hand notation for $\hat{P}_{W|\mathbf{X}}(\omega_{(i)} | \mathbf{X})$, $P_{W|\mathbf{X}}(\omega_{(i)} | \mathbf{X})$, and $\Delta_{W|\mathbf{X}}(\omega_{(1)} | \mathbf{X})$, the bounding inequalities are

$$P(\delta_{\omega_{(1)}}(\mathbf{X}) \leq 0) \leq P(|\delta_{\omega_{(1)}}(\mathbf{X}) - \Delta(\omega_{(1)})| \geq \Delta(\omega_{(1)})) \quad (6.62)$$

$$\leq \frac{\text{Var}[\delta_{\omega_{(1)}}(\mathbf{X})]}{(\Delta(\omega_{(1)}))^2} \quad (6.63)$$

The inequality in (6.62) represents the two-sided upper bound on the one-sided probability, and the inequality in (6.63) is an application of Chebyshev's inequality (e.g., [33, pg. 219]).

Since $\Delta(\omega_{(1)}) = P(\omega_{(1)}) - P(\omega_{(2)})$, and we operate under assumption 6.1, we assume $\delta_{\omega_{(1)}}(\mathbf{X}) = \hat{P}(\omega_{(1)}) - \hat{P}(\omega_{(2)})$. Although the collective empirical face probabilities of the die are multinomially distributed, individual face probabilities are binomially distributed (i.e., in any given cast, the face $\omega_{(i)}$ turns up with probability $P(\omega_{(i)})$ and fails to turn up with probability $P(\neg\omega_{(i)}) = 1 - P(\omega_{(i)})$). Thus, the variance of $\hat{P}(\omega_{(i)})$ is given by

$$\text{Var}[\hat{P}(\omega_{(i)})] = \frac{P(\omega_{(i)}) \cdot (1 - P(\omega_{(i)}))}{n} \quad (6.64)$$

We make the invalid but simplifying assumption that $\hat{P}(\omega_{(1)})$ and $\hat{P}(\omega_{(2)})$ are independent. This allows us to express the variance of $\delta_{\omega_{(1)}}(\mathbf{X})$ by

$$\begin{aligned} \text{Var}[\delta_{\omega_{(1)}}(\mathbf{X})] &\approx \text{Var}[\hat{P}(\omega_{(1)})] + \text{Var}[\hat{P}(\omega_{(2)})] \\ &= \frac{P(\omega_{(1)}) \cdot (1 - P(\omega_{(1)})) + P(\omega_{(2)}) \cdot (1 - P(\omega_{(2)}))}{n} \end{aligned} \quad (6.65)$$

Thus, by (6.62), (6.63), and (6.65), the probability that the discriminant differential will be non-positive (i.e., that the most likely die face will *not* be evident) after n casts of the die is bounded from above as follows:

$$P(\delta_{\omega_{(1)}}(\mathbf{X}) \leq 0) \leq \frac{P(\omega_{(1)}) \cdot (1 - P(\omega_{(1)})) + P(\omega_{(2)}) \cdot (1 - P(\omega_{(2)}))}{n \cdot (P(\omega_{(1)}) - P(\omega_{(2)}))^2} \quad (6.66)$$

If we wish this probability to be no more than $d = 1 - \alpha$, the number of casts n_Δ must be bounded from below as follows:

$$\sim n_\Delta [P(\omega_{(1)}), P(\omega_{(2)}), d] \geq \frac{P(\omega_{(1)}) \cdot (1 - P(\omega_{(1)})) + P(\omega_{(2)}) \cdot (1 - P(\omega_{(2)}))}{d \cdot \underbrace{(P(\omega_{(1)}) - P(\omega_{(2)}))^2}_{\Delta(\omega_{(1)})}} \quad (6.67)$$

It is straightforward to show that the condition of (6.67) is equivalent to requiring that one standard deviation of the discriminant differential $\delta_{\omega_{(1)}}(X)$ not exceed the value $\sqrt{d} \cdot \Delta(\omega_{(1)})$. Thus, if we want the most likely die face to be evident with probability at least $\alpha = 1 - d = .95$, one standard deviation of $\delta_{\omega_{(1)}}(X)$ must not exceed $.224 \cdot \Delta(\omega_{(1)})$. Equivalently,

$$\sim n_\Delta [P(\omega_{(1)}), P(\omega_{(2)}), d] \geq \underbrace{20}_{\zeta} \frac{P(\omega_{(1)}) \cdot (1 - P(\omega_{(1)})) + P(\omega_{(2)}) \cdot (1 - P(\omega_{(2)}))}{(P(\omega_{(1)}) - P(\omega_{(2)}))^2} \quad (6.68)$$

Through Monte Carlo simulations, we have found that the most likely die face is evident with an empirical probability of at least .95 if $\delta_{\omega_{(1)}}(X)$ does not exceed $\frac{1}{3} \Delta(\omega_{(1)})$. That is, ζ in (6.68) can, in practice, be reduced to 9. Appendix J tabulates $\sim n_\Delta [P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$, given $\zeta = 9$, and compares this bound with empirical values of n_Δ (generated via Monte Carlo simulations) above which the most likely die face is evident in 95% of the trials.

6.4.2 A Greatest Lower Bound on n_P

A greatest lower bound on the sample size n_P necessary to guarantee the more rigorous goal of probabilistic learning is derived in a similar manner. Of course, probabilistic learning implies tighter constraints on the variance of *all* the estimated face probabilities, not just $\hat{P}(\omega_{(1)})$ and $\hat{P}(\omega_{(2)})$. As a result, $\sim n_P$ is substantially larger than $\sim n_\Delta$, reflecting the greater information requirements of probabilistic learning.

Let us consider the probability of a single die face $\omega_{(i)}$ turning up on a given cast, versus the probability of any other die face turning up. As mentioned earlier, the estimated probability of this event $\hat{P}(\omega_{(i)})$ is binomially distributed when considered in this manner, because the C -sided die reduces to an unfair coin with face probabilities $P(\omega_{(i)})$ and $P(\neg\omega_{(i)})$. In this context, we wish to know the upper bound on the probability that the estimate $\hat{P}(\omega_{(i)})$ will deviate from the true probability $P(\omega_{(i)})$ by an amount not less than $\epsilon \cdot P(\omega_{(i)})$. Using Chebyshev's inequality once more, the bound is given by

$$P\left(\left|\hat{P}(\omega_{(i)}) - P(\omega_{(i)})\right| \geq \epsilon \cdot P(\omega_{(i)})\right) \leq \frac{\text{Var}[\hat{P}(\omega_{(i)})]}{(\epsilon \cdot P(\omega_{(i)}))^2} = \frac{1 - P(\omega_{(i)})}{n \cdot \epsilon^2 \cdot P(\omega_{(i)})} \quad (6.69)$$

Thus, the greatest lower bound $\sim n_P$ on the number of die casts necessary to ensure with probability at least $\alpha = 1 - d$ that $\hat{P}(\omega_{(i)})$ does not deviate from $P(\omega_{(i)})$ by more than $\epsilon \cdot P(\omega_{(i)})$ is

$$\sim n_P[P(\omega_{(i)}), \epsilon, d] \geq \frac{1 - P(\omega_{(i)})}{d \cdot \epsilon^2 \cdot P(\omega_{(i)})} \quad (6.70)$$

A comparison of (6.70) and (6.67) shows that the probabilistic bound is more than $\frac{1}{\epsilon^2}$ times the differential bound — i.e.,

$$\sim n_P[P(\omega_{(i)}), \epsilon, d] \gtrsim \frac{1}{\epsilon^2} \cdot n_\Delta[P(\omega_{(1)}), P(\omega_{(2)}), d] \quad (6.71)$$

— unless $\Delta(\omega_{(1)})$ is small in (6.67). Therefore, if we wish $\hat{P}(\omega_{(i)})$ to be within, say, five percent of its true value, the number of die casts necessary to meet this goal will usually be at least 400 times the number of casts needed merely to identify the most likely die face. If $P(\omega_{(i)})$ is appreciably smaller than $P(\omega_{(1)})$ and $P(\omega_{(2)})$, the disparity between $\sim n_P[P(\omega_{(i)}), \epsilon, d]$ and $n_\Delta[P(\omega_{(1)}), P(\omega_{(2)}), d]$ is even greater than indicated by (6.71). Thus, the bounds on n_Δ and n_P in (6.67) and (6.70) quantify the assertions of theorem 6.1 and its corollaries.

By (6.71), the training sample sizes of (6.67), necessary to guarantee a specified level of generalization via differential learning, are typically orders of magnitude smaller than those of (6.70), necessary to estimate probabilities with a specified level of precision. This indicates that current probabilistic extensions of the PAC learning paradigm [133] to stochastic concepts on uncountable feature vector domains (e.g., [59, 60, 146]) are likely to over-estimate the training sample sizes necessary for good generalization when the learning objective is merely pattern classification.

6.5 Extending the Rigged Die Paradigm to the General C -class Learning/Pattern Recognition Task

It is straightforward to extend this chapter's information-theoretic paradigm from a single die to both countable and uncountable feature vector spaces. The extension to the countable feature vector space is quite simple, following immediately from the realization that a single die represents a single point on the countable feature vector space $\mathcal{X}$. Thus, we move from a paradigm in which a single rigged dice game is played to one in which a countably finite or infinite number of games are played. The choice of die to be cast is itself modeled as an unfair die with P sides, corresponding to the cardinality $|\mathcal{X}|$ (where P may be infinitely large). The probabilities associated with each of the P faces reflect the probability mass function of the feature vector $\mathbf{X}$. From this point, the analysis is essentially the same as that for the single dice game.

The extension to the uncountable feature vector space follows along the same lines as that for the countable space. We partition the uncountable feature vector space into P disjoint "resolution cells" $\mathcal{X}_1, \dots, \mathcal{X}_P$

such that

$$\bigcup_{p=1}^P \mathcal{X}_p = \mathcal{X} \quad (6.72)$$

We associate some nominal pattern (i.e., value of the feature vector) X_p with each $\mathcal{X}_p$, and view the *a posteriori* class probability $P_{W|X}(\omega_i | X_p)$ as the expectation

$$P_{W|X}(\omega_i | X_p) = \int_{\mathcal{X}_p} P_{W|X}(\omega_i | X) \cdot \rho_X(X) dX \quad (6.73)$$

Through this artifice, the uncountable space looks just like the countable one, and the analysis follows naturally. As the number of die casts grows large, we simply allow the number of resolution cells to grow until

$$\lim_{\substack{n \rightarrow \infty \\ P \rightarrow \infty}} P_{W|X}(\omega_i | X_p) = P_{W|X}(\omega_i | X), \quad |\mathcal{X}_p| = 0 \quad (6.74)$$

...precisely the mechanism we employ in the derivations for probabilistic and differential learning in chapter 2. Thus, the mean values $E_X [\sim n_P [P(\omega_{(1)}), \epsilon, d]]$, and $E_X [n_\Delta [P(\omega_{(1)}), P(\omega_{(2)}), d]]$, can be derived in order to determine the expected number of examples of X needed to learn the most likely class $\omega_* \equiv \omega_{(1)}$ for each pattern in feature vector space. Likewise, $E_X [M_{qp \min}]$ can be derived in order to determine the average minimum functional complexity necessary for probabilistic learning.

Our objective in describing the procedure by which the die paradigm is extended to the general feature vector space is not so much to do actual modeling or sample complexity computations (see [8] and [146] for lovely, probabilistically motivated sample complexity analyses along these lines) as it is to point out that theorem 6.1 and its corollaries hold for the general feature vector space as well. There is at least one important restriction, however. The information-theoretic analysis of this chapter operates under an agnostic assumption. In terms of dice, the assumption holds that information regarding one die conveys nothing about any other die. In terms of feature vector space, the assumption holds that information regarding the probabilistic nature of the feature vector at one point on $\mathcal{X}$ conveys nothing about the probabilistic nature of the feature vector at any other point on $\mathcal{X}$. Clearly, feature vectors for which a proper parametric model exists violate the agnostic assumption, in that information regarding the probabilistic nature of the feature vector at one point in $\mathcal{X}$ conveys information about the probabilistic nature of the feature vector at *all* points in $\mathcal{X}$. Under these gnostic conditions, the information-theoretic predictions of the sample sizes necessary to characterize the feature vector (either probabilistically or differentially) will be pessimistic (i.e., excessive, perhaps by orders of magnitude). Moreover, corollary 6.2, which asserts that differential learning requires the smallest number of die casts to determine the most likely face, will *not* always generalize

to the statement that differential learning requires the smallest sample size necessary to yield Bayesian discrimination. We remind the reader that the gnostic condition (i.e., the case in which the proper parametric model of the feature vector exists) and its relationship to differential and probabilistic learning strategies are treated extensively in chapters 3 and 4.

The minimum-complexity requirements of differential learning do not depend on the existence of a proper parametric model, but hold in all cases. This trait ensures that learning can be done with the simplest model possible, which in turn ensures that the model will generalize well, independent of the feature vector's probabilistic nature (section 3.5).

6.6 Summary

The rigged game of dice lies at the heart of all statistical pattern recognition tasks. By analysing the requirements for identifying the most likely face of the unfair die, we derive information-theoretic proofs that correspond to the estimation and set-theoretic proofs of chapter 3. Those proofs establish the asymptotically efficiency of differential learning, as well as its minimal classifier complexity requirements. The proof that differential learning requires the fewest casts of the die to identify the most likely die face does not extend to a blanket assertion that differential learning is efficient for both small and large training sample sizes, since probabilistic learning can be more efficient for small training samples when paired with a proper parametric model (recall section 3.6). However, the proof does confirm the assertion that differential learning is efficient for small as well as large training sample sizes when the hypothesis class is an improper parametric model.

Part II

Applications

Chapter 7

Implementing Differential Learning

Outline

We describe the pragmatic issues that arise when one implements differential learning via the CFM objective function. Specifically, we discuss regulation of the CFM confidence parameter, the role of CFM and confidence in accepting or rejecting classifications, issues of representation, and discriminator complexity. We use the Iris data collected by E. Anderson, and subsequently used by R. A. Fisher in his celebrated paper on linear discriminants [34] to illustrate the importance of these issues and to describe practical means of addressing them.

7.1 Introduction

Part I describes the theory of differential learning, but it does not discuss the details of *implementing* the theory. The two bodies of knowledge are linked, but there is a point at which scientific rigor inevitably gives way to practical considerations. This chapter discusses such considerations, and serves as a link between the theory of part I and the applications of that theory in the chapters that follow.

We describe three hypothesis classes that we use in the experiments of this chapter and all that follow. We then describe the Iris data collected by E. Anderson [3] and subsequently used by R. A. Fisher in his 1936 paper on linear discriminants [34]. We show that a linear classifier can learn all but two of the 150 Iris examples. We then use this learning task to illustrate the following practical considerations of differential learning:

- How the confidence parameter ψ affects learning, and how it can be regulated during learning to control the level of detail to be learned from the feature vector X .
- How ψ is practically related to subsequent acceptance or rejection of test example classifications.

- How one's representational choice (i.e., one's *a priori* choice of hypothesis class) affects differential learning.
- The relationship between low classifier complexity and efficient learning.

Throughout this chapter and most of those that follow, we contrast differential learning with probabilistic learning under controlled experimental conditions.

7.2 Three Hypothesis Classes

In this and the remaining chapters, we employ classifiers drawn from three hypothesis classes, corresponding to three functional bases. Different functional bases yield different *representations* or *models* of the data. We describe these hypothesis classes in the following three sections.

7.2.1 The Linear Hypothesis Class

The i th discriminant function of a discriminator belonging to the linear hypothesis class is given by

$$g_i(\mathbf{X}|\theta) = \mathbf{X}'^T \theta_i, \quad (7.1)$$

where the notation $\mathbf{z}^T$ denotes the transpose of vector $\mathbf{z}$, $\mathbf{X}$ is the N -dimensional feature vector, and $\mathbf{X}'$ is the $(N+1)$ -dimensional *augmented* feature vector formed by prepending a single element of unit value to $\mathbf{X}$ (e.g., [29, pp. 136-137]):

$$\mathbf{X}' \triangleq \begin{bmatrix} 1 \\ \mathbf{X} \end{bmatrix} \quad (7.2)$$

The parameter vector for the i th discriminant function is part of the over-all parameter vector for the discriminator: $\theta_i \subset \theta \in \Theta$; $\theta_i \in \mathbb{R}^{N+1}$. We refer to the classifier with such a discriminator as a *linear classifier* because it forms piece-wise linear class boundaries on $\mathcal{X}$.

7.2.2 The Logistic Linear Hypothesis Class

The i th discriminant function of a discriminator belonging to the logistic linear hypothesis class is given by

$$g_i(\mathbf{X}|\theta) = \underbrace{\left[1 + \exp(-\mathbf{X}'^T \theta_i) \right]^{-1}}_{\text{logistic function of } \mathbf{X}'^T \theta_i}, \quad (7.3)$$

where $\mathbf{X}$, $\mathbf{X}'$, and θ_i are as described above. We refer to the classifier with such a discriminator as a *logistic linear classifier* because it employs logistic discriminant functions, yet it forms piece-wise linear

class boundaries on $\mathcal{X}$. This latter characteristic is clear in the solution of the ω_i/ω_j boundary equation, given the i th and j th discriminant functions; the solution is linear in $\mathbf{X}$:

$$g_i(\mathbf{X}|\theta) = g_j(\mathbf{X}|\theta) \quad \text{iff} \quad \mathbf{X}^T [\theta_i - \theta_j] = 0 \quad (7.4)$$

Note that when the discriminator is formed by cascaded layers of logistic functions it constitutes a multi-layer perceptron (e.g., [120]), and the resulting class boundaries it forms on $\mathcal{X}$ are non-linear in $\mathbf{X}$.

The Kullback-Leibler-generated logistic linear classifier: *When generated probabilistically via the CE objective function, the logistic linear classifier is identically the logistic discriminant analysis model (i.e., the logistic regression model used for classification). See appendix F for proofs of this assertion, which originate with White and Hjort.*

7.2.3 The Gaussian Radial Basis Hypothesis Class

The i th discriminant function of a discriminator belonging to the Gaussian Radial Basis hypothesis class is given by

$$g_i(\mathbf{X}|\theta) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Sigma_i|^{\frac{1}{2}}} \cdot \exp \left[-\frac{1}{2} (\mathbf{X} - \mu_i)^T \Sigma_i^{-1} (\mathbf{X} - \mu_i) \right], \quad (7.5)$$

where N denotes the dimensionality of the feature vector, and the i th mean μ_i and covariance matrix Σ_i are subsets of the over-all discriminator parameter vector θ . We refer to the classifier with such a discriminator as an *RBF classifier*, recognizing that it constitutes a Gaussian radial basis function (RBF) classifier (e.g., [18, 95, 104, 92]) when the discriminator is formed by cascaded layers of these functions. We also use a modified form of RBF classifier described in appendix K; we refer to the differentially-generated variants as Differential Radial Basis Function (DRBF) classifiers.

7.3 Learning to Identify the Irises of the Gaspé Peninsula¹

Fisher's Iris data is a well-known database consisting of four physical measurements (petal length and width and sepal length and width) taken from 150 Iris specimens, 50 examples from each of three species. E. Anderson collected the data [3], and R. A. Fisher subsequently used it in his seminal paper on linear discriminants [34]. We, in turn, use the data to illustrate the pragmatic issues of differential learning.

Figure 7.1 shows the Iris data projected onto two of the four dimensions of feature vector space $\mathcal{X}$. The figure is based on figure 6.11 of [29]; the reader will note differences in the data locations between figure 7.1

¹We thank Professor Casimir Kulikowski of Rutgers University for providing us with an electronic version of Anderson/Fisher's original Iris data.

and its progenitor; this is because Duda and Hart depicted multiple examples having the same petal length and width by perturbing the data points in their diagram — a procedure that we omit.

The empirical class distributions have a strong correlation with the histograms Fisher generated from his linear discriminant function (cf. figure 1, [34]). Iris setosa is easily distinguishable from the other two species (we will use the terms “class” and “species” synonymously throughout the remainder of this chapter). Iris versicolor (ω_2) and Iris virginica (ω_3) have empirical distributions that overlap to some degree, as projected onto this two-dimensional sub-space of $\mathcal{X}$. The region of overlap is depicted by the blue-to-red color bar underneath the examples. Boundary $B_{2,3}$ separates the empirical distributions of ω_2 and ω_3 with a relatively small number of errors. The color bar underneath is a means of encoding the position of the superimposed examples relative to boundary $B_{2,3}$ on the 2-dimensional (petal length and petal width) sub-space of $\mathcal{X}$: examples in the dark red region are well into the ω_3 side of the boundary; examples in the dark blue region are well into the ω_2 side of the boundary.

In reality, $B_{2,3}$ is the 1-dimensional *projection* of a hypothetical 3-dimensional boundary in $\mathcal{X} = \mathbb{R}^4$. The color bar is the graphical means by which we transform the two-dimensional sub-space of figure 7.1 into a single dimension perpendicular to the projection of $B_{2,3}$. We encode position along this real dimension by color and intensity: examples superimposed on increasingly red portions of the color bar have increasingly positive values (i.e., they are more to the right of $B_{2,3}$); those on increasingly blue portions of the color bar have increasingly negative values (i.e., they are more to the left of $B_{2,3}$). Figure 7.2 shows all the confusable examples (i.e., all those in the vicinity of $B_{2,3}$ in figure 7.1 and, as a result, superimposed on the color bar) in the disjoint 2-dimensional sub-space of $\mathcal{X}$ comprising the sepal length and width features. The true class of each example in figure 7.2 is indicated by its shape. The petal length and width of each example in figure 7.2 is indicated by the color/intensity of its shape, which denotes the position of the example with respect to the projection of boundary $B_{2,3}$ in figure 7.1.

The projection of boundary $B_{2,3}$ onto sepal length/width space in figure 7.2 obviously depends on the values of petal length and width. After some study of figures 7.1 and 7.2, it should be clear that a linear classifier will produce the fewest errors if the projection of $B_{2,3}$ onto sepal length/width space is $B_{2,3A}$ for values of petal length and width corresponding to the blue region of figure 7.1. As we transition from this region of feature space to the one corresponding to the red region of figure 7.1, the projection of $B_{2,3}$ onto sepal length/width space in figure 7.2 transitions from $B_{2,3A}$ to $B_{2,3B}$. As a first approximation to the full 3-dimensional projection of boundary $B_{2,3}$, we can imagine that boundary $B_{2,3A}$ in figure 7.2 applies to all blue-colored examples (i.e., all those to the left of $B_{2,3}$ in figure 7.1), and boundary $B_{2,3B}$ applies to all red-colored examples (i.e., all those to the right of $B_{2,3}$ in figure 7.1). A linear classifier with such a

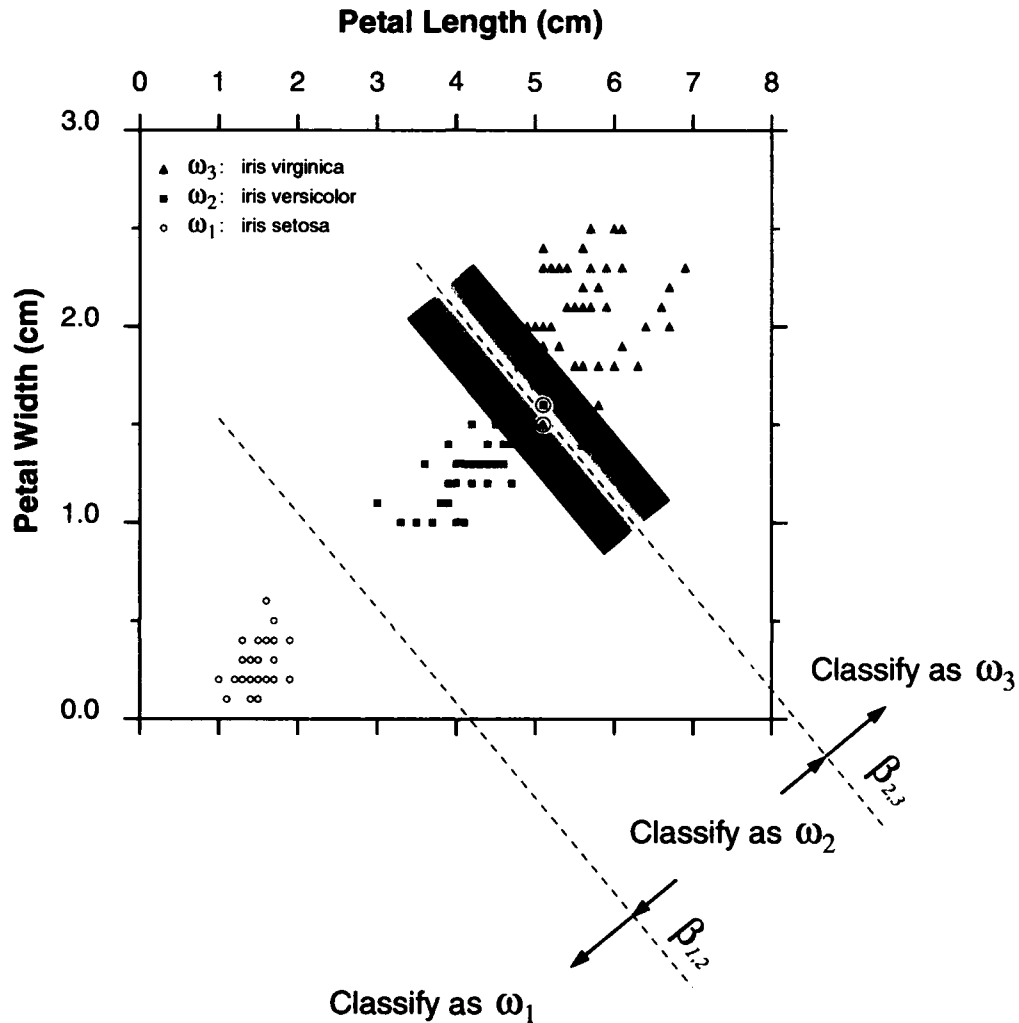


Figure 7.1: Two of the four features (petal length and width) E. Anderson measured on the Irises of the Gaspé Peninsula [3] (see appendix L). This figure is based on figure 6.11 of Duda & Hart [26]. With these two features alone, Iris setosa are clearly distinguishable from the two other species (i.e., classes). Class boundaries $B_{1,2}$ and $B_{2,3}$ separate the classes with relatively few errors. The blue/red color bar denotes the position of the superimposed examples relative to boundary $B_{2,3}$. The color blue denotes the ω_2 side of the boundary; red denotes the ω_3 side of the boundary; the more intense the color, the farther the Euclidean distance a superimposed example is from the boundary. The examples superimposed on the red/blue graphic are the confusable examples of Iris versicolor (ω_2) and Iris virginica (ω_3) because they straddle the optimal linear boundary between these two classes. These confusable examples are shown in figure 7.2. Examples 83 and 133 (circled) cannot be learned by a linear classifier.

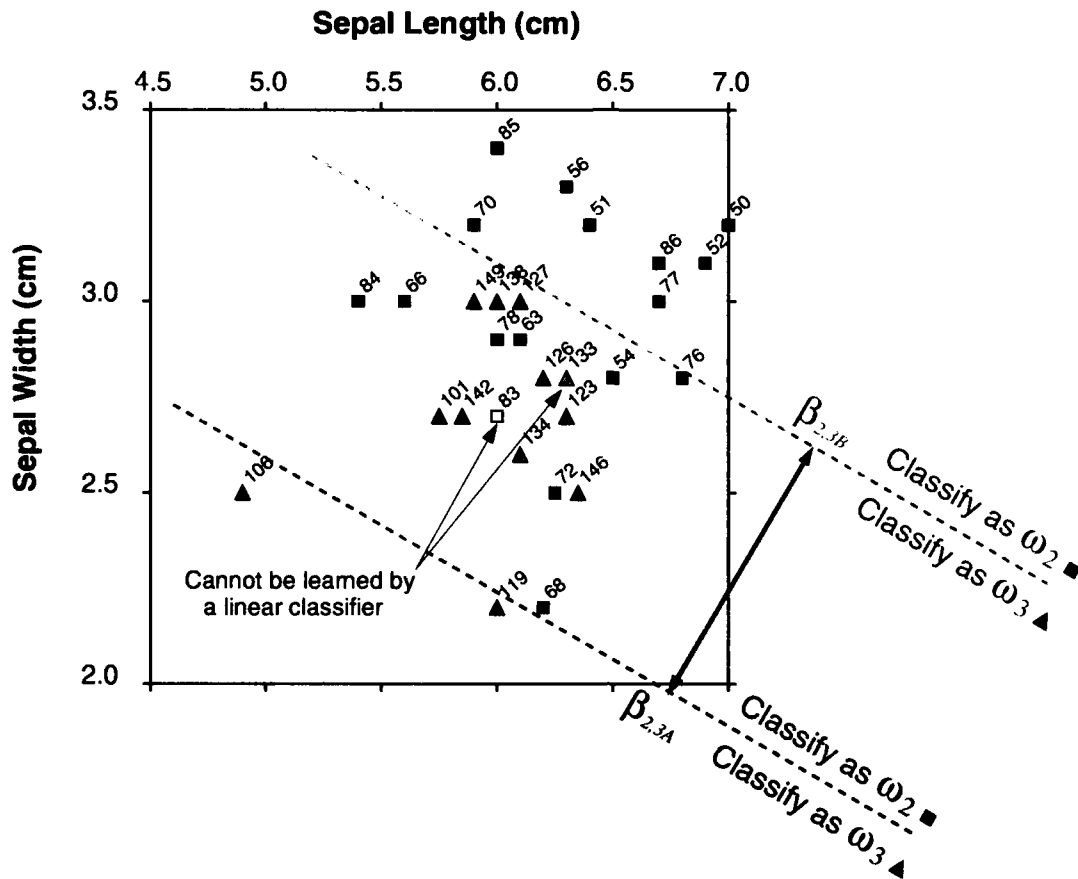


Figure 7.2: The confusable examples of figure 7.1, plotted as a function of the other two features (sepal length and width). Each example's shape denotes its true class. The color and intensity of the shape denote the example's position relative to boundary $B_{2,3}$ in figure 7.1, in accordance with the blue/red color bar of that figure. If the example is blue (i.e., falls to the left of boundary $B_{2,3}$ in figure 7.1), boundary $B_{2,3A}$ applies in this figure. If the example is red (i.e., falls to the right of boundary $B_{2,3}$ in figure 7.1), boundary $B_{2,3B}$ applies in this figure. This linear model correctly classifies all but two of the 150 examples; it cannot learn examples 83 and 133. These two examples are circled in figure 7.1; note that they are the two closest examples to boundary $B_{2,3}$ in that figure.

boundary will misclassify examples 83 and 133,² which lie very close to the projection of $B_{2,3}$ in figure 7.1 (they are the circled examples in that figure).

The reader should note that boundary $B_{2,3}$ and its projections are not unique, but represent a nominal form of the minimum-error linear boundary between Iris classes ω_2 and ω_3 . Thus, we would expect an efficient learning algorithm to produce a linear classifier that correctly identifies all Iris examples except numbers 83 and 133. We illustrate the implementational issues of differential learning by showing that it

²We use the indices $0 \rightarrow 149$ for the 150 examples in the database (see appendix L). Other authors use the indices $1 \rightarrow 150$.

produces just such a linear classifier.

7.4 Controlling the Confidence Parameter

Section D.4 proves that the CFM confidence parameter ψ must be proportional to $P_{W|X}(\omega_* | X)$, the largest *a posteriori* class probability for X , in order for the classifier to learn the Bayes-optimal (i.e., most likely) class ω_* (recall definition 2.1, page 17). Indeed, as $P_{W|X}(\omega_* | X)$ and/or the associated discriminant differential³ $\delta_*(X | \theta^*)$ decrease in the vicinity of the class boundaries, ψ must, by (2.102) and (2.104), be decreased in order to learn ω_* . As stated in sections 5.3.6 and 5.4, the relationship between small values of $P_{W|X}(\omega_* | X)$ and/or $\delta_*(X | \theta^*)$ and small values of ψ accounts for our use of the term “confidence” parameter for ψ . If ψ must be small to learn a training example X^j , then we should literally have low confidence that its associated class label W^j is the Bayes-optimal class ω_* .

In fact, the effect of ψ on learning is not local, as one might infer from sections 2.4 and D.4, rather it is global (i.e., its value for one example affects the learning of all other examples). One’s *a priori* choice of hypothesis class bounds the classifier’s functional complexity, and one can think of the differential learning procedure simply as a means of allocating that complexity in such a way that CFM is maximized. Complexity is allocated proportional to the confidence associated with each training example, so a fixed value of ψ for all training examples determines which examples can be learned and which, if any, cannot be learned, given the hypothesis class. If the training sample size is large, then we are justified in learning all examples — even those in which we have relatively low confidence — since we assume that the sample is representative of the underlying probabilistic nature of X . If on the other hand the training sample is small, we are unwise to learn examples in which we have low confidence, since they may not be representative of X . Instead we would be wise to learn only those examples in which we have relatively high confidence.

The Iris data in figures 7.1 and 7.2 illustrate that training samples usually contain both “easy” examples (i.e., ones that are easily classified) and “hard” examples (i.e., ones that aren’t so easily classified). Recall our definitions of easy and hard examples in section 5.4. Probabilistically, the easy example is found far from the Bayes-optimal class boundaries on $\mathcal{X}$, near a mode of its class-conditional pdf; its *a posteriori* class probability $P_{W|X}(\omega_* | X)$ and the associated discriminant differential $\delta_*(X | \theta^*)$ are therefore large, allowing it to be learned with high confidence ψ . The hard example is found in the vicinity of the class boundaries on $\mathcal{X}$, in a “tail” of its class-conditional pdf. In these tails $P_{W|X}(\omega_* | X)$ and/or $\delta_*(X | \theta^*)$

³Recall from section 2.4 that $\delta_*(X | \theta^*)$ denotes the discriminant differential $g_*(X | \theta^*) - \max_{k \neq *} g_k(X | \theta^*)$, where the subscript $*$ denotes the index of the most likely class ω_* . The notation also indicates that the discriminant differential is the one generated by the discriminator with the CFM-maximizing parameterization θ^* . Throughout the present discussion, we assume that the discriminator possesses sufficient functional complexity to learn the Bayes-optimal classifier of X . As a result, we assume that $\delta_*(X | \theta^*)$ is positive, as long as ψ is sufficiently small.

are relatively small, so the hard example can be learned only with low confidence if it can be learned at all.⁴ The challenge, therefore, is to develop a learning procedure that learns easy examples with high confidence and hard ones with low confidence, allocating the functional complexity of the classifier in commensurate fashion.

Figure 7.3 illustrates the output state of a 15-parameter⁵ logistic linear classifier projected onto reduced discriminator output space (definition 5.2, page 116). This is the reduced output state after the classifier has attempt to learn all 150 examples of the Iris data with $\psi = 1$ (350 epochs). Reduced discriminator output space is shown in the upper right of the figure, and the projection of the CFM objective function onto the reduced discriminant continuum (i.e., the domain of $\delta = y_r - \bar{y}_r$) is shown in the lower left of the figure. The discriminant differentials for most of the training examples are large, corresponding to output states that are nearly binary. Two of the nine examples that are not learned have relatively large discriminant differentials, indicating that the classifier is relatively confident in its incorrect classification. The remaining seven misclassifications engender very small discriminant differentials (these examples appear in the lower left and upper right corner of reduced discriminator output space in figure 7.3). Recall from section 2.4 that the “linear” form of CFM (associated with the highest confidence value of unity) cannot learn examples for which $P_{W|X}(\omega_* | X) \leq \frac{1}{2}$: the discriminant differential that maximizes CFM is zero in these cases. The empirical *a posteriori* class probabilities of ω_2 and ω_3 are approximately $\frac{1}{2}$ in the vicinity of $B_{2,3}$, which accounts for the tiny discriminant differentials exhibited by seven of the nine misclassified examples. Most of the remaining examples are classified with high confidence, as indicated by the large positive discriminant differentials they engender.

Figure 7.4 shows the same classifier after learning with $\psi = 0.6$ (350 epochs). Three examples (70, 83, and 133) are un-learnable at this level of confidence. Note that no examples generate binary output states: the largest discriminant differential is approximately 0.7. Likewise, no learned or transition examples exhibit discriminant differentials less than 0.3, in contrast with figure 7.4. The un-learned examples all exhibit discriminant differentials in the vicinity of -0.4. By reducing the confidence with which the classifier learns, we allow it to allocate its functional complexity in such a way that it learns more of the hard examples.

Figure 7.5 shows the effect of differential learning over 350 epochs when the confidence parameter is gradually decreased from a starting value of 0.6 at epoch 0 to a final value of 0.1 beyond epoch 200. The “gradual” reduction is linear (i.e., ψ is reduced by $\frac{0.6-0.1}{201}$ at the end of each epoch, beginning with epoch 0, and ending with epoch 200). We find that this form of scheduled confidence reduction allows the classifier to learn the easy examples with high confidence and the hard ones with (necessarily) lower confidence. After 350 epochs, only examples 83 and 133 remain un-learned, as our analysis in section 7.3 predicts for an

⁴For stochastic feature vectors with overlapping class-conditional pdfs, the Bayes error rate is non-zero: some example/class label pairs are inevitably un-learnable.

⁵There are $C = 3$ discriminant functions, and the augmented feature vector has $N + 1 = 5$ elements. Therefore the classifier has $3 \cdot 5 = 15$ parameters.

optimally-generated linear classifier. The largest discriminant differential exhibited by a learned example is now approximately 0.68; the minimum discriminant differential is approximately 0.07 for the few learned examples that fall near the reduced discriminant boundary (definition 5.5). Figure 7.6 compares histograms of the classifier outputs corresponding to figures 7.4 and 7.5 respectively. For $\psi = 0.6$, outputs y_1 and y_3 are binary for most examples; output y_2 is approximately normally distributed about a mean of 0.45. When ψ is reduced to 0.1 in the scheduled manner described above, the distribution of y_2 becomes bimodal and outputs y_1 and y_3 become noticeably less binary. These changes are sufficient to learn example 70 (cf. figure 7.4 versus 7.5), one of the three un-learnable examples for $\psi = 0.6$ and, by its proximity to $\mathcal{B}_{2,3}$ and $\mathcal{B}_{2,3B}$ in figures 7.1 and 7.2, one of the hardest examples that a linear classifier can learn.

Differential learning with high confidence focuses on the easy examples because they have a large *a posteriori* probability $P_{W|X}(\omega_* | X)$ associated with the Bayes-optimal class ω_* and they generate a large discriminant differential $\delta_*(X | \theta^*)$ (again, recall the relationships of (2.102) and (2.104)). In turn, $P_{W|X}(\omega_* | X)$ and $\delta_*(X | \theta^*)$ are large where the associated class-conditional pdf $\rho_{X|W}(X | \omega_*)$ peaks — that is, around its mode(s). As learning confidence is reduced, focus shifts from the modes of the training sample's empirical class-conditional distributions (i.e., the easy examples) to the tails (i.e., the hard examples in the vicinity of the class boundaries) — a phenomenon illustrated in figures 7.4 — 7.6. In this sense, one can think of the “outlier” examples of a given class as ones containing the fine details (encoded by X) that distinguish one class from another. By beginning with the easy examples and gradually proceeding to the hard ones, differential learning with scheduled confidence reduction first learns the gross characteristics of each class and then focuses on the details that distinguish each class from all the others.

7.5 Focussing on the Un-Learned Examples

Scheduled reduction of ψ has added benefits

- As $\psi \rightarrow 0^+$ and the synthetic CFM sigmoid becomes steeper, more training examples fall into the “learned” category (i.e., they exhibit positive discriminant differentials that are large enough to generate the maximum CFM value of unity). Since the derivative of the synthetic CFM objective function is zero for learned examples, these examples have no effect on learning.
- Since the synthetic CFM derivative is zero for learned examples, the learning procedure can skip the parameter adjustment phase associated with learned examples. For example, a differentially-generated neural network classifier that uses backpropagation need not backpropagate on the learned examples. As learning proceeds and most examples become learned, this results in substantial computational

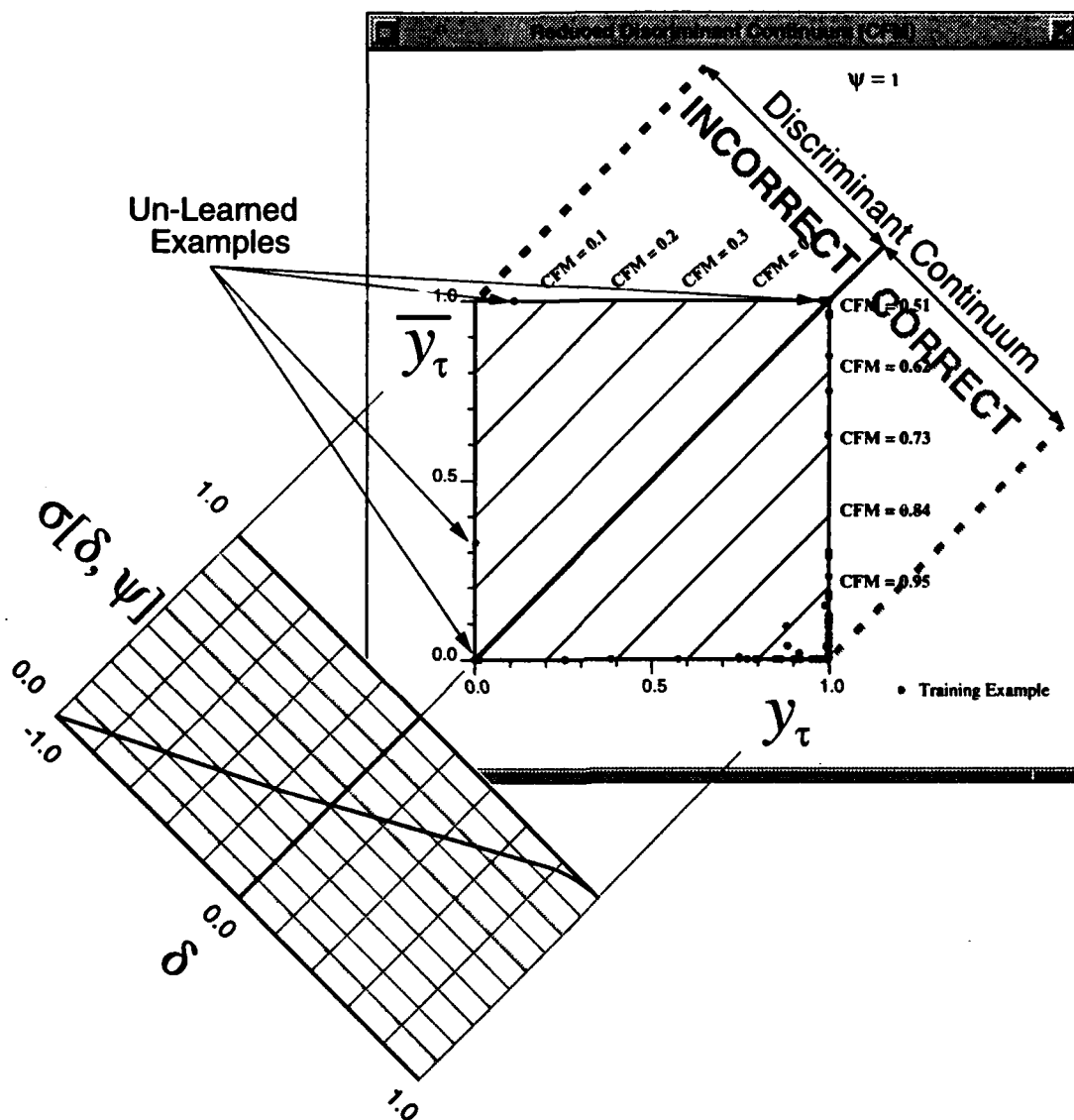


Figure 7.3: The 15-parameter differential logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris data with high confidence. Recall from chapter 5 that y_τ denotes the classifier output corresponding to the correct class for each example, and $y_{\bar{\tau}}$ denotes the largest *other* classifier output. The confidence parameter of 1.0 (set prior to learning) results in a nearly linear form of the CFM objective function (lower left), which tends to engender binary output states in the classifier (most examples appear in the lower right corner of the display). The classifier cannot learn 9 of the 150 examples with this high level of confidence. Seven of these 9 un-learned examples occur at $(y_\tau \cong 0, y_{\bar{\tau}} \cong 0)$ or $(y_\tau \cong 1, y_{\bar{\tau}} \cong 1)$.

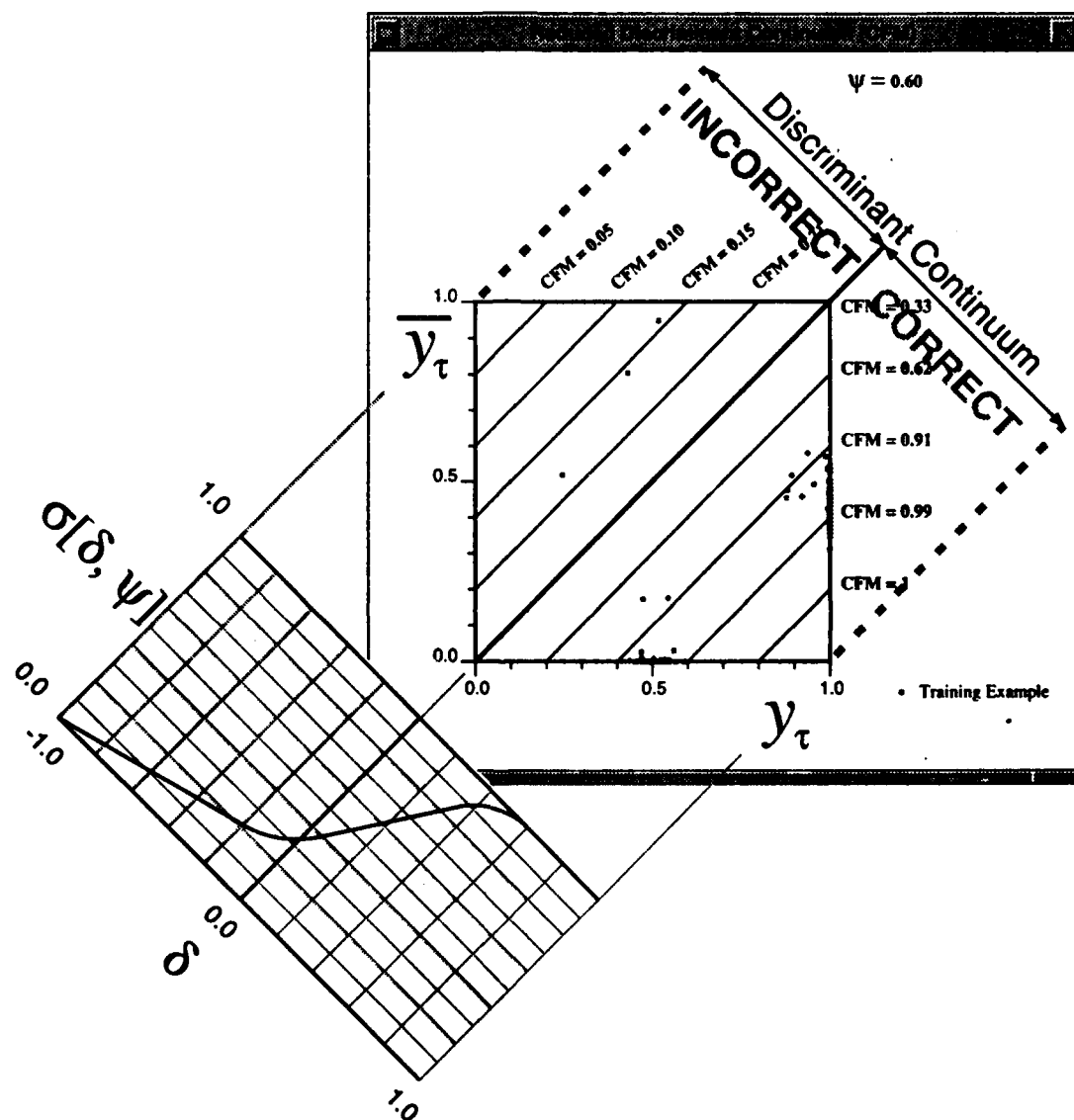


Figure 7.4: The differentially-generated logistic linear classifier's output state after attempting to learn the Iris data with moderate confidence. The confidence parameter of 0.6 allows the classifier to learn all but three of the 150 examples. Note that the output state of the classifier is no longer binary.

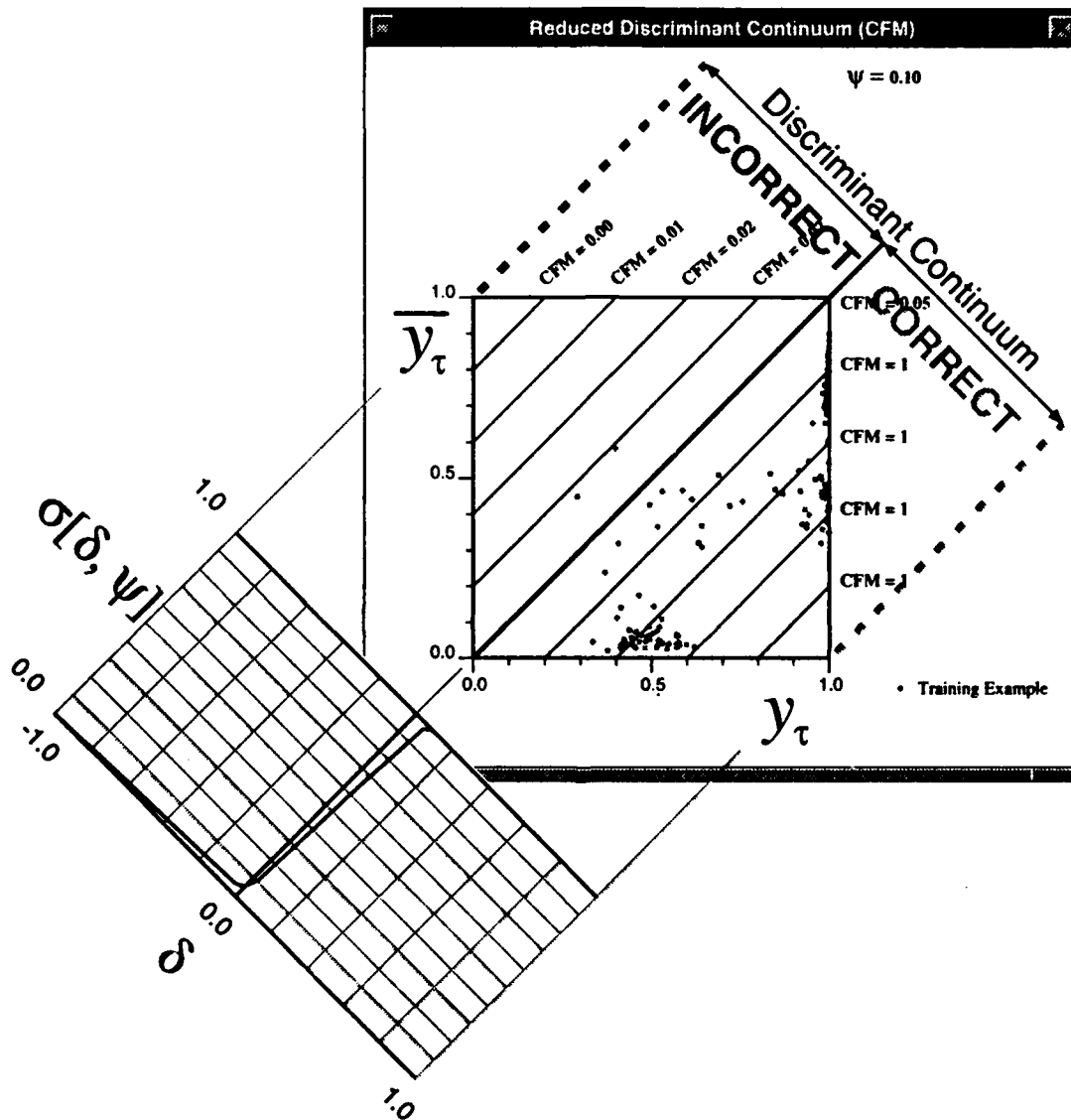


Figure 7.5: The differentially-generated logistic linear classifier's output state after attempting to learn the Iris data with low confidence. The confidence parameter of 0.1 allows the classifier to learn all but two of the 150 examples. Across many independent trials, the two un-learnable examples are consistently 83 and 133: these are the examples shown to be un-learnable by a linear classifier in figure 7.2.

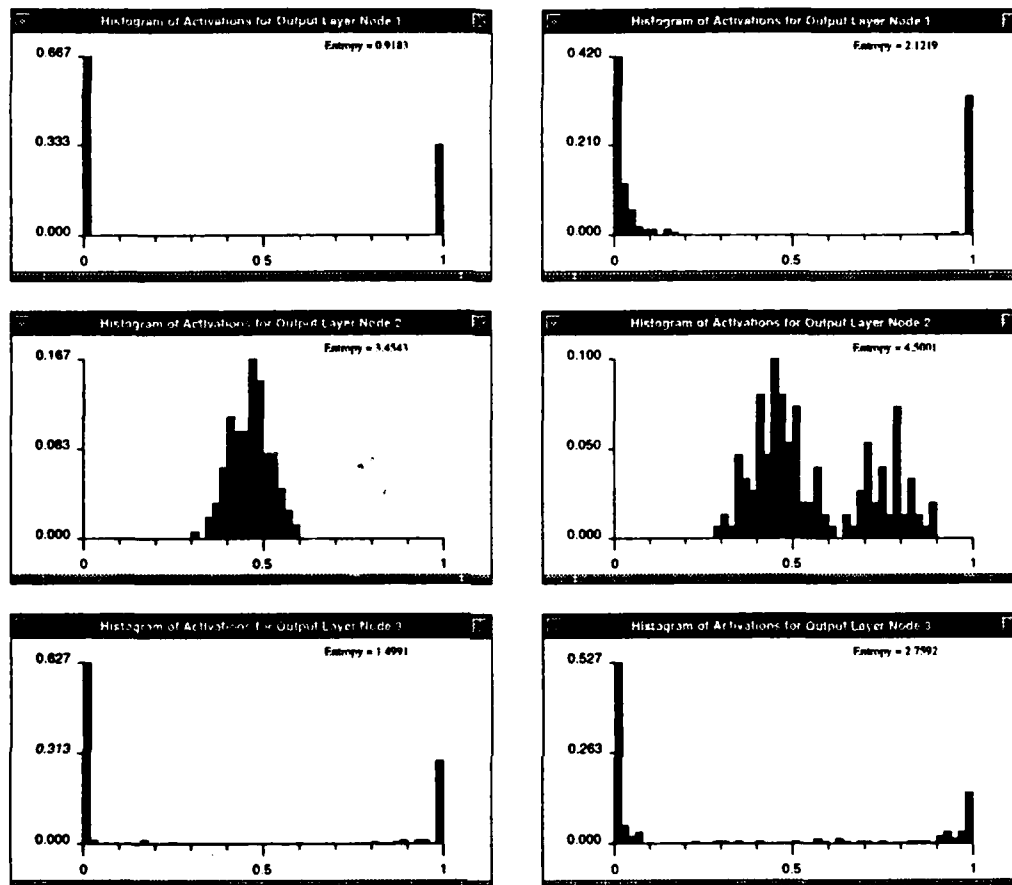


Figure 7.6: **Left:** Histograms of the output states for the classifier in figure 7.4 after 350 learning epochs: $\psi' = 0.6$. These histograms correspond to the reduced discriminator output state in figure 7.4. **Right:** Histograms of the output states for the same classifier (figure 7.5) after 350 learning epochs: ψ' is reduced from 0.6 to 0.1 over the first 200 learning epochs. These histograms correspond to the reduced discriminator output state in figure 7.5.

savings.⁶

In short, *differential learning with scheduled confidence reduction focuses on learning the un-learned examples.*

Figure 7.7 illustrates that the easy learning proceeds relatively quickly, and the hard learning proceeds relatively slowly.⁷ The figure shows the learning curve for the classifier with scheduled confidence reduction,

⁶Haffner et al employ an analogous form of focussed *probabilistic* learning in [48]. They use the mean-squared-error (MSE) objective function and ignore training examples that generate MSE less than a human-specified threshold value. Since there is no monotonic relationship between an example's MSE and whether or not it is correctly classified, the method does not necessarily focus on un-learned examples; rather, it focuses on examples with relatively high MSE. Nevertheless, the motivation for the technique is the computational savings it produces — a motivation that we both acknowledge and share.

⁷Please see section 8.2 for a description of the experimental protocols we employ throughout this text when comparing differentially-generated classifiers with probabilistically-generated controls.

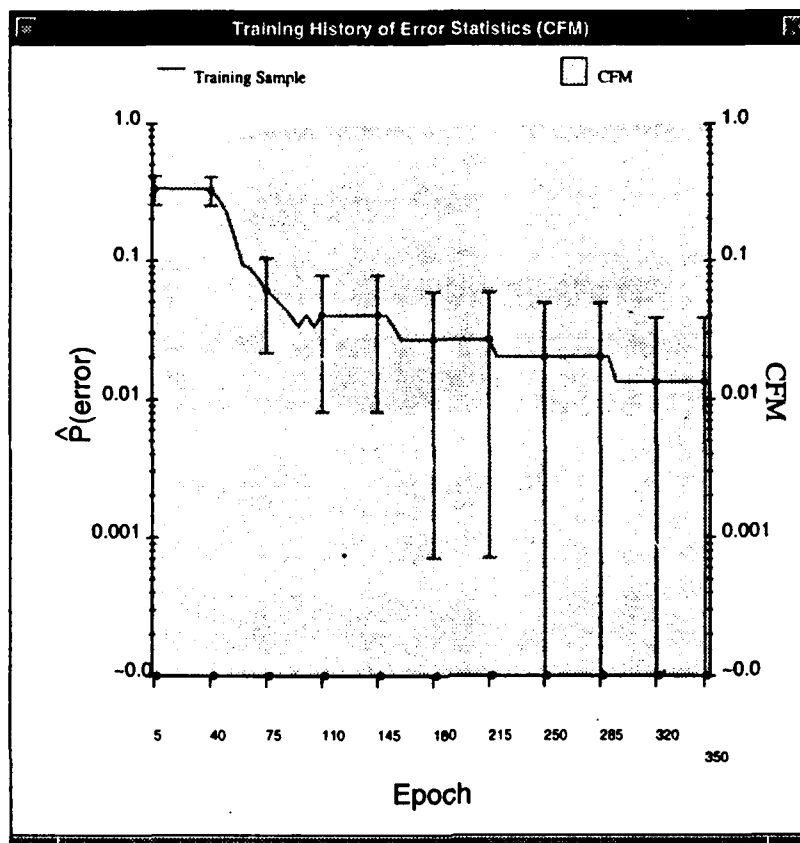


Figure 7.7: The empirical error rates (training sample with all 150 examples) for the 15-parameter logistic linear classifier shown in figure 7.5 as it learns differentially (CFM). The classifier's empirical error rate after 350 learning epochs is 1.3 (+2.5/-1.3)%.

shown in figure 7.5. The dark gray curve shows the classifier's training sample empirical error rate as learning progresses; 95% confidence bounds are superimposed on the curve periodically. The light gray background shows the associated value of CFM as learning proceeds. The classifier learns to distinguish the members of ω_1 from members of the other two classes in fewer than five epochs. By 75 epochs the classifier has learned all but nine of the hard examples; it then requires approximately 220 epochs to learn all but two of those nine hard examples. Owing to the computational savings associated with learned examples, the actual number of computations per epoch decreases proportional to the fraction of learned examples in the training sample. As a result, the last 150 learning epochs (associated with the minimum confidence value of 0.1) proceed more rapidly than the early learning epochs.

Figure 7.8 shows the learning curve for the logistic linear classifier that employs probabilistic learning via the MSE objective function. It learns the easy examples faster, and the hard ones more slowly than its

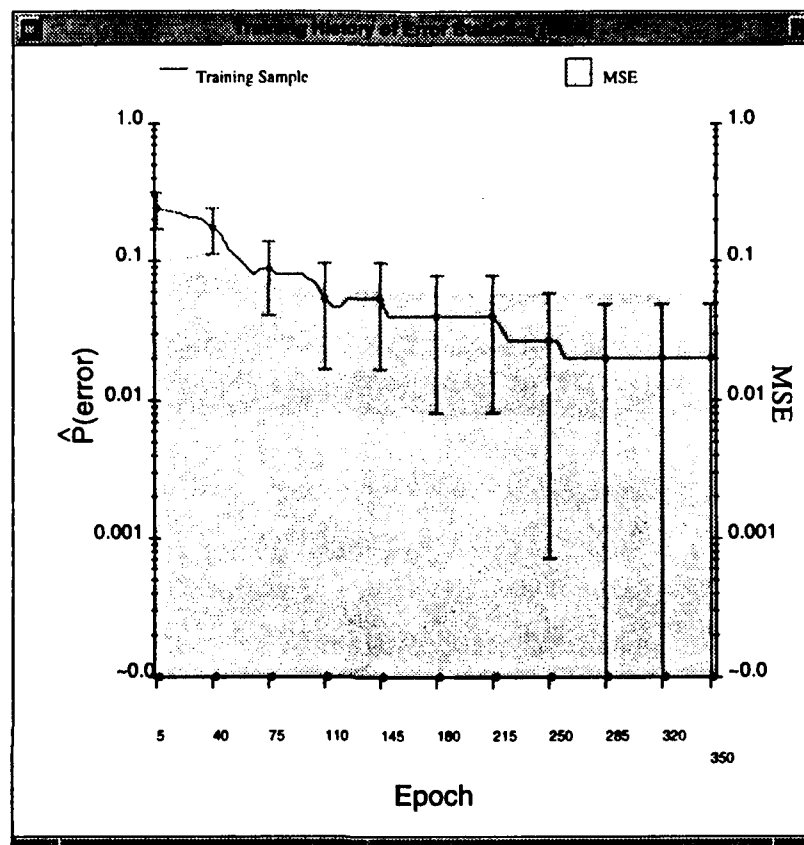


Figure 7.8: The empirical error rates (training sample with all 150 examples) for the 15-parameter logistic linear classifier as it learns probabilistically (MSE). The classifier's empirical error rate after 350 learning epochs is 2.0 (+2.9/-2.0)%.

differentially-generated counterpart — a trend that we find common across a wide range of learning tasks. Figure 7.9 shows the final state of this classifier after 350 learning epochs. All learning conditions are identical to those for the differential model, except that the MSE objective function is minimized in lieu of maximizing the CFM objective function. The MSE-generated classifier cannot learn examples 70, 83, and 133. Note in figure 7.9 that the easy examples (corresponding to the modes of the empirical class-conditional example distributions) dominate the learning procedure: y_T , the output representing the true class of a given training example, is frequently unity, and $\bar{y}_T$, the largest other output, is generally less than 0.5. The harder examples, corresponding to the tails of the empirical class-conditional example distributions, tend to cluster along contours of constant MSE. Many of these examples exhibit zero values of $\bar{y}_T$ (i.e., they appear all along the line $\bar{y}_T = 0$ in figure 7.9). This output state shows that probabilistic learning engenders classifier outputs that approximate the empirical *a posteriori* class probabilities of X to the degree of precision allowed by the

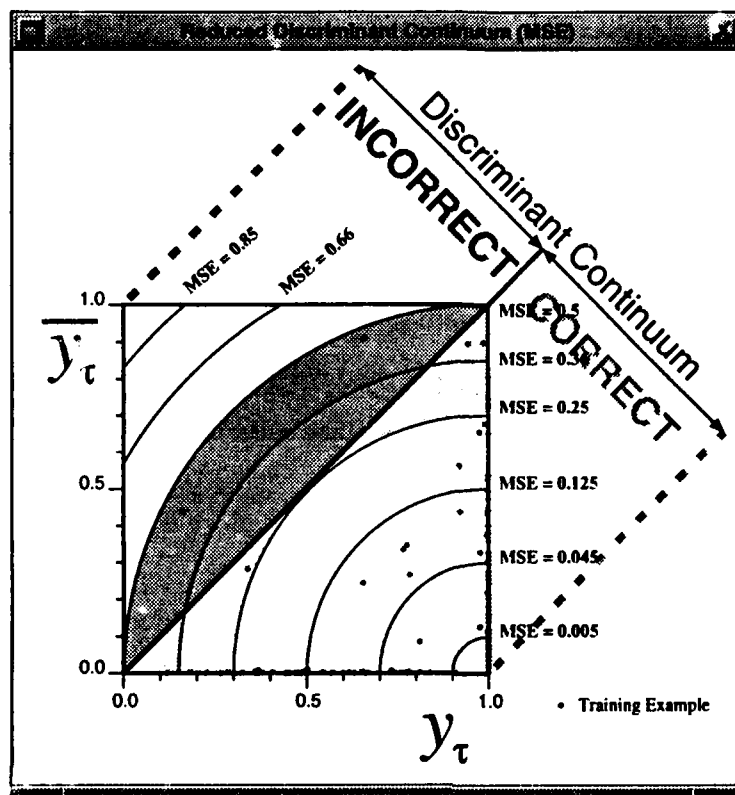


Figure 7.9: The 15-parameter logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (MSE — see figure 7.8).

hypothesis class.

Figure 7.10 shows the learning curve for the logistic linear classifier that employs probabilistic learning via the Kullback-Leibler information distance (CE objective function). Like the MSE-generated classifier, it learns the easy examples faster, and the hard ones more slowly than its differentially-generated counterpart. Figure 7.11 shows the final state of this classifier after 350 learning epochs. All learning conditions are identical to those for the differential model, except that the CE objective function is minimized in lieu of maximizing the CFM objective function. Like the MSE-generated model, the CE-generated classifier cannot learn examples 70, 83, and 133. Note again, the easy examples (corresponding to the modes of the empirical class-conditional example distributions) dominate the learning procedure: y_τ , the output representing the true class of a given training example, is frequently unity, and $\bar{y}_\tau$, the largest other output, is generally less than 0.5. The harder examples, corresponding to the tails of the empirical class-conditional example distributions, tend to cluster along contours of constant CE. Many of these examples exhibit zero values of $\bar{y}_\tau$. Again, this kind of output state reflects the general nature of probabilistic learning.

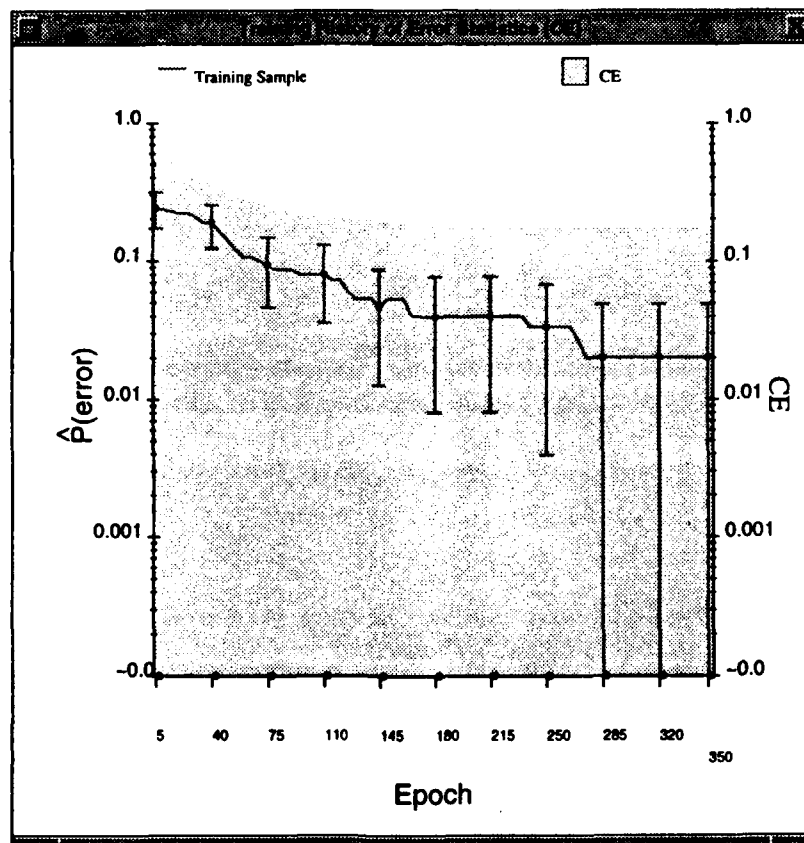


Figure 7.10: The empirical error rates (training sample with all 150 examples) for the 15-parameter logistic linear classifier shown in figure 7.3 as it learns probabilistically (Kullback-Leibler — CE). The classifier's empirical error rate after 350 learning epochs is 2.0 (+2.9/-2.0)%.

In fairness to the probabilistic models, they are not significantly worse than the differential model. Figure 7.1 clearly indicates that the empirical class-conditional example distributions are reasonably well separated and unimodal — conditions for which the logistic linear classifier is a reasonable approximation to a proper parametric model of X (see definition 3.13 and appendix F). As a result, we expect — and obtain — reasonably good discrimination from the probabilistically-generated classifiers. As we shall see in section 7.7, the disparity between differential and probabilistic learning can be significant when the hypothesis class is not a proper parametric model of the data.

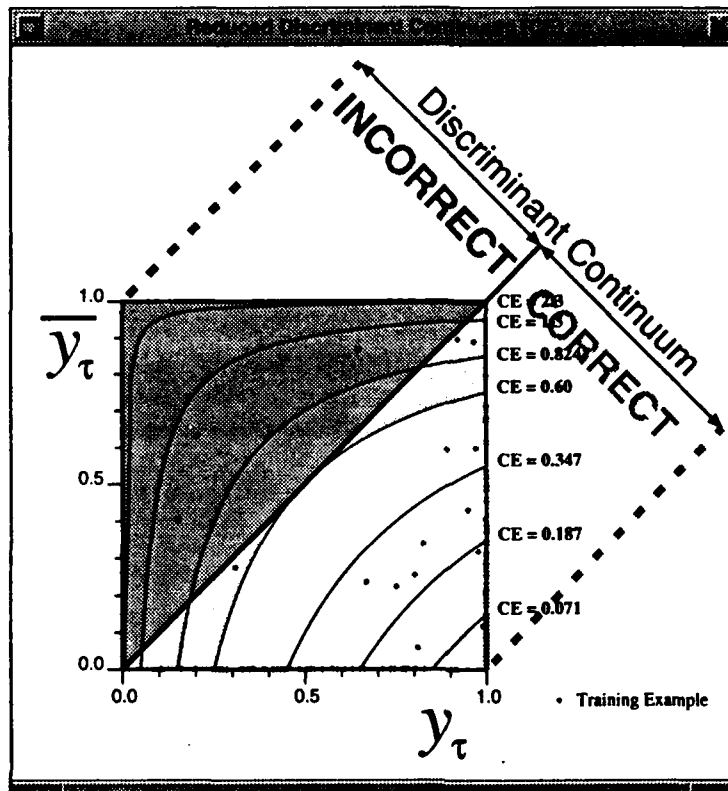


Figure 7.11: The 15-parameter logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (CE — see figure 7.10).

7.6 Rejecting the Classification

Stochastic concepts are sometimes confusable with one another since the class-conditional pdfs of their common feature vector overlap. Just as there are easy and hard learning examples, there are easy and hard test examples. When human subjects cannot identify a concept with high confidence they usually give a “don’t know” classification. Synthetic pattern recognition systems often incorporate an analogous mechanism that rejects test classifications that do not meet a minimum standard of “confidence”. Classical decision theory formalizes the mechanism by which classification hypotheses are rejected. In general the mechanism establishes a *reject region* (e.g., [40, pp. 78-82]) about the class boundaries on $\mathcal{X}$ inside which the classifier always yields a “don’t know” classification.

Since differential learning is a discriminative form of learning, which focuses on estimating the class boundaries on $\mathcal{X}$, it is naturally compatible with the rejection mechanism described above. The reject region on $\mathcal{X}$ maps to reduced discriminator output space in a straightforward manner. Figure 7.12 illustrates this for the differential logistic linear classifier depicted in figure 7.4. The light gray shading on reduced

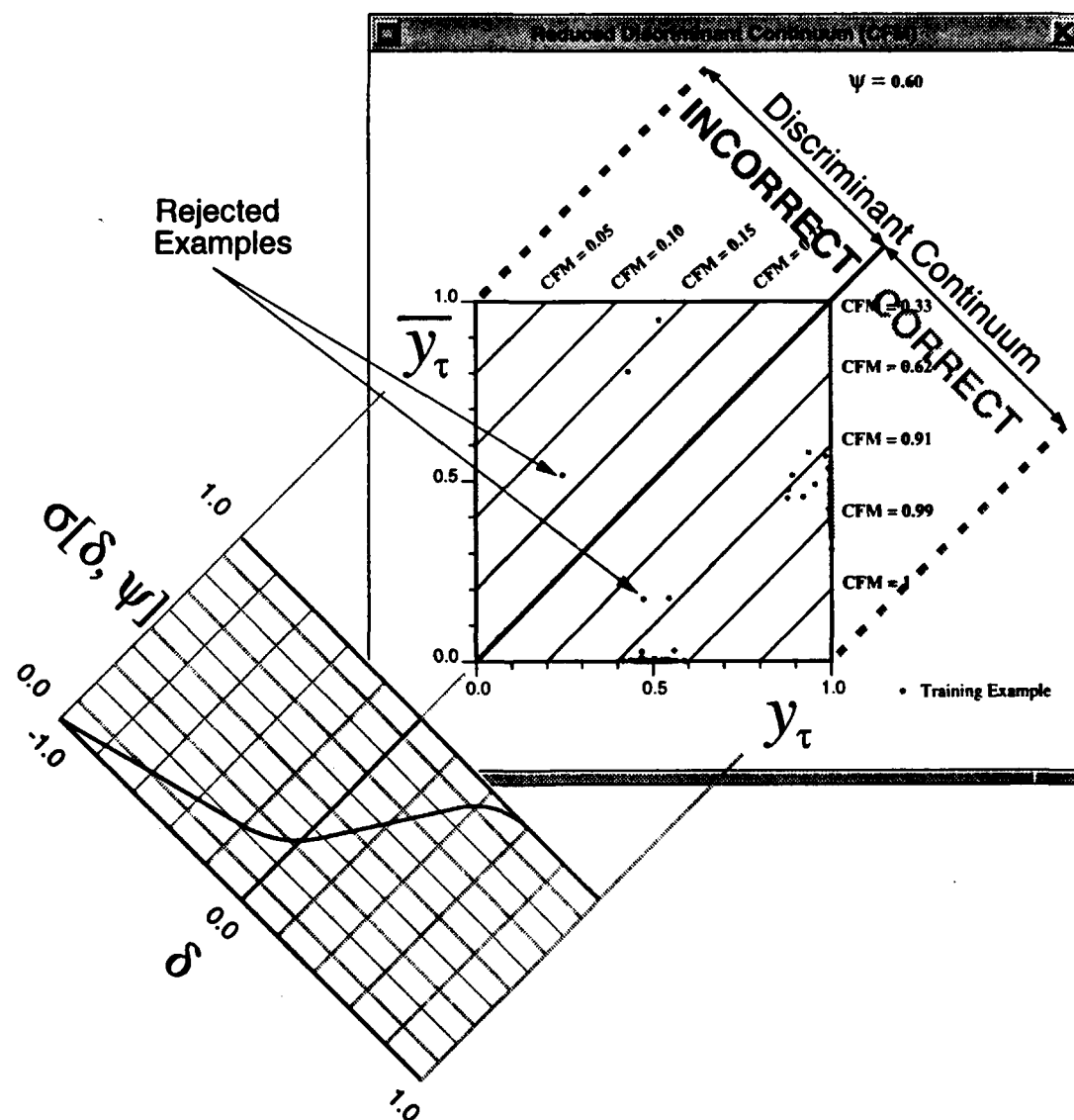


Figure 7.12: Figure 7.4 shown with a rejection threshold of $\delta_{\text{reject}} = 0.35$ (see text) in light gray. For this level of confidence (0.6) and the rejection threshold shown, the classifier rejects 1.3% of the samples and misclassifies 1.3%.

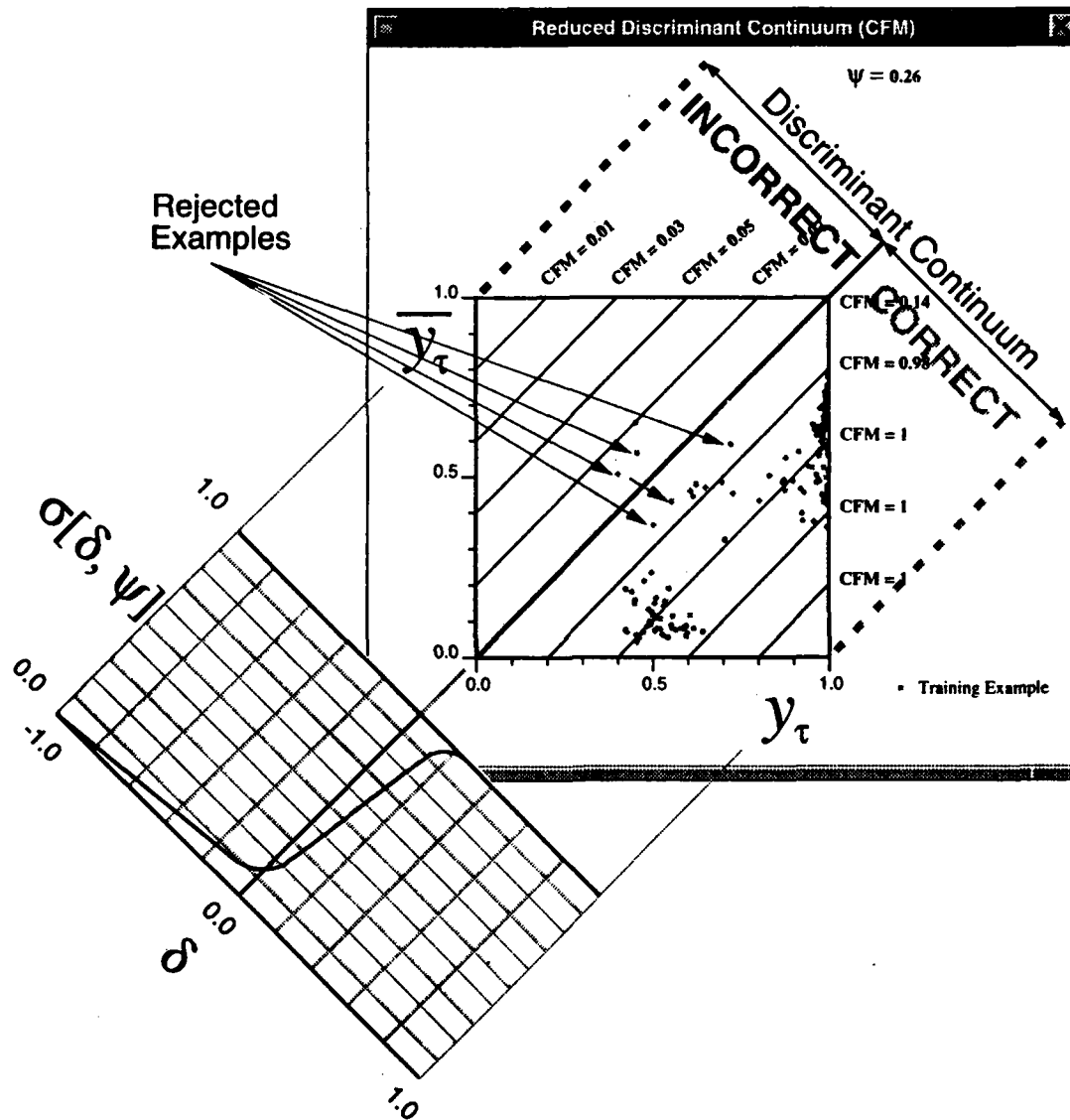


Figure 7.13: The differentially-generated logistic linear classifier's output state after attempting to learn the Iris data with lower confidence (0.26), shown with a rejection threshold of $\delta_{\text{reject}} = 0.15$ (see text) in light gray. For this level of confidence and the rejection threshold shown, the classifier rejects 3.3% of the samples and misclassifies 0.7%.

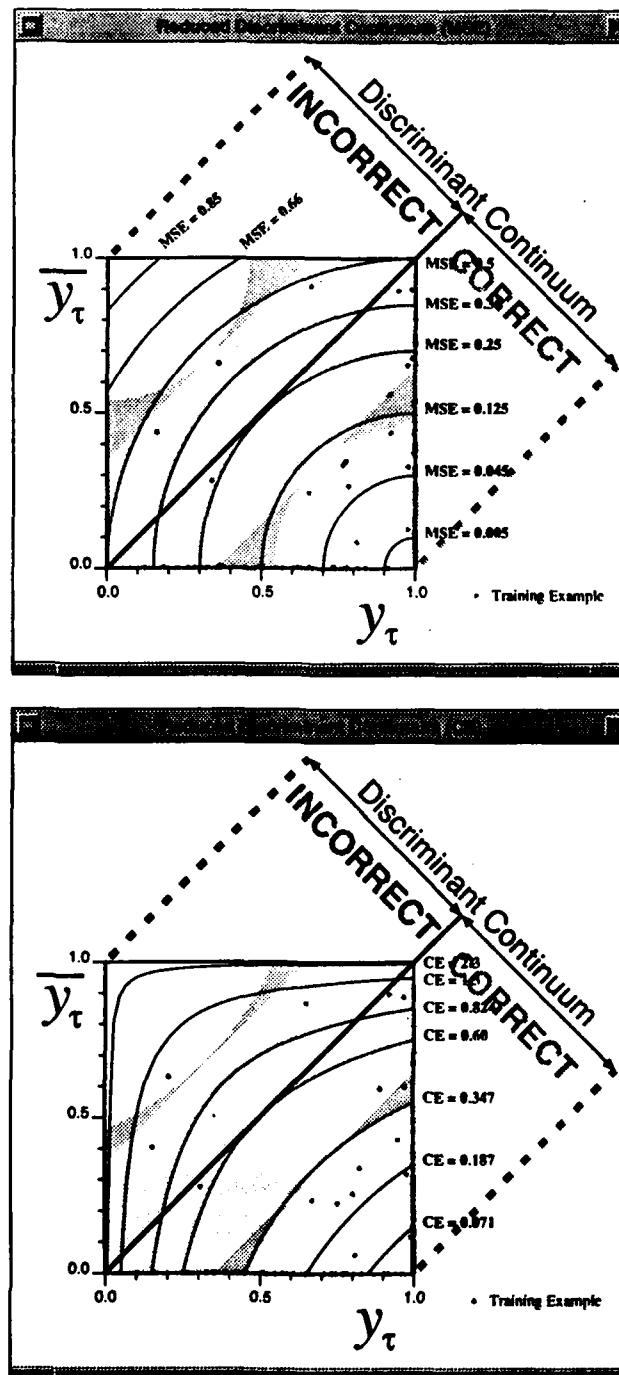


Figure 7.14: Figures 7.9 (MSE, top) and 7.11 (CE, bottom) shown with a rejection threshold of $\delta_{reject} = 0.35$ in light gray. **Top:** The MSE-generated classifier rejects 12.7% and misclassifies 0% of the training sample for the $\delta_{reject} = 0.35$ rejection threshold. The darker gray region corresponds to a (larger) reject region defined in terms of MSE; its rejection/misclassification characteristics are worse. **Bottom:** The CE-generated classifier rejects 14% and misclassifies 0.7% of the training sample for the $\delta_{reject} = 0.35$ rejection threshold. The darker gray region corresponds to a (larger) reject region defined in terms of CE; its rejection/misclassification characteristics are worse.

discriminator output space (upper right) and the reduced discriminant continuum (lower left) denotes the reject region. After learning the training sample, we can set the reject region in such a way that we obtain an acceptable trade-off between error rate and rejection rate (i.e., the percentage of the total sample for which the classification is rejected). The region is specified by a minimum discriminant differential δ_{reject} below which the classification is rejected. Of course, given test cases for which we don't know the true class label, we always assume that the discriminant differential is positive — that is, we always assume the classification is correct. Given the informed perspective from which we know the true class label of each example, the discriminant differential might be negative. Thus, the reject region spans the interval $[-\delta_{reject}, \delta_{reject}]$ in reality. In the experimental chapters that follow, we use a simple default setting for δ_{reject} , based on the value of ψ : δ_{reject} is one half the value of δ at the upper end of the synthetic CFM sigmoid's linear transition leg (i.e., $\frac{1}{2}$ of x_{η} in figure D.1).⁸

Figure 7.12 illustrates a reject region corresponding to a δ_{reject} that is twice the default value. We double the default width of the reject region because the classifier has learned all 150 Iris examples; we set a higher than normal standard of confidence below which we reject the classification for the purpose of illustration. Given this reject region, the classifier rejects 1.3% (2) of the training sample classifications, at the cost of misclassifying 1.3% of the sample. If we decrease the confidence with which we learn to 0.26 (a result not previously shown) and again set a δ_{reject} that is twice the default value, we reject 3.3% (5) of the training sample at the cost of misclassifying 0.7% (1). This scenario is depicted in figure 7.13. Finally, if we apply a δ_{reject} that is twice the default value to the results of figure 7.5, we reject none of the training examples at the cost of misclassifying 1.3% (2).

The tradeoff between error rate and rejection rate in these three scenarios remains both balanced and relatively stable across a wide range of learning confidence parameters, corresponding to a wide range of reject regions. This is not the case for the classifier that employs probabilistic learning. Since the classifier that learns with $\psi = 0.6$ in figures 7.4 and 7.12 exhibits the same empirical error rate as the probabilistically-generated variants in figures 7.9 and 7.11, we apply the reject region shown in figure 7.12 to figures 7.9 and 7.11 as a means of fairly comparing differential and probabilistic learning. Given this reject region, depicted by the light shading in figure 7.14, the MSE-generated classifier rejects 12.7% (19) of the sample at the cost of making no misclassifications; the Kullback-Leibler (CE) variant rejects 14% of the sample at the cost of misclassifying 0.7% (1). Thus, probabilistically-generated classifiers reject a significantly higher proportion of examples without attaining a significantly lower error rate. If the reject region is defined in terms of the MSE or CE that an example (always assumed to be correctly classified) elicits, the reject regions are depicted by the light *and* dark shading in figure 7.14. The resulting rejection/misclassification statistics for

⁸We do not proffer a theoretically justified approach to setting δ_{reject} ; although we believe that this is an important avenue of research, we limit ourselves to this acknowledgement in the interest of bounding the present text's scope.

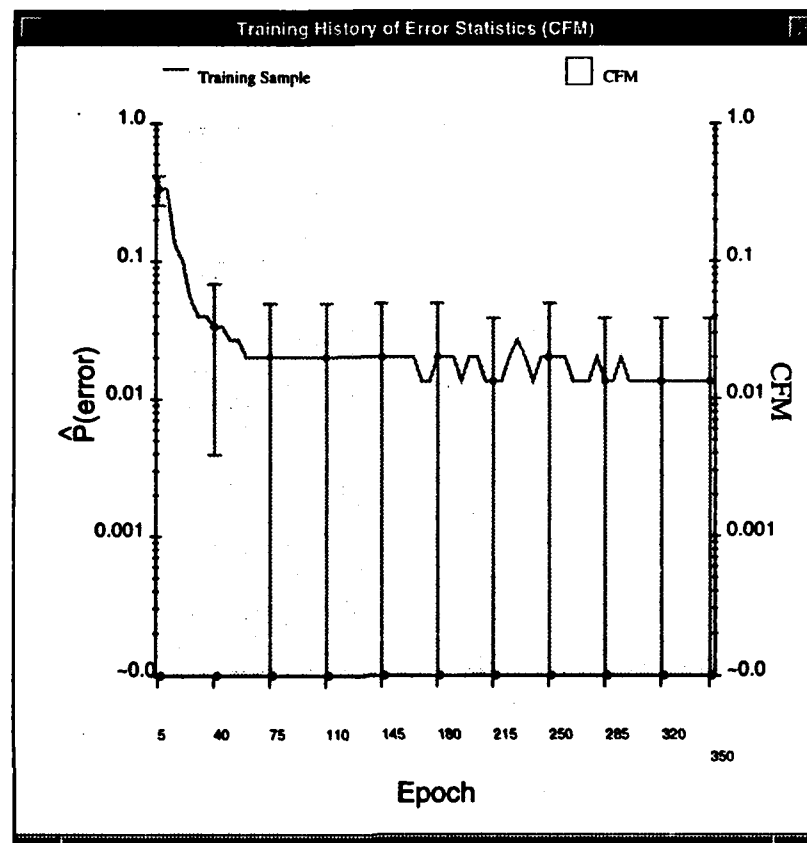


Figure 7.15: The empirical error rates (training sample with all 150 examples) for the 15-parameter linear classifier as it learns differentially (CFM). The classifier's empirical error rate after 350 learning epochs is 1.3 (+2.5/-1.3)%.

these MSE/CE-based reject regions are worse than those for the δ_{reject} -based regions — further evidence that minimizing an error measure is not monotonically related to minimizing the classifier's error rate.

7.7 The Importance of Representational Choices

In section 7.5 we found that differential learning did not produce a logistic linear classifier with a significantly lower empirical error rate than its probabilistically-generated counterparts. We attributed this to the logistic linear hypothesis class, which is a good approximation to the proper parametric model of the Iris feature vector. In this section we explore the effects of changing the hypothesis class from the logistic functional basis to alternative functional bases: linear and Gaussian radial basis.

Figure 7.15 shows the learning curve for a simple linear classifier (i.e. one with discriminant functions

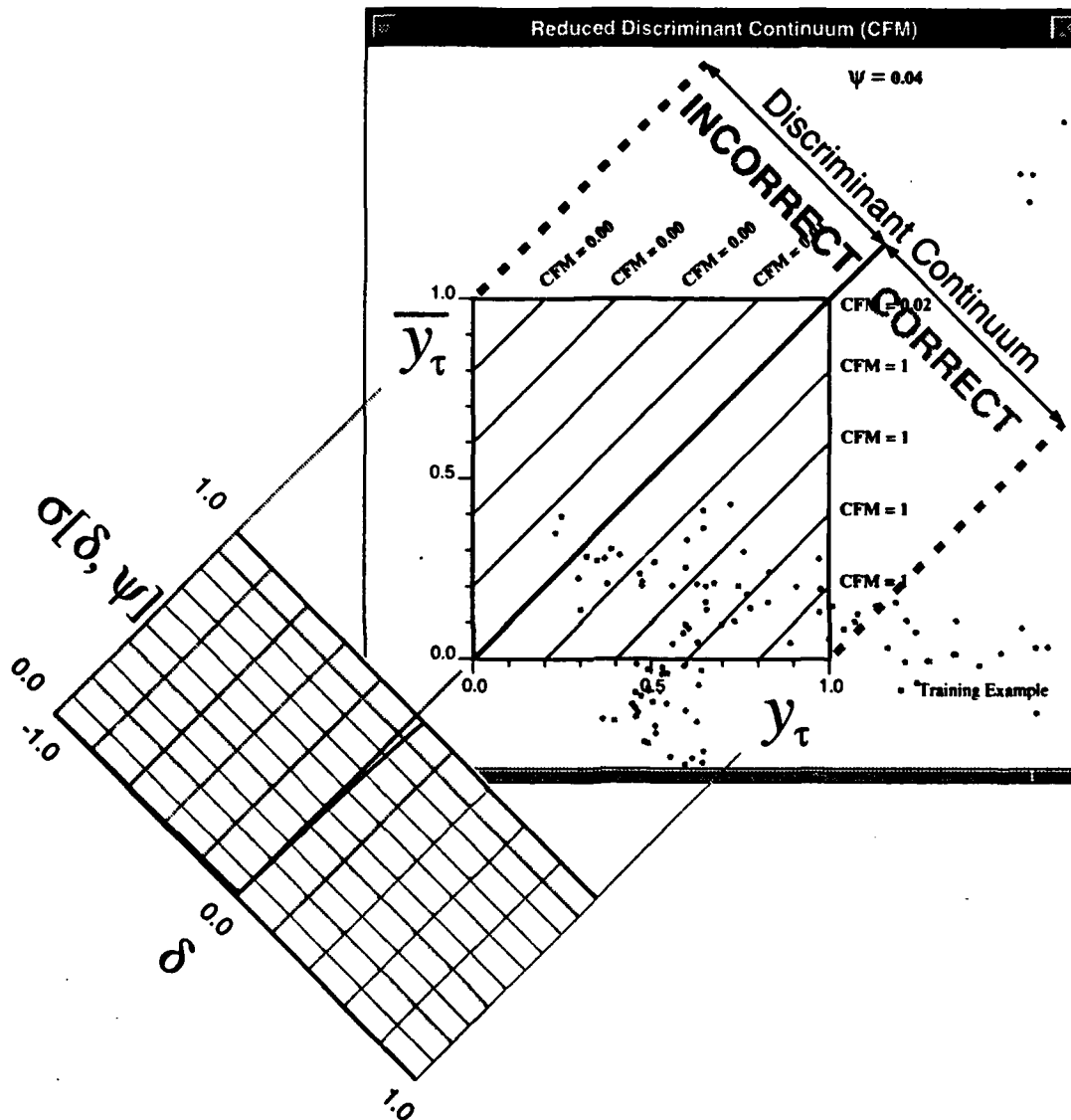


Figure 7.16: The 15-parameter differentially-generated linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris with low confidence. The classifier cannot learn examples 83 and 133 (see figure 7.2).

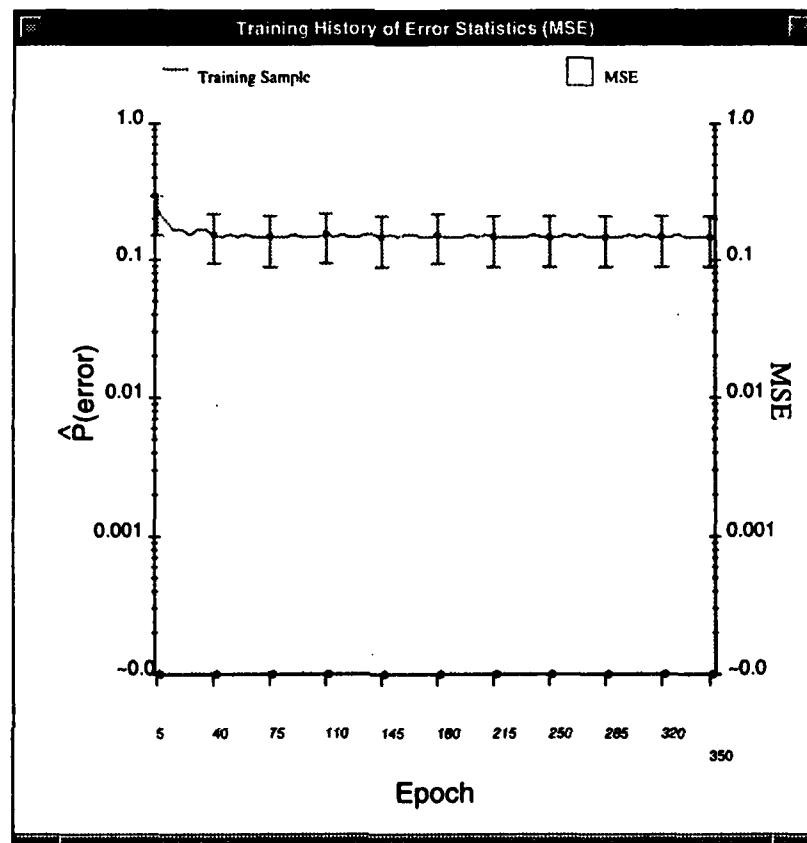


Figure 7.17: The empirical error rates (training sample with all 150 examples) for the 15-parameter linear classifier as it learns probabilistically (MSE). The classifier's empirical error rate after 350 learning epochs is 14.7 (+6.2/-5.8)%.

that are simple linear functions of the feature vector). The classifier employs differential learning with scheduled confidence reduction from 0.6 at epoch zero to 0.04 beyond epoch 200. The only appreciable learning difference between this linear classifier and its logistic linear counterpart shown in figure 7.7 is in their convergence rates. The simple linear model learns the easy examples and most of the hard ones faster than the logistic model. We attribute this phenomenon solely to the change in the classifier's functional basis. The linear model learns at a rate that is approximately linearly proportional to the training sample size (section 5.5.1). The logistic model learns at a rate that is exponentially proportional to the training sample size as its parameters grow large because large parameter values drive the logistic non-linearities towards their limiting step functional form. The proof of this assertion follows directly from section D.3.1, which proves that gradient-based learning via the original logistic sigmoidal form of CFM becomes unreasonably slow as the CFM sigmoid approaches its limiting step functional form. Faster convergence notwithstanding, the

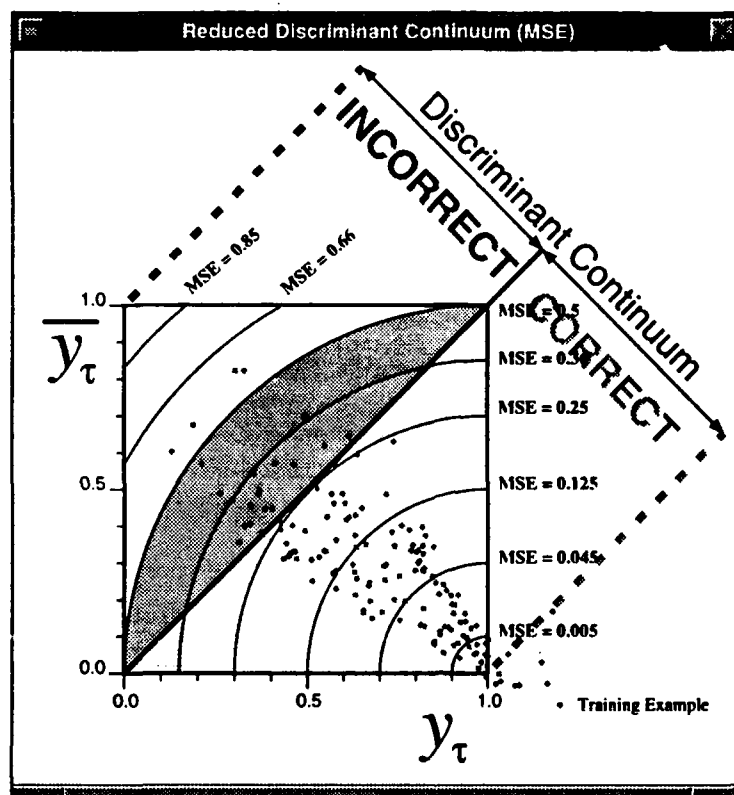


Figure 7.18: The 15-parameter linear classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (MSE). The classifier cannot learn 22 of the examples.

linear and logistic linear classifiers exhibit the same final empirical error rate of 1.3 (+2.5/-1.3)%. Figure 7.16 shows the reduced discriminator output state of the linear classifier after 350 learning epochs. Note that because the classifier is linear in $\mathbf{X}$ its output is on $\mathcal{Y} = \mathbb{R}^3$ rather than $[0,1]^3$. This explains why a number of the training examples appear outside the unit square in the figure. Despite the substantial change in functional basis, this linear classifier exhibits the same learning characteristics as its logistic counterpart: it fails to learn examples 83 and 133.

The same is not true for the simple linear classifier that employs probabilistic learning. Because the linear discriminant functions are a decidedly improper parametric model of the Iris feature vector, they have insufficient functional complexity to approximate the *a posteriori* class probabilities of $\mathbf{X}$. Figure 7.17 shows the probabilistic learning curve for the MSE objective function. The classifier cannot learn 14.7 (+6.2/-5.8)% (22) of the 150 examples, which are clearly visible in figure 7.18. Note that most of these un-learnable training examples fall inside the 0.36 MSE contour. All of these examples are learned with the differential strategy.

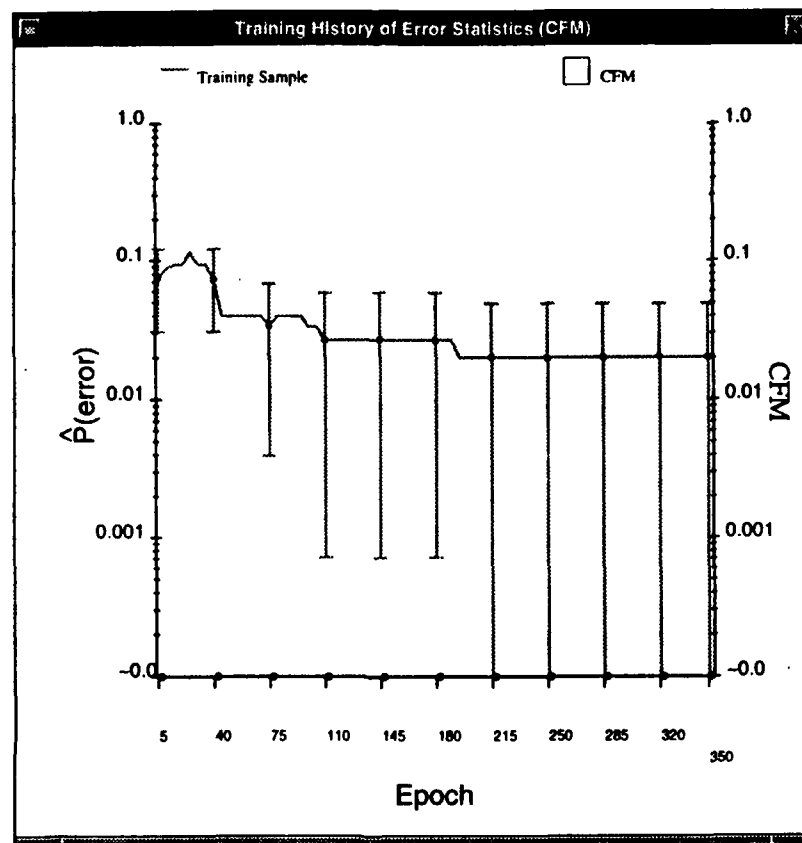


Figure 7.19: The empirical error rates (training sample with all 150 examples) for the 15-parameter modified RBF classifier as it learns differentially (CFM). The classifier's empirical error rate after 350 learning epochs is 2.0 (+2.9/-2.0)%.

Figure 7.19 shows the learning curve for a modified radial basis function (RBF) classifier (see appendix K) that employs differential learning. The classifier has no hidden layer nodes, only three output nodes corresponding to the three discriminant functions. For both differentially and probabilistically-generated variants, the mean vectors of the model $\{\mu_1, \mu_2, \mu_3\}$ are initialized to the empirical class-conditional means of the training sample, and the single variance parameters $\{\sigma_1, \sigma_2, \sigma_3\}$ of (K.3) are set to the average eigenvalue of their corresponding empirical class-conditional covariance matrix. This initialization procedure is not necessary; it simply reduces the learning time for all models.⁹ The differentially-generated model

⁹The critical reader will note that the initialization procedure is fundamentally probabilistic. In this sense the reader might think it logically inconsistent for us to follow such an initialization with differential learning, claiming some advantage over probabilistic learning. Realizing this potential inconsistency, we ran a series of simulations without such initialization. The results for this and other tasks were fundamentally identical to the results with initialization, the only difference being that the initialized models learned much more quickly. In our view, one of the principal weaknesses of radial basis functions is their very local nature, which leads to learning times that increase exponentially as the RBF covariance matrix eigenvalues decrease (i.e., as the RBF nodes become increasingly local).

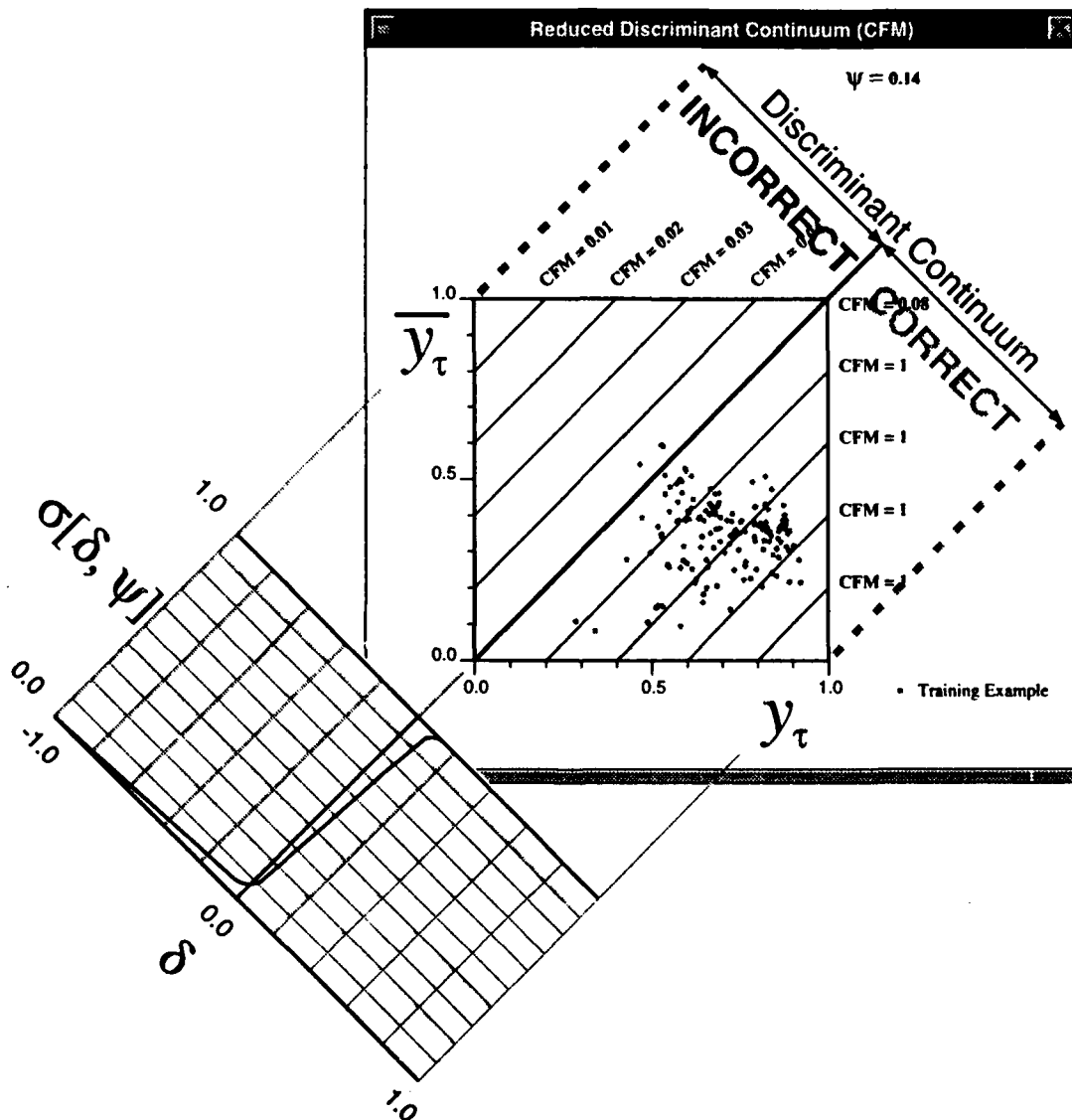


Figure 7.20: The 15-parameter differential modified RBF classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris with low confidence. The classifier cannot learn examples 70, 83, and 133 (see figure 7.2).

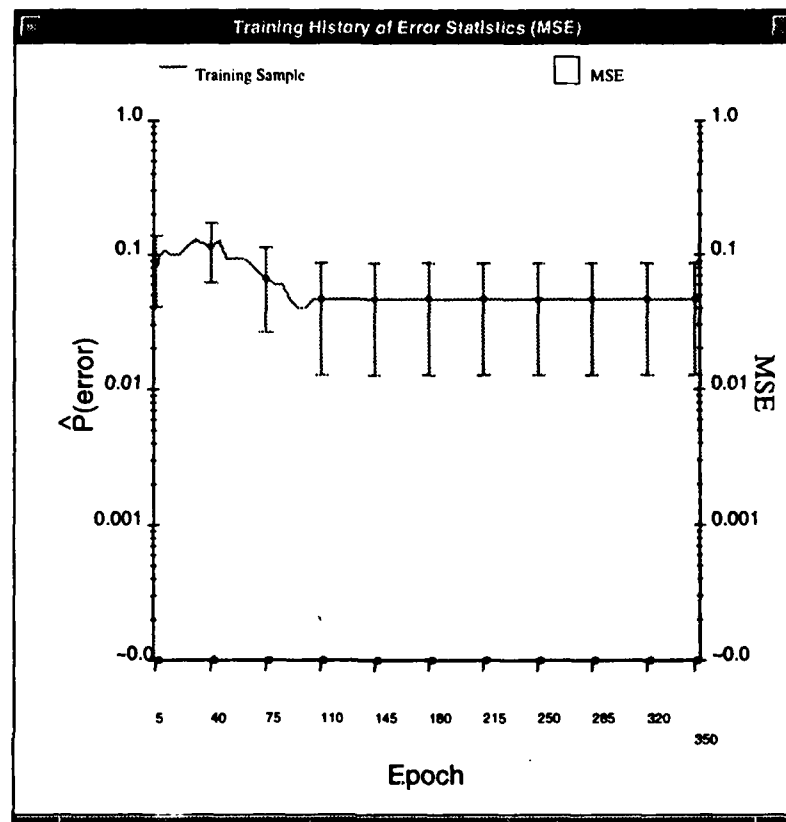


Figure 7.21: The empirical error rates (training sample with all 150 examples) for the 15-parameter modified RBF classifier as it learns probabilistically (MSE). The classifier's empirical error rate after 350 learning epochs is 4.7 (+4.0/-3.4)%.

learns all the examples except 70, 83, and 133 after 350 epochs (confidence is reduced from 0.6 at epoch zero to 0.14 beyond epoch 200). The classifier's final reduced discriminator output state is shown in figure 7.20. Again, this differentially-generated classifier is not substantially worse than its logistic linear counterpart, despite the substantial change in the discriminator's functional basis. These results indicate that differential learning is relatively insensitive to the representational scheme of the hypothesis class (i.e., its functional basis). By the proofs of chapter 3, differential learning will produce the lowest MSDE possible, given the representational scheme. These experiments bear that out.

Figures 7.21 — 7.24 illustrate the learning curves and final reduced discriminator output state for the MSE and Kullback-Leibler probabilistic RBF variants. The MSE-generated classifier exhibits a 4.7 (+4.0/-3.4)%

By placing the radial basis functions in quasi-optimal locations on feature space at the outset, this probabilistic initialization procedure reduces learning times substantially.

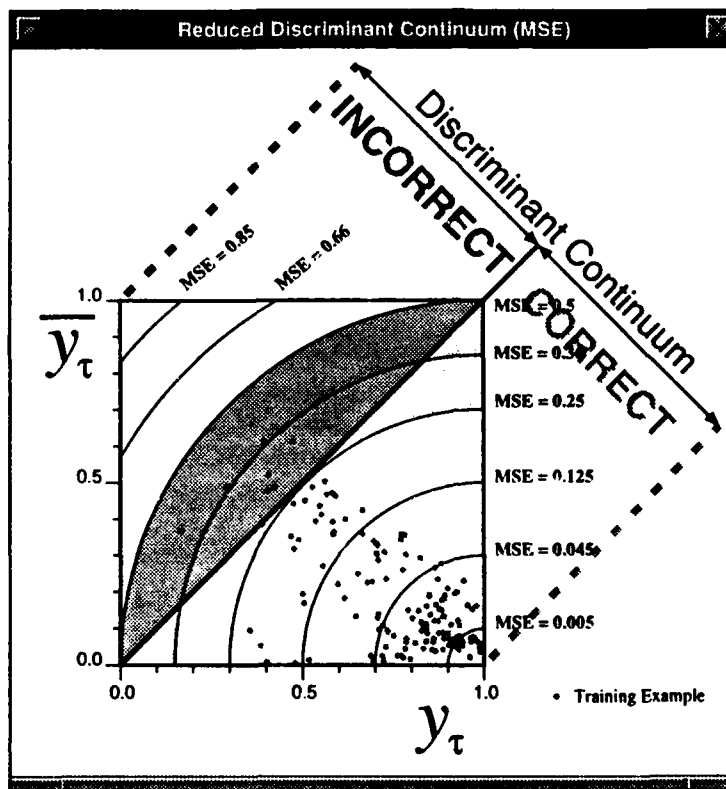


Figure 7.22: The 15-parameter modified RBF classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (MSE). The classifier cannot learn 7 of the examples.

empirical error rate after 350 learning epochs, whereas the Kullback-Leibler-generated classifier exhibits a 6.7 (+4.6/-4.0)% rate after the same number of epochs. Again we see the non-monotonic nature of probabilistic learning when the hypothesis class is not a proper parametric model of the feature vector. Most of the misclassified training examples in figure 7.22 fall inside the 0.36 MSE contour, while most of those in figure 7.24 fall inside the 0.824 CE contour. It is ironic that there are regions on the correct side of reduced discriminant space in which MSE/CE is the same or higher. Clearly, probabilistic learning minimizes the functional error between the discriminator and the training sample without regard to whether an example is learned or un-learned. As a result, the easy examples (i.e., the majority of the training sample) dominate the error minimization, and the hard examples are never learned. When the hypothesis class is a distinctly improper parametric model, as it is in figures 7.19, 7.21, and 7.24, the phenomenon is particularly pronounced; decreasing the functional error paradoxically *increases* the discriminant error.

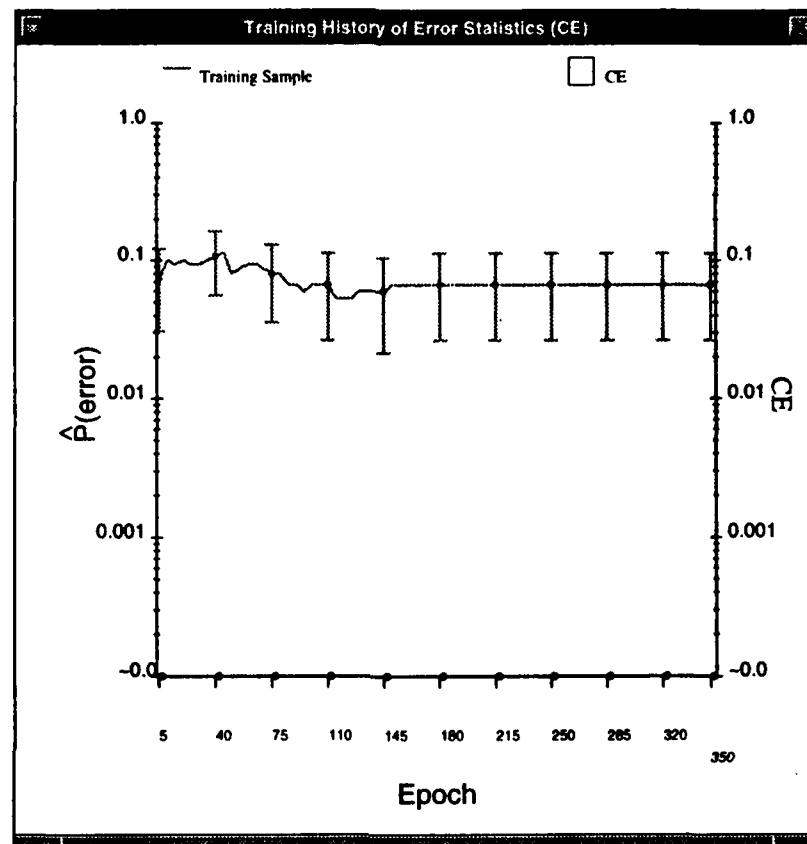


Figure 7.23: The empirical error rates (training sample with all 150 examples) for the 15-parameter modified RBF classifier as it learns probabilistically (Kullback-Leibler — CE). The classifier's empirical error rate after 350 learning epochs is 6.7 (+4.6/-4.0)%.

7.8 Minimizing the Classifier's Complexity

One way to learn all the Iris examples probabilistically is to increase the functional complexity of the hypothesis class enough so that minimizing the resulting classifier's functional error is sure to minimize the empirical error rate for the training sample. If we do this we are likely to cut ourselves on Occam's razor. Specifically, by increasing the classifier's complexity we reduce its discriminant bias *at the cost of increasing its discriminant variance: the net effect for small training sample sizes is an increase in the classifier's mean-squared discriminant error (MSDE)* — a phenomenon we shall see repeatedly in the chapters that follow. The *functional* bias-variance tradeoff is well-known both in detection and estimation theory, and the connectionist literature (e.g., [41]). We remind the reader that we are discussing a very different tradeoff between *discriminant* bias and variance (see chapter 3).

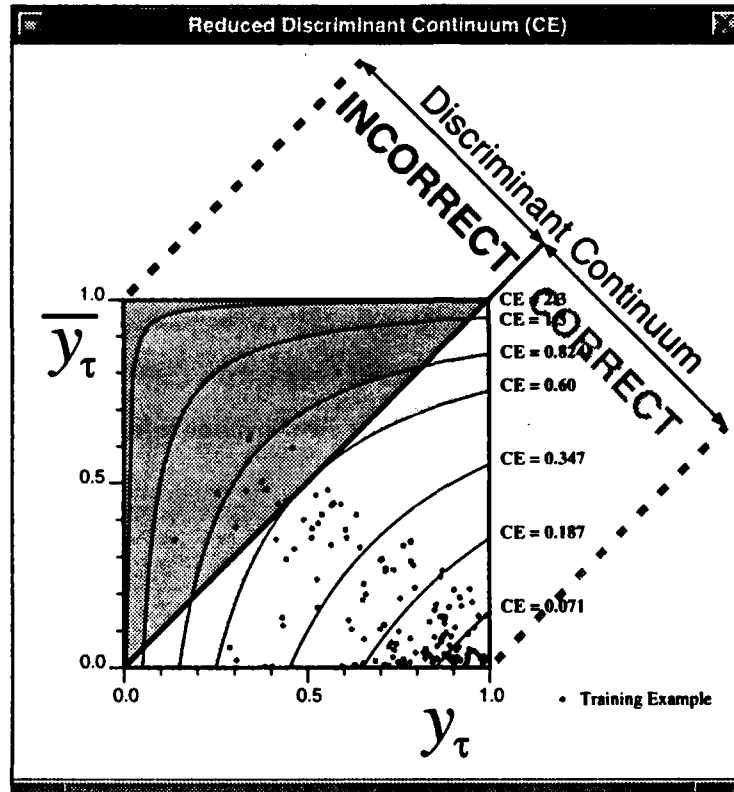


Figure 7.24: The 15-parameter modified RBF classifier's output state — as projected onto the reduced discriminant continuum — after attempting to learn all the Iris probabilistically (Kullback-Leibler — CE). The classifier cannot learn 10 of the examples.

Our Iris experiments illustrate the importance of corollary 3.1:

- because differential learning is asymptotically efficient, it guarantees the lowest discriminant bias possible for a particular choice of hypothesis class.
- because differential learning requires the hypothesis class with the least functional complexity necessary to achieve a specific level of discriminant bias for asymptotically large training samples, it guarantees the least discriminant variance and, as a result, the lowest MSDE possible for small training sample sizes.¹⁰

We shall see in the following chapters that the minimum complexity requirement of differential learning is the key to its producing classifiers with *consistently* lower empirical error rates than those produced by probabilistic learning. We find that many interesting pattern recognition tasks can be learned with simple classifiers, which generalize well to unseen test examples by virtue of their simplicity.

¹⁰The one exception to this guarantee is when the hypothesis class is a minimum-complexity proper parametric model of X , as described in section 3.6 and appendix F.

The 15-parameter logistic linear classifier employing differential learning exhibits a $2.7(+2.2/-2.6)\%$ error rate (4 errors) when it learns and is tested in 150 leave-one-out [84] trials.¹¹ This result suggests that the logistic linear classifier's MSDE is relatively low and that the model generalizes well. The best independent leave-one-out test result for *any* classifier is a statistically equivalent $2.0(+2.9/-2.0)\%$ error rate (3 errors), reported in [139, ch. 6]. The 15-parameter logistic linear classifier employing probabilistic learning via MSE exhibits a $6.0(+4.4/-3.9)\%$ error rate (9 errors) when it learns and is tested in 150 leave-one-out trials. The 15-parameter logistic linear classifier employing probabilistic learning via CE exhibits a $8.0(+5.0/-4.4)\%$ error rate (12 errors) when it learns and is tested in 150 leave-one-out trials.

7.9 Summary

The Iris classification task is an interesting case study because the task is real, not fabricated, the data have been studied extensively by a number of authors, and the classes are not quite linearly separable. As a result, the task is neither trivial nor hard, and it provides a good comparison of differential and probabilistic learning. By visualizing the Iris data in two dimensions, we find that a linear classifier should be able to learn all but two of the 150 examples. The two linear classifiers (ones with linear and logistic functional bases) employing differential learning do indeed learn all but two of the examples. A (non-linear) modified RBF classifier learns all but three of the examples. Comparable probabilistically-generated classifiers cannot learn as many of the examples, illustrating both the inefficiency of probabilistic learning and its sensitivity to the representational scheme.

Because the Iris data constitute a 3-class pattern recognition task with a 4-element feature vector, and because the classes are nearly separable, the learning task is relatively easy. In the chapters that follow, we explore learning tasks that are somewhat harder. Throughout these experiments, we find the results of this chapter repeated: differential learning is efficient, producing the classifier that generalizes best for a given choice of hypothesis class.

¹¹The 95% confidence bounds we give are based on the assumption that each leave-one-out trial is a Bernoulli trial [62].

Chapter 8

Optical Character Recognition with Differential Learning

Outline

We use a linear classifier employing differential learning to recognize handwritten digits of the AT&T "little" (DB1) database.¹ The classifier has 650 total parameters for this optical character recognition (OCR) task. After learning the benchmark training sample, the classifier exhibits a 1.3% error rate on the benchmark test sample. Its probabilistically-generated counterparts exhibit twice this error rate, as does the the best independently-developed linear classifier. A differentially-generated simple Gaussian radial basis function (RBF) classifier achieves a 2.0% error rate on the benchmark test sample — not substantially worse than the linear model, despite the substantial "representational" change (i.e., the change in functional basis). An identical probabilistically-generated RBF exhibits a 10.3% error rate on the benchmark test sample. We use noisy versions of the DB1 database to illustrate the special (and readily discernible) conditions under which differential learning might not produce the best-generalizing classifier for small training sample sizes.

8.1 Introduction

The AT&T DB1 database contains 1200 handwritten digits: ten examples of each digit, obtained from each of twelve different subjects [47]. Figure 8.1 illustrates 40 examples from the database. Each example is a 256-pixel (16×16) binary image (i.e., pixels are either black = -1 or white = +1). The examples are well-defined to the human eye and have uniform scale and orientation. Since its introduction, the database has become a benchmark standard for evaluating learning procedures and neural network architectures in the optical character recognition (OCR) domain. We in turn use the database to illustrate the theoretical arguments of part I.

¹We thank Dr. Isabelle Guyon of AT&T for providing us with the DB1 database. Readers interested in previous research on this database should review [46, 41, 47, 16].

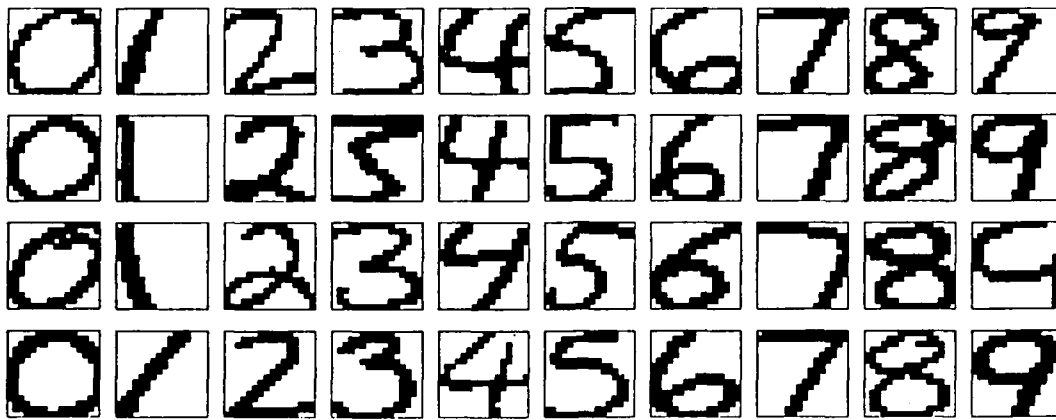


Figure 8.1: Forty digits randomly chosen from the AT&T DB1 database.

We show that compressing the 256-pixel (16×16) binary images to 64-pixel (8×8) 5-state images allows us to employ less complex classifiers, which exhibit lower test sample error rates than those designed for the un-compressed images. Specifically, simple linear and non-linear classifiers with 650 parameters² (65/digit) — one fourth the number of parameters necessary for the 256-pixel images — classify the compressed images with test sample error rates on the order of 2%. We compare these classifiers, which learn differentially, with counterparts that learn probabilistically. The latter exhibit error rates that are between 1.7 and 3.5 times the differentially-generated models' rates, depending on the classifier's functional basis. We conclude by extending the experiments of [41] in which the original 256-pixel binary images are corrupted by noise that takes the form of randomly inverted pixel states. We derive simple signal-to-noise ratio (SNR) expressions for the noisy images. We then use compressed versions of the noisy images to illustrate the special (and readily discernible) circumstances under which differential learning might *not* generate the relatively efficient classifier (definition 3.15 — i.e., the one with the lowest MSDE allowed by the choice of hypothesis class) for *small* training sample sizes.

8.1.1 A Word Regarding Training and Test Samples

Throughout this chapter we refer to a “benchmark split” of the DB1 database. This term refers to the partitioning of the database into a training sample and test sample. Both samples contain 600 examples. The benchmark training sample comprises the first five examples of each digit, obtained from each of the twelve subjects. The benchmark test sample comprises the last five examples of each digit, obtained from each of

²There are $C = 10$ discriminant functions, and the augmented feature vector has $N + 1 = 65$ elements. Therefore the classifier has $10 \cdot 65 = 650$ total parameters.

the twelve subjects. This benchmark split has been used in a number of previous papers on the database; we use it in order to compare our results with previously published ones. We also run multiple trials in which the training examples are selected randomly from the entire 1200-example database with probability $\frac{1}{2}$ (see sections 8.2.1 and 8.2.2). Since these random splits of the database do not guarantee a balanced number of examples of each digit for each subject, the empirical test sample error rates of classifiers generated/tested with them are typically higher than the rate for the benchmark split.

8.2 Test and Evaluation Protocols

Classifier comparisons are strictly controlled: *Throughout this entire text, when we compare classifiers that employ differential learning with those that employ probabilistic learning, all experimental conditions in a given trial are identical except for the objective function used to drive the learning procedure. Learning rates, momentum terms,³ weight decay and or weight smoothing constants (see appendix M), training and test samples, the hypothesis class and its associated parameter space, etc. — all of these factors are identical: only the objective function is different. Furthermore, learning is completely automated after task/classifier setup, so there is no human intervention during the actual learning process. These controls aim at an un-biased comparison of differential and probabilistic learning strategies.*

8.2.1 Estimating Error Rates

All estimated error rates quoted in this text are based on classification results for test samples that have no examples in common with the training sample used for learning.

Definition 8.1 *Estimated error rate:* Given a single test sample of size η , the estimated error rate — which we denote by $\hat{P}_e(\cdot, \eta)$ — for the classifier with discriminator $\mathcal{G}(\mathbf{X}|\theta)$ is simply the ratio of test sample errors $\Xi(\eta)$ to the total number of test examples η :

$$\hat{P}_e(\mathcal{G}|\theta, \eta) \triangleq \frac{\Xi(\eta)}{\eta} = \frac{\text{number of test sample errors}}{\text{test sample size}} \quad (8.1)$$

Remark: We sometimes refer to $\hat{P}_e(\mathcal{G}|\theta, \eta)$ as the classifier's *empirical test sample error rate*. It is valid to view $\hat{P}_e(\mathcal{G}|\theta, \eta)$ as an asymptotically unbiased, maximum-likelihood estimator of the classifier's

³Learning is a search over parameter space for the parameterization that maximizes the objective function, given the differentiable supervised classifier and the training sample. We employ a variant of the simple gradient-based search algorithm with "momentum" typically associated with the backpropagation algorithm (e.g., [119, 120]).

true error rate $P_e(\mathcal{G}|\theta)$. Indeed, $\hat{P}_e(\mathcal{G}|\theta, \eta)$ itself is a binomially-distributed random variable with mean $p = P_e(\mathcal{G}|\theta)$. We assume that $p \cong \hat{P}_e(\mathcal{G}|\theta, \eta)$ in order to compute 95% confidence bounds on $\hat{P}_e(\mathcal{G}|\theta, \eta)$ in the manner described by Highleyman [62].

We judge the error rates of two different classifier/learning strategy combinations, estimated from a single learning/test trial (involving a single training/test sample), to be significantly different if their 95% confidence intervals do not overlap. When the classifier/learning strategies are compared over a series of independent trials (each involving an independent, randomly-selected training/test sample), we consider them to be significantly different if one classifier's empirical test sample error rate is *consistently* lower than the other classifier's.

It is important to clarify the nature of our, "independent, randomly-selected training/test samples." In this chapter we have 1200 total digit examples. Each randomly-selected training sample contains approximately 600 examples; the associated test sample contains all the examples in the original set of 1200 that are *not* in the training sample. Different training/test samples are independent to the extent that they contain different randomly-selected sub-sets of the original 1200-example database; the selection procedures are independent across trials. We denote the size of the k th training sample by n_k and the size of the associated k th test sample by η_k . Thus, our 25 independent learning/test trials in this chapter (and similar trials in chapter 9) constitute 25 repetitions of a 2-fold cross validation procedure (e.g., [139, 91]) by which we estimate the classifier's true error rate $P_e(\mathcal{G}|\theta)$.

Cross validation: *In general, 2-fold cross validation is done by dividing all the labeled examples of the feature vector into a training sample and a test sample of approximately equal size (i.e., $n_k \approx \eta_k$ — a 50/50 partitioning, or "split", of the entire data sample). We use this protocol throughout this text unless otherwise stated.*

Repeated 2-fold cross validation generates an error rate estimator (and, as a result, a discriminant bias estimator) with relatively low bias and variance.⁴ The procedure also allows us to estimate the classifier's discriminant variance and MSDE.

8.2.2 Estimating a Classifier's MSDE

Given K independent, randomly-selected 2-fold cross validation training samples with sizes $\{n_1, \dots, n_K\}$ and associated test samples with sizes $\{\eta_1, \dots, \eta_K\}$, we define the following estimators of the expectations (defined in section 3.2) for the classifier's error rate, discriminant bias, discriminant variance, and mean-squared discriminant error (MSDE). The notation $\Xi(\eta)_k$ denotes the number of misclassifications made on the k th test sample of size η_k .

⁴The reader will find a very readable overview of the extensive literature relating to classifier error rate estimation in [139, sec. 2.5]. Those seeking a more detailed treatment of this material will find it, along with extensive references to the literature, in [91, ch. 10].

Definition 8.2 *Average estimated error rate:* Given the classifier repeatedly generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ over K 2-fold cross validation trials, its average estimated error rate is simply the average of the estimated error rate in (8.1) across all trials:

$$\hat{P}_e(\mathcal{G}|\theta, \{\eta_1, \dots, \eta_K\}) \triangleq \frac{1}{K} \sum_{k=1}^K \hat{P}_e(\mathcal{G}|\theta, \eta_k) = \frac{1}{K} \sum_{k=1}^K \frac{\Xi(\eta)_k}{\eta_k} \quad (8.2)$$

Remark: We sometimes refer to $\hat{P}_e(\mathcal{G}|\theta, \{\eta_1, \dots, \eta_K\})$ as the classifier's *average (empirical) test sample error rate*. Note that (8.2) is an estimate based on the average of K 2-fold cross validation trials, whereas (8.1) is based on a single 2-fold cross validation trial.

Definition 8.3 *Estimated discriminant bias:* Given the classifier repeatedly generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ over K 2-fold cross validation trials, its estimated discriminant bias is its average estimated error rate minus the estimated Bayes error rate $\hat{P}_e(\mathcal{F}_{\text{Bayes}})$:

$$\widehat{\text{DBias}}[\mathcal{G}|\{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda] \triangleq \hat{P}_e(\mathcal{G}|\theta, \{\eta_1, \dots, \eta_K\}) - \hat{P}_e(\mathcal{F}_{\text{Bayes}}) \quad (8.3)$$

Remark: In general we do not know the Bayes error rate for $\mathbf{X}$. The most conservative estimate is that $P_e(\mathcal{F}_{\text{Bayes}}) = 0$ (i.e., the Bayes-optimal classifier can classify examples of $\mathbf{X}$ without error). Note from the definitions of section 3.2 that a classifier's discriminant bias and MSDE are maximized (as a function of $P_e(\mathcal{F}_{\text{Bayes}})$) when $P_e(\mathcal{F}_{\text{Bayes}}) = 0$. Thus, if we assume $P_e(\mathcal{F}_{\text{Bayes}}) = 0$, we are, if anything, over-estimating the classifier's discriminant bias and MSDE. In the case of the DB1 digit recognition task, humans typically recognize all 1200 examples without error, so we assume that the Bayes error rate for this task is indeed zero.

Definition 8.4 *Estimated discriminant variance:* Given the classifier repeatedly generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ over K 2-fold cross validation trials, its estimated discriminant variance is the "sample variance" of its estimated error rate:

$$\widehat{\text{DVar}}[\mathcal{G}|\{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda] \triangleq \frac{1}{K} \sum_{k=1}^K \left(\hat{P}_e(\mathcal{G}|\theta, \eta_k) - \hat{P}_e(\mathcal{G}|\theta, \{\eta_1, \dots, \eta_K\}) \right)^2 \quad (8.4)$$

Definition 8.5 *Estimated mean-squared discriminant error (MSDE):* Given the classifier repeatedly generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ over K 2-fold cross validation trials,

its estimated MSDE is the sum of its estimated discriminant bias squared and its estimated discriminant variance:

$$\widehat{\text{MSDE}} [\mathcal{G} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda] \triangleq \left(\widehat{\text{DBias}} [\mathcal{G} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda] \right)^2 + \widehat{\text{DVar}} [\mathcal{G} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda] \quad (8.5)$$

We use these estimators to assess and compare different classifier/learning strategy combinations across a series of 2-fold cross validation trials. We often use the term “empirical” when referring to our estimates (e.g., the term “empirical MSDE” refers to the estimated MSDE defined above).

8.2.3 Graphical Statistical Summaries

We display single and multi-trial statistics based on the estimators described above using two simple graphics. Both graphics illustrate the set of empirical error rates obtained over a series of 2-fold cross validation trials, and one can be used to characterize the result of a single trial.

The first graphic is the box plot [131, ch. 2], which is described in detail in appendix C. In brief (see, for example, figure 8.3 on page 226), the box of each plot has vertical extrema that match the first and third quartiles of the ranked empirical error rates; the horizontal line dividing the box delineates the median error rate; the inner and (if shown) outer “T”-shaped “fences” of each plot depict the nominal lower bound of the first quartile and nominal upper bound of the fourth quartile, given the ranked empirical error rates. Any extreme first/fourth quartile values falling beyond the outer fence(s) are plotted as dots. The box plot therefore displays the results of *all* trials, emphasizing the median empirical error rate and a quartile partitioning of the results.

The second graphic we use is the familiar whisker plot. In the case of a single trial (see, for example, figure 8.8 on page 231), the dot of the plot delineates the estimated error rate of (8.1), and the upper and lower fences represent the upper and lower bounds of a 95% confidence interval about this estimate. The computation of this confidence interval is described above in section 8.2.1. In the case of multiple trials (see, for example, figure 8.3 on page 226), the dot of the plot delineates the average estimated error rate of (8.2), and the fences represent one upper and one lower standard deviation (derived from (8.4)) about the average estimated error rate.

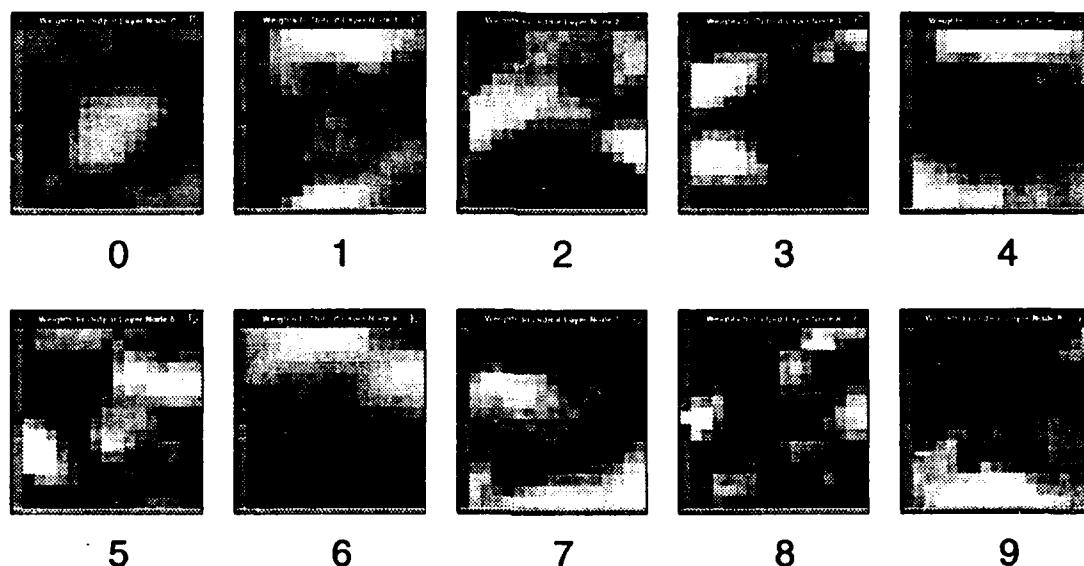


Figure 8.2: Parameters or *weights* of the logistic linear classifier after learning the DB1 database's benchmark training sample differentially. Dark weights are negative and detect dark regions common to training examples of the digit with which they are associated; light weights are positive and detect dark regions common to training examples of *other* digits.

8.3 Compressing the Data to Improve Generalization

Figure 8.2 illustrates the parameters (or *weights*) of a linear classifier with 10 logistic linear discriminant functions of the form described in section 7.2.2. The classifier has 2570 total parameters (257/digit)⁵ and it learns the benchmark training sample differentially. When tested on the benchmark test sample, it exhibits a 2.7 (+1.4/-1.3)% empirical error rate.⁶ Each weight display in figure 8.2 corresponds to the discriminant function for the digit beneath the display. Dark pixels in the display represent negative weights, and light pixels represent positive weights. The far-left column of each display contains only one vertically-centered pixel. This pixel represents the "bias" parameter corresponding to the unit-value element prepended to $\mathbf{X}$ in order to form the augmented feature vector of (7.2). The gray shade of the far-left pixel column represents the value zero (for reference). A dark (negative) weight corresponds to a region that is typically dark (-1) in the training examples of the digit with which the weight's discriminant function is associated. A light (positive) weight corresponds to a region that is typically dark in the training examples of any digit with which the weight's discriminant function is *not* associated. For example, a diagonally-skewed dark image of the digit zero is clearly visible in the weight display for the digit zero discriminant function. Likewise, a dark

⁵There are $C = 10$ discriminant functions, and the augmented feature vector has $N + 1 = 257$ elements. Therefore the classifier has $10 \cdot 257 = 2570$ total parameters.

⁶Unless otherwise noted, error rates are given with 95% confidence intervals. These intervals are computed on the assumption that the empirical test sample error rate is binomially distributed [62].

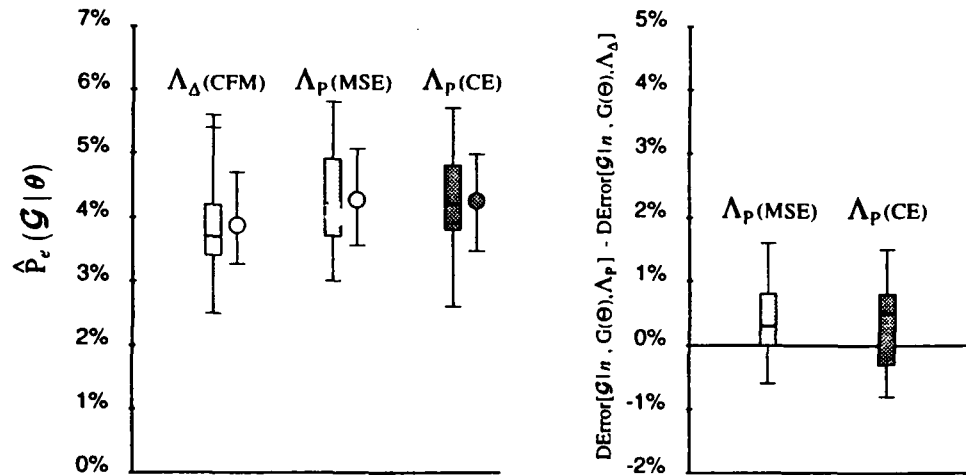


Figure 8.3: **Left:** Test sample classification summaries for the 2570-parameter logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). The summaries are based on 25 independent trials. In each trial, training examples are drawn randomly from the set of 1200 images with probability $\frac{1}{2}$; those not chosen for training form the test sample. The box plots are a non-parametric depiction of the empirical test sample error rate's distribution over the 25 trials; the whisker plots depict the average empirical test sample error rate plus and minus one standard deviation, thereby characterizing each classifier's MSDE. **Right:** The difference between the probabilistically-generated models' empirical error rates and the differentially-generated model's rate on a trial-by-trial basis. These box plots show that differential learning doesn't always produce the classifier with the lowest empirical error rate; this is because the hypothesis class has excessive functional complexity for the task.

image of the digit one is visible at the left edge of the weight display for the digit one discriminant function; however, a light image of the digit three is also clearly visible in the center of this weight display. To a first approximation, the discriminant function for the digit one therefore detects "one and *not* 3" images. Similar characteristics can be found in all the weight displays, although the representations tend to be quite abstract to the human eye.

When generated and tested with 25 different random splits of the DB1 database, the 2570-parameter differentially-generated logistic linear classifier exhibits a median empirical test sample error rate of 3.7%. Probabilistically-generated variants (both MSE and Kullback-Leibler (CE) objective functions) exhibit a slightly higher median rate of 4.2%. Figure 8.3 (left) displays the 25 trial empirical test sample error rate statistics for the three objective functions. The results are shown in box plot [131, ch. 2] (see appendix C) and whisker plot statistical summaries. The right-hand side of figure 8.3 compares the two probabilistic learning strategies with the differential strategy on a trial-by-trial basis. These box plots summarize the difference between the the MSE/CE-generated classifiers' and the CFM-generated classifier's empirical test sample error rates for each of the 25 trials. Positive values indicate that the differentially-generated classifier exhibits a lower empirical test sample error rate than the probabilistically-generated one for the

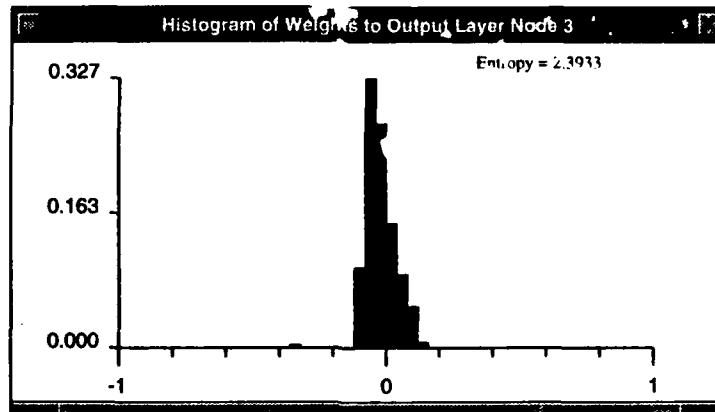


Figure 8.4: The distribution of parameter values in the 257-parameter logistic linear discriminant function representing the digit "3" (cf. figure 8.2). The parametric entropy of these weights is 2.39, corresponding to the relatively low variance in the distribution. The parametric entropy of all weights in the 2570-parameter model is 2.30.

trial; negative values indicate that the differentially-generated classifier exhibits a higher empirical test sample error rate than the probabilistically-generated one for the trial. From figure 8.3 (left) we see that the differentially-generated model's discriminant bias, indicated by its average empirical test sample error rate, is slightly lower than the probabilistically-generated models'. This is also evident in the trial-by-trial statistics on the right side of the figure: in 3/4 of the trials, the differential model exhibits a lower error rate than its MSE-generated counterpart; in 2/3 of the trials, the differential model exhibits a lower error rate than its CE-generated counterpart. The discriminant variance of the classifiers produced by differential learning and both probabilistic learning procedures is indicated by the vertical span of their respective whisker plots. Figure 8.3 (left) indicates that the differentially-generated model's discriminant variance is approximately the same as the probabilistically-generated models'.

Figure 8.3 illustrates that the differentially-generated model's empirical MSDE (as indicated by the whisker plot) is not significantly lower than the probabilistically-generated models'. This is because the hypothesis class (i.e., the 2570-parameter logistic linear discriminator) has excess functional complexity for the task. Figure 8.2 helps to explain why this is so. The weights of the figure are blurred looking because the classifier employs weight smoothing (described in section M.2) during learning. This is done in order to minimize the *parametric entropy* (definition M.1) of the classifier's weight vector, an empirical measure that we use to gauge the weight vector's information content. Weight smoothing therefore reduces the classifier's discriminant variance, since only the information essential to learning is retained. Figure 8.4 shows a histogram of the weights in the discriminant function for the digit "3" (again, the weights themselves are shown in figure 8.2). Because the classifier learns all of the 600 training examples with a large amount

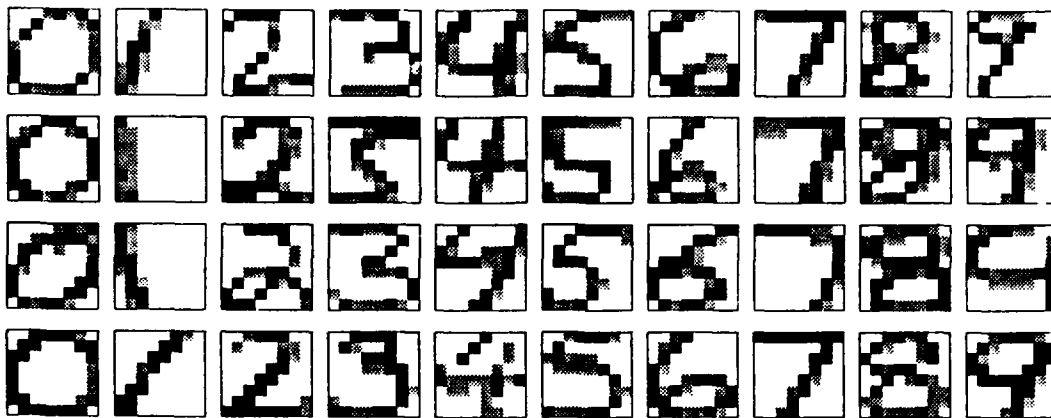


Figure 8.5: The same digits shown in figure 8.1, linearly compressed from 256- to 64-pixel images.

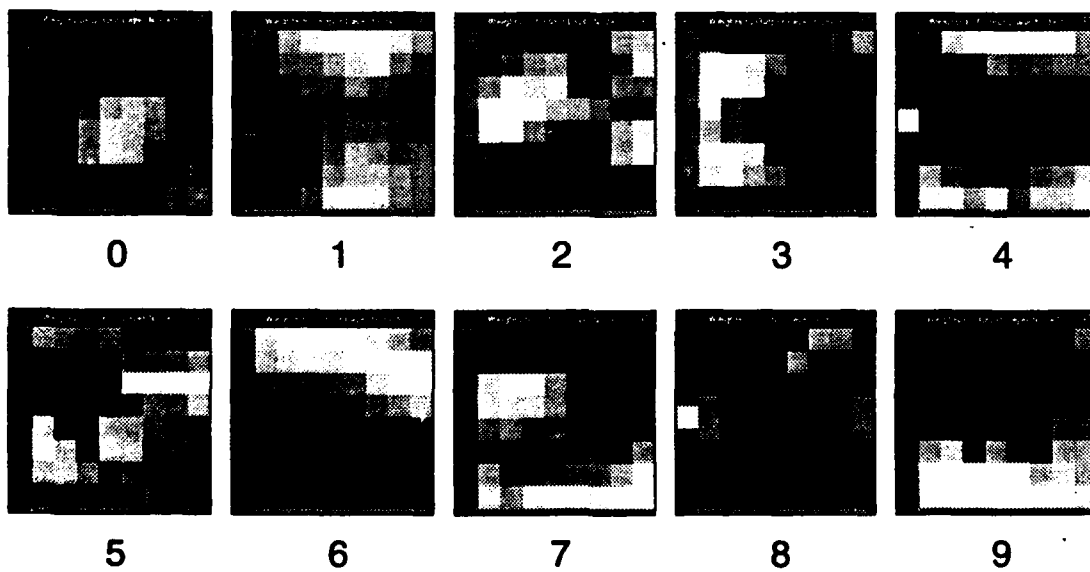


Figure 8.6: Parameters or *weights* of the 650-parameter logistic linear classifier after learning the DB1 database's benchmark training sample differentially.

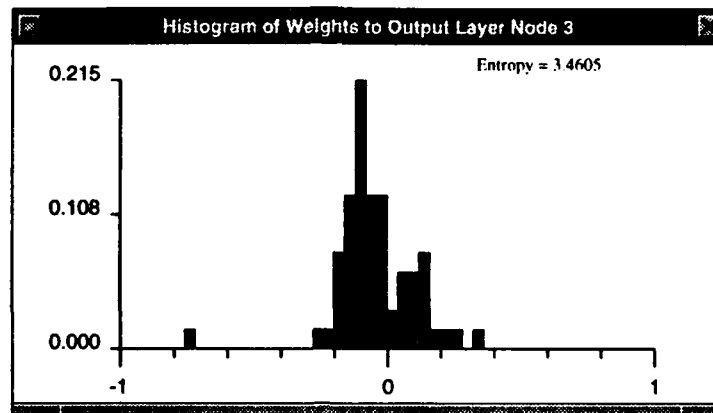


Figure 8.7: The distribution of parameter values in the 65-parameter logistic linear discriminant function representing the digit “3” (cf. figure 8.6). The parametric entropy of these weights is 3.46; the parametric entropy of all weights in the 650-parameter model is 3.18, compared with 2.30 for the 2570-parameter model. The increased variance in these weights compared to those in figures 8.2 and 8.4 reflects the classifier’s lower functional complexity: each weight now contains more information for the discrimination task.

of weight smoothing⁷ ($\kappa = 0.128$), the “3” discriminant function’s weight vector has relatively low parametric entropy (2.39). This low entropy reflects the low variance in the histogram of the weights. The parametric entropy of all the classifier’s weights is 2.3, a relatively small value suggesting that 256-pixel images can be compressed without an appreciable loss of information. That is, we would expect that compressing the images would increase the parametric entropy of the resulting lower-complexity classifier, and that this increase would not be so large as to cause an increase in the classifier’s error rate. Our expectation is based on the notion that the classifier encodes all the information required to classify all the training examples correctly; this information can be measured in terms of the total number of bits necessary to describe the classifier’s discriminator. As the number of parameters in the discriminator is decreased, the same amount of information must be encoded with fewer parameters, so the parametric entropy — our measure of the average amount of information in a single parameter of the discriminator — increases. We remind the reader that parametric entropy is an ad-hoc measure of the information content in a retinotopic parameter vector (see section M.1.1). Given this measure, we hypothesize — but have not proven — that there is an upper bound on the classifier’s information capacity, which corresponds to an upper bound on the parametric entropy of its weight vector. Beyond this upper bound, the classifier fails to encode all of the information in the training sample essential to robust discrimination. Below this upper bound, the classifier has more than sufficient capacity to encode all the information necessary for robust discrimination.

Our belief that the classifier complexity can be reduced without an appreciable information loss is validated by figure 8.5, which shows the images of figure 8.1 after they are compressed using the procedure

⁷The weight smoothing parameter κ has a value between zero and one (see appendix M). A value of zero results in no smoothing; a value of one forces all weights to have the same value. From a qualitative perspective, any value of $\kappa > 0.1$ is large.

described in section M.3. The images are still quite legible to the human eye, despite their having one fourth the number of pixels. Figure 8.6 shows the weights of a 650-parameter logistic linear classifier after it learns the compressed benchmark training sample differentially. Again, the classifier learns with a weight smoothing coefficient of $\kappa = 0.128$. Figure 8.7 shows that the “3” discriminant function’s parametric entropy has increased from 2.39 (for the 2570-parameter classifier) to 3.45, while the entropy of all discriminant functions has increased from 2.30 to 3.18. That is, each weight in the 650-parameter model encodes more information (as measured by the classifier’s parametric entropy) than each weight in the 2570-parameter model. At the same time, the lower-complexity classifier’s empirical benchmark test sample error rate has *dropped* from 2.7 (+1.4/-1.3)% to 1.3 (+1.1/-0.9)% — a 52% reduction, which indicates the improved generalization of the lower-complexity classifier.

Figure 8.8 compares the empirical benchmark test sample error rates for the 650-parameter logistic linear classifier employing differential learning with those of two probabilistically-generated counterparts. The differentially-generated model’s 1.3 (+1.1/-0.9)% error rate is approximately one half the MSE-generated model’s rate of 2.7 (+1.4/-1.3)%, and it is approximately one third the CE-generated model’s rate of 4.0 (+1.7/-1.6)%. Also shown are the benchmark test sample error rates of the best independently-developed linear classifier and the best independently-developed non-linear classifier. Both of these independent results are described in [16]. These classifiers learn a subset of the *un-compressed* benchmark training sample, which has had unrepresentative examples removed by a culling procedure described in [16]. The independently-developed linear classifier shown learns the culled training sample using a discriminative learning procedure also described in [16]; it exhibits an empirical benchmark test sample error rate of 3.2 (+1.6/-1.4)%. The independently-developed non-linear classifier shown learns the culled training sample after all its examples have been heavily filtered using a Gaussian smoothing kernel; it exhibits an error rate of 0.3 (+0.6/-0.3)%.

8.4 Recognition Results

Figure 8.8 shows that the differentially-generated logistic linear classifier makes fewer recognition errors on the benchmark test sample than all but the best independently-developed non-linear classifier. Based on this single trial, the differentially-generated linear model is not significantly better than the other linear models, nor is it significantly worse than the non-linear model. Since the independent results are based on a single trial using the benchmark test sample, the only multi-trial comparisons we can make are with our own probabilistically-generated models.⁸

⁸Geman et al have run multiple independent learning/testing trials using random data splits [41], but the error rates of their probabilistically-generated classifiers are considerably higher than those of our probabilistic controls. We therefore restrict our multi-trial comparisons to our own experiments in order to give probabilistic learning a fair evaluation.

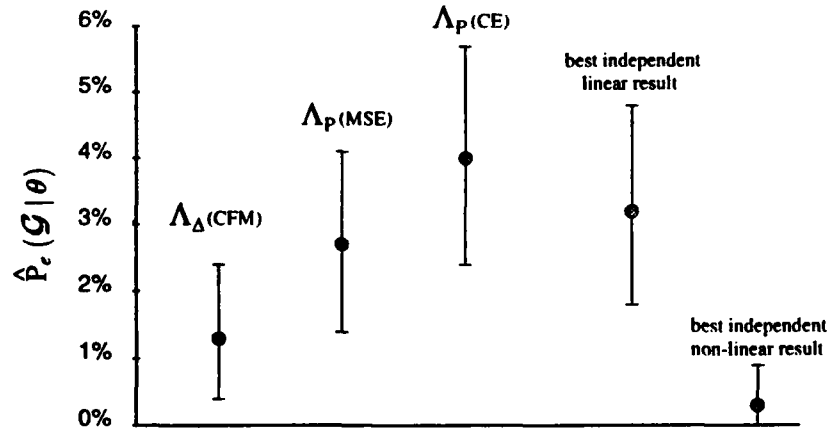


Figure 8.8: Test sample empirical error rates with 95% confidence intervals for the DB1 database's benchmark split of training/testing examples. The differentially-generated logistic linear classifier (Λ_{Δ}) is shown with two probabilistically-generated controls (Λ_P), the best independent linear result [16], and the best independent non-linear result [16].

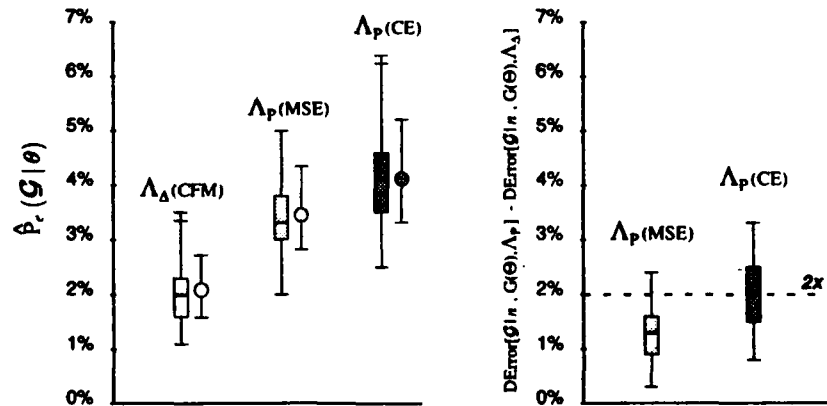


Figure 8.9: **Left:** Test sample classification summaries for the 650-parameter logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). The summaries are based on 25 independent trials in which the DB1 database is randomly partitioned into training and test samples, each containing approximately 600 examples. The box plots are a non-parametric depiction of the empirical test sample error rate's distribution over the 25 trials; the whisker plots depict the average empirical test sample error rate plus and minus one standard deviation, thereby characterizing each classifier's MSDE. **Right:** The difference between the probabilistically-generated models' empirical error rate and the differentially-generated model's rate on a trial-by-trial basis. An increase of 2% represents a doubling of the differentially-generated classifier's median empirical error rate. These box plots show that differential learning always produces the classifier with the lowest empirical error rate. Moreover, the lower-complexity differentially-generated logistic linear classifier generalizes better than all of the higher-complexity classifiers in figure 8.3.

Estimated MSDE (25 trials)			
Classifier	Learning Strategy		
	Λ_{Δ} (CFM)	Λ_P (MSE)	Λ_P (CE)
2570-Parameter	1.6×10^{-3}	1.9×10^{-3}	1.9×10^{-3}
650-Parameter	4.7×10^{-4}	1.2×10^{-3}	1.8×10^{-3}

Table 8.1: Estimated MSDE for the high and low-complexity logistic linear classifiers employing differential learning (Λ_{Δ}) via the CFM objective function and probabilistic learning (Λ_P) via the MSE and CE objective functions. Estimates are based on 25 independent learning/testing trials. Reducing the classifier's complexity by compressing the digit images has the beneficial effect of reducing the classifier's estimated MSDE. The reduction is most pronounced in the differentially-generated model, as predicted by theory.

8.4.1 Experiments with the Logistic Linear Hypothesis Class

Figure 8.9 compares the 650-parameter logistic linear classifier employing differential learning with controls that employ probabilistic learning (MSE and CE objective functions). These comparisons are for the same 25 random splits of the DB1 database used for the 2570-parameter classifier tests shown in figure 8.3. All the low-complexity models exhibit lower empirical MSDE than their higher-complexity counterparts, as indicated by the lower average error rates and slightly reduced whisker plot spans of figure 8.9 (left) versus figure 8.3 (left). Table 8.1 summarizes the estimated MSDE for the high and low complexity classifiers, given the three learning strategies. The differentially-generated model exhibits the largest reduction in MSDE: its average empirical test sample error rate drops from 3.9% to 2.1%, while the standard deviation of this statistic drops from 0.71% to 0.57%. Assuming a Bayes error rate of zero for the DB1 database, the 2570-parameter differentially-generated model's empirical MSDE is, by (3.9), 1.6×10^{-3} ; the 650-parameter differentially-generated model's empirical MSDE is 4.7×10^{-4} — approximately one fourth that of the higher-complexity model. Thus, reducing the classifier's complexity by compressing the image feature vector by a factor of 4 : 1 reduces the differentially-generated model's MSDE by approximately the same ratio. For probabilistic learning via MSE, the higher-complexity model's MSDE is 1.9×10^{-3} , and the lower-complexity model's is 1.2×10^{-3} . For probabilistic learning via the Kullback-Leibler information distance (CE), the higher-complexity model's MSDE is 1.9×10^{-3} , and the lower-complexity model's is virtually unchanged at 1.8×10^{-3} . Figure 8.9 (right) also shows that the reduced-complexity differentially-generated classifier consistently exhibits a lower empirical test sample error rate than its probabilistic counterparts. The MSE-generated classifier's error rate is typically 1.3% greater than (or 1.65 times) the CFM-generated classifier's. The CE-generated classifier's error rate is typically 2.0% greater than (or two times) the CFM-generated classifier's. Thus, reducing the classifier's

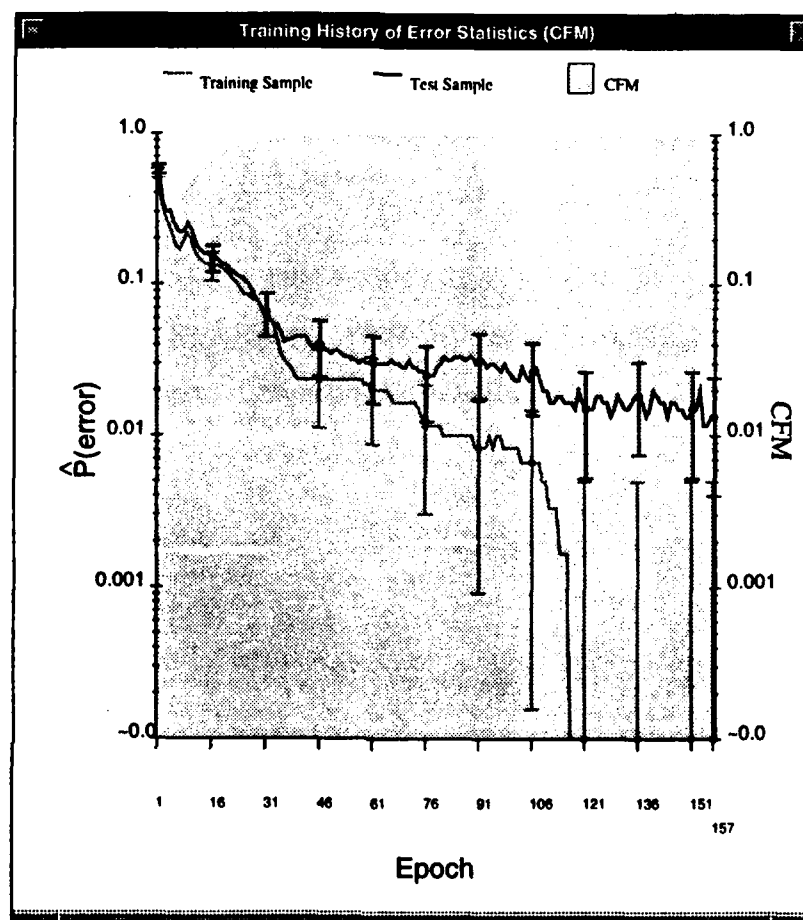


Figure 8.10: The empirical error rates (training sample in gray and test sample in black) for the 650-parameter logistic linear classifier as it learns the benchmark training sample differentially. The classifier's empirical test sample error rate is 1.3 (+1.1/-0.9)% after 157 learning epochs.

complexity reduces the MSDE of all the classifiers, but the reduction realized by the differentially-generated model is significantly greater.

Comparing Learning Strategies for the Benchmark Training/Test Sample

In simple terms, the reduced-complexity differentially-generated model realizes the biggest reduction in MSDE because (as proven in chapter 3) differential learning 1) is asymptotically efficient, regardless of the choice of hypothesis class, and 2) it requires the minimum-complexity hypothesis class necessary for Bayesian discrimination. Figures 8.10 – 8.15 demonstrate these characteristics for the logistic linear classifier learning the benchmark training sample. Figure 8.10 shows both the training sample (gray) and test sample (black)

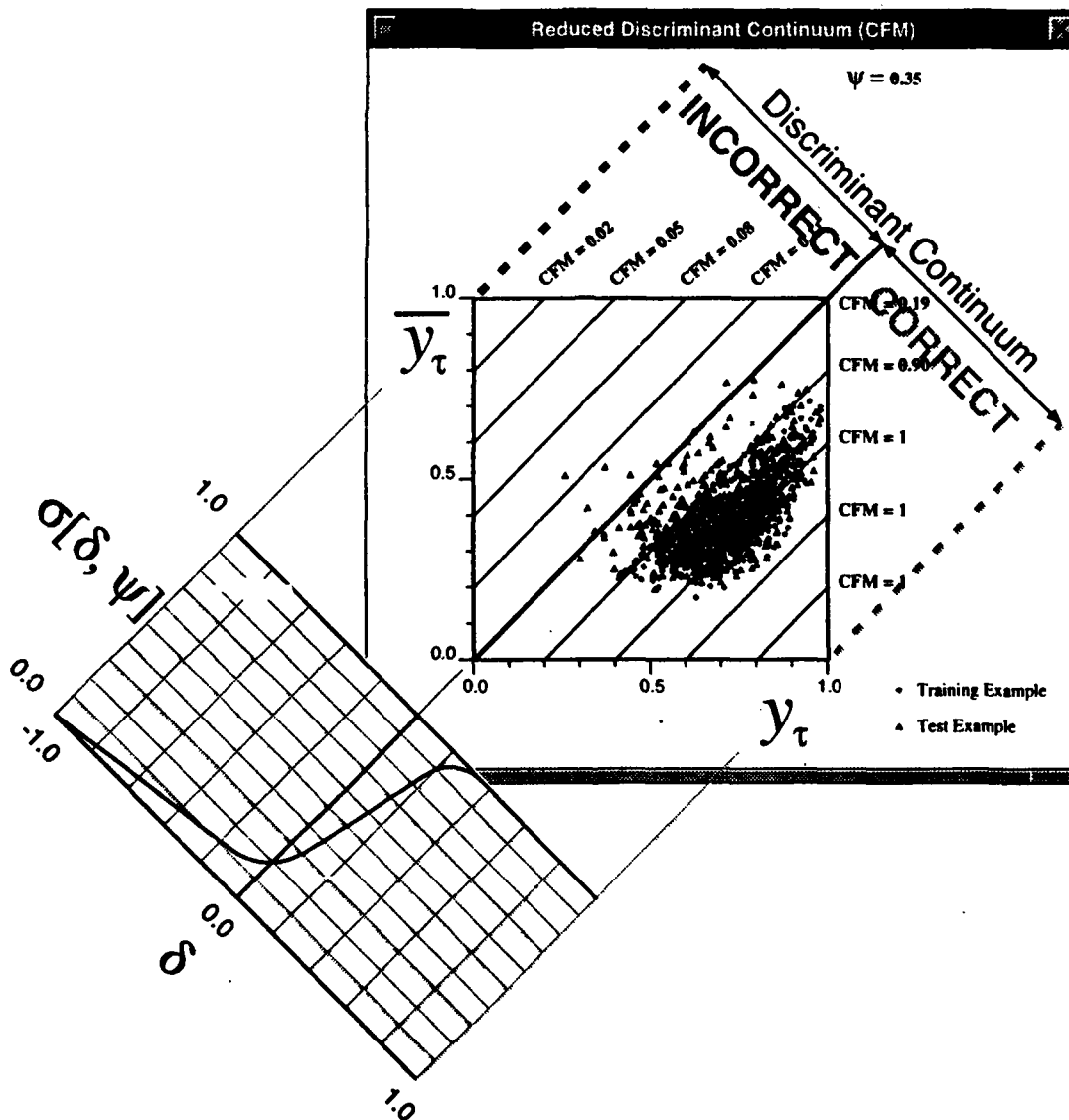


Figure 8.11: The 650-parameter logistic linear classifier's output state — as projected onto reduced discriminator output space — after learning the 600 benchmark training examples differentially. This output state corresponds to the parameters shown in figure 8.6. Note how most of the test examples (black triangles) and all of the training examples (gray dots underneath the test examples) are aligned with the contours of constant CFM on reduced discriminator output space. These constant CFM contours are parallel to the reduced discriminant boundary (definition 5.5) — a necessary condition for efficient learning.

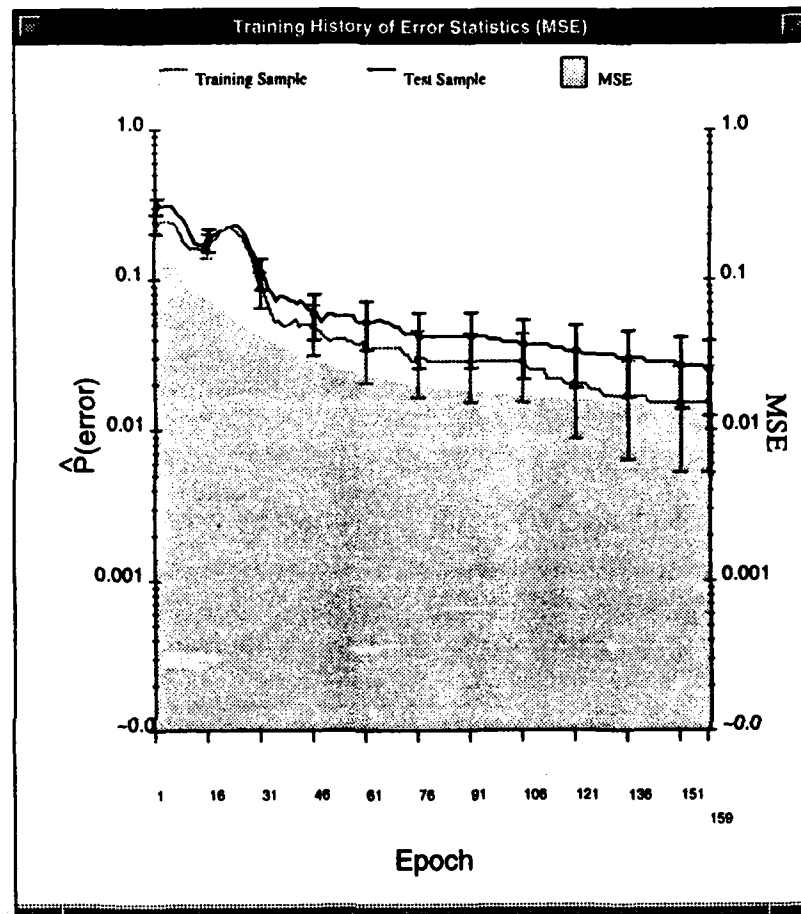


Figure 8.12: The empirical error rates (training sample in gray and test sample in black) for the 650-parameter logistic linear classifier as it learns the benchmark training sample probabilistically (MSE objective function). The classifier's empirical test sample error rate is 2.7 (+1.4/-1.3)% after 159 learning epochs.

empirical error rates as differential learning progresses through approximately 160 learning epochs. The objective function's value is plotted as a light gray background in the figure. Ninety-five percent confidence intervals on the error rates are plotted at periodic intervals. From these one can see that the training sample error rate is representative of the test sample error rate up to ninety differential learning epochs. Beyond this point the empirical training sample error rate is significantly lower than the test sample error rate. During differential learning, ψ' is reduced from a value of 0.48 at epoch zero to 0.35 beyond epoch 100. The final output state of the logistic linear classifier is shown on reduced discriminator output space (definition 5.2) in figure 8.11. Test examples are shown as black triangles, and training examples are shown as gray dots. After approximately 160 epochs, all the training examples lie parallel to the CFM = 0.90 contour, as do most of

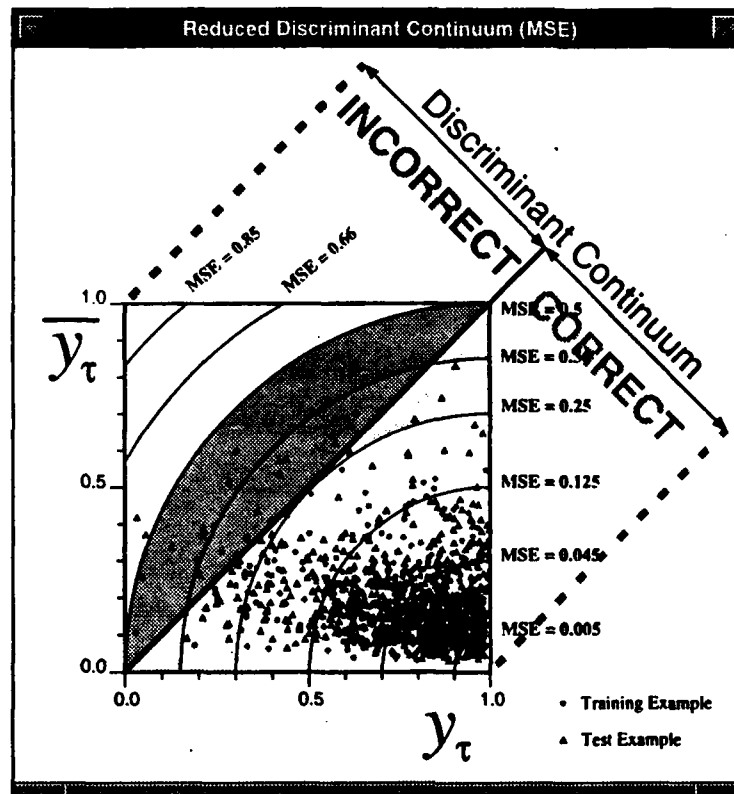


Figure 8.13: The 650-parameter logistic linear classifier's output state — as projected onto reduced discriminator output space — after it attempts to learn the 600 benchmark training examples probabilistically. Note that the MSE-generated classifier cannot learn all the training examples, given the low-complexity logistic linear hypothesis class.

the test examples. The remaining test examples — ones that are hard for the classifier to discriminate — fall close to the reduced discriminant boundary (definition 5.5). Owing to the monotonic nature of the CFM objective function, these examples are also parallel to the reduced discriminant boundary, and most of them are on the correct side of the boundary.

Figure 8.12 shows the empirical training and test sample error rates for the logistic linear classifier that learns the benchmark training sample probabilistically via the MSE objective function. Its empirical training sample error rate remains representative of the test sample error rate through all 160 epochs, although both error rates are higher than those for the differentially-generated model. Unfortunately the 650-parameter classifier that learns probabilistically with a weight smoothing coefficient of $\kappa = .128$ has insufficient functional complexity to learn the training sample as well as its differentially-generated counterpart (cf. figure 8.13 versus figure 8.11). As a result, the non-monotonic nature of the MSE objective function leads the classifier to learn the majority of easy examples with high confidence (i.e., to minimize the MSE

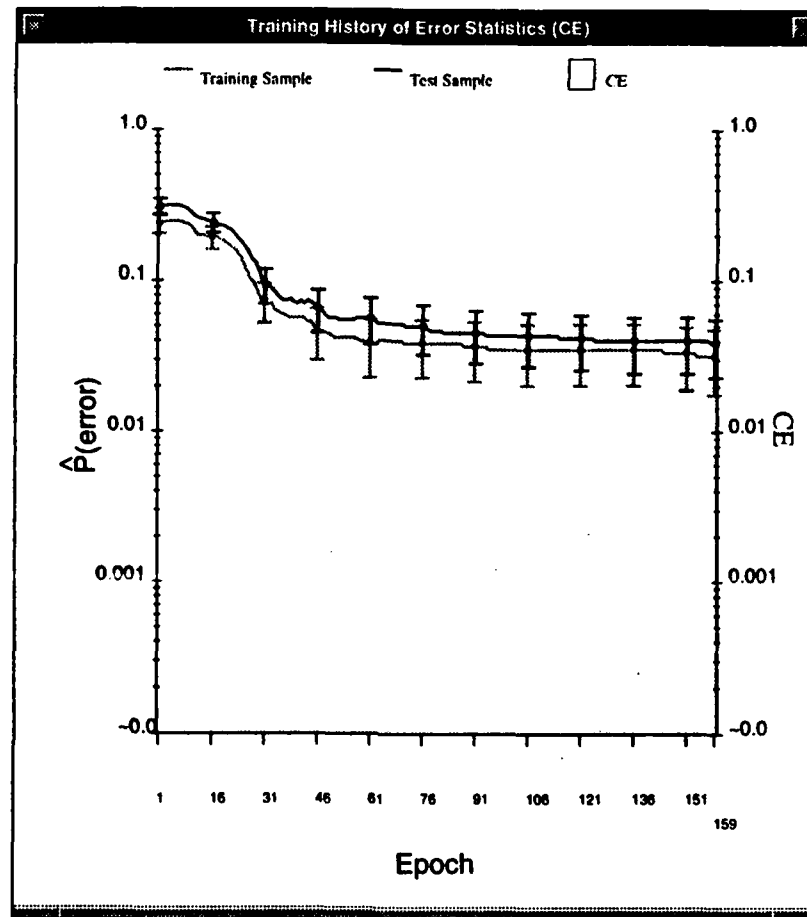


Figure 8.14: The empirical error rates (training sample in gray and test sample in black) for the 650-parameter logistic linear classifier as it learns the benchmark training sample probabilistically (CE objective function). The classifier's empirical test sample error rate is 4.0 (+1.7/-1.6)% after 159 learning epochs.

between its outputs and the easy examples' binary target vectors), while it fails to learn the minority of hard examples. This phenomenon is clearly depicted in figure 8.13. The MSE-generated classifier's training and test examples are aligned with constant contours of MSE; the harder the example, the higher the MSE. However the contours of constant MSE are not parallel to the reduced discriminant boundary (i.e., MSE is not a monotonic objective function — definition 5.10). As a result, a larger proportion of hard examples fall on the incorrect side of the boundary, and the classifier exhibits higher empirical training *and* test sample error rates than its differentially-generated counterpart (see figure 8.8, page 231).

The classifier that employs probabilistic learning via the Kullback-Leibler information distance (CE objective function) exhibits the same inefficient behavior that the MSE-generated classifier exhibits. Figure 8.14 shows that the CE-generated classifier's empirical training sample error rate remains representative

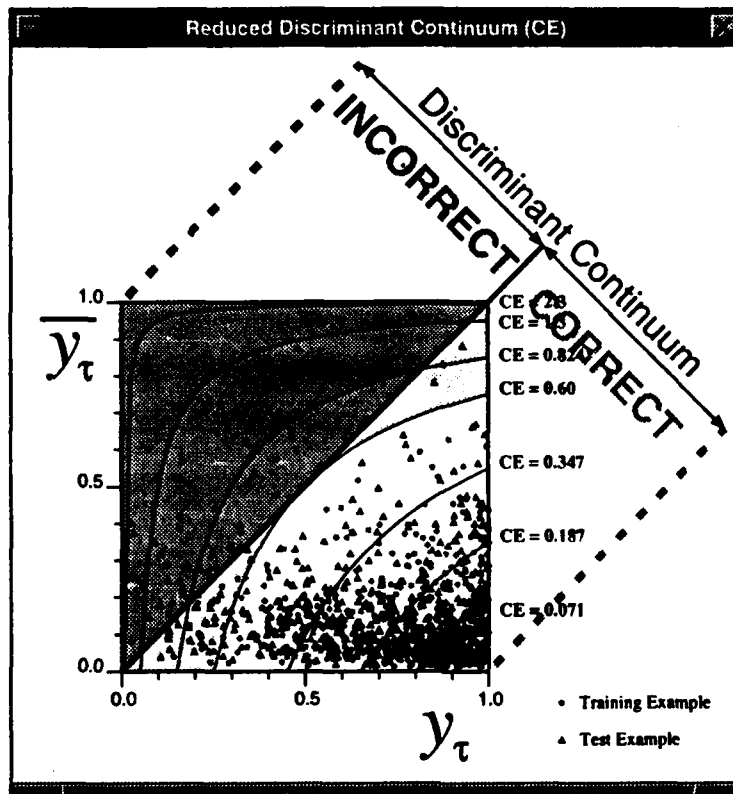


Figure 8.15: The 650-parameter logistic linear classifier's output state — as projected onto reduced discriminator output space — after it attempts to learn the 600 benchmark training examples probabilistically. Note that the Kullback-Leibler (CE) generated classifier cannot learn all the training examples, given the low-complexity logistic linear hypothesis class.

of the test sample error rate through all 160 epochs, although both error rates are higher than those for the differentially-generated model. Again, the 650-parameter classifier that learns probabilistically with a weight smoothing coefficient of $\kappa = .128$ has insufficient functional complexity to learn the training sample as well as its differentially-generated counterpart. As a result, the CE objective function induces the classifier to learn the easy examples with high confidence while it fails to learn the minority of hard examples. Since the contours of constant CE have even more curvature than those of the MSE objective function (i.e., CE is even less monotonic than the MSE objective function — cf. figures 8.15 versus 8.13), a proportionally greater number of hard examples fall on the incorrect side of the reduced discriminant boundary. This is reflected in the CE-generated classifier's elevated empirical training and test sample error rates (again, see figure 8.8, page 231).

8.4.2 Experiments with Alternative Hypothesis Classes

The inefficiency of probabilistic learning impacts the classifier's MSDE to a degree that varies with the choice of hypothesis class. If the hypothesis class is a reasonably good approximation to a proper parametric model of the feature vector (definition 3.13, page 63), the MSDE of classifiers produced from the hypothesis class by probabilistic learning might be lower than the MSDE of their differentially-generated counterparts for small training sample sizes. However, if the hypothesis class is a distinctly improper parametric model of the feature vector, the MSDE of the probabilistically-generated classifier will be substantially higher than that of the differentially-generated classifier (recall that chapter 4 illustrates this phenomenon for a simulated random feature variable).

The Linear Hypothesis Class

Figures 8.16 — 8.20 provide a detailed comparison of probabilistic and differential learning when the classifier is generated from a 650-parameter linear hypothesis class of the form described in section 7.2.1. Learning with the linear hypothesis class proceeds faster than it does with the logistic linear hypothesis class for the reasons outlined in section 5.5.1 (all learning parameters for the linear hypothesis class are identical to those for the logistic linear hypothesis class). This is evident from the history of the benchmark empirical training and test sample error rates shown in figure 8.16. As with the logistic linear hypothesis class, the training sample error rate is significantly lower than the test sample error rate beyond a certain point in the differential learning trial (40 epochs in this case). The linear classifier's output state after 75 differential learning epochs is shown in figure 8.17. Many training and test examples fall outside the unit square on reduced discriminator output space because the linear classifier's outputs are not bounded. That is, $\mathcal{Y} = \mathcal{R}^C$ rather than $[0, 1]^C$. Nevertheless, we see the same general trends displayed by the logistic linear classifier in figure 8.11: All of the training examples are learned, since they engender discriminant differentials that are greater than $\delta = \delta_{\text{learned}} \approx 0.2$. Most of the test examples also exhibit relatively large positive discriminant differentials. The harder test examples with negative or relatively small positive discriminant differentials are parallel to the reduced discriminant boundary. The differentially-generated linear classifier's benchmark empirical test sample error rate is 2.3 (+1.4/-1.3)% — not significantly higher than the differentially-generated logistic linear classifier's rate.

Figure 8.18 shows the benchmark empirical training and test sample error rates of the linear classifier during probabilistic learning via MSE. The training sample error rate remains representative of the test sample error rate throughout the learning trial. Indeed, the error rates converge to their final values after approximately 35 epochs. The final empirical test sample error rate is 5.0 (+1.9/-1.8)% — more than twice the differentially-generated model's rate. Being an improper parametric model of $\mathbf{X}$ and having insufficient functional complexity to model the empirical *a posteriori* class probabilities of $\mathbf{X}$ accurately, the probabilistically-generated linear model minimizes the MSE between its output state and the training sample

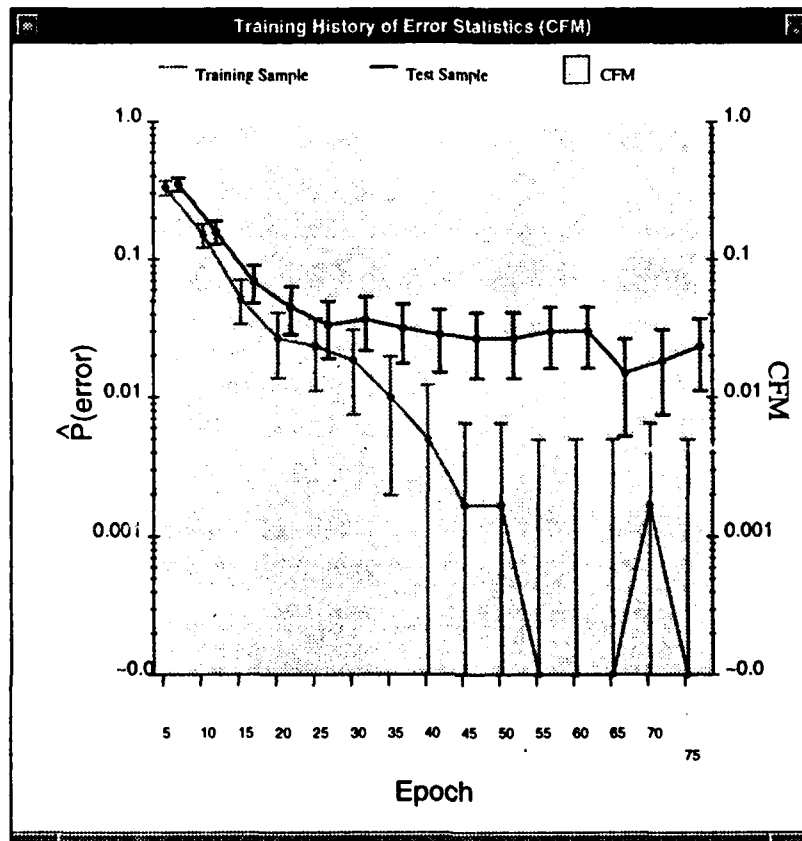


Figure 8.16: The empirical error rates (training sample in gray and test sample in black) for the 650-parameter linear classifier as it learns the benchmark training sample differentially. Learning is predictably faster, albeit somewhat less stable than it is with the logistic linear hypothesis class. The classifier's empirical test sample error rate is 2.3 (+1.4/-1.2)% after 75 learning epochs. The critical reader should note that the empirical test sample error rate settles at this value beyond 75 learning epochs.

target vectors as best it can. Figure 8.19 shows the classifier's post-learning output state as projected onto reduced discriminator output space. Note that both training and test samples are aligned with the contours of constant MSE. This is particularly clear for the misclassified examples, which fall on the incorrect side of the reduced discriminant boundary. All of the training examples and most of the test examples that fall inside the $\text{MSE} = 0.36$ contour are learned or correctly classified by the differentially-generated model in figure 8.17. Figure 8.20 shows that the probabilistically-generated linear classifier's error rate is consistently higher than the differentially-generated model's across the 25 random splits of the DB1 database. Indeed, the MSE-generated model's error rate is typically more than three times the CFM-generated model's. Experiments

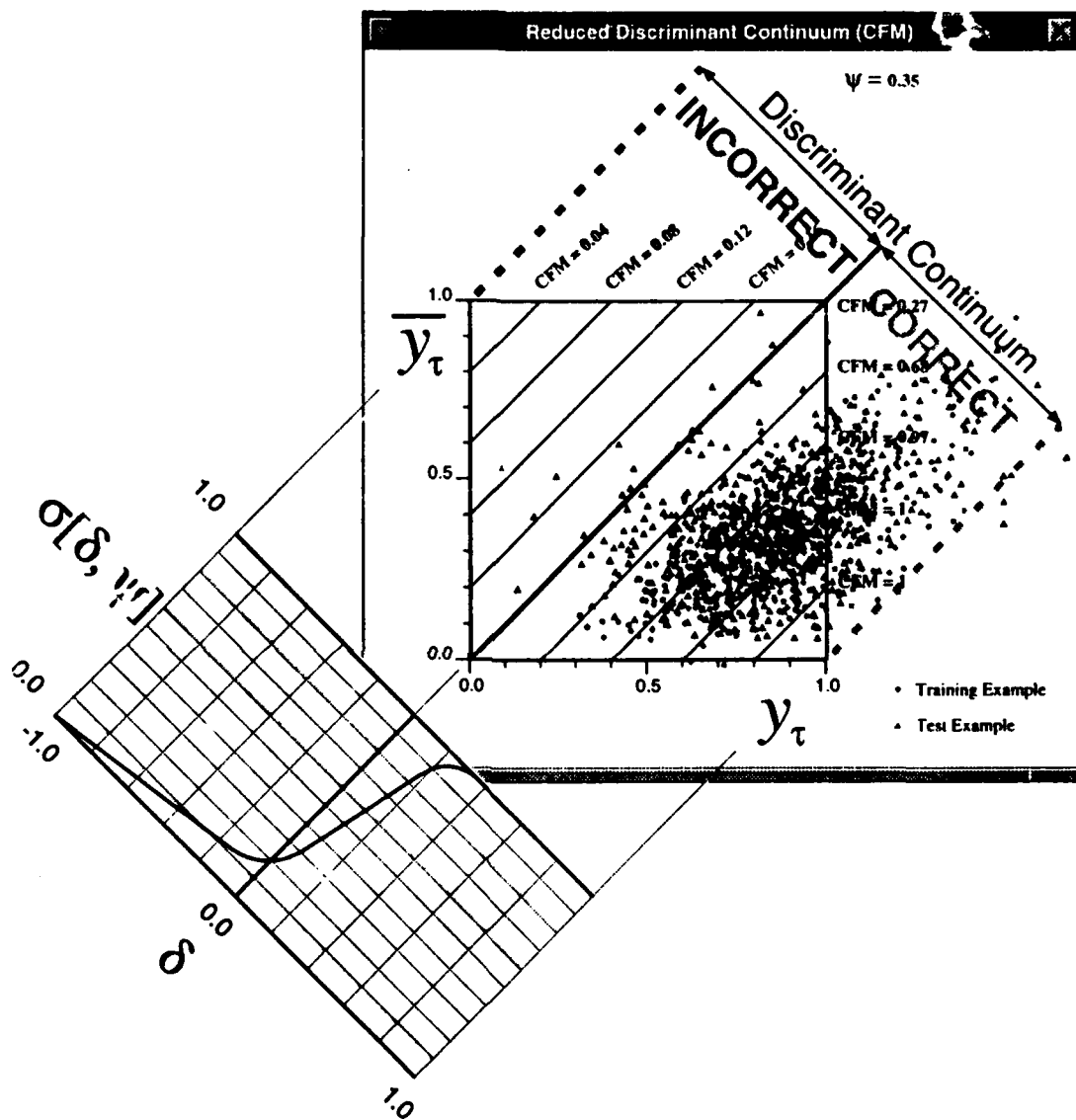


Figure 8.17: The 650-parameter linear classifier's output state — as projected onto reduced discriminator output space — after learning the 600 benchmark training examples differentially. Since the linear classifier's outputs are not bounded on $[0, 1]$, significant fractions of the training and test samples engender output states that fall outside the unit hypercube $[0, 1]^{C=10}$. Such examples therefore fall outside the unit square in this figure.

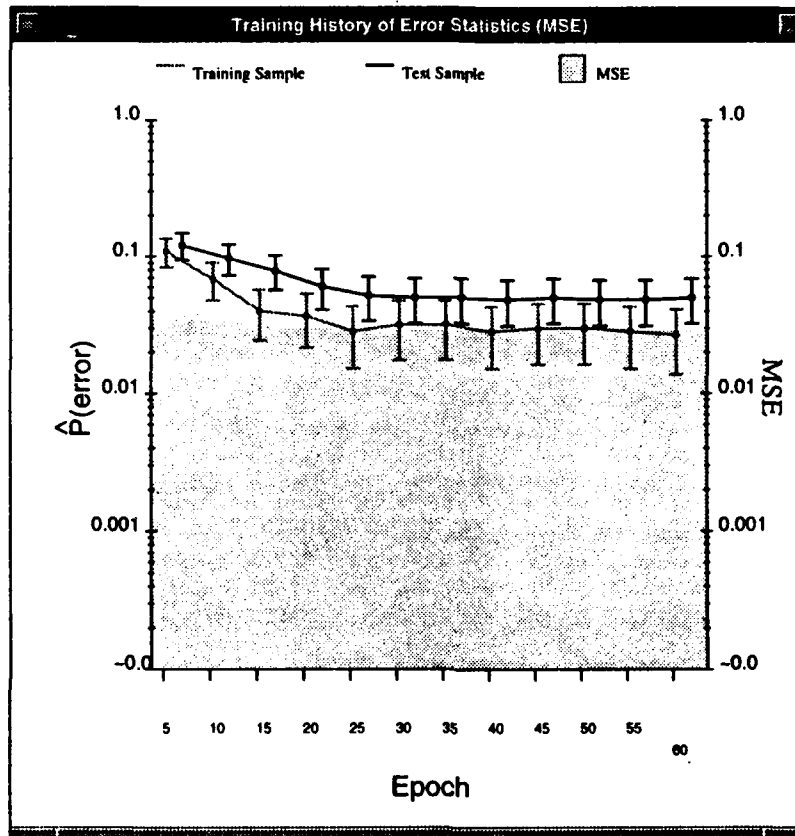


Figure 8.18: The empirical error rates (training sample in gray and test sample in black) for the 650-parameter linear classifier as it learns the benchmark training sample probabilistically (MSE objective function). The classifier's empirical test sample error rate is 5.0 (+1.9/-1.8)% after 60 learning epochs. No further learning occurs beyond 60 epochs.

with the CE objective function are not possible because the linear classifier's outputs are unbounded; this violates the conditions necessary for learning via the CE objective function (see section 2.3.2).

The Modified Gaussian RBF Hypothesis Class

Figure 8.21 summarizes the results of 25 learning trials, given a 650-parameter modified Gaussian RBF hypothesis class of the form described in appendix K. Results are summarized for differential learning via the CFM objective function and probabilistic learning via the MSE and CE objective functions. These comparisons are for the same 25 random splits of the DB1 database used for all the previous multi-trail experiments. The modified RBF hypothesis class is an improper parametric model of X . In addition, the

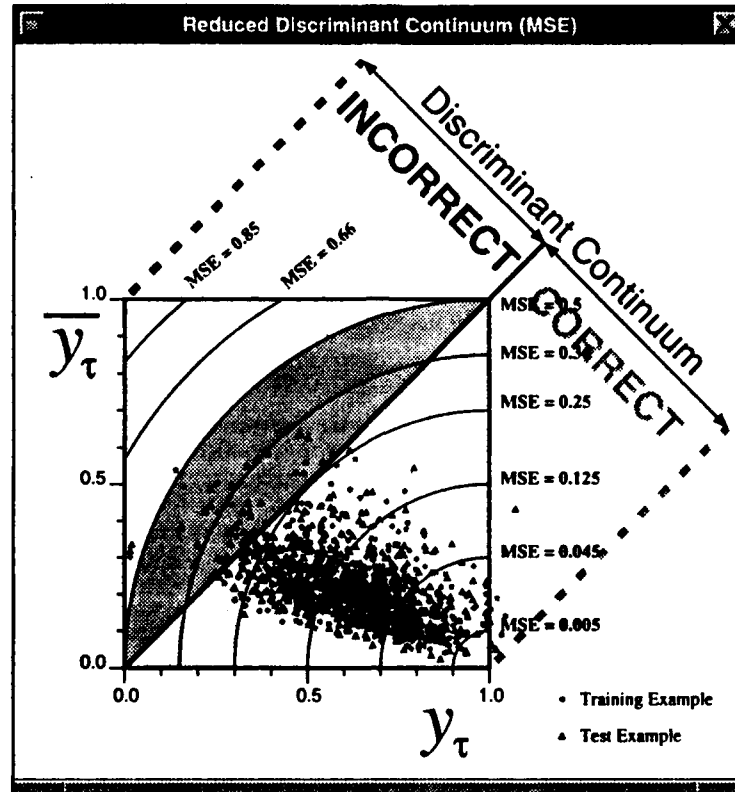


Figure 8.19: The 650-parameter linear classifier's output state — as projected onto the reduced discriminator output space — after it attempts to learn the 600 benchmark training examples probabilistically (MSE objective function). Note that the MSE-generated classifier cannot learn all of the training examples, given the low-complexity linear hypothesis class.

hypothesis class cannot form the same piece-wise linear boundaries on $\mathcal{X}$ that the linear hypothesis classes can.⁹ As a result, the modified RBF hypothesis class has insufficient functional complexity to match the linear hypothesis class error rates. This is evident in the 3.8% median error rate of the differentially-generated classifier, which is approximately twice the differentially-generated linear classifiers' median empirical error rates. The probabilistically-generated RBF classifiers fare worse in comparison to their linear counterparts; their median empirical error rates increase to 12% (MSE) and 10% (CE). The probabilistically-generated RBF classifiers consistently exhibit empirical error rates that are between two and four times the median rate for the differentially-generated classifier.

8.4.3 Interpretation of Results

⁹A formal proof of this assertion would require a number of pages, and would not lend anything of substance to our line of argument. Therefore, we ask the reader to accept this assertion on faith.

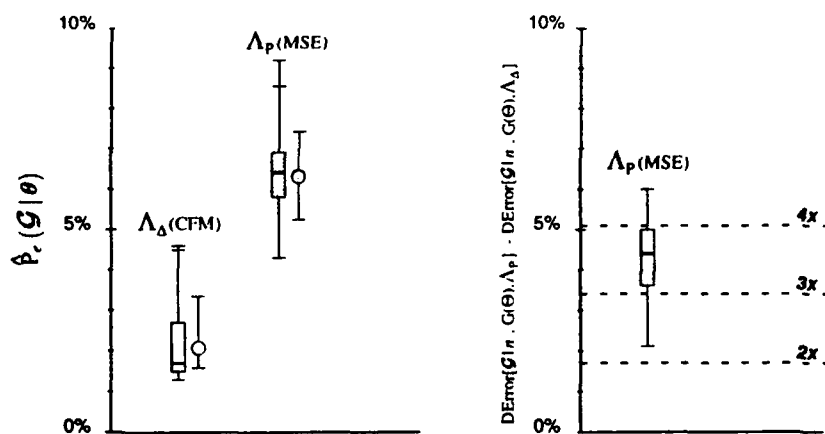


Figure 8.20: **Left:** Test sample classification summaries for the 650-parameter linear classifier employing differential learning (Λ_Δ) and the MSE form of probabilistic learning (Λ_P). The summaries are based on 25 independent trials in which the DB1 database is randomly partitioned into training and test samples, each containing approximately 600 examples. The box plots are a non-parametric depiction of the empirical test sample error rate's distribution over the 25 trials; the whisker plots depict the average empirical test sample error rate plus and minus one standard deviation, thereby characterizing each classifier's MSDE. **Right:** The difference between the probabilistically-generated models' empirical error rate and the differentially-generated model's rate on a trial-by-trial basis. This box plot shows that differential learning always produces the classifier with the lowest empirical error rate. Probabilistic learning engenders linear classifiers with error rates that are typically more than three times those of the differentially-generated linear classifier.

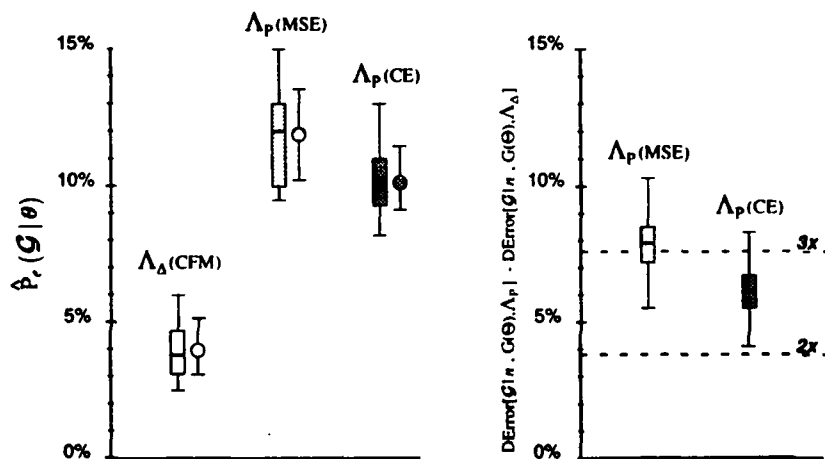


Figure 8.21: **Left:** Test sample classification summaries for the 650-parameter modified RBF classifier employing differential learning (Λ_Δ) and two forms of probabilistic learning (Λ_P). The summaries are based on 25 independent trials in which the DB1 database is randomly partitioned into training and test samples, each containing approximately 600 examples. The box plots are a non-parametric depiction of the empirical test sample error rate's distribution over the 25 trials; the whisker plots depict the average empirical test sample error rate plus and minus one standard deviation, thereby characterizing each classifier's MSDE. **Right:** The difference between the probabilistically-generated models' empirical error rate and the differentially-generated model's rate on a trial-by-trial basis. These box plots show that differential learning always produces the classifier with the lowest empirical error rate, as is the case with both 650-parameter linear hypothesis classes. Probabilistic learning engenders RBF classifiers with error rates that are typically two and a half to three times those of the differentially-generated RBF classifier.

Estimated DBias, DVar, and MSDE (25 trials)				
Hypothesis	Learning Strategy			
Class		Λ_{Δ} (CFM)	Λ_P (MSE)	Λ_P (CE)
Linear	DBias	2.1×10^{-2}	6.3×10^{-2}	N/A
	DVar	6.3×10^{-5}	1.2×10^{-4}	N/A
	MSDE	4.9×10^{-4}	4.1×10^{-3}	N/A
Logistic Linear	DBias	2.1×10^{-2}	3.4×10^{-2}	4.1×10^{-2}
	DVar	3.2×10^{-5}	5.6×10^{-5}	8.7×10^{-5}
	MSDE	4.7×10^{-4}	1.2×10^{-3}	1.8×10^{-3}
Modified RBF	DBias	4.0×10^{-2}	11.9×10^{-2}	10.1×10^{-2}
	DVar	1.1×10^{-4}	2.7×10^{-4}	1.3×10^{-4}
	MSDE	1.7×10^{-3}	1.4×10^{-2}	1.0×10^{-2}

Table 8.2: Estimated discriminant bias, discriminant variance, and MSDE for 650-parameter classifiers generated from the linear, logistic linear, and modified RBF hypothesis classes by differential learning (Λ_{Δ}) via the CFM objective function and probabilistic learning (Λ_P) via the MSE and CE objective functions. Estimates are based on 25 independent learning/testing trials in which the DB1 database is randomly partitioned into training and test samples, each containing approximately 600 examples. The differentially generated classifier's MSDE is $\mathcal{O}[1/10]$ that of its probabilistically generated counterparts' for all three hypothesis classes.

Table 8.2 summarizes the estimated discriminant bias, discriminant variance, and MSDE of the classifiers generated from the linear, logistic linear, and modified RBF hypothesis classes. Results are given for each learning strategy, as appropriate. All values are based on the assumption that the Bayes error rate for the DB1 task is zero, given the compressed images (see section 8.2). The MSE-generated logistic linear classifier's empirical MSDE is 2.5 times the differentially-generated classifier's; the CE-generated logistic linear classifier's empirical MSDE is 4 times the differentially-generated classifier's. Most of the increase is due to increased discriminant bias. For the linear and modified RBF hypothesis classes, the probabilistically-generated classifiers' MSDE is an order of magnitude higher than the differentially-generated classifier's (recall from chapter 3 that a classifier with lower MSDE constitutes a better approximation to the Bayes-optimal classifier). Although most of the MSDE increase is due to increased discriminant bias, an appreciable fraction of it is due to the increased discriminant variance of these hypothesis classes when paired with probabilistic learning.

These findings are consistent with the theoretical predictions of chapter 3. The non-monotonic nature of error measures clearly plays a role in the inefficient behavior of probabilistic learning strategies and explains

in geometric terms why minimizing the classifier's functional error does not minimize its discriminant error — proofs of which are given in section 3.4. The chapter 3 proof that differential learning produces the relatively efficient classifier, regardless of the choice of hypothesis class, is clearly demonstrated by the statistics in table 8.2.

8.4.4 Rejecting Classifications After Learning

Table 8.3 reviews the empirical error rates for the benchmark test sample. Rates are given for classifiers generated from the linear, logistic linear, and modified Gaussian RBF hypothesis classes with differential learning via CFM and probabilistic learning via MSE and CE. The logistic linear classifiers' error rates correspond to figures 8.11 (CFM), 8.13 (MSE), and 8.15 (CE). The linear classifiers' error rates correspond to figures 8.17 (CFM) and 8.19 (MSE). The modified Gaussian RBF classifiers' error rates correspond to the text of section 8.4.2.

Table 8.4 shows the results of rejecting marginal classifications (i.e., those close to the reduced discriminant boundary) for each of these classifiers. As described in section 7.6, the classifier rejects all test examples that generate a top-ranked discriminant differential $\delta_{(1)}(\mathbf{X}|\theta)$ that is less than the default rejection threshold δ_{reject} :

$$\begin{aligned} \text{reject classification iff } \underbrace{y_{(1)} - y_{(2)}}_{\delta_{(1)}(\mathbf{X}|\theta)} &\leq \delta_{\text{reject}}; \\ \delta_{\text{reject}} &= \frac{1}{2} \underbrace{x_{\eta}}_{(D.11)} \end{aligned} \tag{8.6}$$

Note that x_{η} is shown diagrammatically in figure D.1. We use this differential rejection threshold for the probabilistically-generated classifiers as well as the differentially-generated ones, since it yields better results than MSE or CE-based thresholds (recall section 7.6).

The differentially generated classifiers consistently reject no more than 6% of the test sample and exhibit no more than a 1% error rate on the remaining (i.e., un-rejected) test examples. The consistency of these rejection/error rates does not hold for the probabilistically-generated classifiers. Both probabilistically-generated logistic linear classifiers reject no more than 6% of the test sample and exhibit no more than a 2% error rate on the remaining test examples, but the linear and modified Gaussian RBF classifiers' rejection/error statistics are considerably worse. The MSE-generated linear classifier rejects approximately 9% of the test sample, and misclassifies approximately 2% of the remaining test examples. The MSE-generated RBF classifier rejects approximately 15% of the test sample, and misclassifies approximately 5% of the remaining test examples. The CE-generated RBF classifier rejects approximately 16% of the test sample, and misclassifies approximately 3% of the remaining test examples.

Benchmark Empirical Error Rates			
Hypothesis	Learning Strategy		
Class	Λ_{Δ} (CFM)	Λ_P (MSE)	Λ_P (CE)
Linear	2.3 (+1.4/-1.2)%	5.0 (+1.9/-1.8)%	N/A
Logistic Linear	1.3 (+1.1/-0.9)%	2.7 (+1.4/-1.3)%	4.0 (+1.7/-1.5)%
Modified RBF	2.0 (+1.3/-1.1)%	11.5 (+2.7/-2.6)%	10.3 (+2.6/-2.4)%

Table 8.3: Empirical benchmark test sample error rates for 650-parameter classifiers produced from the linear, logistic linear, and modified RBF hypothesis classes by differential learning (Λ_{Δ}) via the CFM objective function and probabilistic learning (Λ_P) via the MSE and CE objective functions. Ninety five percent confidence intervals are based on the assumption that the error rates are binomially distributed [62].

Benchmark Rejection Rates / Error Rates				
Hypothesis		Learning Strategy		
Class		Λ_{Δ} (CFM)	Λ_P (MSE)	Λ_P (CE)
Linear	Fraction of Test Sample Rejected	4.2 (+1.7/-1.6)%	8.7 (+2.6/-2.3)%	N/A
	Fraction of Un-Rejected Test Sample Misclassified	0.9 (+0.9/-0.8)%	1.8 (+1.3/-1.1)%	N/A
Logistic Linear	Fraction of Test Sample Rejected	5.8 (+2.1/-1.9)%	4.8 (+1.9/-1.7)%	5.5 (+2.0/-1.8)%
	Fraction of Un-Rejected Test Sample Misclassified	0.4 (+0.6/-0.4)%	1.2 (+1.1/-0.9)%	1.8 (+1.2/-1.1)%
Modified RBF	Fraction of Test Sample Rejected	5.8 (+2.1/-1.9)%	15.2 (+3.0/-2.9)%	16.5 (+3.1/-3.0)%
	Fraction of Un-Rejected Test Sample Misclassified	0.4 (+0.6/-0.4)%	5.3 (+2.1/-2.0)%	3.0 (+1.7/-1.5)%

Table 8.4: Benchmark test sample rejection/empirical error rate statistics for 650-parameter classifiers produced from the linear, logistic linear, and modified RBF hypothesis classes by differential learning (Λ_{Δ}) via the CFM objective function and probabilistic learning (Λ_P) via the MSE and CE objective functions. Ninety five percent confidence intervals are based on the assumption that the error rates are binomially distributed [62].

As in section 7.6, we see that the inefficiency of probabilistic learning has a deleterious effect on the process by which the classification hypothesis is accepted or rejected. Moreover, the rejection/error statistics of probabilistically-generated classifiers are quite sensitive to the choice of hypothesis class. This phenomenon is consistent with the theoretical arguments of section 3.4, in which we make a clear distinction between minimizing discriminant error via differential learning and minimizing functional error via probabilistic learning. The efficiency of differential learning ensures consistent rejection/error statistics across a wide range of hypothesis classes.

8.5 Recognition Results in the Presence of Noise

As described in section 3.6, there are special cases in which probabilistic learning generates the efficient classifier of $\mathbf{X}$ for small training sample sizes, whereas differential learning does so only for large training sample sizes. Specifically, when the hypothesis class is a proper parametric model of $\mathbf{X}$ probabilistic learning will generate the efficient classifier of $\mathbf{X}$.

This is illustrated in the case of the DBI OCR task when the original 256-pixel binary images are corrupted by noise that takes the form of random independent pixel inversions throughout the image. This form of noise is originally described in [41]. When the probability of pixel inversion becomes relatively high and the noise-corrupted images are subsequently compressed using the simple linear lossy scheme described in section M.3, the resulting feature vector exhibits class-conditional pdfs that are very nearly Gaussian with homoscedastic covariance matrices. As a result, both fully-parametric and partially-parametric proper models exist for these compressed noisy characters.

We characterize the independent noise source as an additive one in order to derive a simple expression for the signal-to-noise ratio (SNR) of the un-compressed noise-corrupted images. We then prove by an application of the central limit theorem that compressing the noisy images generates an approximately homoscedastic Gaussian feature vector, the approximation being better as the SNR drops toward -0.8 dB. We find that differential learning generates a more efficient classifier, given any choice of hypothesis class, as long as the un-compressed image SNR remains above approximately 2 dB. When the SNR drops to 1.2 dB the logistic linear hypothesis class becomes a good approximation to the proper parametric model of the compressed noisy feature vector. As a result, classifiers generated probabilistically from this hypothesis class with training sample sizes of $n \approx 600$ are more efficient than their differentially generated counterparts.

8.5.1 Signal-to-Noise Ratio (SNR) Computations

Recall from section 8.1 that the feature vector for the original un-compressed DBI digits is a 256-pixel (16×16) binary vector with elements of $+1/-1$, where black = -1 and white = $+1$. In order to simplify our SNR expressions, we view this binary vector $\mathbf{X}$ as a simple affine transformation of another binary vector

$\mathcal{X}$ with elements of $\{0, 1\}$, where black = 1 and white = 0:

$$\begin{aligned} \mathbf{X} &\in \mathcal{X} = \{-1, 1\}^{256}, \\ \mathbf{X} &= -2\mathcal{X} + \mathbf{I}; \\ \mathcal{X} &= \{x_1, \dots, x_{256}\}, \mathcal{X} \in \{0, 1\}^{256} \end{aligned} \quad (8.7)$$

Note that $\mathbf{I}$ denotes the identity vector. We refer to the un-compressed vector $\mathcal{X}$ rather than $\mathbf{X}$ throughout the remainder of this chapter, trusting that the resulting mathematical simplifications are worth the sleight-of-hand in (8.7).

Given $\mathcal{X}$ described above, we can characterize a noise-corrupted version of it, in which pixels are independently and randomly inverted; the expression models the noise source as an additive, Bernoulli-distributed random vector

$$\boldsymbol{\varsigma} = \{\varsigma_1, \dots, \varsigma_{256}\} \in \{-1, 0\}^{256} \quad (8.8)$$

with diagonal covariance matrix $\Sigma = p_{\varsigma} q_{\varsigma} \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix, and the Bernoulli probabilities p_{ς} and q_{ς} are given by¹⁰

$$\begin{aligned} p_{\varsigma} &= P(\varsigma_i = -1) \\ q_{\varsigma} &= 1 - p_{\varsigma} = P(\varsigma_i = 0) \end{aligned} \quad (8.9)$$

Given this expression for the random noise vector, the noise-corrupted version of $\mathcal{X}$, which we denote by $\boldsymbol{\nu}$, is given by

$$\begin{aligned} \boldsymbol{\nu} &= |\mathcal{X} + \boldsymbol{\varsigma}| \\ \text{s.t. } \nu_i &= |x_i + \varsigma_i| \quad \forall i \end{aligned} \quad (8.10)$$

In simple terms, all 256 noise vector elements are independent and identically distributed (i.i.d.) Bernoulli random variables: the probability of pixel inversion is p_{ς} , and, by (8.8) and (8.10), when $\varsigma_i = -1$ the noise-corrupted pixel ν_i is the inverse of its un-corrupted counterpart x_i :

$$\nu_i = \begin{cases} x_i, & \varsigma_i = 0 \\ 1, & \varsigma_i = -1 \cap x_i = 0 \\ 0, & \varsigma_i = -1 \cap x_i = 1 \end{cases} \quad (8.11)$$

¹⁰Note: a Bernoulli random variable that assumes the value -1 with probability p and the value 0 with probability $q = 1 - p$ will have a mean value of $-p$ and a variance $pq = p(1 - p)$. See [28, pg. 244] for an example of the first and second moment computations underlying these expressions.

In order to derive a simple expression for the SNR of the noise-corrupted feature vector, we make the simplifying assumption that each element of $\mathcal{X}$ is itself a Bernoulli random variable when considered over the set of all 10 digits, and that all the elements of $\mathcal{X}$ are i.i.d. This assumption is of course invalid, but we argue that it is permissible for the purpose of computing a simple measure of the noise-corrupted images' SNR. Based on the statistics of the un-corrupted DB1 database, the probability of a black pixel p_x is 0.30, and the probability of a white pixel q_x is 0.70:

$$\begin{aligned} p_x &= \underbrace{P(x_i = 1)}_{\text{black pixel}} = 0.3 \\ q_x &= 1 - p_x = \underbrace{P(x_i = 0)}_{\text{white pixel}} = 0.7 \end{aligned} \quad (8.12)$$

As a result, we can express the SNR of the noise-corrupted DB1 database in terms of the variances of the Bernoulli random variables x_i and ς_i thus:

$$\begin{aligned} \text{SNR} &= 10 \log_{10} \left(\frac{\text{Var}[x_i]}{\text{Var}[\varsigma_i]} \right) \text{ dB} \\ &= 10 \log_{10} \left(\frac{p_x q_x}{p_\varsigma q_\varsigma} \right) \text{ dB} \\ &= 10 \log_{10} \left(\frac{0.21}{p_\varsigma q_\varsigma} \right) \text{ dB} \end{aligned} \quad (8.13)$$

8.5.2 The Compressed Noisy Feature Vector is Approximately Homoscedastic Gaussian

Using the compression scheme of section M.3, we compress four neighboring binary pixels into one 5-state pixel. In so doing, the (16×16) image is compressed to an (8×8) image. Consider a single cluster of four noise-corrupted pixels $\{\nu_1, \nu_2, \nu_3, \nu_4\}$ in the original 256-pixel image. This group of four pixels forms a single pixel ν' in the compressed image by the following transformation:

$$\nu' = \frac{1}{4} \sum_{i=1}^4 \nu_i = \frac{1}{4} \sum_{i=1}^4 |x_i + \varsigma_i| \quad (8.14)$$

Sixty-four of these compressed pixels form the compressed noise-corrupted feature vector $\mathcal{V}'$. We omit a subscript when discussing individual pixels (i.e., individual elements of $\mathcal{V}'$), opting instead for the generic pixel notation ν' in (8.14). We do this in the interest of notational simplicity, assuming that the reader understands the relationship between the vector $\mathcal{V}'$ and one of its 64 constituent pixels ν' .

If we consider that the value of a pixel (black or white) in the original un-corrupted 256-pixel binary image depends on the digit (i.e., the class that the image represents), we can express (8.14) in its class-conditional form:

$$\nu'|\omega_j = \frac{1}{4} \sum_{i=1}^4 \nu_i|\omega_j = \frac{1}{4} \sum_{i=1}^4 |x_i|\omega_j + \varsigma_i| \quad (8.15)$$

In fact it is legitimate to consider the i th pixel x_i in the original 256-pixel un-corrupted binary image as a Bernoulli-distributed random variable with a class-conditional parameter $p_{x_i|\omega_j}$. That is, for a given digit ω_j , x_i will be black with probability $p_{x_i|\omega_j}$ and white with probability $q_{x_i|\omega_j} = 1 - p_{x_i|\omega_j}$, such that the pdf of x_i , given ω_j , is¹¹

$$\rho_{x_i|\omega_j}(x_i|\omega_j) = q_{x_i|\omega_j} \delta(x) + p_{x_i|\omega_j} \delta(x - 1), \quad (8.16)$$

where δ denotes the Dirac delta function (e.g., [80, pg. 266]). From (8.8) and (8.9) the pdf of the noise vector element ς_i is

$$\rho_{\varsigma}(\varsigma_i) = p_{\varsigma} \delta(\varsigma + 1) + q_{\varsigma} \delta(\varsigma) \quad (8.17)$$

Since x_i and ς_i are independent, the pdf of $x_i + \varsigma_i$ is the convolution of their individual pdfs (e.g., see [25, pg. 36]). As a result, the pdf of $\nu_i|\omega_j$ is, by (8.10), (8.16), and (8.17),

$$\rho_{\nu|\omega_j}(\nu_i|\omega_j) = \underbrace{\left(p_{\varsigma} p_{x_i|\omega_j} + q_{\varsigma} q_{x_i|\omega_j} \right)}_{p_{\nu_i|\omega_j}} \delta(\nu) + \underbrace{\left(p_{\varsigma} q_{x_i|\omega_j} + p_{x_i|\omega_j} q_{\varsigma} \right)}_{q_{\nu_i|\omega_j}} \delta(\nu - 1) \quad (8.18)$$

Equation (8.18) shows that ν_i is itself a Bernoulli-distributed random variable with parameter $p_{\nu_i|\omega_j}$.

Although all the ς_i are independent, the x_i are not. The hand-strokes that generated the original DBI images induced a fair amount of spatial correlation in the pixels. We make the incorrect but simplifying assumption that the pixels are indeed independent so that we can derive a tractable approximation to the pdf of the compressed, class-conditional noise-corrupted pixel $\nu'|\omega_j$. By assuming statistical independence among the original class-conditional constituent pixels $x_1|\omega_j, \dots, x_4|\omega_j$, we can express the class-conditional pdf $\rho_{\nu'|\omega_j}(\nu'|\omega_j)$ as the convolution of $\rho_{\nu_1|\omega_j}(\nu_1|\omega_j), \dots, \rho_{\nu_4|\omega_j}(\nu_4|\omega_j)$, where $*$ denotes the convolution operator:

¹¹Since x_i is countable, it has a probability mass function (pmf) rather than a probability density function (pdf). We consider the pmf a special case of the pdf in which the pdf is expressed as a sum of Dirac delta functions, thus the expression in (8.16).

$$\rho_{\nu'|\mathcal{W}}(\nu'|\omega_j) \cong \rho_{\nu|\mathcal{W}}(\nu_1|\omega_j) * \rho_{\nu|\mathcal{W}}(\nu_2|\omega_j) * \rho_{\nu|\mathcal{W}}(\nu_3|\omega_j) * \rho_{\nu|\mathcal{W}}(\nu_4|\omega_j) \quad (8.19)$$

$$\begin{aligned} &= \prod_{i=1}^4 p_{\nu_i|\omega_j} \delta(\nu') + \sum_{k=1}^4 \left(\prod_{\substack{i=1 \\ i \neq k}}^4 p_{\nu_i|\omega_j} q_{\nu_k|\omega_j} \right) \delta(\nu' - \frac{1}{4}) \\ &\quad + \left(\begin{aligned} &p_{\nu_1|\omega_j} p_{\nu_2|\omega_j} q_{\nu_3|\omega_j} q_{\nu_4|\omega_j} + p_{\nu_2|\omega_j} p_{\nu_3|\omega_j} q_{\nu_1|\omega_j} q_{\nu_4|\omega_j} \\ &+ p_{\nu_3|\omega_j} p_{\nu_4|\omega_j} q_{\nu_1|\omega_j} q_{\nu_2|\omega_j} + p_{\nu_1|\omega_j} p_{\nu_3|\omega_j} q_{\nu_2|\omega_j} q_{\nu_4|\omega_j} \\ &+ p_{\nu_1|\omega_j} p_{\nu_4|\omega_j} q_{\nu_2|\omega_j} q_{\nu_3|\omega_j} + p_{\nu_2|\omega_j} p_{\nu_4|\omega_j} q_{\nu_1|\omega_j} q_{\nu_3|\omega_j} \end{aligned} \right) \delta(\nu' - \frac{1}{2}) \\ &\quad + \sum_{k=1}^4 \left(\prod_{\substack{i=1 \\ i \neq k}}^4 p_{\nu_k|\omega_j} q_{\nu_i|\omega_j} \right) \delta(\nu' - \frac{3}{4}) + \prod_{i=1}^4 p_{\nu_i|\omega_j} \delta(\nu' - 1) \end{aligned} \quad (8.20)$$

Equation (8.20) is, in fact, a kind of binomial pdf (or pmf, strictly speaking) for the noisy compressed class-conditional pixel $\nu'|\omega_j$. The expression reduces to the familiar binomial form ($n = 4, p = p_{x_i|\omega_j}$) if all the $p_{x_i|\omega_j}$ in (8.16) — and, as a result, all the $p_{\nu_i|\omega_j}$ in (8.20) — are equal. The central limit theorem assures us that the sum of a large number of independent random variables will have a pdf that is approximately Gaussian. Indeed, the DeMoivre — Laplace approximation can be viewed as a special expression of the central limit theorem, by which binomial-distributed random variables are shown to be very nearly Gaussian. The approximation is fairly good when the number of Bernoulli trials giving rise to the binomial distribution is $\mathcal{O}[10]$ or greater and the binomial parameter $p \approx \frac{1}{2}$ (e.g., [63, pg. 186]). Note that since our compressed image pixel is the sum of four binary image pixels (i.e., the sum of a total of four Bernoulli trials), the pdf $\rho_{\nu'|\mathcal{W}}(\nu'|\omega_j)$ is an increasingly good approximation to a Gaussian random variable as the noise probability p_c approaches $\frac{1}{2}$ (this corresponds to a signal-to-noise ratio of $\text{SNR} \rightarrow -0.8$ dB). Again, the goodness of the Gaussian approximation is diminished somewhat by the invalid assumption that the original image pixels are independent; nevertheless, the approximation proves to be reasonably good as the image SNR drops below 2 dB (i.e., when $p_c > 0.13$).

Figure 8.22 illustrates the pdf $\rho_{\nu'|\mathcal{W}}(\nu'|\omega_j)$: the black arrows denote the Dirac delta functions of the expression in (8.20) when the probability of pixel inversion is $p_c = 0.2$ and $p_{\nu_i|\omega_j}$ is assumed to be 0.3 for all i . Under these conditions the OCR image signal to noise ratio is $\text{SNR} = 1.2$ dB, and $\rho_{\nu'|\mathcal{W}}(\nu'|\omega_j)$ is indeed a good approximation to the Gaussian pdf shown in light gray.

As the SNR drops below 2 dB, the compressed image pixels are increasingly independent of one another, since the noise statistics begin to dominate the image. Under these circumstances, the noise-corrupted class-conditional feature *vectors*, which we denote by $\boldsymbol{\nu}'|\omega_j$ ($j = 1, \dots, 10$), become homoscedastic

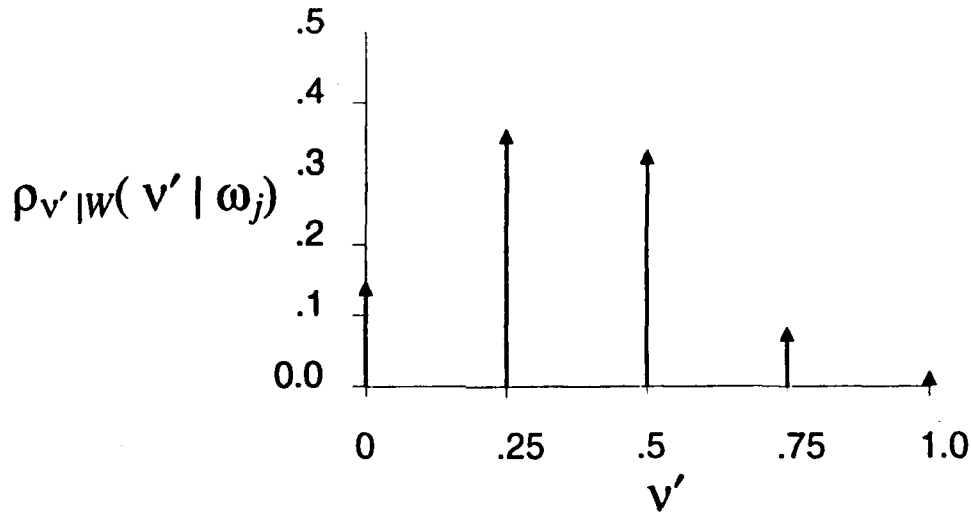


Figure 8.22: The probability density function $\rho_{v'|w}(v' | \omega_j)$ (shown with black arrows denoting the Dirac delta functions of the pdf) for the noisy compressed DB1 image pixel $v' | \omega_j$ when the probability of pixel inversion is $p_c = 0.2$, so that the OCR image signal to noise ratio is $\text{SNR} = 1.2$ dB. For the purpose of this particular graphic, it is assumed that each noise-free pixel in the four being compressed ($\{x_1, \dots, x_4\}$) has a 30% probability of being black (i.e., $p_{v_i | \omega_j}$ is assumed to be 0.3 for $i = 1, \dots, 4$). The gray-shaded background is the Gaussian pdf that $\rho_{v'|w}(v' | \omega_j)$ approximates.

Gaussian-distributed. They satisfy the conditions of appendix F such that the logistic linear hypothesis class constitutes the partially-parametric proper model of $\mathcal{V}'$. This phenomenon is demonstrated in the experiments that follow.

8.5.3 Recognition Results for a Moderate SNR

Figure 8.23 shows moderately noise-corrupted versions of the compressed DB1 digits shown in figure 8.5. The images are arranged in a random order so that the reader has no order-based cues for recognizing the digits. The compressed images are derived from the original 256-pixel binary images after the latter have been corrupted by a noise source with $p_c = 0.1$. That is, the binary pixels of the original images are inverted (or “flipped”) with probability 0.1, a moderate amount of noise. By (8.13), the noisy image SNR is therefore 3.7 dB. The noise-corrupted 256-pixel images — from which those in figure 8.23 are formed by 64 compression operations of the form in (8.14) — are shown in figure 8.33 (page 269). The sequence of digits in figure 8.33 is different from that in figure 8.23. Both sequences are given on page 270, although we ask the reader to indulge us by not peeking at the answers for a while.

Figure 8.24 shows the parameters of the logistic linear classifier generated by differential learning from the first of the 25 moderately noise-corrupted training samples. The parametric entropy of all the classifier's parameters is 3.46, versus 3.18 for the parameters of the logistic linear classifier differentially generated from the noise-free benchmark training sample (cf. figures 8.24 and 8.6, page 228). The increased parametric

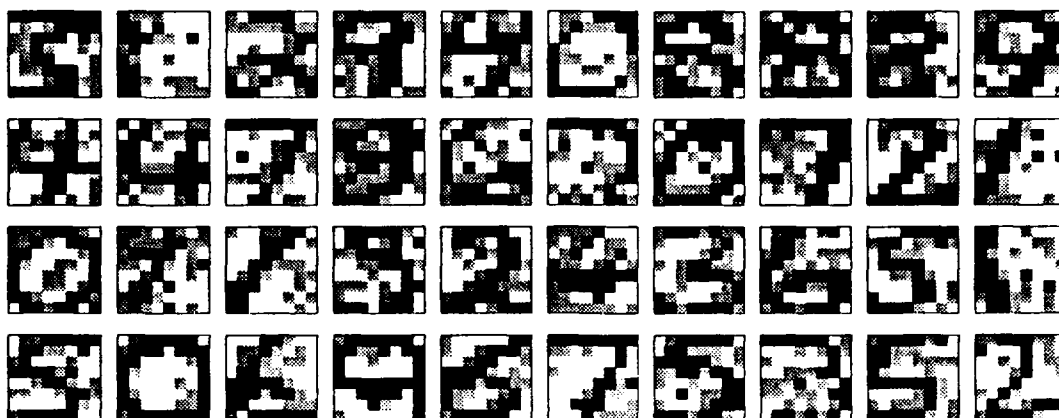


Figure 8.23: Moderately noisy versions of the digits shown in figure 8.5. The order of the images has been randomized so that the digits they represent cannot be inferred from their position on the page. The correct labels for these examples are shown in figure 8.35. We encourage the reader to classify these images prior to looking at the correct labels. Figure 8.33 shows the original 256-pixel noisy images from which these linearly compressed images were derived (the image sequence is different).

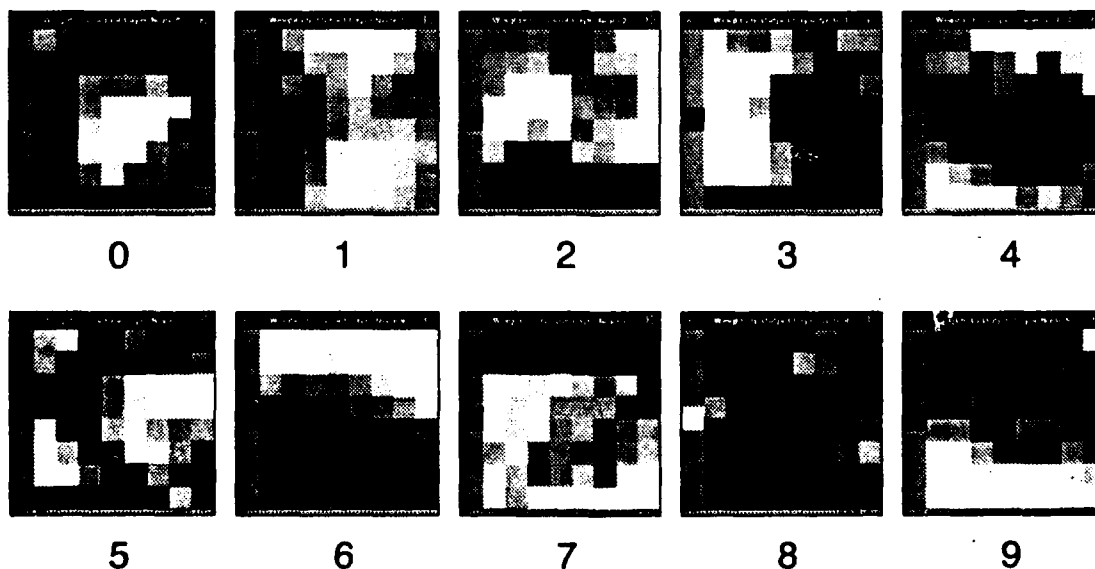


Figure 8.24: Parameters of the differential logistic linear classifier generated from the first of 25 moderately noisy DB1 database training samples. Each of the 50 samples (25 training and 25 test, each containing approximately 600 randomly selected examples) is corrupted with a different realization of the moderate-intensity noise source. The entropy of these parameters is 3.46, reflecting the increased entropy of the noise-corrupted images versus the original images (cf. figure 8.6).

entropy reflects the increased information content of the training sample; the information increase stems from the noise added to the images. Figure 8.25 summarizes the empirical test sample error rates for classifiers generated from the logistic linear hypothesis class by differential and probabilistic learning. The statistics are obtained from the same 25 independent randomly-generated partitions of the DB1 database used in the earlier noise-free experiments. The noise corruption for each of the 25 different training/test samples is independent of that for any and all other trials. Figure 8.25 shows that the differentially-generated logistic linear classifier is somewhat more efficient than its probabilistically-generated counterparts for this nominal training sample size and this moderate amount of noise corruption ($\text{SNR} = 3.7 \text{ dB}$). The average empirical test sample error rate is 6.6% for the differentially-generated classifier, versus 7.0% (MSE) and 7.5% (CE) for the probabilistically-generated classifiers. All classifiers exhibit about the same empirical discriminant variance ($\sim 1.2 \times 10^{-4}$). The right-hand side of figure 8.25 shows that, although the differentially-generated classifier's empirical MSDE is lower than the probabilistically-generated classifiers', differential learning does not *consistently* generate the classifier with the lowest empirical test sample error rate. Probabilistic learning via MSE generates a classifier with a lower empirical test sample error rate for about one third of all trials, and probabilistic learning via CE generates a classifier with a lower empirical test sample error rate for about one quarter of all trials. The SNR of 3.7 dB resulting from the noise-corruption has altered the class-conditional pdfs of the digits in such a way that they have become better approximations to homoscedastic Gaussian pdfs. As a result, the probabilistically-generated logistic linear classifier can learn the noisy compressed images more efficiently *relative to its differentially-generated counterpart* than it can when the images are noise-free. Specifically, consider the estimated relative efficiency of one learning strategy versus another, given a specific hypothesis class and a specific set of training/test samples:

Definition 8.6 The estimated relative efficiency of one learning strategy versus another: *The estimated relative efficiency (ERE or $\widehat{\text{RE}}$ [$\Lambda, \Lambda' | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)$]) of one learning strategy Λ versus another Λ' , given a specific hypothesis class $\mathbf{G}(\Theta)$ and a specific set of K training/test samples of sizes $\{n_1, \dots, n_K\}$ and $\{n'_1, \dots, n'_K\}$ respectively, is simply the ratio of their estimated MSDEs (definition 8.5):*

$$\widehat{\text{RE}} [\Lambda, \Lambda' | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] \triangleq \frac{\widehat{\text{MSDE}} [\mathcal{G} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda]}{\widehat{\text{MSDE}} [\mathcal{G} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda']} \quad (8.21)$$

For $K = 25$ learning and testing trials (sample sizes are $n_i \approx n'_i \approx 600$ $i = 1, \dots, 25$) with the noise-free compressed digits and the logistic linear hypothesis class $\mathbf{G}(\Theta)$, the ERE of differential learning (Λ_Δ) versus probabilistic learning via CE ($\Lambda_{\text{P,CE}}$) is $\widehat{\text{RE}} [\Lambda_\Delta, \Lambda_{\text{P,CE}} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] =$

0.26.¹² The ERE of differential learning versus probabilistic learning via MSE (Λ_{P-MSE}) is $\widehat{RE}[\Lambda_{\Delta}, \Lambda_{P-MSE} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 0.38$. These ERE figures are shown in table 8.5, and they indicate that the differentially-generated logistic linear classifier is three to four times as efficient as its probabilistically-generated counterparts for a nominal, noise-free training sample size of $n \approx 600$ compressed digits.

Estimating the Bayes error rate for the noisy, compressed images: *There is strong evidence that the Bayes error rate for the noise-free, compressed images is indistinguishable from zero. However, we are merely guessing the Bayes error rate for the noisy, compressed images; it is important that the reader understand this, because our guess affects the EREs that we quote below. We assume that $\hat{P}_e(\mathcal{F}_{Bayes}) = 5\%$ for the moderately noisy, compressed images; we assume that $\hat{P}_e(\mathcal{F}_{Bayes}) = 12\%$ for the very noisy, compressed images described in section 8.5.4. These estimates correspond to the lowest empirical test sample error rate exhibited by any classifier in any of the 25 learning/testing trials run at each SNR. Clearly, if the estimates are low, then our MSDE estimates based on them will be high; if the estimates are high, then our MSDE estimates based on them will be low. Bias in our MSDE estimates will, of course, introduce bias in our ERE estimates. In simple terms, if we have over-estimated the Bayes error rate, our ERE values will be biased away from a value of unity; that is, all MSDE estimates will be lower than their actual values, a phenomenon that exaggerates the difference between learning strategy efficiencies. If we have under-estimated the Bayes error rate, our ERE values will be biased towards a value of unity; that is, all MSDE estimates will be higher than their actual values, a phenomenon that obscures any differences between learning strategy efficiencies. We believe that our Bayes error rate estimates are reasonable; nevertheless, we urge the reader to interpret the resulting EREs conservatively.*

If we assume that the estimated Bayes error rate is 5% for the moderately noisy compressed images (we stress that this number is a guess), the resulting EREs are $\widehat{RE}[\Lambda_{\Delta}, \Lambda_{P-CE} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 0.48$ and $\widehat{RE}[\Lambda_{\Delta}, \Lambda_{P-MSE} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 0.75$ (see table 8.5). These ERE figures indicate that the differentially-generated logistic linear classifier is between 1.3 and 2 times as efficient as its probabilistically-generated counterparts for a nominal training sample size of 600 moderately noisy, compressed digits. Note that the ERE of differential learning, given the logistic linear hypothesis class, has increased with decreasing SNR — evidence that the compressed noisy images are more Gaussian-like (and, as a result, more properly modeled by the logistic linear hypothesis class) than the noise-free compressed images were.

¹²We remind the reader that the estimated Bayes error rate $\hat{P}_e(\mathcal{F}_{Bayes})$ for the noise-free compressed images is 0%.

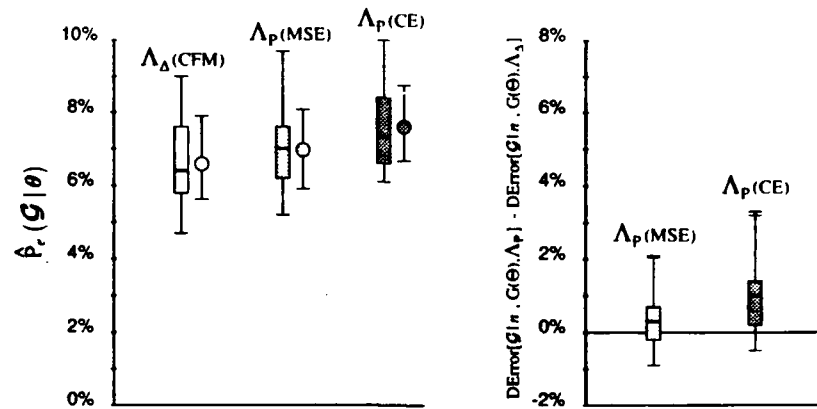


Figure 8.25: **Left:** Test sample classification summaries for the 650-parameter logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). The summaries are based on 25 independent trials in which the DB1 database is randomly partitioned into moderately noisy training and test samples, each containing approximately 600 examples. The box plots are a non-parametric depiction of the empirical test sample error rate's distribution over the 25 trials; the whisker plots depict the average empirical test sample error rate plus and minus one standard deviation, thereby characterizing each classifier's MSDE. **Right:** The increase in the discriminant error of the two probabilistically-generated models over the differentially-generated model on a trial-by-trial basis. These box plots show that differential learning does not always generate the classifier with the lowest empirical error rate. This is due to the low (3.7dB) signal-to-noise ratio (SNR) of the examples and the compression scheme we employ (see figure 8.23): the post-compression pdf of the noisy feature vector becomes increasingly Gaussian-like as the SNR drops, so the probabilistically-generated logistic linear classifier becomes an increasingly good approximation to the proper parametric model of the noisy, compressed digits.

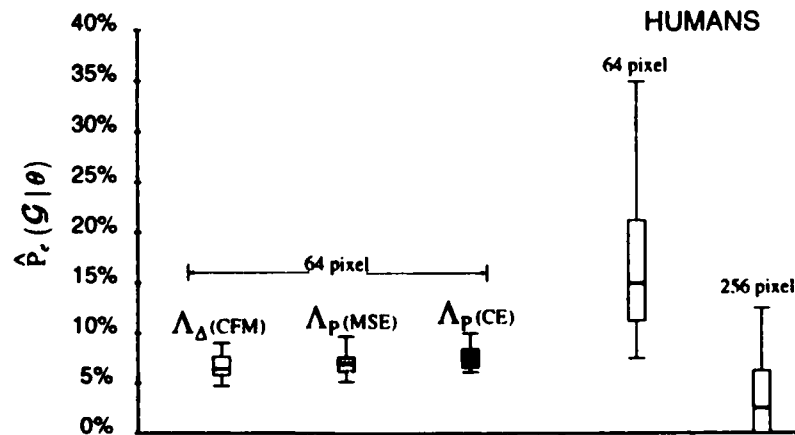


Figure 8.26: **Left:** The test sample classification summaries shown in figure 8.25 for the 650-parameter (64-pixels/digit) logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). **Right:** Classification summaries for fifteen human subjects asked to classify the 40 64-pixel examples shown in figure 8.23. **Far Right:** Classification summaries for fifteen different human subjects asked to classify the 40 256-pixel examples shown in figure 8.33, from which the compressed versions in figure 8.23 are derived.

Estimated Relative Efficiency (ERE) of Differential Learning Λ_{Δ} $n \approx 600$			
Hypothesis	SNR	Alternative Learning Strategy	
Class		Λ_P (MSE)	Λ_P (CE)
Linear	∞ dB ^a	0.12	N/A
	3.7 dB ^b	0.11	N/A
	1.2 dB ^c	0.40	N/A
Logistic Linear	∞ dB ^a	0.38	0.26
	3.7 dB ^b	0.75	0.48
	1.2 dB ^c	2.01	2.28
Modified RBF	∞ dB ^a	0.12	0.16
	3.7 dB	—	—
	1.2 dB	—	—

Table 8.5: The estimated relative efficiency (ERE) — see definition 8.6 — of differential versus probabilistic learning for the linear, logistic linear, and modified Gaussian RBF hypothesis classes (the nominal training sample size is $n \approx 600$ compressed digits). If $ERE < 1$, differential learning generates a more efficient classifier from the hypothesis class than the alternative learning strategy does, given the SNR. If $ERE > 1$, differential learning generates a less efficient classifier from the hypothesis class than the alternative learning strategy does, given the SNR. ERE estimates are based on 25 independent learning/testing trials in which the DB1 database is randomly partitioned into training and test samples, each containing approximately 600 examples. Estimates are shown for all hypothesis classes, given the noise-free digits ($SNR = \infty$). Estimates are shown only for the linear and logistic linear hypothesis classes, given the noisy digits ($SNR = 3.7$ dB and 1.2 dB). Note that the only case in which differential learning does not generate the most efficient classifier for a nominal training sample size of 600 is when the $SNR = 1.2$ dB and the logistic linear hypothesis class is employed. Under these conditions, the logistic linear hypothesis class is a good approximation to the proper parametric model of the very noisy, compressed digits, so the CE-generated logistic linear classifier is a good approximation to the efficient classifier.

^aEstimated Bayes error rate (i.e., $\hat{P}_e(\mathcal{F}_{Bayes})$) is assumed to be 0% for the noise-free compressed images.

^bEstimated Bayes error rate is assumed to be 5% for the moderately noisy compressed images.

^cEstimated Bayes error rate is assumed to be 12% for the very noisy compressed images.

Human Recognition of the Moderately Noisy Images

Figure 8.26 replicates the box plots in the left-hand side of figure 8.25 alongside box plots that summarize the empirical error rates of fifteen human subjects. The human subjects were asked to classify the forty images of figure 8.23, and their empirical error rates were computed according to (8.1) using $\eta = 40$. The human experiment was not rigorously matched against the machine experiments. The humans were *not* given any training examples; instead they relied solely on their prior knowledge of digit forms to perform the classification task. Also, the humans made their classifications by viewing a laser-printed version of

the 40 examples in figure 8.23, whereas the machine learned and subsequently recognized numeric feature vectors. Presumably the printed character images do not contain the same information as their numeric representations, owing to non-linearities in the production and perception of the gray-scale representations of the compressed feature vector. In short, the experiment was almost surely biased against the human subjects. These handicaps notwithstanding, all subjects were electrical and computer engineering graduate students, who, tending to be fundamentally insecure over-achievers, were motivated to perform well on the task.¹³

Fifteen subjects classified the 64-pixel images in figure 8.23, and fifteen different subjects classified the 256-pixel "parent" images in figure 8.33. Figure 8.26 (right) shows that the median empirical test sample error rate for humans was 15% for the compressed images — approximately twice the differentially-generated logistic linear classifier's median rate. Moreover, the human subjects' discriminant variance was more than an order of magnitude higher than the logistic linear classifiers', as indicated by the comparative spans of the box plots for the compressed image experiments. At this point, we encourage the reader to classify the images in figures 8.23 and 8.33; then determine your empirical error rates by comparing your classifications with the answers in figures 8.35 and 8.37.

Note that the human subjects who classified the un-compressed noisy images had a much lower median empirical error rate of 2.5% than the median human rate for the compressed images (figure 8.26, far right), a phenomenon that we discuss further in section 8.5.4.

Learning and Recognizing the Moderately Noisy Images with the Linear Hypothesis Class

Figure 8.27 summarizes the empirical test sample error rates for classifiers generated from the linear hypothesis class by differential and probabilistic learning. The statistics are obtained from the same 25 independent randomly-generated partitions of the DB1 database used in the logistic linear experiments. Figure 8.27 shows that the differentially-generated linear classifier is more efficient than its probabilistically-generated counterpart for this moderate amount of noise corruption (SNR = 3.7 dB). The average empirical test sample error rate is 6.1% for the differentially-generated classifier, versus 9.7% for the probabilistically-(MSE)-generated classifier. Both classifiers exhibit about the same empirical discriminant variance ($\sim 1.3 \times 10^{-4}$). The right-hand side of figure 8.27 shows that the differentially-generated classifier *consistently* generates the classifier with the lowest empirical test sample error rate. Probabilistic learning via MSE generates a classifier with an empirical test sample error rate that is typically about 1.3 times that of its differentially-generated counterpart. The MSE-generated classifier's empirical MSDE is about 10 times the differentially-generated classifier's. This is reflected in the differential learning strategy's ERE, which is $\widehat{\text{RE}}[\Lambda_{\Delta}, \Lambda_{\text{P-MSE}} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 0.11$ for the linear hypothesis class and the 3.7 dB SNR.

¹³We were surprised by the number of subjects who took the experiment as a real challenge and wanted to know how well they had done relative to the whole subject population. When told that the machine did far better than they did, these subjects seemed uniformly relieved to know that good performance on the task didn't require "real" intelligence.

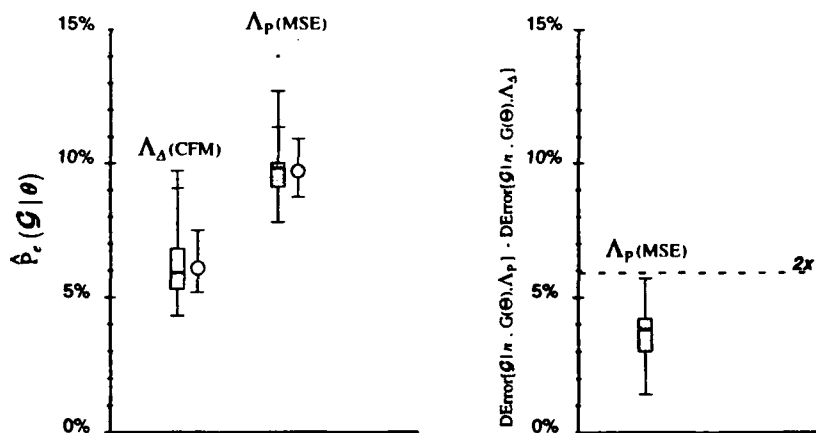


Figure 8.27: **Left:** Test sample classification summaries for the 650-parameter linear classifier employing differential learning (Λ_Δ) and the MSE form of probabilistic learning (Λ_P). The summaries are based on 25 independent trials in which the DB1 database is randomly partitioned into moderately noisy training and test samples, each containing approximately 600 examples. The box plots are a non-parametric depiction of the empirical test sample error rate's distribution over the 25 trials; the whisker plots depict the average empirical test sample error rate plus and minus one standard deviation, thereby characterizing each classifier's MSDE. **Right:** The increase in the discriminant error of the probabilistic model over the differentially-generated model on a trial-by-trial basis. Because the probabilistically-generated linear classifier remains an improper parametric model of the compressed digits with decreasing SNR, differential learning always generates the classifier with the lowest empirical error rate/MSDE.

These results stand in contrast to those for the logistic linear classifier (cf. figures 8.25 and 8.27). The probabilistically-generated linear classifier is consistently worse than its differentially-generated counterpart because the linear hypothesis class remains an improper parametric model of the noisy DB1 feature vector for all SNR values. As a result, differential learning's asymptotic efficiency, which holds for any and all hypothesis classes, generates the relatively efficient linear classifier for these training sample sizes of $n \approx 600$. Experiments were not conducted with the modified Gaussian RBF hypothesis class because it is clearly an improper parametric model of both the noise-free and noisy digits. For this reason, differential learning would surely generate the most efficient RBF classifier, regardless of the SNR, just as it does for the linear hypothesis class.

Finally, note that the test sample error rates, empirical MSDE, etc. for the differentially-generated logistic linear and linear classifiers are virtually identical (cf. figures 8.25 and 8.27). This phenomenon follows the trends we see in the noise-free experiments, and reflects the asymptotic efficiency of differential learning. It generates classifiers with the same empirical MSDE from these two hypothesis classes because both hypothesis classes are capable of forming the same piece-wise linear boundaries on feature vector space, despite their functional differences.

8.5.4 Recognition Results for a Low SNR

Figure 8.28 shows highly noise-corrupted versions of the compressed DB1 digits shown in figure 8.5. Like their moderately noisy counterparts, these images are arranged in a random order so that the reader has no order-based cues for recognizing the digits. The compressed images are derived from the original 256-pixel binary images after the latter have been corrupted by a noise source with $p_c = 0.2$. That is, the binary pixels of the original images are inverted (or “flipped”) with probability 0.2, a high amount of noise. By (8.13), the noisy image SNR is therefore 1.2 dB. The noise-corrupted 256-pixel images — from which those in figure 8.28 are formed by 64 compression operations of the form in (8.14) — are shown in figure 8.34 (page 269). The sequence of digits in figure 8.34 is different from that in figure 8.28. Both sequences are given on page 270; again, we ask the reader to indulge us by not peeking at the answers for a while.

Figure 8.29 shows the parameters of the logistic linear classifier generated by differential learning from the first of the 25 highly noise-corrupted training samples. The parametric entropy of all the classifier’s parameters is 3.85, versus 3.46 for the parameters of the logistic linear classifier differentially generated from the analogous moderately noisy training sample, and 3.18 for the parameters of the logistic linear classifier differentially generated from the noise-free benchmark training sample (cf. figures 8.29, 8.24, page 254, and 8.6, page 228). The increased parametric entropy reflects the increased information content of the training sample stemming from the higher level of noise in the images. Figure 8.30 summarizes the empirical test sample error rates for classifiers generated from the logistic linear hypothesis class by differential and probabilistic learning. The statistics are obtained from the same 25 independent randomly-generated partitions of the DB1 database used in the earlier noise-free and moderate noise experiments, the only difference being the increased level of noise corruption. The noise corruption for each of the 25 different training/test samples is independent of that for any and all other trials. Figure 8.30 and table 8.5 show that the differentially-generated logistic linear classifier is *less* efficient than its probabilistically-generated counterparts for this nominal training sample size and this high amount of noise corruption (SNR = 1.2 dB). The average empirical test sample error rate is 16.0% for the differentially-generated classifier, versus 14.7% (MSE) and 14.4% (CE) for the probabilistically-generated classifiers. The differentially-generated classifier exhibits an empirical discriminant variance of 1.7×10^{-4} , and both probabilistically-generated classifier exhibit an empirical discriminant variance of approximately 1.8×10^{-4} . The right-hand side of figure 8.30 shows that differential learning never generates the classifier with the lowest empirical test sample error rate: instead, probabilistic learning via the Kullback-Leibler information distance (CE) does. This is because the SNR of 1.2 dB resulting from the noise-corruption with $p_c = 0.2$ has altered the class-conditional pdfs of the digits in such a way that they have become reasonably good approximation to homoscedastic Gaussian pdfs. As a result, the CE-generated logistic linear classifier is a good approximation to the proper parametric model of the very noisy compressed feature vector: it can learn the very noisy compressed images more efficiently than its differentially-generated counterpart can (recall section 3.6). Indeed, although the

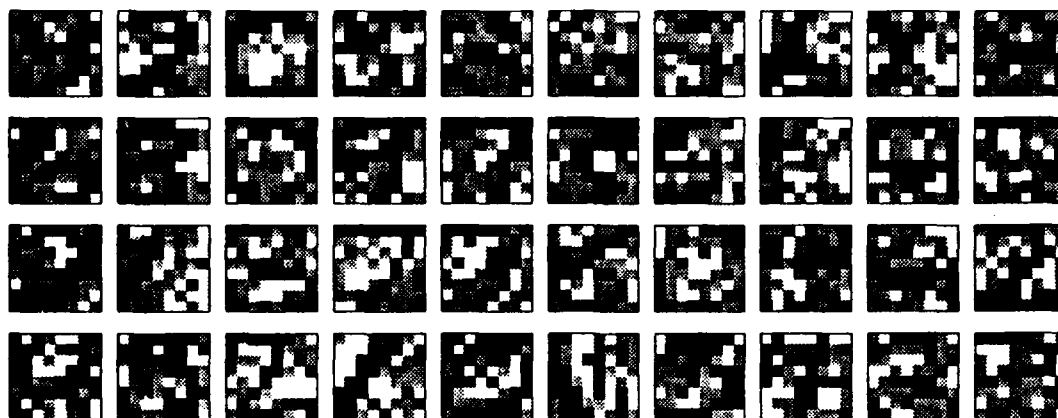


Figure 8.28: Very noisy versions of the digits shown in figure 8.5. The order of the images has been randomized (the sequence differs from those in figures 8.28 and 8.34) so that the digits they represent cannot be inferred from their position on the page. The correct labels for these examples are shown in figure 8.36. We encourage the reader to classify these images prior to looking at the correct labels. Figure 8.34 shows the original 256-pixel noisy images from which these linearly compressed images were derived (the image sequence is different).

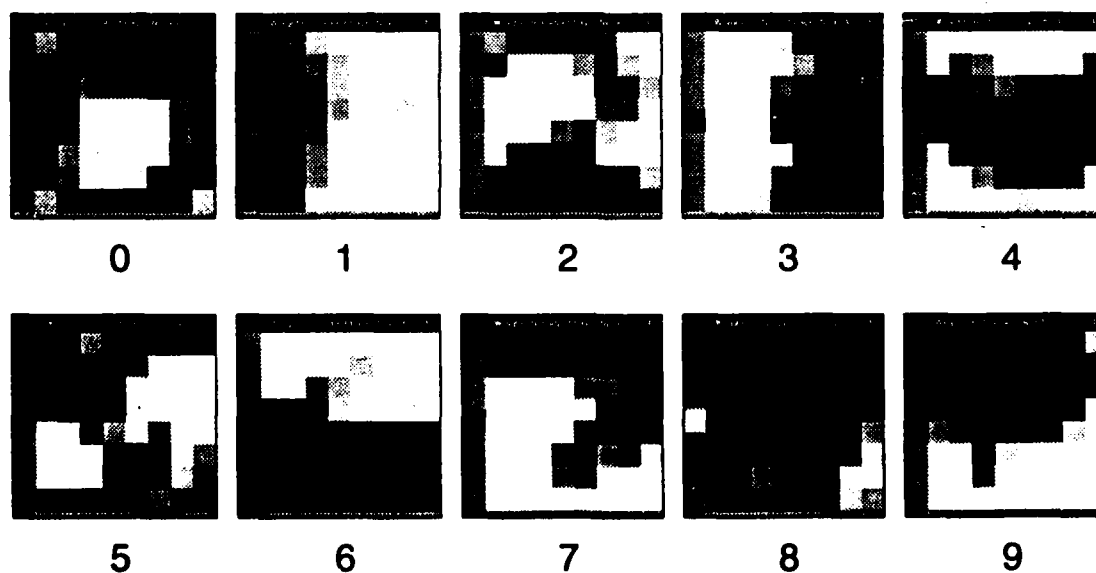


Figure 8.29: Parameters of the differential logistic linear classifier generated from the first of 25 very noisy DB1 database training samples. Each of the 50 samples (25 training and 25 test, each containing approximately 600 randomly selected examples) is corrupted with a different realization of the high-intensity noise source. The entropy of these parameters is 3.85, reflecting the increased entropy of the noise-corrupted images versus those with less or no noise (cf. figures 8.24 and 8.6).

MSE-generated logistic linear classifier is not, strictly speaking, the proper parametric model of the noisy compressed feature vector, it is also a good approximation thereto. As a result, the MSE-generated logistic linear classifier consistently exhibits a lower empirical test sample error rate than the differentially-generated classifier's. Table 8.5 confirms these facts by showing that the differential learning strategy's EREs are $\widehat{\text{RE}}[\Lambda_{\Delta}, \Lambda_{\text{P-MSE}} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 2.01$ and $\widehat{\text{RE}}[\Lambda_{\Delta}, \Lambda_{\text{P-CE}} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 2.28$ for the logistic linear hypothesis class and the 1.2 dB SNR.

As we have stated before, the statistical literature details an extensive collection of hypothesis testing procedures (e.g., see [140]) by which it is possible to determine whether or not a chosen hypothesis class is a good approximation to the proper parametric model of the feature vector. If the improper hypothesis is rejected (i.e., if the parametric model is determined to be proper), then we are justified in using the appropriate form of probabilistic learning and we are justified in expecting that the resulting classifier will be a good approximation to the efficient classifier for small training sample sizes. If, on the other hand, the improper hypothesis is not rejected (i.e., if the parametric model is determined to be improper), then we are better off using differential learning, which is *guaranteed* to produce the relatively efficient classifier for large training sample sizes (and generally does so for small training sample sizes).

Human Recognition of the Very Noisy Images

Figure 8.31 replicates the box plots in the left-hand side of figure 8.30 alongside box plots that summarize the empirical error rates of fifteen human subjects. The human subjects were asked to classify the forty images of figure 8.28, and their empirical error rates were computed according to (8.1) using $\eta = 40$. Like the moderately noisy experiment, the very noisy human experiment was not rigorously matched against the machine experiments, so it was almost surely biased against the human subjects for the reasons described earlier.

Fifteen subjects classified the 64-pixel images in figure 8.28, and fifteen different subjects classified the 256-pixel "parent" images in figure 8.34. Figure 8.31 (right) shows that the median empirical test sample error rate for humans was 37% for the compressed images — more than $2\frac{1}{2}$ times the CE-generated logistic linear classifier's median rate. As in the moderately noisy experiment, the human subjects' discriminant variance was more than an order of magnitude higher than the logistic linear classifiers' (indicated by the comparative spans of the box plots for the compressed image experiments). At this point, we encourage the reader to classify the images in figures 8.28 and 8.34; then determine your empirical error rates by comparing your classifications with the answers in figures 8.36 and 8.38.

Note that the disparity between the error rates of the human subjects who classified the un-compressed noisy images and those who classified the compressed images is substantially less than it was for the moderately noisy images (the median empirical error rate was 30% for the un-compressed high-noise images, versus 37% for the compressed high-noise images: see figure 8.31, far right and right, respectively). Simply

put, when the image SNR is greater than 2 dB, the human subjects distinguish the digits in the un-compressed noisy images more easily than they can in the compressed versions. As the SNR drops to 1.2 dB, the un-compressed images become nearly as hard to recognize as their compressed counterparts.

We are not qualified to ponder whether or not humans have and use proper parametric models for learning — a question that goes well beyond our interest and expertise. We have included these relatively un-controlled human experiments simply to illustrate that the differences among the three logistic linear classifiers are insignificant when compared to the differences between the machine and human experiments. All the machine learning approaches out-classified the human subjects by a substantial margin when the digits were corrupted by large amounts of noise. We find this result interesting in its implications for future comparisons of synthetic (i.e., machine-based) and organic (e.g., human) learning systems.

Learning and Recognizing the Very Noisy Images with the Linear Hypothesis Class

Figure 8.32 summarizes the empirical test sample error rates for classifiers generated from the linear hypothesis class by differential and probabilistic learning. The statistics are obtained from the same 25 independent randomly-generated partitions of the DB1 database used in the high-noise logistic linear experiments. Figure 8.32 and table 8.5 show that the differentially-generated linear classifier is more efficient than its probabilistically-generated counterpart for this high amount of noise corruption (SNR = 1.2 dB). The average empirical test sample error rate is 15.4% for the differentially-generated classifier, versus 17.1% for the probabilistically- (MSE)-generated classifier. Both classifiers exhibit about the same empirical discriminant variance ($\sim 1.6 \times 10^{-4}$). The right-hand side of figure 8.32 shows that the differentially-generated linear classifier *consistently* generates the classifier with the lowest empirical test sample error rate. Probabilistic learning via MSE generates a classifier with an empirical test sample error rate that is typically about 1.8% higher than (or 1.13 times) that of its differentially-generated counterpart. The MSE-generated classifier's empirical MSDE is about $2\frac{1}{2}$ times the differentially-generated classifier's. This is reflected in the differential learning strategy's ERE, which is $\widehat{\text{RE}}[\Lambda_{\Delta}, \Lambda_{\text{P-MSE}} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 0.40$ for the linear hypothesis class and the 1.2 dB SNR.

These results stand in contrast to those for the logistic linear classifier (cf. figures 8.30 and 8.32). The probabilistically-generated linear classifier is consistently worse than its differentially-generated counterpart because, as in the moderately noisy experiments, the linear hypothesis class remains an improper parametric model of the noisy, compressed DB1 feature vector for all SNR values. As a result, differential learning's asymptotic efficiency, which holds for any and all hypothesis classes, generates the relatively efficient linear classifier for these training sample sizes of $n \approx 600$. Again, experiments were not conducted with the modified Gaussian RBF hypothesis class because it is clearly an improper parametric model of both the noise-free and noisy digits — differential learning would surely generate the most efficient RBF classifier.

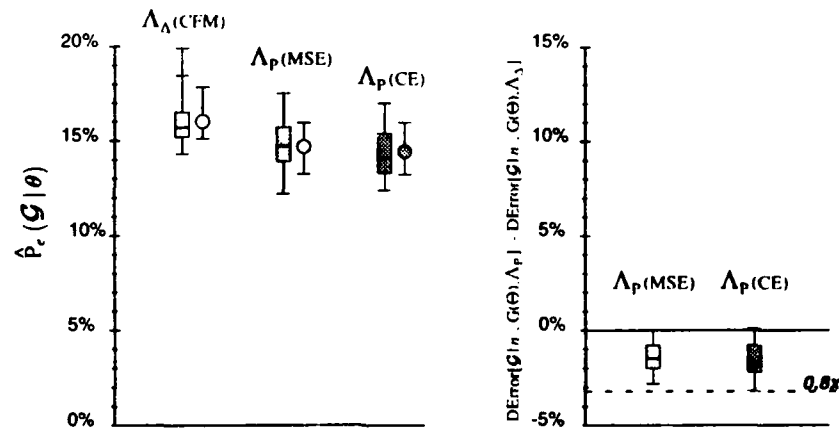


Figure 8.30: **Left:** Test sample classification summaries for the 650-parameter logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). The summaries are based on 25 independent trials in which the DB1 database is randomly partitioned into very noisy training and test samples, each containing approximately 600 examples. The box plots are a non-parametric depiction of the empirical test sample error rate's distribution over the 25 trials; the whisker plots depict the average empirical test sample error rate plus and minus one standard deviation, thereby characterizing each classifier's MSDE. **Right:** The increase in the discriminant error of the two probabilistically-generated models over the differentially-generated model on a trial-by-trial basis. These box plots show that differential learning never generates the classifier with the lowest empirical error rate. This is because the logistic linear hypothesis class employing probabilistic learning approximates the proper parametric model of the very noisy feature vector: the class-conditional pdf for each digit is now dominated by noise ($\text{SNR} = 1.2\text{dB}$) and is approximately Gaussian distributed, owing to the compression scheme we employ (see sections 8.5.1 and 8.5.2).

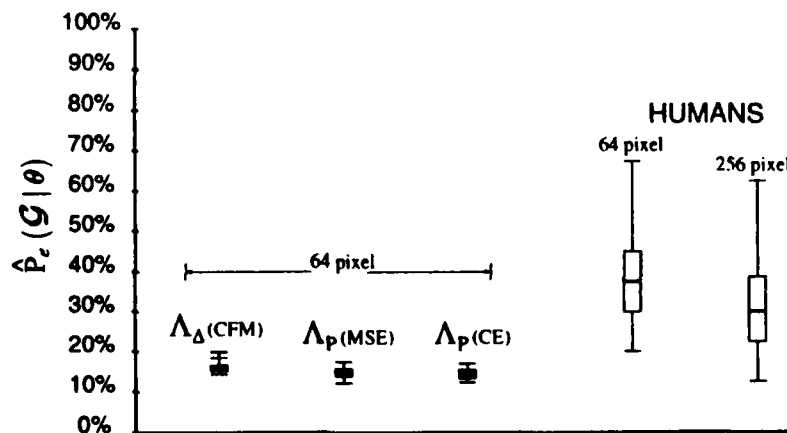


Figure 8.31: **Left:** The test sample classification summaries shown in figure 8.30 for the 650-parameter (64-pixels/digit) logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). **Right:** Classification summaries for fifteen human subjects asked to classify the 40 64-pixel examples shown in figure 8.28. **Far Right:** Classification summaries for fifteen different human subjects asked to classify the 40 256-pixel examples shown in figure 8.28, from which the compressed versions in figure 8.28 are derived.

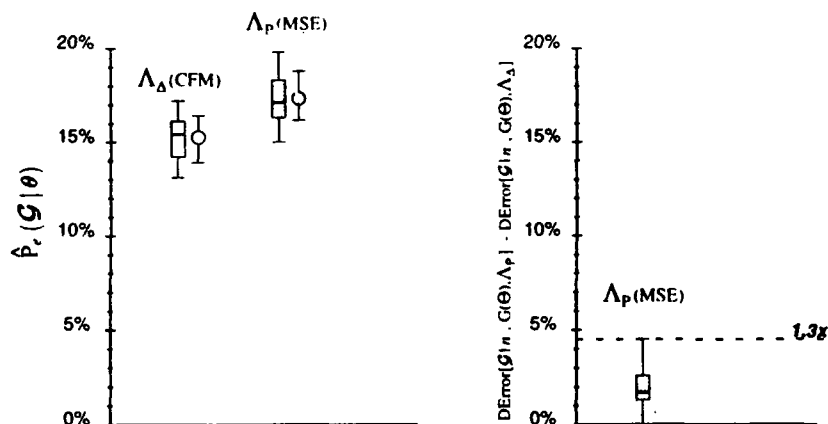


Figure 8.32: **Left:** Test sample classification summaries for the 650-parameter linear classifier employing differential learning (Λ_{Δ}) and the MSE form of probabilistic learning (Λ_P). The summaries are based on 25 independent trials in which the DB1 database is randomly partitioned into very noisy training and test samples, each containing approximately 600 examples. The box plots are a non-parametric depiction of the empirical test sample error rate's distribution over the 25 trials; the whisker plots depict the average empirical test sample error rate plus and minus one standard deviation, thereby characterizing each classifier's MSDE. **Right:** The increase in the discriminant error of the probabilistic model over the differentially-generated model on a trial-by-trial basis. Because the probabilistically-generated linear classifier remains an improper parametric model of the compressed digits with decreasing SNR, differential learning always generates the linear classifier with the lowest empirical error rate. Although differential learning yields the *relatively* efficient classifier for the linear hypothesis class, it does *not* generate the *efficient* classifier. The efficient classifier is approximated by the logistic linear hypothesis class/probabilistic learning (Kullback-Leibler) combination, a proper parametric model approximation that exhibits a slightly lower average empirical error rate (cf. figure 8.30).

Finally, note that the test sample error rates, empirical MSDE, etc. for the differentially-generated linear classifier are all slightly lower (i.e., better) than those for the differentially-generated logistic linear classifier (cf. figures 8.30 and 8.32). This phenomenon is at odds with the trends we see in the noise-free experiments. We suspect (but have not substantiated) that it might relate to the issue of model complexity versus training sample size. That is, the functional complexity of the logistic linear hypothesis class might be somewhat higher than that of the linear hypothesis class. If this is true, then the logistic linear classifier might have excessive functional complexity for a small training sample of very noisy digits, resulting in its slightly higher empirical MSDE.

8.6 Summary

The AT&T DB1 optical character recognition (OCR) task serves to illustrate the theoretical findings of part I. We have shown that compressing the digit images allows us to reduce the complexity of the classifiers we employ in the OCR task. By lowering the classifier complexity we improve the generalization of all classifiers, but the improvement is most pronounced for the differentially-generated classifiers. This is

because differential learning is *guaranteed* to generate the relatively efficient classifier, regardless of the choice of hypothesis class. Although large training sample sizes are necessary to *guarantee* that differential learning will be efficient, the trait generally holds for small training sample sizes as well (i.e., as long as the hypothesis class is an improper parametric model of the data).

Learning to recognize noisy versions of the compressed DBI digits provides us with the conditions under which differential learning is *not* the most efficient learning strategy for small training sample sizes. Because the very noisy compressed digits are nearly Gaussian distributed with homoscedastic class-conditional pdfs, the CE-generated logistic linear classifier constitutes a proper parametric model of the noisy data; probabilistic learning is therefore the more efficient learning strategy. As we have mentioned a number of times already, there are well-known procedures for assessing whether or not the hypothesis class is a proper parametric model of the feature vector, so it is relatively straightforward to detect the circumstances under which differential learning might not be the most efficient strategy. Absent a strong indication to the contrary, differential learning is the prudent, efficient choice.

Readers familiar with Geman, Bienenstock, and Doursat's lovely paper on regression and classification with neural networks [41] (we will refer to the authors as "GBD" henceforth) will recognize our use of their noise protocol to corrupt the DBI digits. Those readers will also note that our noise-free average empirical error rates of 2% and very noisy rates of 14.5% are substantially lower than GBD's, which were approximately 16% and 40%, respectively.¹⁴ Part of the disparity surely stems from their using training sample sizes of only 200: ours were 600. However, we conclude by virtue of our own control experiments with probabilistic learning that much of the disparity stems from their use of probabilistically-generated high-complexity classifiers. Indeed, their typical MSE-generated classifier had $O[6700]$ parameters, whereas ours have only 650 parameters. To be sure, our simple classifiers are generally incapable of modeling the DBI data with low *functional* error, but our objective is merely to model the data with low *discriminant* error — an objective that we achieve consistently well with differentially-generated low-complexity classifiers.

Our results lead us to dispute GBD's implication that a "good representation", determined *a priori* by careful engineering, is a prerequisite for good generalization when the ultimate objective is pattern classification.¹⁵ Instead, we argue that an efficient learning strategy is the most important prerequisite for good generalization. Given this, one can employ simple, potentially *improper* parametric models (i.e., potentially poor representations, as defined by GBD), yet achieve robust generalization — a theoretical fact by the proofs of part I, clearly illustrated by the DBI experiments of this chapter. Indeed, the robust beauty of differentially-generated models is that they need not be proper to yield robust generalization. Thus, one's choice of representation is no longer a critical factor on which the classifier's ability or failure to discriminate

¹⁴Our error rates are for the 650-parameter logistic linear classifier, which is the simplest form of multi-layer perceptron (MLP), one having no hidden layer units. The logistic linear classifier recognizes 64-pixel images. GBD's error rates are for an MLP with 25 hidden layer units, a substantially more complex hypothesis class possessing 6685 parameters. Their MLP recognized 256-pixel images.

¹⁵We wholeheartedly support their implication when the objective is function approximation, as it is in regression tasks. We remind the reader, however, that regression and pattern classification are two very different tasks.

well hinges.

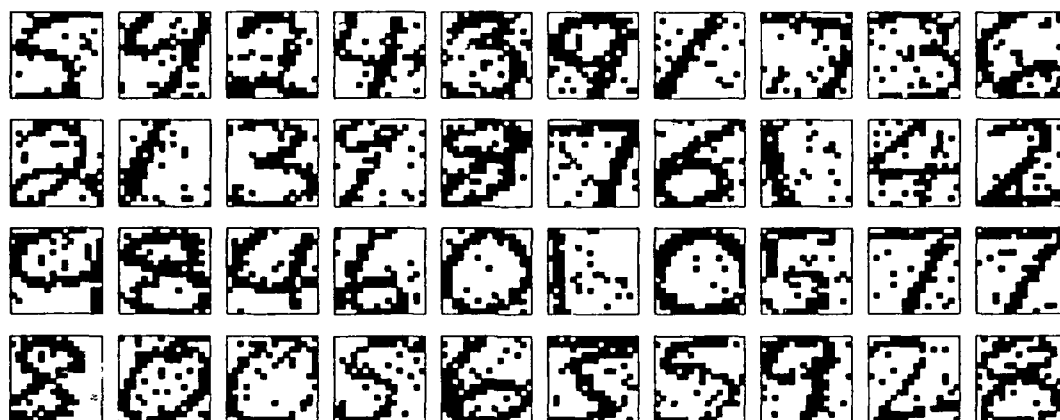


Figure 8.33: Moderately noisy 256-pixel digits formed by flipping the binary pixels of the original DB1 database with probability 0.1. This form of artificial noise corruption is described in [41]. The signal-to-noise ratio (SNR) of these images is 3.7dB. The digits shown in figure 8.23 are generated by linearly compressing these images. The correct labels for the digits are shown in figure 8.37. We encourage the reader to classify these images prior to looking at the correct labels.

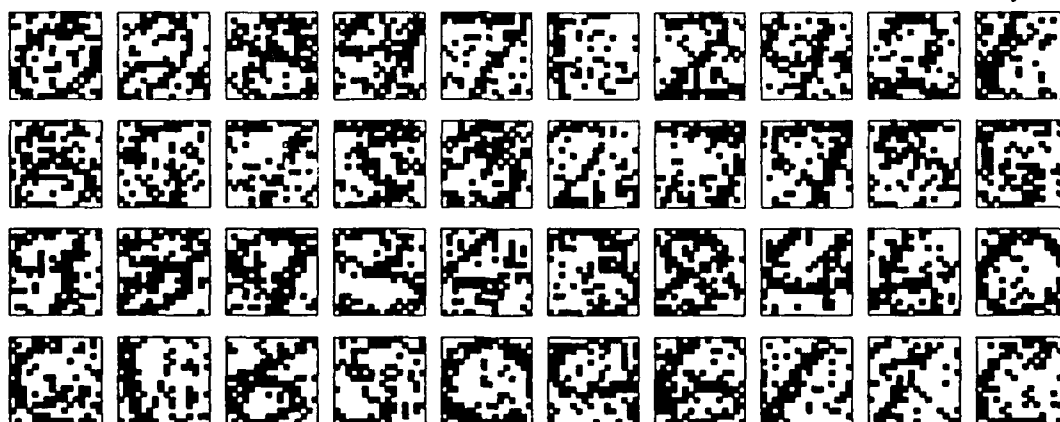


Figure 8.34: Very noisy 256-pixel digits formed by flipping the binary pixels of the original DB1 database with probability 0.2. The signal-to-noise ratio (SNR) of these images is 1.2dB. The digits shown in figure 8.28 are generated by linearly compressing these images. The correct labels for the digits are shown in figure 8.38. We encourage the reader to classify these images prior to looking at the correct labels.

3	1	2	9	4	0	8	8	8	9
4	4	7	8	6	7	0	7	2	1
0	9	1	3	2	4	3	6	5	1
5	0	6	9	6	7	5	3	5	2

Figure 8.35: Correct labels for the digits in figures 8.23.

8	2	0	5	8	6	5	6	9	8
6	8	0	9	7	3	5	1	9	7
9	1	3	7	7	6	0	4	4	2
4	3	2	1	0	1	5	4	2	3

Figure 8.36: Correct labels for the digits in figures 8.28.

5	4	2	4	3	9	1	7	3	6
2	1	3	9	8	7	6	1	4	2
9	8	4	6	0	1	0	5	7	7
8	0	0	5	6	3	5	9	2	8

Figure 8.37: Correct labels for the digits in figures 8.33.

0	2	8	4	7	1	2	4	2	1
8	5	7	3	3	2	7	7	5	5
9	8	9	3	4	3	8	4	6	0
6	1	6	5	0	9	6	1	9	0

Figure 8.38: Correct labels for the digits in figures 8.34.

Chapter 9

Medical Diagnosis with Differential Learning

Outline

We use a simple logistic linear classifier employing differential learning to diagnose avascular necrosis (AVN) of the femoral head, a potentially crippling hip joint disorder. The diagnosis is rendered from magnetic resonance images. Specifically, we repeat the experiments of Manduca, Christy, and Ehman [90]; we compare the diagnostic accuracy of a differentially-generated logistic linear classifier and two probabilistically-generated controls (including the logistic regression model) with their original results. When presented with approximately sixty training images and subsequently evaluated on the same number of test images, the differentially-generated logistic linear classifier discriminates between healthy and AVN compromised femoral heads with a 5.9% error rate. This error rate is slightly lower than the 7.5% error rate of humans without formal training in radiology, reported in [90]. The differentially-generated logistic linear classifier generalizes better than the probabilistic controls and the best previous machine model, a multi-layer perceptron having approximately 24 times the number of parameters (6,164, versus 257 for the logistic linear classifier).

9.1 Introduction

Avascular necrosis of the femoral head is a disease in which the blood supply to the head of the femur is restricted, causing loss of bone marrow. Manduca, Christy, and Ehman have used neural network classifiers to detect the presence of this disorder from magnetic resonance images (MRIs) of 40 adult patients [90]. The image database they generated for the task contains 125 images, 51% of which indicate the presence of AVN. Details of the database are given in [90]. Figure 9.1 shows fourteen examples of these MRI images: each

¹We thank Dr. Armando Manduca of the Mayo Clinic/Foundation for providing us with the magnetic resonance image data for this task along with statistics and insights from the original experiments in [90].

image is a 1024-pixel (32×32) image, and the pixels have four-bits of precision (i.e., they have 16 possible values: dark image pixels represent negative values, and light pixels represent positive values). The image is of the femoral head, the ball at the top of the femur that mates with the hip socket. As figure 9.1 illustrates, the presence of AVN is manifest as dark regions in the light-shaded oval-shaped bone mass. These dark regions indicate the absence of water-containing bone marrow and fat — conclusive evidence of AVN. The dark annular region surrounding the marrow and fat is cortical bone (i.e., the bone's hard outer surface). The surrounding light material is fluid in the joint space; the surrounding dark material is generally cartilage.² It is obvious from these images that there is some variance in the size of the femoral head among subjects. As a result, a correct diagnosis in part hinges on being able to distinguish the cortical bone and cartilage (present in all of the images) from AVN sites.

We begin by learning all 125 images with a simple classifier possessing the single logistic linear discriminant function (section 7.2.2)³

$$g_1(\mathbf{X}|\boldsymbol{\theta}) = \left[1 + \exp(-\mathbf{X}'^T \boldsymbol{\theta})\right]^{-1}. \quad (9.1)$$

where $\mathbf{X}$ is the $N = 1024$ pixel image vector, $\mathbf{X}'$ is the $N + 1 = 1025$ dimensional *augmented* feature vector formed by adding a single element of unit value to $\mathbf{X}$ (see (7.2)), and $\boldsymbol{\theta} \in \Theta = \mathbb{R}^{N+1}$ is the $N + 1$ dimensional parameter or *weight* vector. The single discriminant function $g_1(\mathbf{X}|\boldsymbol{\theta}) \in [0,1]$ is associated with a healthy diagnosis. That is, when $g_1(\mathbf{X}|\boldsymbol{\theta}) > 0.5$, the classifier makes a healthy diagnosis; when $g_1(\mathbf{X}|\boldsymbol{\theta}) \leq 0.5$ the classifier makes an AVN diagnosis. We can cast this single output classifier into the canonical form described in section 2.2.1 simply by imagining a second discriminant function $g_2(\mathbf{X}|\boldsymbol{\theta}) = 1 - g_1(\mathbf{X}|\boldsymbol{\theta})$ associated with an AVN diagnosis.

Figure 9.2 (left) illustrates the parameters (or *weights*) formed when this logistic linear classifier learns to diagnose all 125 images correctly using differential learning. Dark pixels in the left display represent negative weights, and light pixels represent positive weights. The far-left column of the left display contains only one vertically-centered pixel. This pixel represents the “bias” parameter corresponding to the unit-value element prepended to $\mathbf{X}$ in order to form the augmented feature vector $\mathbf{X}'$ of (7.2). The gray shade of the far-left pixel column represents the value zero (for reference). The lighter weights have positive values and correlate with dark AVN-compromised regions in the MRI images. The darker weights have negative values and correlate with dark regions in the MRI images that are common to healthy examples (e.g., cortical bone and cartilage). A visual comparison of figure 9.2 (left) with the AVN-compromised examples in figure 9.1 confirms these relationships. One notable aspect of the 1025 weight image is its low

²We thank Dr. Martha McDaniel, M.D., of the VA Medical Center, White River Junction, VT, and the Dartmouth-Hitchcock Medical Center, Lebanon, NH, for her primer on femoral head MRI interpretation. In truth, the process of image interpretation is complex, performed by experts with extensive training. In addition, the spin sequence used in generating the MRI has a significant impact on how the image is interpreted. Our description of the image composition is therefore a general one.

³We use a single discriminant function for this $C = 2$ -class problem because it has one-half the number of parameters a classifier with two discriminant functions would have. Excess complexity is *anathema*.

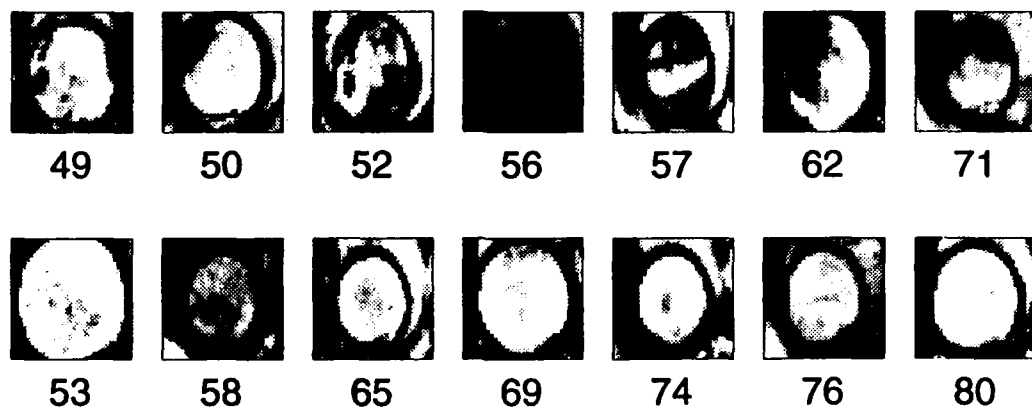


Figure 9.1: Fourteen 1024-pixel examples of healthy (bottom row) and AVN compromised (top row) femoral heads. The number below each image is its index number in the 125-image database described in [90] (our indices run from 0 $\rightarrow$ 124).

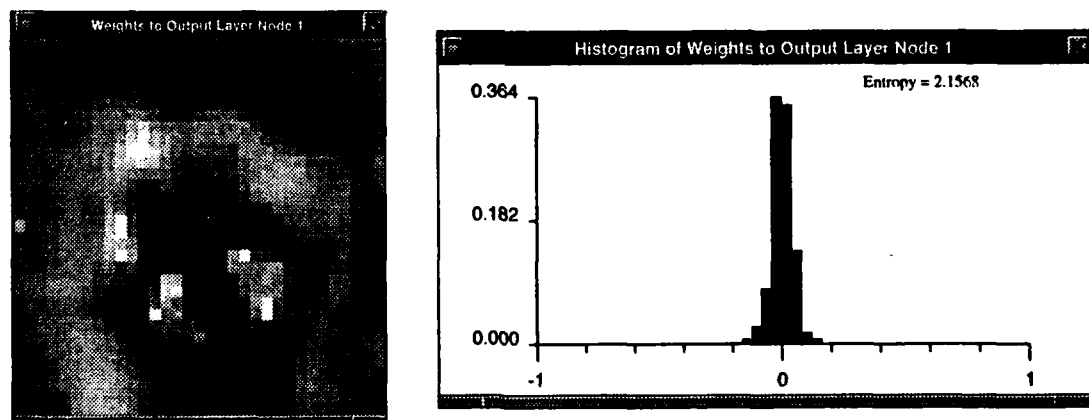


Figure 9.2: **Left:** The parameters of a 1024-pixel logistic linear classifier, obtained by differentially learning all 125 example images. Light parameters (or *weights*) are positive and detect AVN-related dark regions in the image; dark weights are negative and detect dark regions in the image (cortical bone and cartilage) *not* associated with AVN. Weight smoothing (described in section M.2) is applied during learning to minimize the parametric entropy (definition M.1) of the weights (i.e., the amount of information they store). This in turn reduces discriminant variance across learning trials. **Right:** A histogram of the weights in the left figure. Note the parametric entropy of the weights is 2.2, reflecting the low variance in the distribution of weights caused by smoothing. This low variance/parametric entropy accounts for the low-contrast in the weight display on the left.

contrast: the gray scale of weights changes abruptly in only a few regions of the display. This is because the classifier employs weight smoothing (described in section M.2) during learning in order to minimize the *parametric entropy* (definition M.1) of its weight vector, an empirical measure that we use to gauge the weight vector's information content. Weight smoothing reduces the classifier's discriminant variance, since only the information essential to learning is retained. Because the classifier learns all 125 examples with a moderate amount of weight smoothing ($\kappa = 0.05$),⁴ its weight vector's parametric entropy is a relatively low 2.16, which reflects the low variance in the histogram of the weights (figure 9.2, right).

These factors indicate that the *effective number of classifier parameters* [97] is much lower than 1025. Furthermore, they suggest that the degree of image resolution necessary for a correct diagnosis is relatively low. As a result, we generate a database of linearly compressed 256-pixel (16×16) images from the original database of 1024-pixel (32×32) images. The linear lossy compression algorithm is described in section M.3. Figure 9.3 shows the compressed versions of the MRIs in figure 9.1. Figure 9.4 (left) illustrates the weights formed when a 257-parameter⁵ logistic linear classifier learns to diagnose all 125 compressed images correctly using differential learning. This classifier also employs weight smoothing ($\kappa = 0.02$). As with the higher-resolution classifier, the lighter weights have positive values that correlate with dark AVN-compromised regions in the MRI images, and the darker weights have negative values that correlate with dark regions in the MRI images that are common to healthy examples. Figure 9.4 (right) indicates that the 257-element parameter vector has higher parametric entropy than its 1025-element counterpart. This is reflected in the increased variance of the 257 weights' histogram and suggests that the average weight in the lower complexity classifier encodes more discriminant information than the average weight in the higher complexity classifier.

The lower resolution MRIs are definitely harder to learn than the high-resolution images. The 1025-parameter classifier learns all the examples with $\psi^1 \approx 0.3$, whereas the 257-parameter classifier must reduce ψ^1 to 0.07 before it can learn all 125 examples. Figure 9.5 shows the final state of the low-complexity (257-parameter) classifier after learning. All examples lie on the reduced discriminant continuum (i.e., the line between the upper left and lower right corners of reduced discriminator output space) because the classifier has only one output; our earlier specification of the phantom second output ensures this. Note that the hardest examples (including examples 48 and 49) engender very small positive discriminant differentials; the classifier has low confidence in its diagnosis of these examples.

⁴The weight smoothing parameter κ has a value between zero and one (see appendix M). A value of zero results in no smoothing; a value of one forces all weights to have the same value. From a qualitative perspective, any value of $\kappa > 0.1$ is large, any value of $\kappa > 0.01$ is moderate, and any value of $\kappa < 0.01$ is small.

⁵There is one discriminant function, and the augmented compressed feature vector has $N + 1 = 257$ elements. Therefore the classifier has 257 total parameters.

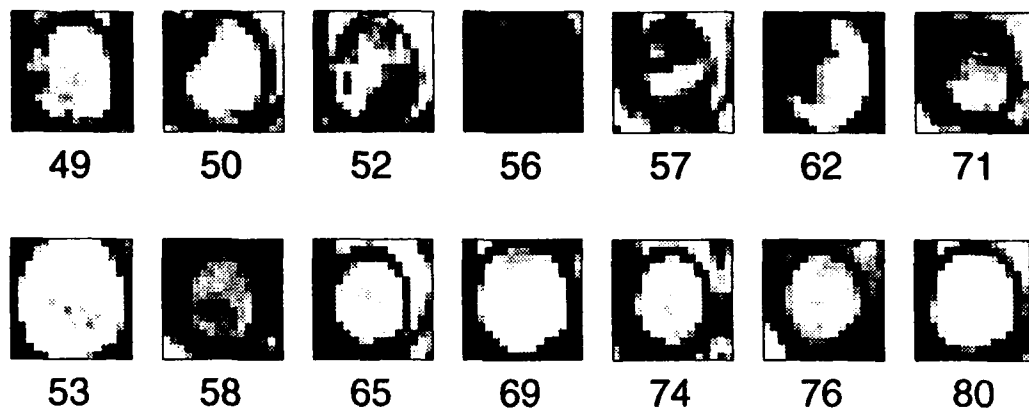


Figure 9.3: The images in figure 9.1, linearly compressed to 256 pixels.

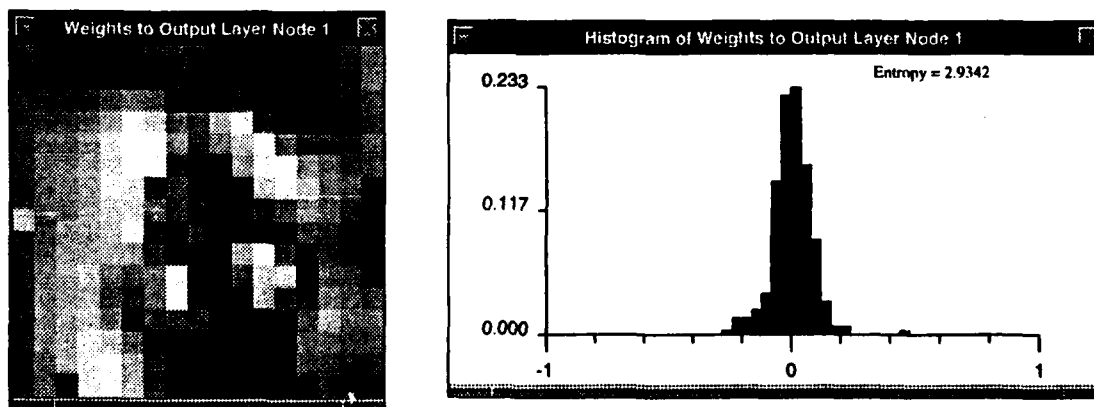


Figure 9.4: **Left:** The parameters of a 256-pixel differentially-generated logistic linear classifier, obtained by learning all 125 example images. Weight smoothing is applied during learning to reduce discriminant variance across learning trials. **Right:** A histogram of the weights in the left figure. Note the parametric entropy of the weights is 2.9, reflecting a moderate increase in the weight distribution's variance (cf. figure 9.2). This increased variance/parametric entropy accounts for the slightly increased contrast in the lower-resolution weight display.

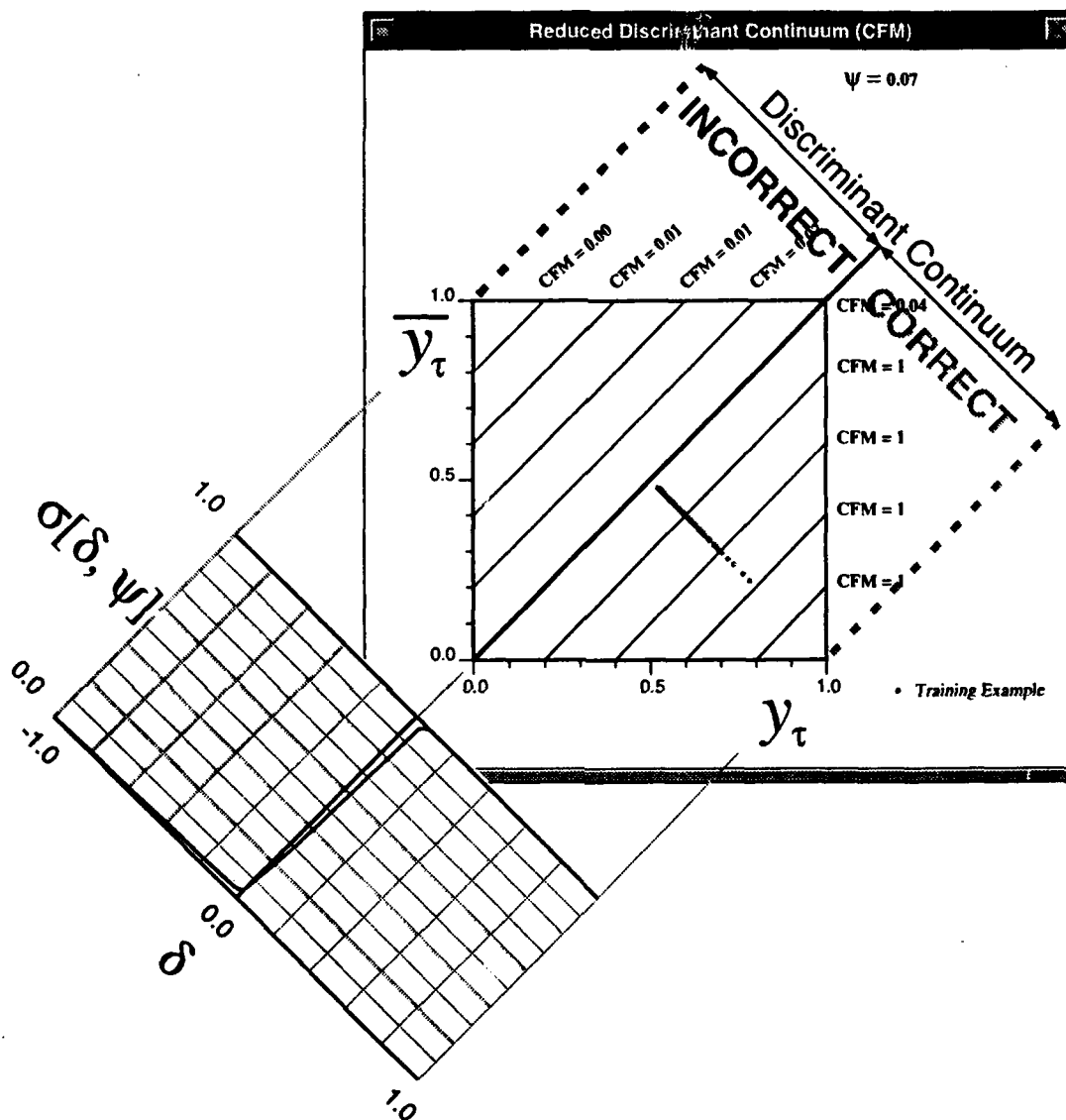


Figure 9.5: The 257-parameter differentially-generated logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after learning all 125 AVN examples. This output state corresponds to the parameters shown in figure 9.4 (right). Note that the low CFM confidence parameter $\psi = .077$ (resulting in a steep sigmoidal form of CFM) is necessary to learn examples 48 and 49 (see figure 9.8). These examples generate very small positive discriminant differentials, corresponding to the classifier's low confidence in its diagnosis.

9.2 Recognition Results

In order to assess how well the low-complexity logistic linear classifier generalizes, we repeat an experiment described in [90] in which the 125 example database is randomly partitioned into training and test samples of approximately equal sizes. Specifically, we run 55 2-fold cross validation trials; in each trial, each example is randomly assigned to the training sample with probability $\frac{1}{2}$; those examples not assigned to the training sample constitute the test sample. The classifier learns for 125 epochs or for 5 epochs beyond the point at which all training examples are classified correctly, whichever is less. All the trials are conducted according to the general protocols set forth in section 8.2. The confidence parameter is set to $\psi = 1$, and the weight smoothing factor is $\kappa = 0.02$. These measures are taken to reduce the classifier's empirical discriminant variance across trials: training sample sizes average 62 examples, a small number even for the compressed images; as a result, we want the classifier to learn only those examples in which it has high confidence. Figure 9.6 illustrates the final reduced discriminator output state after a typical learning trial for which the training sample size is 58 and the test sample size is 67. Training examples are shown as dark gray dots and test examples are shown as black triangles. The classifier learns all the training examples (most with high confidence), but misclassifies three test examples.

From a diagnostic perspective, the *null hypothesis* is that the image is of a healthy femur; the *alternative hypothesis* is that the image is of an AVN-compromised femur. The classifier's *sensitivity* to the alternative hypothesis is the probability that it will detect AVN when it is indeed present. The classifier's *specificity* is the probability that it will not incorrectly classify a healthy image as AVN-compromised. Figure 9.7 (left) shows the estimated sensitivity and specificity (with 95% confidence bounds) for the classifier in figure 9.6. This classifier's sensitivity is 91.4 (+8.6/-11.7)%, so it fails to detect 8.6 (+11.7/-8.6)% of the AVN-compromised test examples. The classifier's specificity is 100 (+0.0/-8.9)%, so it never incorrectly classifies a healthy test example as AVN-compromised (modulo the confidence bound). Figure 9.7 (right) shows the receiver operating characteristic (ROC) [134, sec. 2.2.2] for the classifier's ability to detect AVN. The discontinuities in the ROC are due to the small test sample size (67) on which it is based. Taken in the context of this small sample size and the large confidence bounds on the sensitivity/specificity statistics, the ROC power of 0.93 cannot be viewed as anything more than a gross measure of the classifier's detection capabilities. That is, we know that the classifier has reasonably good AVN detection characteristics, but there is insufficient data to state specifically how good these characteristics are.

One fact is certain from the trials: the information loss due to image compression has an impact on the classifier's ability to detect AVN. Figure 9.8 illustrates why. The figure shows the original 1024-pixel images (top) of the three examples misclassified by the classifier in figure 9.6. The three images below the high-resolution ones are their compressed versions. After compression it is difficult to discern the signs of AVN in examples 37 and 48; example 49 looks compromised to the human eye, but the low complexity

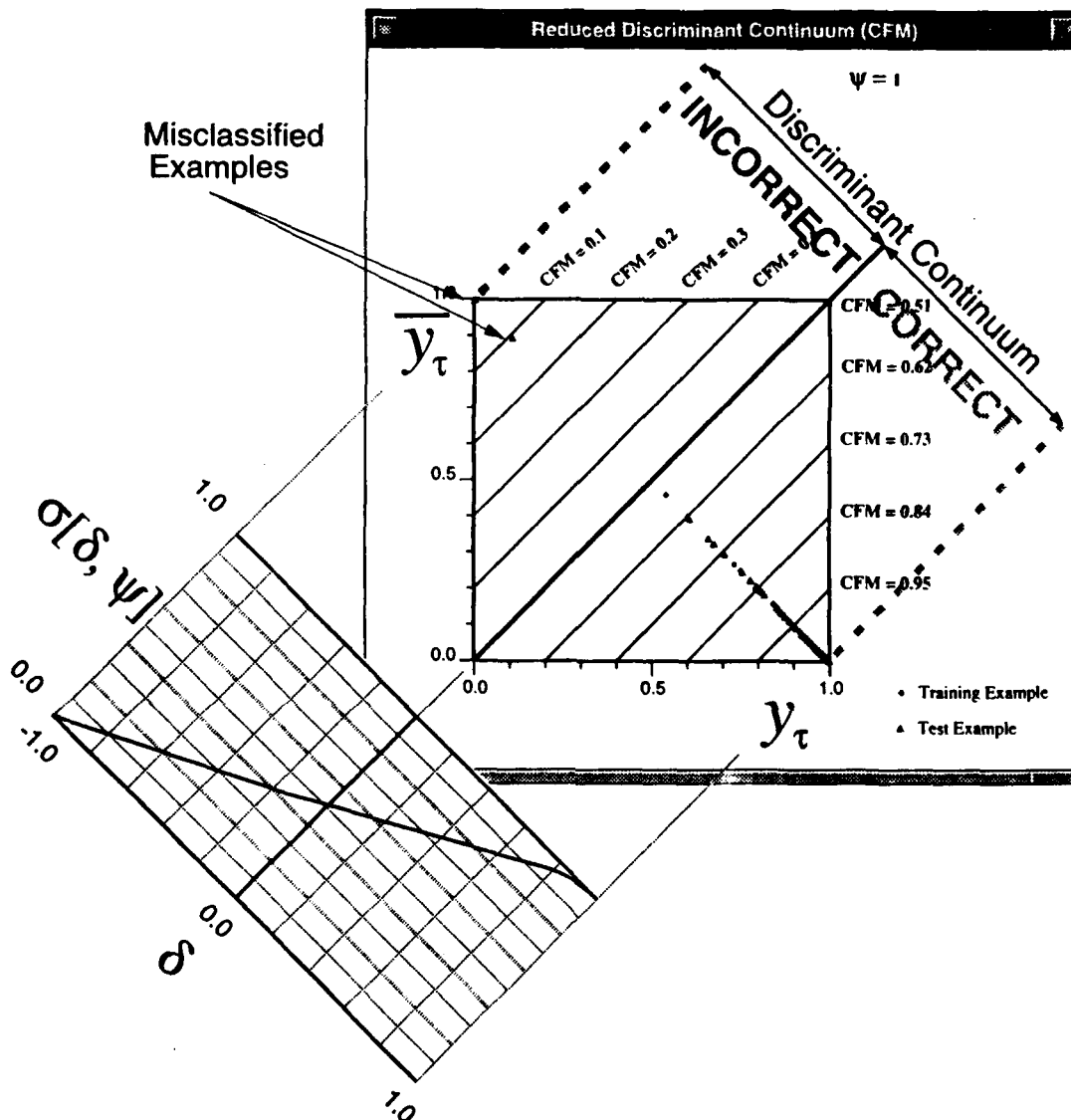


Figure 9.6: The 257-parameter differentially-generated logistic linear classifier's output state — as projected onto the reduced discriminant continuum — after a typical learning trial for which the training sample size is 58 images selected randomly from the pool of 125 total images. The model learns all training examples, but misclassifies three of the 67 test examples: 37, 48, and 49 (see figure 9.8). These misclassifications appear on the "incorrect" side of the reduced discriminant boundary; their discriminant differentials are $-.89$, $-.99$, and $-.99$ respectively. The large negative differentials indicate that the classifier is relatively confident in its incorrect diagnosis, based on the 58 training examples it has learned.

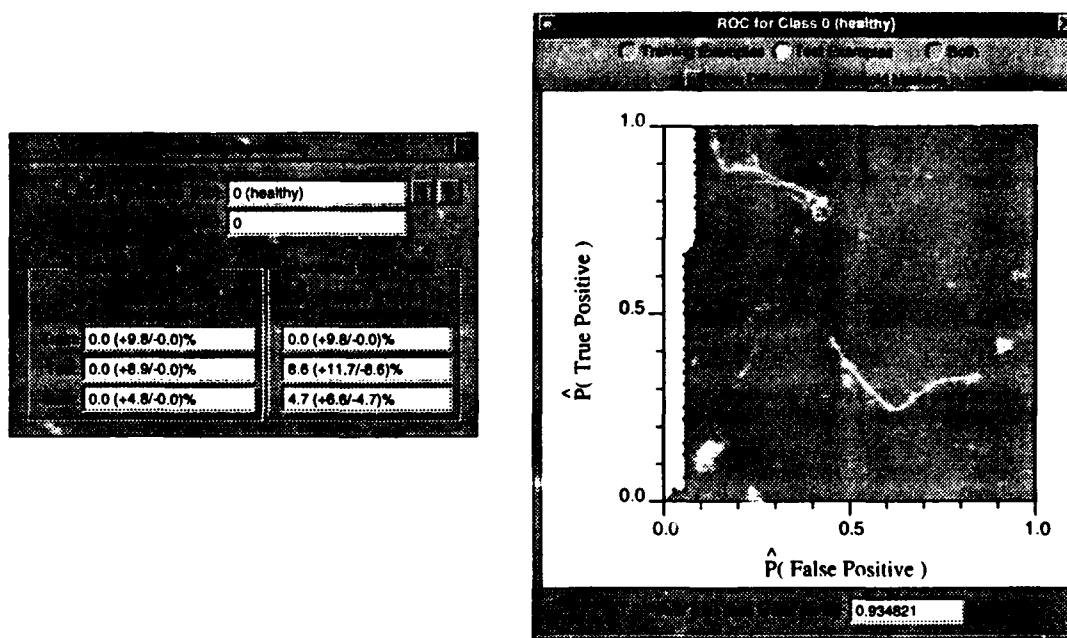


Figure 9.7: Left: The differentially-generated logistic linear classifier's sensitivity and specificity for the trial depicted in figure 9.6; 95% confidence bounds are given for α and β , and are computed as described in [62]. Right: The associated receiver operator characteristic for detecting an AVN-compromised femoral head (based on the test sample for this trial — middle row of the table on the left).

(i.e., 257-parameter) logistic linear classifier consistently evaluates this example as healthy. In reducing the classifier complexity by a 4 : 1 compression of the feature vector's dimensionality, we reduce the classifier's discriminant variance. This ensures that the classifier's error rate remains reasonably consistent across independently drawn training/test samples. The price we pay for this decreased discriminant variance is an increase in discriminant bias: some examples (e.g., 37, 48, and 49) become difficult to classify at lower resolution.

This reveals an interesting manifestation of the bias/variance tradeoff common to all estimators. In the case of AVN diagnosis from MRI images, the nature of the disease sometimes requires high-resolution imagery for a correct diagnosis (i.e., for low discriminant bias). This requirement, in turn, dictates higher classifier complexity. The increased complexity demands more data to ensure that the classifier's discriminant variance is low (i.e., that the more complex classifier will be sure to make consistent diagnoses).

Figure 9.9 summarizes the results of the 55 2-fold cross validation trials run on the compressed AVN images. The left side of the figure compares differential learning with two forms of probabilistic learning: the 257-parameter logistic linear classifier is used in each case, and all aspects of learning are identical except for the objective function (learning strategy) employed (see section 8.2). The classifier employing differential learning (Δ) has an average empirical test sample error rate of 5.9%, compared with 7.4% and 7.9% for the MSE and CE-generated variants. The classifier's empirical discriminant variance is approximately 8×10^{-4}

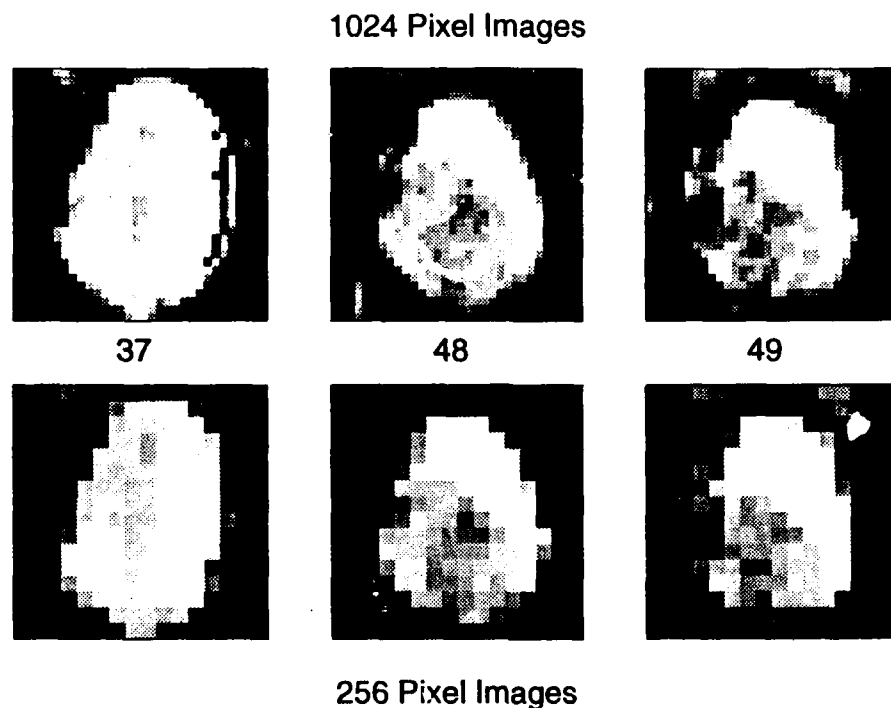


Figure 9.8: The three test images that the differentially-generated logistic linear classifier of figure 9.6 misclassifies. The classifier is presented the 256-pixel images in the bottom row of the figure: for these particular examples, compression from 1024 to 256 pixels results in the loss of information critical to a correct classification. In fact, the low-complexity linear classifier consistently misclassifies examples 48 and 49; it misclassifies example 37 in many of the 55 2-fold cross validation trials.

with differential learning and 9×10^{-4} with probabilistic learning. The right side of the figure compares the probabilistically generated classifiers with the differentially-generated one on a trial-by-trial basis. On average, the MSE-generated classifier's empirical test sample error rate is 1.4% greater than (or 1.2 times) the differentially-generated classifier's, with upper and lower standard deviations as shown. In half of the trials the differentially-generated classifier does no better than the MSE-generated classifier; in three trials it does worse (by as much as 1.9%); in the remaining trials it does better (by as much as 9.0%). On average, the Kullback-Leibler (CE)-generated classifier's empirical test sample error rate is 1.9% greater than (or 1.3 times) the differentially-generated classifier's, with upper and lower standard deviations as shown. In one quarter of the trials the differentially-generated classifier does no better than the CE-generated classifier; in four trials it does worse (by as much as 5.2%); in the remaining trials it does better (by as much as 8.1%).

We surmise that the differentially-generated model is not always better than its probabilistic counterparts due to its high complexity, given the average training sample size of $n = 62$; even the "low-complexity" classifier has a large number of parameters. As a result, all of the models exhibit high discriminant variance. Since differential learning guarantees *asymptotic* efficiency only, n is not big enough for the differentially-

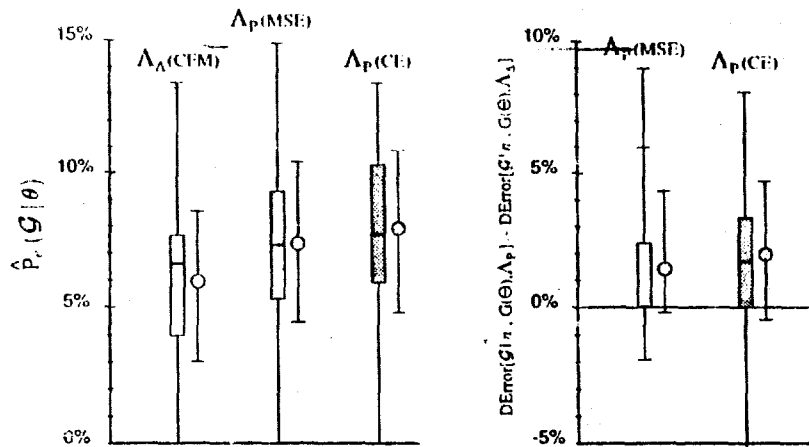


Figure 9.9: Left: Test sample classification summaries for the 257-parameter logistic linear classifier employing differential learning (Λ_{Δ}) and two forms of probabilistic learning (Λ_P). The summaries are based on 55 independent trials. In each trial, training examples are drawn randomly from the set of 125 images with probability $\frac{1}{2}$; those not chosen for training formed the test sample. Note that the CE-generated logistic linear classifier is identically the logistic regression model for this classification task (see appendix F). Right: The increase in the discriminant error of the two probabilistic models over the differentially-generated model on a trial-by-trial basis.

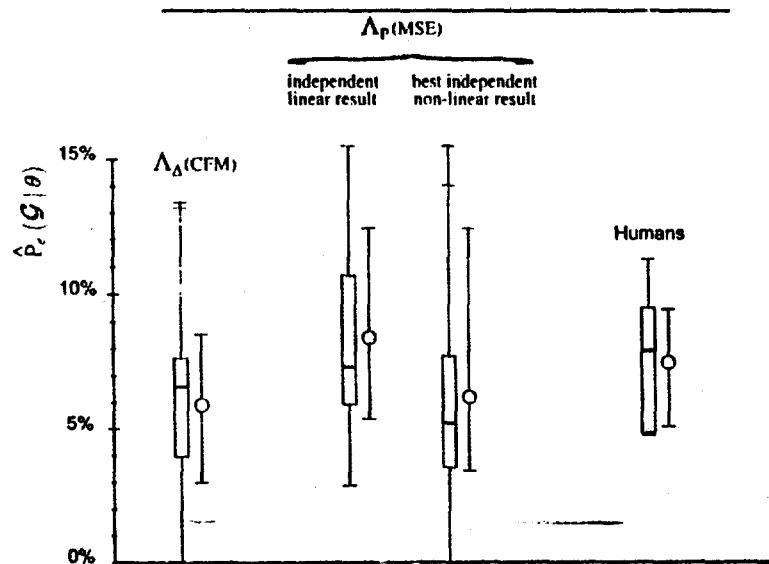


Figure 9.10: Test sample classification summaries for the low-complexity (257-parameter) differentially-generated logistic linear classifier (far left), Manduca, Christy, and Ehman's [90] 2050-parameter MSE-generated logistic linear classifier (middle left), and their best MSE-generated non-linear classifier (middle right), a high-complexity multi-layer perceptron with 6 hidden units and 6,164 parameters. Note that the differentially-generated model's average estimated discriminant error is the same as the high-complexity probabilistic model's, but its estimated discriminant variance is one-half that of the high-complexity model (cf. table 9.1). Each of ten human subjects performed one 2-fold cross validation trial (far right); the differentially-generated logistic linear classifier compares favorably with the humans.

Estimated DBias, DVar, and MSDE				
Hypothesis	Learning Strategy			
Class		Λ_{Δ} (CFM)	Λ_P (MSE)	Λ_P (CE)
55 trials, 256-pixel images				
Logistic Linear (257 parameters)	DBias	5.9×10^{-2}	7.4×10^{-2}	7.9×10^{-2}
	DVar	7.7×10^{-4}	8.8×10^{-4}	9.1×10^{-4}
	MSDE	4.3×10^{-3}	6.3×10^{-3}	7.2×10^{-3}
	ERE	—	0.68	0.60
Manduca, Christy, and Ehman [90]: 24 trials, 1024-pixel images ^a				
Logistic Linear (2050 parameters)	DBias	—	8.4×10^{-2}	—
	DVar	—	1.2×10^{-3}	—
	MSDE	—	8.3×10^{-3}	—
MLP (6164 parameters)	DBias	—	6.2×10^{-2}	—
	DVar	—	1.7×10^{-3}	—
	MSDE	—	5.6×10^{-3}	—
Manduca, Christy, and Ehman [90]: 10 human subjects, 1 trial each ^a				
Human (10 subjects)	DBias	—	7.5×10^{-2}	—
	DVar	—	4.6×10^{-4}	—
	MSDE	—	6.1×10^{-3}	—

Table 9.1: Estimated discriminant bias, discriminant variance, and MSDE for the 257-parameter logistic linear classifiers generated by differential learning (Λ_{Δ}) via the CFM objective function and probabilistic learning (Λ_P) via the MSE and CE objective functions. Estimates are based on 55 2-fold cross-validation trials in which the AVN database is randomly partitioned into training and test samples, each containing approximately 62 examples. Estimates are also shown for Manduca, Christy, and Ehman's 2050-parameter logistic linear and 6164-parameter multi-layer perceptron (MLP) classifiers, both generated probabilistically with the MSE objective function [90]. Those estimates are based on 24 2-fold cross validation trials; the training/test sample partitions for their trials are different from ours. Finally, estimates are shown for 10 Human subjects [90]: each performed one 2-fold cross validation trial.

^aResults from Manduca, Christy, and Ehman [90] used with permission.

generated model to exhibit the lowest empirical error rate in *all* trials. By making the feature vector more Gaussian-like, compressing the MRI images makes the logistic linear classifier a better approximation to the proper parametric model of the feature vector. This also diminishes the relative efficiency of differential learning vis-a-vis probabilistic learning. Nevertheless, the differentially-generated model is rarely worse, and on average better than its probabilistically-generated counterparts. In short, its empirical MSDE is lower. Table 9.1 shows that the differentially-generated 257-parameter logistic linear classifier's ERE (definition 8.6, page 255) is $\widehat{RE}[\Lambda_{\Delta}, \Lambda_{P-MSE} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 0.68$ versus probabilistic learning via MSE and $\widehat{RE}[\Lambda_{\Delta}, \Lambda_{P-CE} | \{n_1, \dots, n_K\}, \mathbf{G}(\Theta)] = 0.60$ versus probabilistic learning via CE. The comparative efficiency of differential learning stems mainly from its lower discriminant bias (5.9% for differential learning, versus 7.4% (MSE) and 7.9% (CE) for probabilistic learning).⁶

Finally, we compare our results for these 55 trials with Manduca, Christy, and Ehman's results for 24 trials in which the data is split into training and test samples as described above [90]. Figure 9.9 summarizes the over-all comparison (results cannot be compared on a trial-by-trial basis, because we use different randomly partitioned training/test samples). Manduca, Christy, and Ehman's logistic linear classifier uses the original 1024-pixel images and has an output for each class, so it has 2050 parameters. It learns probabilistically with the MSE objective function and a conjugate gradient search algorithm. Its average empirical test sample error rate⁷ (and, as a result, its empirical discriminant bias) is 8.4%; its empirical discriminant variance is 1.2×10^{-3} . Their best multi-layer perceptron (MLP) classifier for the 50/50 training/test sample splits has 6 hidden units, two output units, and uses the original 1024-pixel images. As a result, the model has 6,164 parameters. Like its logistic linear counterpart, it learns probabilistically with the MSE objective function and a conjugate gradient search algorithm. Its average empirical test sample error rate (and empirical discriminant bias) is 6.2%; its empirical discriminant variance is 1.7×10^{-3} . As described earlier, our logistic linear classifier has 257 parameters; it learns with the CFM objective function and simple gradient ascent search algorithm. Its average empirical test sample error rate (and empirical discriminant bias) is 5.9%; its empirical discriminant variance is 7.7×10^{-4} .

Compared with Manduca, Christy, and Ehman's linear classifier, our differentially-generated model has a lower average empirical test sample error rate, and somewhat lower empirical discriminant variance. Compared with Manduca, Christy, and Ehman's best non-linear classifier, our differentially-generated logistic linear model has the same empirical test sample error rate (our 5.9% is not significantly better than their 6.2%) and approximately one half the discriminant variance. Virtually none of this empirical discriminant variance difference is attributable to our larger number of trials (as the critical reader might suspect): we have

⁶We assume an estimated Bayes error rate of $\widehat{P}_e(\mathcal{F}_{Bayes}) = 0\%$ for the AVN diagnosis task. The actual number is probably non-zero since Manduca, Christy, and Ehman report that human radiology experts will not commit to a diagnosis on all 125 images in the database [90].

⁷All statistics attributed to Manduca, Christy, and Ehman are either published in [90] or have been provided to the author by Dr. Manduca via personal correspondence.

looked at the empirical discriminant variance of several randomly selected 24-trial sub-sets of our 55 trials, and it remains fairly steady about 8×10^{-4} . Thus, the differentially-generated model combines the low error rate of the complex probabilistic model and the consistency of the simple probabilistic model.

Manduca, Christy, and Ehman ran trials in which 10 humans not trained in radiology learned to diagnose AVN from half of the 125 images; they were subsequently tested on the other half. The human subjects had an average empirical test sample error rate (and empirical discriminant bias) of 7.5%; their discriminant variance was 4.6×10^{-4} . Thus, the low complexity logistic linear classifier employing differential learning is, on average, at least as good as the human novice for this limited diagnosis task.

9.3 Summary

We find that a relatively simple logistic linear classifier learns to diagnose avascular necrosis of the femoral head from a single low-resolution MRI image with an error rate of 5.9 (+4.9/-4.2)%.⁸ The classifier is more consistent than the best independently developed (probabilistic) model, which exhibits approximately twice as much discriminant variance across independent test trials. In addition, the classifier's error rate compares favorably with the 7.5% error rate of human novices who are provided with the same training and testing data.

Learning to diagnose AVN presents an unavoidable tradeoff between discriminant bias and discriminant variance. The simple classifier exhibits lower discriminant variance at the cost of increased discriminant bias when complexity is reduced by compressing the original high-resolution MRI images into lower-resolution ones; the details essential to a correct diagnosis are simply lost in the more difficult examples. This leads us to acknowledge that there are learning tasks in which consistently low recognition error rates demand large training samples. In such tasks, the key to consistently low error rates is in subtle details, which can be gleaned only from a large training sample. This does not diminish the advantages of differential learning, but it does show that a tradeoff between discriminant bias and variance is sometimes inevitable, no matter what learning strategy is employed.

⁸The upper and lower bounds on this error rate are (more rigorous) 95% confidence bounds, rather than the standard deviations quoted in section 9.2.

Chapter 10

Remote Sensing with Differential Learning¹

Outline

We describe a series of remote sensing experiments conducted in collaboration with the Digital Mapping Laboratory, School of Computer Science, Carnegie Mellon University. We use a modified RBF classifier employing differential learning (DRBF) to interpret multi-spectral imagery from the Daedalus airborne (remote sensing) scanner system. The interpretation procedure involves classifying individual image pixels, which represent 64 square meters of earth surface material, into eleven categories of natural and man-made materials — a preliminary step in automated map generation and various environmental analysis tasks. The DRBF classifier has 132 parameters and exhibits a 29% error rate on the interpretation task. The maximum-likelihood (probabilistic) model currently used for this task has 847 parameters and exhibits a 46% error rate. Most of the DRBF's reduced error rate is attributable to its sub-sampling the training data during learning; 12% of the reduction is attributable to differential learning/lower model complexity.

10.1 Introduction

The interpretation of remote sensing imagery is an integral part of a diverse set of earth sciences (e.g., map generation, crop analysis, forestry, land use, assessing the environmental effects of airborne and water-borne pollution, etc.). The imagery is obtained from satellite and airborne optical sensors, which are sensitive to visible as well as near infrared and short-wave infrared light reflected from the earth's surface. The Digital Mapping Laboratory, School of Computer Science, Carnegie Mellon University, is developing computer systems for the automated interpretation of remote sensing imagery obtained from the Daedalus airborne

¹ We thank David McKeown and Stephen Ford of the Digital Mapping Laboratory for providing us with the multi-spectral imagery for this task, introducing us to the business of remote sensing, and providing us with both the DRBF and maximum-likelihood classifier test results.

multi-spectral imaging system (e.g., [36]). Daedalus generates images comprising eleven spectral bands (ten span the continuum from short-wave visible light to short-wave infrared light and one is in the thermal region — see [36, fig. 1]). The image has a ground sample distance of approximately 8 meters, so a single pixel represents a patch of earth with an area of about 64 meters squared.

As a preliminary step in generating maps by machine, each pixel in an image is classified according to its “ground truth” class. Eleven classes are used in the current system: asphalt, concrete, coniferous tree, deciduous tree, deep water, grass, shadow, shallow water, soil, tile, and turbid water. The pixel-by-pixel interpretation is performed by a classifier that has previously learned examples of each ground truth class. The feature vector $\mathbf{X}$ representing each pixel has eleven elements, corresponding to the eleven bandpass sensor outputs of the Daedalus system. Both the test imagery and training pixels are taken from a 3000×700 pixel (134 square kilometer) image of the Washington, D.C. metropolitan area.

The remote sensing community frequently uses Gaussian maximum-likelihood (ML) classifiers to interpret multi-spectral imagery. In the case of the Daedalus scanner data, each of the ML model’s eleven discriminant functions has the form

$$g_i(\mathbf{X}|\theta) = \frac{f_i(\mathbf{X})}{\sum_{j=1}^C f_j(\mathbf{X})}, \quad (10.1)$$

where

$$f_i(\mathbf{X}) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Sigma_i|^{\frac{1}{2}}} \cdot \exp \left[-\frac{1}{2} (\mathbf{X} - \mu_i)^T \Sigma_i^{-1} (\mathbf{X} - \mu_i) \right], \quad (10.2)$$

and the i th mean μ_i and covariance matrix Σ_i are subsets of the over-all discriminator parameter vector θ . Since there are $C = 11$ discriminant functions for the eleven ground truth classes,² the classifier has a total of $11(11 + 66) = 847$ parameters³ (i.e., $\theta \in \Theta = \mathbb{R}^{847}$). The parameters of μ_i and Σ_i are estimated from the training sample by the method of maximum-likelihood (e.g., [28, sec. 6.5]). The resulting classifier therefore learns probabilistically and is the fully parametric counterpart to the partially parametric logistic linear hypothesis class described in section 7.2.2. In the jargon of parametric statistical pattern recognition, the fully-parametric maximum likelihood classifier is the normal-based linear discriminant analysis paradigm, and its partially-parametric counterpart, the logistic linear classifier, is known as the logistic discriminant analysis paradigm (e.g., [91]).

²It is purely coincidental that the number of ground truth classes C and the feature vector dimensionality N are both 11.

³The classifier comprises $C = 11$ discriminant functions, associated with the same number of ground truth classes. Each discriminant function has 11 parameters that correspond to the $N = 11$ -dimensional class-conditional feature vector mean. The covariance matrix has $N^2 = 121$ parameters, but it is symmetric, so it effectively has only $N(N + 1)/2 = 66$ parameters. Therefore, the classifier has a total of $11(11 + 66) = 847$ parameters.

We compare the maximum-likelihood classifier with a differentially-generated modified radial basis function classifier (appendix K) having no hidden layer units. Each of the DRBF's eleven discriminant functions has the form

$$g_i(\mathbf{X}|\theta) = \exp \left[-\frac{1}{2\sigma_i^2} (\mathbf{X} - \mu_i)^T (\mathbf{X} - \mu_i) \right], \quad (10.3)$$

where σ_i^2 denotes the discriminant function's single variance parameter. As a result, the DRBF classifier has a total of $11(11 + 1) = 132$ parameters (i.e., $\theta \in \Theta = \Re^{132}$) — fewer than one sixth as many as the maximum-likelihood model.

10.2 Training Data

The single remote sensing training sample comprises 10,616 pixels of the eleven ground-truth classes, taken from the 3000×700 pixel image of the Washington, D.C. area described earlier. Although the four small test sites described in section 10.3 are taken from the same over-all image, all of the training data is from different locations in the image, so the test and training samples are disjoint. Consequently, this is a site-dependent classification task (if the training data were taken from, say, Atlanta, Georgia, and the test data were taken from Washington, D.C., the task would be truly site-independent). Table 10.1 shows the number of example pixels n_i for each ground truth class ω_i ($i = 1, \dots, C$) in the training sample. The maximum likelihood classifier learns simply by computing the maximum-likelihood estimates of the eleven discriminant functions' class-conditional means and covariances based on this sample. By (10.1) and (10.2), the maximum-likelihood classifier implicitly assumes that all class prior probabilities are equal. The DRBF classifier learns differentially by maximizing the CFM objective function with respect to θ over the set of all training examples; this is done by an iterative search that employs gradient ascent and the backpropagation algorithm (e.g., [119, 120]), altered for use with the modified RBF non-linearities of (10.3). In order to speed learning convergence, the DRBF classifier's parameters are initialized as described on page 211: the class-conditional mean vectors are initialized to their corresponding class-conditional training sample average, while the class-conditional variance parameters are initialized to their corresponding class-conditional sample variances.

A differential *learning epoch* comprises one iteration of the learning algorithm for each example in the training sample. Iterative statistical learning procedures such as differential learning require that the empirical class prior probabilities of the training sample match those of the test sample. At the same time, a large training sample size for each ground-truth class is desirable, since it better characterizes the class-conditional probability density function (pdf). Because the ground-truth classes do not all have the same prior probabilities in the Washington, D.C. area, the DRBF does not attempt to learn every training example each epoch (see table 10.1). Instead it learns a fraction of each ground-truth class in a given epoch.

Training Sample Sizes (n_i)		
Ground Truth Class	n_i	DRBF's Learning Probability p_i
asphalt	1000	0.5
concrete	1000	1.0
coniferous tree	616	0.04
deciduous tree	1000	1.0
deep water	1000	1.0
grass	1000	1.0
shadow	1000	1.0
shallow water	1000	0.3
soil	1000	0.01
tile	1000	0.01
turbid water	1000	1.0

Table 10.1: The training sample sizes n_i ($i = 1, \dots, C$) for both the maximum-likelihood and DRBF classifiers. The far-right column applies to the DRBF classifier only. In any given epoch, the DRBF randomly selects with probability p_i each of the n_i training examples of class ω_i ; the examples selected in a given epoch are learned during that epoch; all other examples are ignored in that epoch. This form of randomly sub-sampling the training data each epoch effectively alters the empirical class prior probabilities of the training sample so that they are more representative of their true underlying values.

At the beginning of each epoch the DRBF randomly selects with probability p_i (or ignores with probability $1 - p_i$) each of the n_i training examples of class ω_i ; the examples selected in a given epoch are learned during that epoch; all other examples are ignored in that epoch. This form of randomly sub-sampling the training data each epoch effectively alters the empirical class prior probabilities of the training sample so that they are more representative of their true underlying values. That is, the probabilities $\{p_1, \dots, p_C\}$ in table 10.1 have been chosen so that they approximate the prior probabilities of the ground truth classes in the Washington, D.C. area (i.e., $\{P_W(\omega_1), \dots, P_W(\omega_C)\}$) thus:

$$\frac{p_i n_i}{\sum_{j=1}^C (p_j n_j)} \cong P_W(\omega_i) \quad (10.4)$$

Reference [35] shows that this technique of sub-sampling the training sample each learning epoch reduces the DRBF's empirical test sample error rate by about 15%. Additionally, it accounts for a significant fraction of the test differences between the DRBF and ML classifiers (see section 10.3).

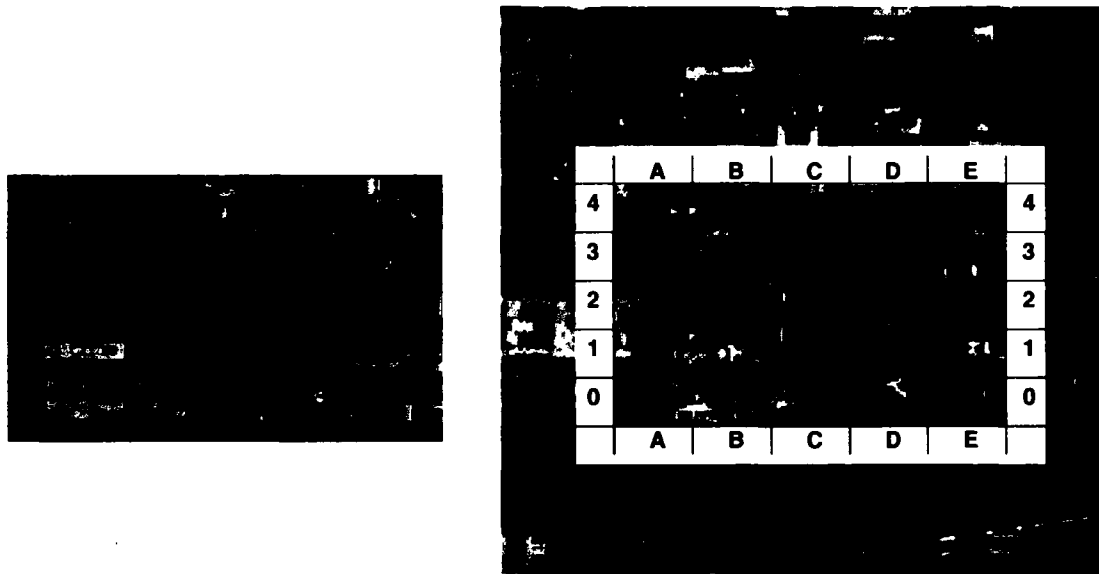


Figure 10.1: **Top Left:** Panchromatic image of the **civill** site (1.2 meter resolution). **Top Right:** Composite of the multi-spectral data for the **civill** site (8 meter resolution), which the classifiers interpret.

10.3 Experimental Results

Figures 10.1 – 10.6 pertain to four sites in downtown Washington, D.C., on which we have tested the DRBF and maximum-likelihood classifiers. Figure 10.1 shows the vicinity of the Civil Service and Department of the Interior buildings. The image on the left is a high-resolution panchromatic image of the site; the image on the right is a lower resolution composite of the 11-band multi-spectral image that the classifiers interpret.⁴ There is no explicit relationship between the colors in the multi-spectral images of figure 10.2 (right) and those in the classification maps of figure 10.2. The three images in figure 10.2 depict the ground truth for the site (middle — generated by a human using an interactive image classification tool) and the DRBF (top) and maximum-likelihood (bottom) classifiers' interpretations of the multi-spectral image in figure 10.1. Classification errors occur at pixels for which the color of a classifier's interpretation differs from the ground truth image color. The color legend to the right in figure 10.2 explains the color-scheme used in the ground-truth and classification maps.

Tables 10.2 and 10.3 summarize the DRBF classifier's test results at the **civill** site, and tables 10.4 and 10.5 summarize the maximum-likelihood classifier's test results. Tables 10.2 and 10.4 match the ground truth class names with their class labels ($\omega_1, \dots, \omega_{11}$),⁵ and they show the top ten confusions made by the

⁴All of the panchromatic images were taken with a resolution of 1.2 meters per pixel side (i.e., a pixel represents 1.44 square meters of surface area); all of the multi-spectral images were taken approximately seven years later at 8 meters/pixel side (i.e., a pixel represents 64 square meters of surface area).

⁵These label/name lists are give in both tables for quick reference.

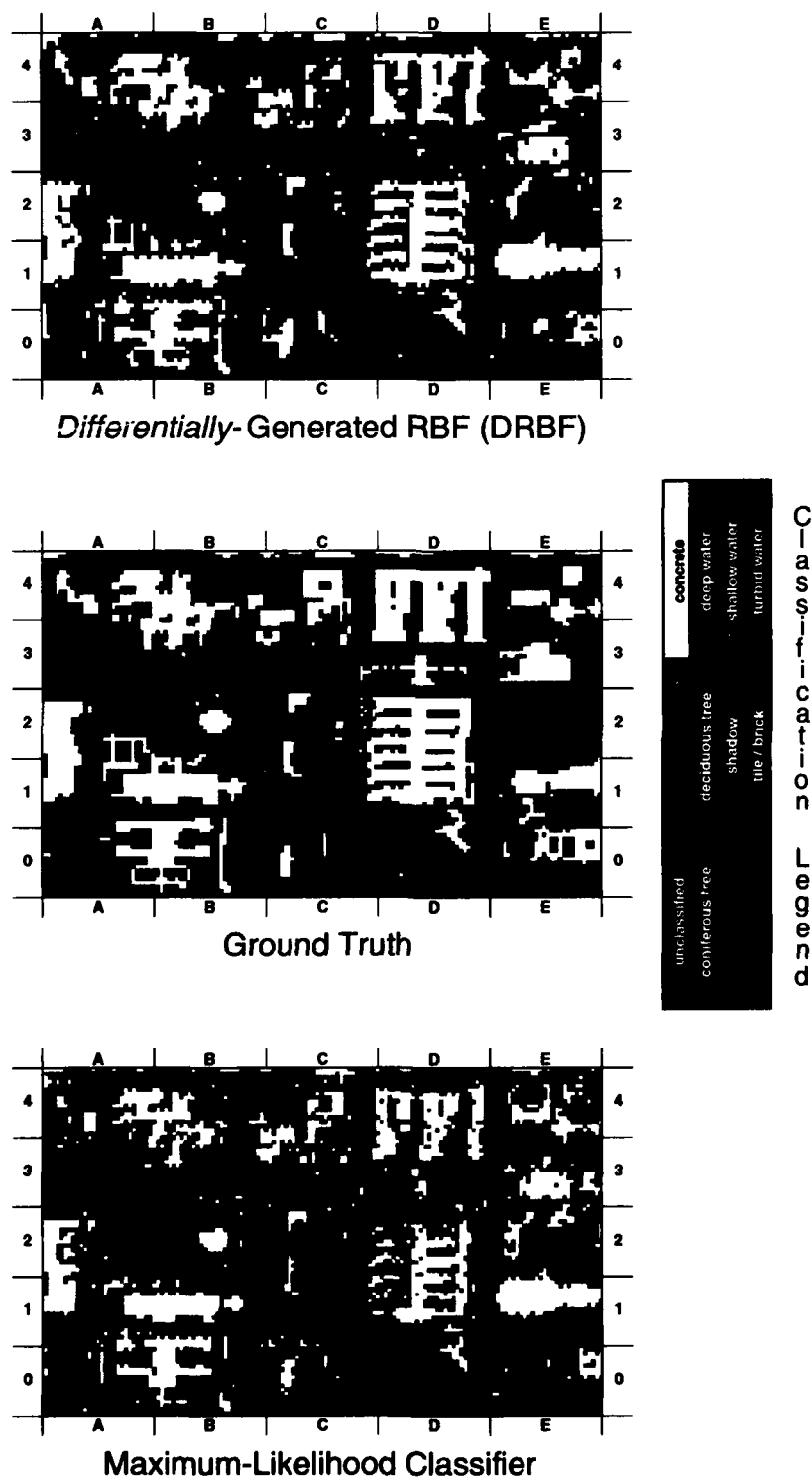


Figure 10.2: **Top:** The DRBF classifier's interpretation of the civil1 site. **Middle** The ground truth for the civil1 site. **Bottom** The maximum-likelihood (ML) classifier's interpretation of the civil1 site.

civill Ground Truth Classes		DRBF Top Ten Confusions			
		True Class	Misclassified as	Count	Percent
ω_1	asphalt	deciduous tree	grass	800	27.3
ω_2	concrete	deciduous tree	asphalt	570	19.5
ω_3	deciduous tree	concrete	asphalt	548	18.8
ω_4	grass	deciduous tree	shadow	284	9.7
ω_5	shadow	asphalt	concrete	244	5.7
ω_6	tile	asphalt	shadow	180	4.2
ω_7	coniferous tree	shadow	asphalt	163	20.1
ω_8	deep water	grass	deciduous tree	156	13.4
ω_9	shallow water	grass	asphalt	144	12.4
ω_{10}	soil	concrete	grass	45	1.5
ω_{11}	turbid water				

Table 10.2: Left: Class labels assigned to the 11 ground truth classes. Right: Top ten confusions made by the DRBF classifier over the **civill** site.

DRBF Ground Truth Class Confusion Matrix														
Detected Class	True Class											Total	Percent Correct	False Detection Rate
	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}	ω_{11}			
ω_1	3809	548	570	144	163	33	0	0	0	0	0	5267	72.3	27.7
ω_2	244	2268	25	24	4	5	0	0	0	0	0	2570	88.2	11.8
ω_3	5	0	1197	156	1	0	0	0	0	0	0	1359	88.1	11.9
ω_4	16	45	800	815	0	1	0	0	0	0	0	1677	48.6	51.4
ω_5	180	12	284	17	625	0	0	0	0	0	0	1118	55.9	44.1
ω_6	38	29	7	0	6	24	0	0	0	0	0	104	23.1	76.9
ω_7	0	3	43	5	1	0	0	0	0	0	0	52	0.0	100.0
ω_8	4	0	1	0	3	0	0	0	0	0	0	8	0.0	100.0
ω_9	0	0	0	0	1	0	0	0	0	0	0	1	0.0	100.0
ω_{10}	0	1	0	0	0	0	0	0	0	0	0	1	0.0	100.0
ω_{11}	9	9	0	0	5	0	0	0	0	0	0	23	0.0	100.0
Total	4305	2915	2927	1161	809	63	0	0	0	0	0	12180	71.7	28.3
% Correct	88.5	77.8	40.9	70.2	77.3	38.1	-	-	-	-	-	71.7		
False Negative Rate	11.5	22.2	59.1	29.8	22.7	61.9	-	-	-	-	-	28.3		

Table 10.3: Confusion matrix for the DRBF classifier over the **civill** site.

civill Ground Truth Classes		ML Top Ten Confusions			
		True Class	Misclassified as	Count	Percent
ω_1	asphalt	deciduous tree	grass	893	30.5
ω_2	concrete	asphalt	tile	876	20.3
ω_3	deciduous tree	deciduous tree	coniferous tree	831	28.4
ω_4	grass	deciduous tree	soil	766	26.2
ω_5	shadow	concrete	soil	634	21.7
ω_6	tile	asphalt	soil	480	11.1
ω_7	coniferous tree	grass	soil	330	28.4
ω_8	deep water	asphalt	concrete	318	7.4
ω_9	shallow water	concrete	tile	283	9.7
ω_{10}	soil	concrete	asphalt	178	6.1
ω_{11}	turbid water				

Table 10.4: **Left:** Class labels assigned to the 11 ground truth classes. **Right:** Top ten confusions made by the maximum-likelihood (ML) classifier over the **civill** site.

ML Ground Truth Class Confusion Matrix														
Detected Class	True Class											Total	Percent Correct	False Detection Rate
	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}	ω_{11}			
ω_1	2504	178	12	0	67	0	0	0	0	0	0	2761	90.7	9.3
ω_2	318	1808	2	0	5	1	0	0	0	0	0	2134	84.7	15.3
ω_3	0	0	252	11	2	0	0	0	0	0	0	265	95.1	4.9
ω_4	6	0	893	739	0	0	0	0	0	0	0	1638	45.1	54.9
ω_5	119	9	43	0	550	0	0	0	0	0	0	721	76.3	23.7
ω_6	876	283	128	11	152	49	0	0	0	0	0	1499	3.3	96.7
ω_7	2	3	831	70	15	0	0	0	0	0	0	921	0.0	100.0
ω_8	0	0	0	0	0	0	0	0	0	0	0	0	-	-
ω_9	0	0	0	0	14	0	0	0	0	0	0	14	0.0	100.0
ω_{10}	480	634	766	330	4	13	0	0	0	0	0	2227	0.0	100.0
ω_{11}	0	0	0	0	0	0	0	0	0	0	0	0	-	-
Total	4305	2915	2927	1161	809	63	0	0	0	0	0	12180	48.5	51.5
% Correct	58.2	62.0	8.6	63.7	68.0	77.8	-	-	-	-	-	48.5		
False Negative Rate	41.8	38.0	91.4	36.3	32.0	22.2	-	-	-	-	-	51.5		

Table 10.5: Confusion matrix for the maximum-likelihood (ML) classifier over the **civill** site.

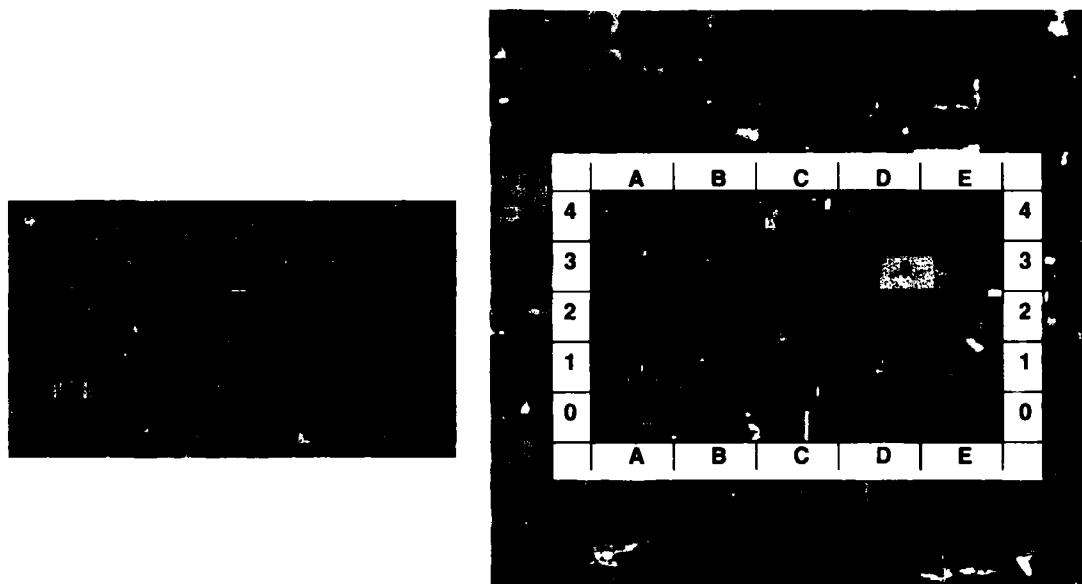


Figure 10.3: **Top Left:** Panchromatic image of the **gaol** site (1.2 meter resolution). **Top Right:** Composite of the multi-spectral data for the **gaol** site (8 meter resolution), which the classifiers interpret.

classifier. The percent confused is equal to the number in the "count" column divided by the total number of ground truth examples occurring in the test sample (image). Tables 10.3 and 10.5 are confusion matrices: they list the ground truth example totals at the bottom, in the "total" row. The class labels across the top of the confusion matrices indicate the actual (or *true*) ground truth class, and the labels in the left-most column denote the class detected by the classifier. The diagonal elements of the confusion matrix show the number of ground truth examples correctly classified or *detected* by the classifier; the off-diagonal elements show the number of examples misclassified (i.e., incorrectly detected as examples of *another* class). As an example, the bottom entry of table 10.3 in the ω_1 column indicates the percentage of asphalt pixels that the DRBF misclassifies as some other ground truth class. The right-most column of the table's ω_1 row indicates what percentage of pixels classified as asphalt actually represent some other ground truth class.

The two bold-face numbers in the lower right corner of tables 10.3 and 10.5 show what percentage of the image pixels are correctly classified and what percentage are misclassified by the classifier. Table 10.3 shows that the DRBF classifier exhibits a 28.3 (+/- 0.8)% empirical error rate on the **civil** site; table 10.5 shows that the maximum-likelihood classifier exhibits a 51.5 (+/- 0.9)% empirical error rate.⁶

⁶We remind the reader that empirical test sample error rates include 95% confidence intervals: these are estimated under the assumption that the error rate is binomially distributed [53]. Please see section 8.2 for details.

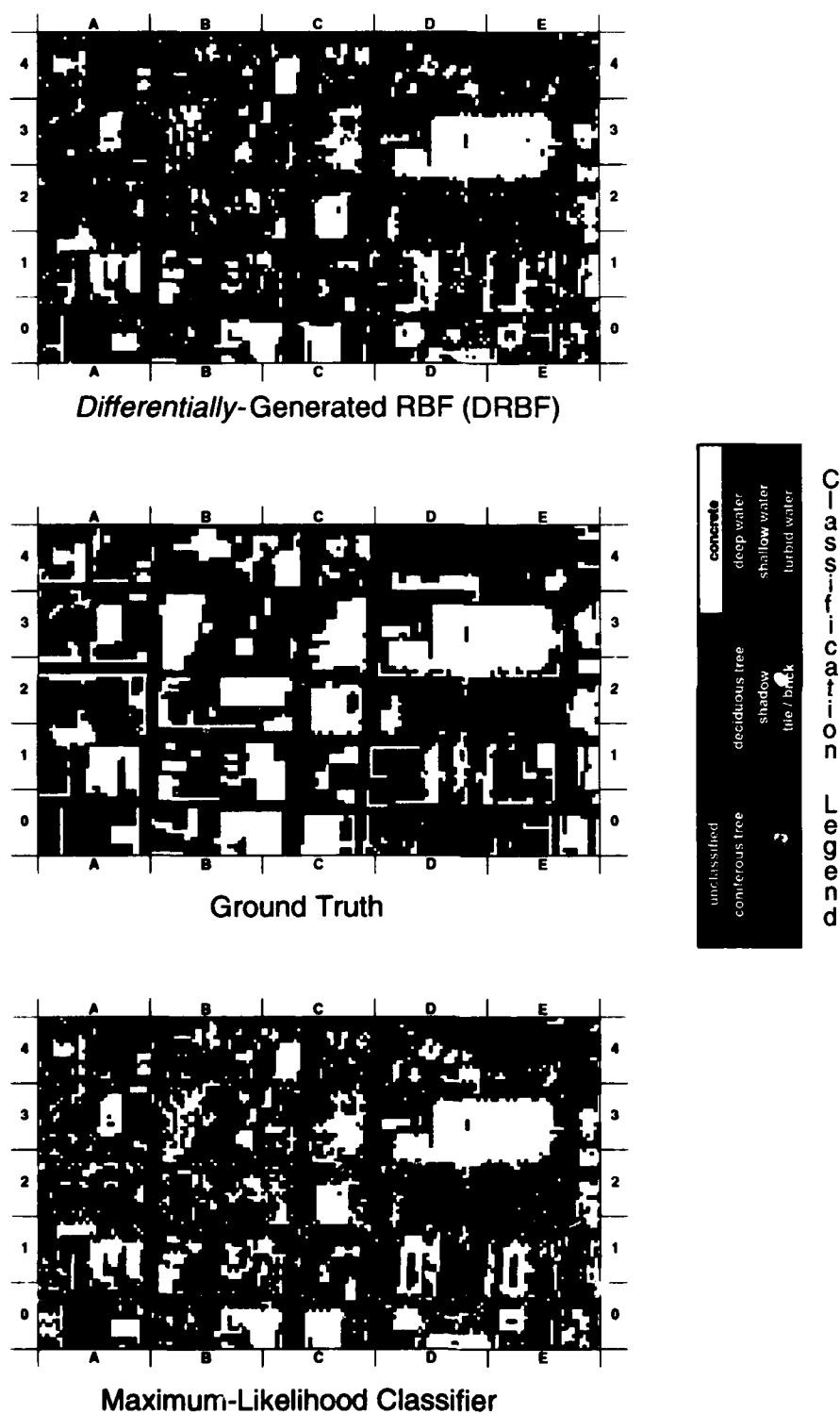


Figure 10.4: Top: The DRBF classifier's interpretation of the gaol site. **Middle** The ground truth for the gaol site. **Bottom** The maximum-likelihood (ML) classifier's interpretation of the gaol site.

gao1 Ground Truth Classes	
ω_1	asphalt
ω_2	concrete
ω_3	deciduous tree
ω_4	grass
ω_5	shadow
ω_6	soil
ω_7	tile
ω_8	coniferous tree
ω_9	deep water
ω_{10}	shallow water
ω_{11}	turbid water

DRBF Top Ten Confusions			
True Class	Misclassified as	Count	Percent
concrete	asphalt	1645	39.2
deciduous tree	asphalt	569	44.1
asphalt	concrete	386	6.1
deciduous tree	grass	365	28.3
shadow	asphalt	349	27.8
tile	asphalt	273	77.1
asphalt	shadow	182	2.9
deciduous tree	shadow	97	7.5
concrete	grass	52	1.2
shadow	deep water	35	2.8

Table 10.6: Left: Class labels assigned to the 11 ground truth classes. Right: Top ten confusions made by the DRBF classifier over the gao1 site.

DRBF Ground Truth Class Confusion Matrix														
Detected Class	True Class											Total	Percent Correct	False Detection Rate
	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}	ω_{11}			
ω_1	5719	1645	569	35	349	0	273	0	0	0	0	8590	66.6	33.4
ω_2	386	2466	12	5	1	0	9	0	0	0	0	2879	85.7	14.3
ω_3	0	0	233	33	5	0	0	0	0	0	0	271	86.0	14.0
ω_4	9	52	365	491	0	0	0	0	0	0	0	917	53.5	46.5
ω_5	182	9	97	2	824	0	4	0	0	0	0	1118	73.7	26.3
ω_6	0	0	0	0	0	0	0	0	0	0	0	0	-	-
ω_7	12	14	1	0	0	0	68	0	0	0	0	95	71.6	28.4
ω_8	0	1	13	2	0	0	0	0	0	0	0	16	0.0	100.0
ω_9	23	0	0	0	35	0	0	0	0	0	0	58	0.0	100.0
ω_{10}	12	0	0	0	28	0	0	0	0	0	0	40	0.0	100.0
ω_{11}	12	6	0	0	12	0	0	0	0	0	0	30	0.0	100.0
Total	6355	4193	1290	568	1254	0	354	0	0	0	0	14014	69.9	30.1
% Correct	90.0	58.8	18.1	86.4	65.7	-	19.2	-	-	-	-	69.9		
False Negative Rate	10.0	41.2	81.9	13.6	34.3	-	80.8	-	-	-	-	30.1		

Table 10.7: Confusion matrix for the DRBF classifier over the gao1 site.

gao1 Ground Truth Classes	
ω_1	asphalt
ω_2	concrete
ω_3	deciduous tree
ω_4	grass
ω_5	shadow
ω_6	soil
ω_7	tile
ω_8	coniferous tree
ω_9	deep water
ω_{10}	shallow water
ω_{11}	turbid water

ML Top Ten Confusions			
True Class	Misclassified as	Count	Percent
concrete	asphalt	965	23.0
asphalt	concrete	733	11.5
asphalt	tile	615	9.7
deciduous tree	soil	545	42.2
shadow	asphalt	393	31.3
deciduous tree	grass	346	26.8
concrete	soil	298	7.1
concrete	tile	246	5.9
asphalt	soil	234	3.7
deciduous tree	coniferous tree	202	15.7

Table 10.8: Left: Class labels assigned to the 11 ground truth classes. Right: Top ten confusions made by the maximum-likelihood (ML) classifier over the gao1 site.

ML Ground Truth Class Confusion Matrix														
Detected Class	True Class											Total	Percent Correct	False Detection Rate
	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9	ω_{10}	ω_{11}			
ω_1	4650	965	27	0	393	0	8	0	0	0	0	6043	76.9	23.1
ω_2	733	2667	0	0	3	0	1	0	0	0	0	3404	78.3	21.7
ω_3	0	0	13	0	0	0	0	0	0	0	0	13	100.0	0.0
ω_4	3	0	346	388	0	0	0	0	0	0	0	737	52.6	47.4
ω_5	116	17	15	0	639	0	2	0	0	0	0	789	81.0	19.0
ω_6	234	298	545	162	4	0	3	0	0	0	0	1246	0.0	100.0
ω_7	615	246	142	1	53	0	340	0	0	0	0	1397	24.3	75.7
ω_8	3	0	202	17	27	0	0	0	0	0	0	249	0.0	100.0
ω_9	0	0	0	0	0	0	0	0	0	0	0	0	-	-
ω_{10}	1	0	0	0	135	0	0	0	0	0	0	136	0.0	100.0
ω_{11}	0	0	0	0	0	0	0	0	0	0	0	0	-	-
Total	6355	4193	1290	568	1254	0	354	0	0	0	0	14014	62.1	37.9
% Correct	73.2	63.6	1.0	68.3	51.0	-	96.0	-	-	-	-	62.1		
False Negative Rate	26.8	36.4	99.0	31.7	49.0	-	4.0	-	-	-	-	37.9		

Table 10.9: Confusion matrix for the maximum-likelihood (ML) classifier over the gao1 site.

DRBF error rate reduction: *The error rate reduction realized by employing the DRBF classifier in lieu of the maximum-likelihood (ML) classifier is $\frac{\Xi(\eta)_{ML} - \Xi(\eta)_{DRBF}}{\Xi(\eta)_{ML}}$, where $\Xi(\eta)_{ML}$ denotes the number of test sample errors made by the maximum-likelihood classifier and $\Xi(\eta)_{DRBF}$ denotes the number made by the DRBF classifier.*

The DRBF classifier therefore reduces the maximum-likelihood classifier's empirical error rate by 45% at the **civill** site. A review of figure 10.2 and tables 10.2 — 10.5 shows that the maximum-likelihood classifier has high false detection rates for soil, tile, and coniferous trees. The DRBF classifier frequently fails to detect tile. Both classifiers misclassify deciduous trees as grass roughly 30% of the time.

Figures 10.3 and 10.4 and tables 10.6 — 10.9 compare the two classifiers for the **gao1** site, which includes the General Accounting Office building. Table 10.7 shows that the DRBF classifier exhibits a 30.1 (+/- 0.8)% empirical error rate at the **gao1** site; table 10.9 shows that the maximum-likelihood classifier exhibits a 37.9 (+/- 0.8)% empirical error rate. The DRBF classifier therefore reduces the maximum-likelihood classifier's empirical error rate by 21% at the **gao1** site. A review of figure 10.4 and tables 10.6 — 10.9 shows that the classifiers exhibit the same general trends over the **gao1** site as they do over the **civill** site, with a few notable exceptions. The DRBF's insensitivity to tile is more evident, owing to the higher prior probability of that ground truth class at the **gao1** site. More than half of the DRBF errors occur when it confuses asphalt and concrete (we attribute this phenomenon to the large number of parking lots with vehicles present at the site, the surfaces of which exhibit both asphalt and concrete-like spectral characteristics). As a result, the disparity between the two classifiers' empirical error rates is considerably lower at the **gao1** site.

Figures 10.5 and 10.6 compare the classifiers over the White House (**whouse1**) and the Federal Bureau of Engraving (**engrave1**), respectively. Human-generated ground truth are not shown for these sites, and we omit a detailed statistical comparison of the classifiers in the interest of brevity. A visual comparison indicates that the general trends exhibited over the **civill** and **gao1** sites are manifest at the **whouse1** and **engrave1** sites as well. This is confirmed by a comparison of the two classifiers' empirical error rates over these two sites, which are shown in table 10.10. The table lists the error rates of both classifiers on all four sites, along with 95% confidence bounds computed as described in section 8.2 and [62]. The DRBF classifier exhibits a 28.4 (+/- 0.8)% empirical error rate at the **whouse1** site; the maximum-likelihood classifier exhibits a 51.6 (+/- 0.9)% empirical error rate. The DRBF classifier therefore reduces the maximum-likelihood classifier's empirical error rate by 45% at the **whouse1** site. The DRBF classifier exhibits a 29.6 (+/- 0.7)% empirical error rate at the **engrave1** site; the maximum-likelihood classifier exhibits a 44.7 (+/- 0.8)% empirical error rate. The DRBF classifier therefore reduces the maximum-likelihood classifier's empirical error rate by 34% at the **engrave1** site. Based on the combined test results from all four sites, the DRBF classifier exhibits a 29.2 (+/- 0.4)% empirical error rate; the maximum likelihood classifier exhibits

Estimated Error Rates				
Site	Test Sample Size n	Maximum Likelihood Classifier	DRBF	Percent Error Rate Reduction, DRBF versus ML
civil1	12,180	51.5 (+/- 0.9)%	28.3 (+/- 0.8)%	45%
gao1	14,014	37.9 (+/- 0.8)%	30.1 (+/- 0.8)%	21%
whouse1	12,155	51.6 (+/- 0.9)%	28.4 (+/- 0.8)%	45%
engravel	15,704	44.7 (+/- 0.8)%	29.6 (+/- 0.7)%	34%
all sites	54,053	46.1 (+/- 0.4)%	29.2 (+/- 0.4)%	37%

Table 10.10: A summary of the empirical test sample error rates for both the maximum-likelihood and DRBF classifiers. The far-right column shows the percent reduction in error rate realized by employing the DRBF in lieu of the maximum-likelihood classifier.

a 46.1 (+/- 0.4)% empirical error rate. The DRBF classifier therefore reduces the maximum-likelihood classifier's empirical error rate by 37% over all four sites.

10.3.1 Interpretation of Test Results

The reduced complexity and differential learning strategy of the DRBF classifier account for part of the 37% improvement over the maximum-likelihood classifier, but they do not account for all of it. Recall from section 10.2 that the DRBF classifier sub-samples the training data when learning — a procedure that effectively alters the ground truth class prior probabilities and reduces the DRBF's error rate by 15%. For this reason, we suspect that more than half of the DRBF's improvement over the maximum-likelihood classifier has nothing to do with differential learning and the reduced classifier complexity it allows. Indeed, if the DRBF classifier learns without sub-sampling the training data, its empirical error rate over the **civil1** and **gao1** sites increases to 40.4 (+/- 0.6)%.⁷ Thus, the DRBF classifier without sub-sampled training data reduces the maximum-likelihood classifier's empirical error rate by only 12% — a statistically significant but not substantial amount. Put another way, it is reasonable to believe that incorporating estimates of the ground-truth class prior probabilities into the maximum-likelihood classifier via (10.1) — altering that equation so that it is an expression of Bayes' rule — would reduce the maximum-likelihood classifier's error rate by about 25%, leaving only a 12% advantage to the DRBF classifier.

These differences notwithstanding, we suspect that the DRBF's 29% error rate is close to the (minimum) Bayes error rate for X , an 11-element vector describing a single image pixel. The human expert generates ground truth for each site by looking at the over-all image; thus, (s)he exploits contextual information (e.g., the appearance of neighboring pixels) to which the DRBF does not have access. Given such information

⁷Comparable results for the **whouse1** and **engravel** sites have not been compiled.

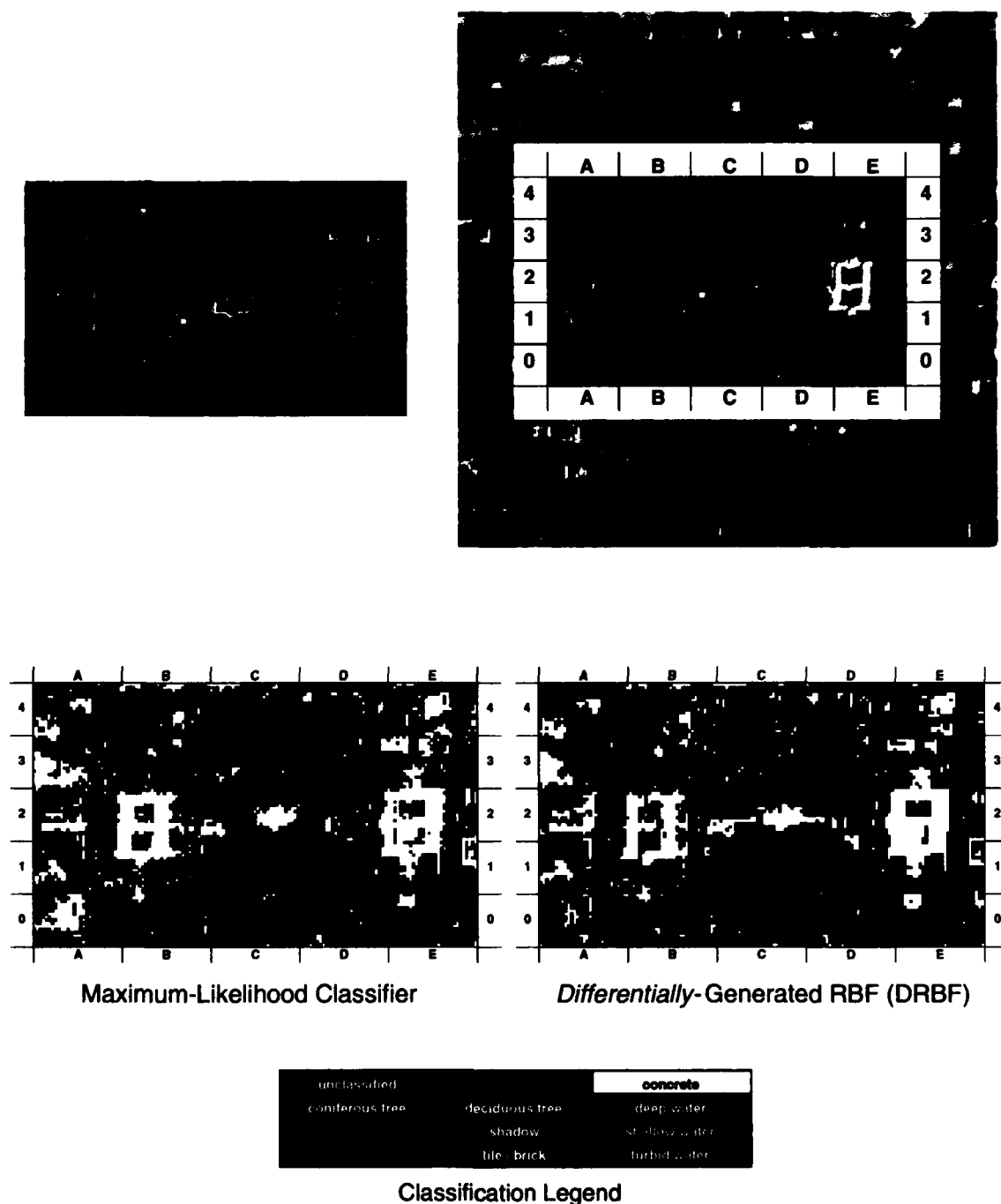


Figure 10.5: **Top Left:** Panchromatic image of the **whouse1** site (1.2 meter resolution). **Top Right:** Composite of the multi-spectral data for the **whouse1** site (8 meter resolution), which the classifiers interpret. **Bottom Left:** The maximum-likelihood classifier's interpretation of the **whouse1** site. **Bottom Right:** The DRBF classifier's interpretation of the **whouse1** site.

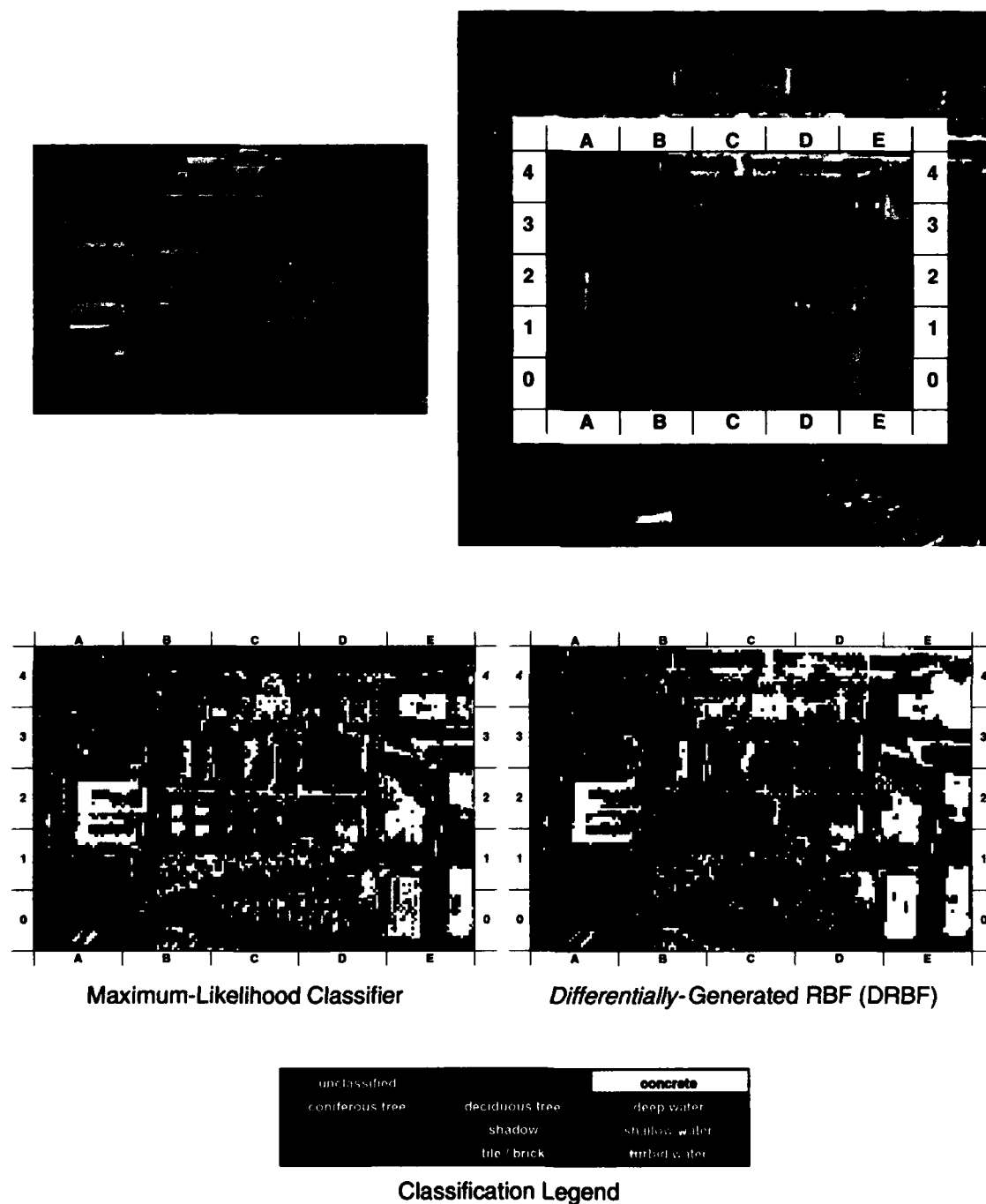


Figure 10.6: **Top Left:** Panchromatic image of the **engrave1** site (1.2 meter resolution). **Top Right:** Composite of the multi-spectral data for the **engrave1** site (8 meter resolution), which the classifiers interpret. **Bottom Left:** The maximum-likelihood classifier's interpretation of the **engrave1** site. **Bottom Right:** The DRBF classifier's interpretation of the **engrave1** site.

in the form of a feature vector that includes the pixel of interest *and* its surrounding neighbors, the DRBF classifier would probably exhibit a significantly lower error rate. Owing to the geometric increase in the number of parameters a comparable maximum-likelihood model would require for the expanded feature vector,⁸ we doubt such a model would exhibit a lower error rate.

10.4 Summary

We have used a DRBF classifier with 132 parameters to interpret multi-spectral images of the Washington, D.C. area, pixel-by-pixel. The DRBF exhibits a 29% error rate, 37% lower than the 46% error rate exhibited by the 847-parameter maximum-likelihood (probabilistic) classifier currently used for this task. The DRBF classifier's reduced complexity and differential learning strategy account for approximately 12% of the improvement over the maximum-likelihood model; the remaining 25% of the improvement is due to the DRBF classifier's method of sub-sampling the training data during learning. This sub-sampling allows the DRBF classifier to adjust the training sample's empirical class prior probabilities so that they match those of the test sample, thereby ensuring that the training sample is representative of the test sample.

⁸The number of DRBF classifier parameters increases linearly with N , the feature vector dimensionality. The number of maximum-likelihood classifier parameters increases as N^2 , dooming the paradigm to Bellman's curse of dimensionality [13].

Chapter 11

Conclusions

11.1 Scientific Contributions

We began this text by stating that the research herein is motivated by three convictions: the first is that it is not necessary to estimate probabilities in order to perform robust statistical pattern recognition; the second is that there are many real-world pattern recognition tasks for which a proper parametric model either does not exist or cannot be determined; the third is that the simplest model of the data is generally the best one — Occam's razor. These convictions motivated our research and serve as the backdrop to what we believe is our principal contribution to the fields of machine learning and statistical pattern recognition:

differential learning — a discriminative learning strategy for differentiable supervised classifiers that guarantees the best-generalizing classifier allowed by the choice of hypothesis class, whatever that choice is; the guarantee always holds for large training sample sizes and it usually holds (i.e., it holds for all improper parametric models) when the training sample size is small.

Lesser contributions — Chapter-by-chapter, the lesser contributions are as follows:

- Defining two strategies by which differentiable supervised classifiers can learn: probabilistic and differential (chapter 2).
- Defining two fundamental forms of the Bayesian discriminant function that correspond to the probabilistic and differential learning strategies (chapter 2).
- Developing an estimation-theoretic view of generalization (chapter 3):
 - Defining the classifier as an estimator of the Bayes-optimal classifier.
 - Defining estimation-theoretic measures of generalization:
 - * Discriminant bias
 - * Discriminant variance

- * Mean-squared discriminant error (MSDE)
 - Defining the efficient and relatively efficient classifiers.
 - Defining the efficient and asymptotically efficient learning strategies.
- Proving that differential learning is accomplished by maximizing the CFM objective function (chapter 2).
- Proving that differential learning is asymptotically efficient (chapter 3).
- Proving that differential learning requires the minimum-complexity hypothesis class necessary for Bayesian discrimination (chapters 3 and 6).
- Proving that minimizing a classifier's functional error does *not* equate to minimizing its error rate; that is, error measures are non-monotonic (chapters 3 and 5).
- Proving that probabilistic learning is accomplished by minimizing error measure objective functions (chapter 2).¹
- Proving that probabilistic learning is inefficient (chapter 3).
- Defining proper and improper parametric models (chapter 3).
- Sketching and illustrating the proof that probabilistically-generated proper parametric models can be more efficient than their differentially-generated counterparts for small training sample sizes (chapters 3 and 4).
- Developing the classification figure-of-merit (CFM) objective function [55] and deriving a synthetic form of it that engenders reasonably fast learning (chapter 5, and appendix D).
- Developing distribution-dependent bounds on the training sample size requirements for good generalization via differential and probabilistic learning ([51] and chapter 6).
- Showing that the minimum-complexity requirements of differential learning are consistent with the tenets of VC theory (section 3.5).

11.2 Philosophical Implications of Differential Learning

There are at least six "philosophical" implications of differential learning that warrant discussion; the most strenuous objections to differential learning that we have fielded to date pertain to them.

¹We remind the reader that our claim of originality here is restricted to our definition of and proofs relating to the general error measure, a significant fraction of which is due to Barak Pearlmuter. Proofs pertaining to specific error measures precede our work, and we make no claim to them.

Estimating probabilities — Differential learning seeks only to learn the identity of the most likely class of the feature vector over its domain — a discriminative strategy that equates to learning the Bayes-optimal class boundaries on the feature vector's domain. This learning objective is substantially less rigorous than that of probabilistic learning, which seeks to learn all the *a posteriori* class probabilities of the feature vector over its domain. Indeed, the lower degree of rigor accounts in part for the efficiency of differential learning. By accepting differential learning we abandon the goal of estimating probabilities. This is heresy to some traditionalists. In fairness to them, we acknowledge that there are statistical pattern recognition tasks for which probabilistic estimates are essential. If, for example, we are going to caution a potential coronary bypass surgery candidate against an operation because our computer diagnostic system indicates that she will not survive the procedure, then we might require a firm probabilistic estimate of mortality (both with and without surgery) on which to base the ultimate decision to operate or not. Hidden Markov Models (HMMs) and Markov Random Fields, which are used to recognize patterns that evolve over space and/or time, rely heavily on robust probabilistic estimates. Since differential learning does not generate these, it is not likely that it can work well in hybrid systems that use neural network classifiers to estimate the probabilities for HMM systems (e.g., [126]).

If we need a probabilistic estimate, *we are compelled* to allocate the resources necessary for a *robust* estimate. On the basis of section 6.4, this might require extremely large training sample sizes (depending on whether or not our parametric model is a good approximation to the proper one). If the model is approximately proper and/or the data can be collected, then we need only expend the time, money, and effort necessary for the collection. If, on the other hand, the model is improper and there is no plausible way to obtain the data, then we must face the fact that reliable probabilistic estimates simply cannot be made. Under these constraints, classification will be more reliable if we abandon the untenable goal of estimating probabilities and employ differential learning.

Ultimately, we should consider whether or not probabilistic estimates are essential to our objectives in the context of whether or not our parametric model is approximately proper. The decision science literature is full of studies showing that humans — indeed human experts — are remarkably bad at estimating probabilities and applying them consistently to their process of decision making (see for example [69, 27]). Nevertheless, we revere our human pattern recognition capabilities and have great confidence in their ability to guide us to rational decisions. This paradox might be explained by the differences between probabilistic and differential learning, although we make no claim of biological plausibility. We leave it to the reader to decide when probabilistic learning is imperative; absent this imperative and/or the knowledge of a proper parametric model of the data, we suggest the differential learning strategy.

How Easy is it to Approximate the Proper Parametric Model — We have sketched the proof that probabilistic learning generates the efficient classifier when the hypothesis class is a proper parametric model

of the data (section 3.6); the experiments of sections 4.2 and 8.5.4 confirm the phenomenon. In all other cases — that is, when the hypothesis class is an improper parametric model of the data — differential learning generates the relatively efficient classifier for both small and large training sample sizes. Thus, the choice between probabilistic and differential learning hinges on whether or not the hypothesis class is a proper parametric model of the feature vector. We remind the reader that if the proper parametric model exists — indeed it does not always exist — it is unique. In practice, however, we need only approximate it in order to obtain efficient probabilistic learning. The wide acceptance and use of logistic regression models and multi-layer perceptrons follows from their use of logistic non-linearities. Many feature vectors have well-separated, unimodal class-conditional densities; consequently, the logistic function allows a good approximation to the feature vector's *a posteriori* class probabilities — the model is approximately proper. Thus, the real question that should decide between probabilistic and differential learning is whether this reasonable approximation holds in a given case. We encourage the reader to ask this question by rigorous hypothesis testing, and act on the answer as appropriate.

The bias/variance tradeoff — As discussed in chapter 3, there is a difference between a classifier's *functional* bias and variance and its *discriminant* bias and variance. In the context of proper parametric models a third type of *parametric* bias and variance arises, which is closely related to its functional counterpart. Assuming that the model is *believed* to be proper (regardless of whether or not it really is), differential learning trades an increase in the classifier's parametric and functional bias for a decrease in both its discriminant bias and variance. Whether or not the trade is a good one from a classification perspective depends upon whether or not the parametric model is indeed proper. If it is, the trade is not a good one for small training sample sizes because probabilistic learning can probably generate a more efficient classifier with lower parametric and functional bias/variance as well. If it is not, the trade is a good one for both small and large training sample sizes because differential learning generates the relatively efficient classifier, whereas probabilistic learning generates a distinctly inefficient classifier.

Interpreting models — One of the many criticisms leveled against neural network classifiers is that, owing to their complexity, they do not reveal readily discernible relationships between the feature vector and the Bayes-optimal classification. Indeed, part of the appeal of parametric models (here we use the term in the traditional sense) is their simple structure, which lends itself to straightforward interpretation. Our experience is that physicians, for example, prefer a bad model that is readily interpretable to a good one that is not. From an engineering perspective this seems silly, but from a medical perspective — people's lives may depend on the classifier's discrimination — the decision favors certainty over uncertainty; this is a rational, defensible preference. By implicitly assuming that the parametric model of the data is improper, differential learning adds one more layer of uncertainty in the eyes of some potential users. We must therefore develop better theories and tools for understanding these models (see section 11.3) if we are to eliminate the uncertainty and

exploit the superior generalization afforded by differential learning.

Representation — Finally, the renaissance of connectionism has brought the issue of representation to the fore. In our terminology, the choice of hypothesis class is the choice of representation. This choice and how it is made are topics of active debate. There seems to be strong consensus that a good representation of the data, carefully engineered prior to learning, is the best assurance of good generalization. As we state in chapter 8, we dispute this conclusion and offer both our theoretical and experimental results as evidence that representation is *not* such an important issue.² Complexity issues are undeniably important, as Vapnik has clearly proven, but the specific functional basis with which the data is modeled is secondary when differential learning is employed. As long as the hypothesis class has sufficient functional complexity to approximate the Bayes-optimal class boundaries on feature vector space, the representation is adequate. This is obviously *not* the case with probabilistic learning, since it seeks to model the feature vector's *a posteriori* class probabilities in addition to its class boundaries — a function approximation task for which representation is a key issue.

Weighting Risks — Not all classification tasks weight classifications equally. The magnetic resonance image (MRI) interpretation task of chapter 9 is a good example. The risk of failing to detect avascular necrosis (AVN) should be weighted more heavily than the risk of a "false positive". Different weightings are incorporated into probabilistic models by a simple application of Bayes rule after the classifier has learned the training sample. Although this same procedure can be used with differentially generated classifiers, it is *not* theoretically defensible, since the classifier's outputs do not represent probabilistic estimates. Altering the empirical class prior probabilities of the training sample to account for the Bayesian risk formalism is a more defensible approach with differential learning; equivalently, the step size of the iterative search algorithm used for differential learning can be weighted in proportion to the risk associated with each class. Both of these techniques can be shown to implement the Bayesian risk weighting formalism.

11.3 Future Research

We view the learning process as one in which the learner must

- choose a strategy for learning the training data with a model,
- choose a means of implementing the learning strategy (i.e. a specific algorithm that implements the strategy),
- acquire the training data and choose the manner in which that data is represented,

²We do not make this statement without regard to the effect of representation on the learning rate. The choice of representation can mean the difference between reasonably fast learning and unreasonably slow learning, so it is an important choice. Indeed, the end of section 5.5.1 implies that differential learning frees us to choose models that yield Bayesian discrimination *and* learn reasonably fast over those that might be better probabilistic representations of the data but take unreasonably long to learn.

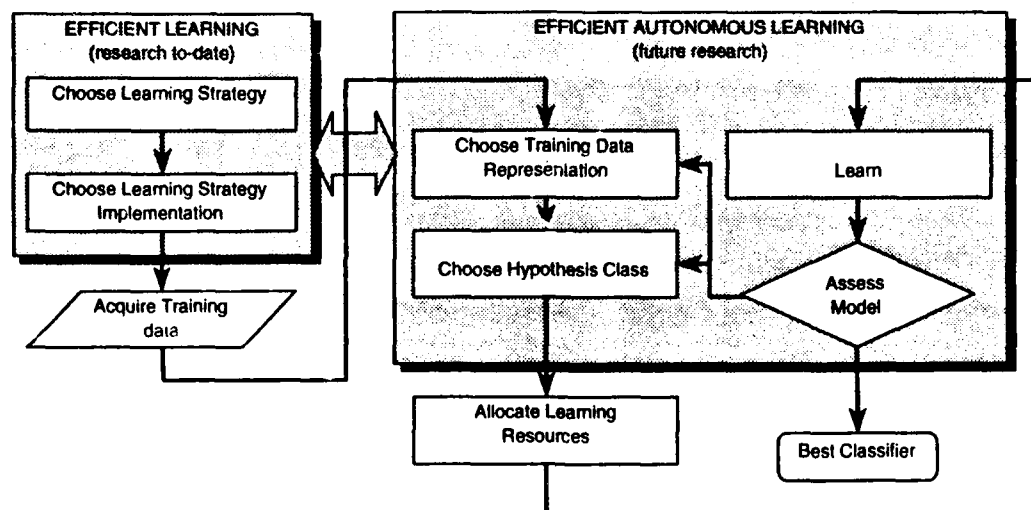


Figure 11.1: A simplified view of efficient autonomous learning.

- choose a hypothesis class (i.e., a limited set of choices from which the classifier will be generated),
- allocate the storage and computational resources necessary for the data, the hypothesis class, and the learning strategy as implemented,
- learn, and
- assess the resulting model in terms of alternative plausible models.

Figure 11.1 illustrates this view of learning. Our research to date has focussed on proving that differential learning is an optimal strategy, in that it guarantees the best generalization allowed by the choice of hypothesis class. We have also developed a computationally efficient implementation of differential learning. Given Occam's razor, the efficiency and minimum-complexity requirements of differential learning are significant: the simplest model that explains the data will generalize best, given a finite number of training examples, and this model can in principle be generated with differential learning.

Since Kolmogorov's theorem [77] can be interpreted as a proof that the minimum-complexity Bayes-optimal classifier can be determined only by exhaustive search, we are faced with the challenge of differentially learning a reasonable approximation to that classifier and comparing it with other plausible models. Since the classifier's complexity is directly related to the training data's form (i.e., the specific form of the feature vector), the challenge of finding a reasonable approximation to the minimum-complexity classifier involves choosing a form for the data and generating a model that explains the data. In turn, the process of choosing a data form and generating a model can be viewed as an iterative search on the joint space of all possible data forms and models. In our research to date, we have chosen the data form and the

hypothesis class (i.e., the set of allowed models) by a procedure that requires substantial human oversight. The training data form has been fixed prior to learning. Likewise, hypothesis class selection has been done by humans prior to learning and has remained fixed during learning. Finally, learning rates, CFM confidence parameter reduction schedules, and regularization (e.g., weight decay and weight smoothing factors) have been fixed by humans prior to learning. If the learning machine is to be truly autonomous, all of these choices must be controlled by the machine *during* learning. Clearly then, future work should entail theory and procedures by which the learning machine can continuously manipulate the training data, the hypothesis class, and the learning search procedure in a manner that both exploits and is consistent with the efficient, minimum-complexity nature of differential learning.

The critical reader will note that numerous researchers are exploring both theories and algorithms for automatically regulating model complexity during learning. The theoretical work of MacKay (e.g., [88]) and the cascade correlation [32] and optimal brain damage (OBD) [85] algorithms are well-known works in the connectionist literature. Since all of these works derive from an inefficient probabilistic view of learning, they can all be shown to be inefficient paradigms for model complexity regulation. Nevertheless, they can be adapted to differential learning (a process that we have already begun with encouraging results). We therefore believe that differential variants of these techniques hold promise for autonomous differential learning.

Finally, a statistically rigorous method of testing/rejecting classification hypotheses made by the differentially-generated classifier *after* learning needs to be developed. Such a testing procedure could form the basis of more sophisticated hypothesis testing procedures for model interpretation (recall section 11.2) as well as classification assessment.

Appendix A

Glossary of Notation

We employ a mixture of the general notational conventions of [45, 29, 117, 100]. The list below contains a comprehensive list of notation used in the text; common symbols are omitted.

Symbol	Meaning
$\triangleq$	Read, "... is defined as ..."
$\therefore$	Read, "Therefore."
$\notin$	Read, "... is not in ..."
$\nexists$	Read, "There does not exist ..."
$\gtrsim$	Read, "... approximately greater than or equal to ..."
$\lesssim$	Read, "... approximately less than or equal to ..."
$\nabla_{\mathbf{Z}} (f(\mathbf{Z}))$	The gradient of $f(\mathbf{Z})$ with respect to the vector $\mathbf{Z}$.
$\underline{0}$	The zero vector (the number of elements in the vector is context-dependent).
$ \cdot $	The cardinality of a set; the absolute value of a number; the determinant of a matrix.
$\ \cdot\ $	The magnitude of a vector.
$\text{ARE}_{n \rightarrow \infty}$	Asymptotic relative efficiency (see definition 3.18).
B_{ij}	The boundary on $\mathcal{X}$ between classes ω_i and ω_j .
C	The number of classes (i.e., concepts) in a pattern recognition task.
CE	The Kullback-Leibler information distance [82, 81], also known as the "cross entropy" objective function.
$\text{CE}(\mathcal{S}^n \theta)$	The CE generated by the training sample $\mathcal{S}^n$, given the discriminator $\mathcal{G}(\mathbf{X} \theta)$ with parameterization θ .
$\mathcal{CF}$	The correct fraction of discriminator output space (see definition 5.12).
$\mathcal{CF}_{\text{mono}}$	The monotonic correct fraction of discriminator output space (see definition 5.15).
$\Phi \mathcal{CF}_{\text{mono}}(C)$	The monotonic correct fraction of discriminator output space associated with the objective function Φ , given a C -dimensional discriminator output space (i.e., a C -class learning task).

Symbol	Meaning
$\mathcal{CF}_{\neg mono}$	The non-monotonic correct fraction of discriminator output space (see definition 5.14).
CFM	The CFM objective function.
CFM ($S^n \theta$)	The CFM generated by the training sample S^n , given the discriminator $\mathcal{G}(\mathbf{X} \theta)$ with parameterization θ .
CFM ($S^n \theta[k]$)	The CFM generated by the training sample S^n , given the discriminator $\mathcal{G}(\mathbf{X} \theta)$ with parameterization $\theta[k]$ at learning iteration k .
$\dim_{VC}(\cdot)$	The Vapnik-Chervonenkis (VC) dimension [137, 136], a measure of classifier complexity (see section 3.5).
δ	The <i>discriminant differential</i> (see definition 2.7). Note: the somewhat smaller notation $\delta(\cdot)$ denotes the Dirac delta function (e.g., [80, pg. 266]); the use is made clear in the text.
δ_τ	Given the example/class label pair $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, $\delta_\tau = y_\tau - \bar{y}_\tau$ is the discriminant differential associated with the class $\mathcal{W}^j$ of the example $\mathbf{X}^j$ (see section 2.2.4).
δ_{reject}	The reject threshold value of the discriminant differential: if a test example generates a discriminant differential less than this value, the classification is rejected as invalid.
$\delta_{learned}$	The learned threshold value of the discriminant differential: if a training example generates a discriminant differential greater than or equal to this value, the example has been learned.
$\delta_i(\mathbf{X} \theta)$	The discriminant differential associated with the i th discriminant function $g_i(\mathbf{X} \theta)$ (see definition 2.7).
$\delta_i(\mathbf{X} \theta^*)$	The discriminant differential associated with the i th discriminant function $g_i(\mathbf{X} \theta^*)$ (see definition 2.7). This notation indicates the the discriminator's parameterization θ^* is optimal to the extent that it maximizes the CFM objective function.
D	The high-state target value associated with an error measure objective function (see section 2.3).
$\neg D$	The low-state target value associated with an error measure objective function (see section 2.3).
$\mathcal{D}(\mathbf{X})$	A classifier of the random vector $\mathbf{X}$.
$\mathcal{D}(\mathbf{X})_{Bayes}$	A Bayes-optimal classifier of $\mathbf{X}$ (see definition 2.1).
$\mathcal{D}^*(\mathbf{X})$	The efficient classifier of $\mathbf{X}$ (see definition 3.12).
$\mathcal{D}^*(\mathbf{X})$	The <i>relatively</i> efficient classifier of $\mathbf{X}$ (see definition 3.15), that is, the classifier that exhibits the lowest MSDE allowed by the hypothesis class from which it is generated.
$\mathcal{D}(\mathbf{X} \theta)$	The class label (or classification) assigned to the feature vector $\mathbf{X}$ by the classifier.
DError [$\mathcal{G} \theta$]	The discriminant error of the classifier generated from the hypothesis class $\mathbf{G}(\theta)$ by the learning strategy Λ , given a training sample size of n (see definition 3.6).
DBias [$\mathcal{G} n, \mathbf{G}(\theta), \Lambda$]	The discriminant bias of the classifier generated from the hypothesis class $\mathbf{G}(\theta)$ by the learning strategy Λ , given a training sample size of n (see definition 3.7).
DVar [$\mathcal{G} n, \mathbf{G}(\theta), \Lambda$]	The discriminant variance of the classifier generated from the hypothesis class $\mathbf{G}(\theta)$ by the learning strategy Λ , given a training sample size of n (see definition 3.8).

Symbol	Meaning
$\widehat{\text{DBias}}[\mathcal{G} \{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda]$	The <i>estimated</i> discriminant bias of the classifier repeatedly generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ , given K training samples of sizes $\{n_1, \dots, n_K\}$.
$\widehat{\text{DVar}}[\mathcal{G} \{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda]$	The <i>estimated</i> discriminant variance of the classifier repeatedly generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ , given K training samples of sizes $\{n_1, \dots, n_K\}$.
$\Delta_{W X}(\omega_i \mathbf{X})$	The <i>a posteriori</i> differential of class ω_i , given the feature vector $\mathbf{X}$ (see definition 2.5).
EM	The general error measure objective function.
$\text{EM}(\mathcal{S}^n \theta)$	The EM generated by the training sample $\mathcal{S}^n$, given the discriminator $\mathcal{G}(\mathbf{X} \theta)$ with parameterization θ .
$\text{E}_X[f(\mathbf{X})]$	The expectation of the function $f(\mathbf{X})$, taken over the domain of the random vector (or variable) $\mathbf{X}$.
$\mathcal{F}(\mathbf{X})$	A discriminant function (more precisely, a set of $\mathcal{C}$ discriminant functions) for $\mathbf{X}$.
$\mathcal{F}(\mathbf{X})_{\text{Bayes}}$	The Bayesian discriminant function (BDF) of $\mathbf{X}$ (in any of its forms — see definition 2.2).
$\mathcal{F}(\mathbf{X})_{\text{Bayes-Probabilistic}}$	A probabilistic form of the BDF (definition 2.4).
$\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Probabilistic}}$	A strictly probabilistic form of the BDF (definition 2.3).
$\mathcal{F}(\mathbf{X})_{\text{Bayes-Differential}}$	A differential form of the BDF (definition 2.6).
$\mathcal{F}(\mathbf{X})_{\text{Bayes-Strictly Differential}}$	A strictly differential form of the BDF (definition 2.5).
$\mathbf{F}_{\text{Bayes}}$	The set of all BDFs of $\mathbf{X}$.
$\mathbf{F}_{\text{Bayes-Probabilistic}}$	The set of all probabilistic forms of the BDF of $\mathbf{X}$.
$\mathbf{F}_{\text{Bayes-Strictly Probabilistic}}$	The set of all strictly probabilistic forms of the BDF of $\mathbf{X}$.
$\mathbf{F}_{\text{Bayes-Differential}}$	The set of all differential forms of the BDF of $\mathbf{X}$.
$\mathbf{F}_{\text{Bayes-Strictly Differential}}$	The set of all strictly differential forms of the BDF of $\mathbf{X}$.
Φ	The general objective function (or empirical risk measure).
$g_i(\mathbf{X} \theta)$	The classifier's discriminant function for class ω_i ; the parameterization of the over-all discriminator is θ .
$g_i(\mathbf{X} \theta^*)$	The classifier's discriminant function for class ω_i ; the parameterization of the over-all discriminator is θ^* , which is optimal by some objective function.
$\mathcal{G}(\mathbf{X} \theta)$	The classifier's discriminator (i.e., the set of $\mathcal{C}$ discriminant functions); the discriminator's parameterization is θ .
$\mathcal{G}(\mathbf{X} \theta^*)$	The classifier's discriminator (i.e., the set of $\mathcal{C}$ discriminant functions); the discriminator's parameterization is θ^* , which is optimal by some objective function.
$\mathcal{G}(\mathbf{X} \theta)_{\text{Bayes}}$	A Bayesian discriminant function (BDF) of $\mathbf{X}$ contained in the hypothesis class $\mathbf{G}(\Theta)$.
$\mathcal{G}(\mathbf{X} \theta)_{\text{Bayes-Probabilistic}}$	A probabilistic form of the BDF of $\mathbf{X}$ contained in the hypothesis class $\mathbf{G}(\Theta)$.
$\mathcal{G}(\mathbf{X} \theta)_{\text{Bayes-Strictly Probabilistic}}$	A strictly probabilistic form of the BDF of $\mathbf{X}$ contained in the hypothesis class $\mathbf{G}(\Theta)$.
$\mathcal{G}(\mathbf{X} \theta)_{\text{Bayes-Differential}}$	A differential form of the BDF of $\mathbf{X}$ contained in the hypothesis class $\mathbf{G}(\Theta)$.

Symbol	Meaning
$\mathcal{G}(X \theta)$ <i>Bayes-Strictly Differential</i>	A strictly differential form of the BDF of X contained in the hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}$	The functional basis of the hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta)$	The hypothesis class with functional basis $\mathcal{G}$ and parameter space Θ .
$\mathcal{G}$	The set of all hypothesis classes.
$\mathcal{G}(\Theta\downarrow)$	The hypothesis class (in the set of all hypothesis classes) with the minimum functional complexity necessary to perform a particular pattern recognition task with a specified level of generalization. The generalization of a classifier generated from $\mathcal{G}(\Theta\downarrow)$ with a training sample size of n is measured in terms of its mean-squared discriminant error (MSDE — see definition 3.9).
$\mathcal{G}(\Theta, X)$ <i>proper</i>	The proper parametric model of X (see definition 3.13).
$\mathcal{G}(\Theta)$ <i>Bayes</i>	The set of all BDFs of X in the hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta)$ <i>Bayes-Probabilistic</i>	The set of all probabilistic forms of the BDF of X in the hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta)$ <i>Bayes-Strictly Probabilistic</i>	The set of all strictly probabilistic forms of the BDF of X in the hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta)$ <i>Bayes-Differential</i>	The set of all differential forms of the BDF of X in the hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta)$ <i>Bayes-Strictly Differential</i>	The set of all strictly differential forms of the BDF of X in the hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta\downarrow)$ <i>Bayes</i>	The set of all BDFs of X in the minimum-complexity hypothesis class $\mathcal{G}(\Theta)$ (i.e., the hypothesis class with the least functional complexity necessary for Bayesian discrimination).
$\mathcal{G}(\Theta\downarrow)$ <i>Bayes-Probabilistic</i>	The set of all probabilistic forms of the BDF of X in the minimum-complexity hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta\downarrow)$ <i>Bayes-Strictly Probabilistic</i>	The set of all strictly probabilistic forms of the BDF of X in the minimum-complexity hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta\downarrow)$ <i>Bayes-Differential</i>	The set of all differential forms of the BDF of X in the minimum-complexity hypothesis class $\mathcal{G}(\Theta)$.
$\mathcal{G}(\Theta\downarrow)$ <i>Bayes-Strictly Differential</i>	The set of all strictly differential forms of the BDF of X in the minimum-complexity hypothesis class $\mathcal{G}(\Theta)$.
$\Upsilon[\cdot]$	The general classifier functional complexity measure of section 3.5, page 74.
$\Upsilon_{\max}[\cdot]$	The upper bound on $\Upsilon[\cdot]$ for a particular choice of hypothesis class (see section 3.5).
$\mathbf{H}_Z(f(Z))$	The Hessian (i.e., the matrix of second-order derivatives) of $f(Z)$ with respect to the vector Z .
<i>iff</i>	Read, "... if and only if ..."
$\mathbf{I}$	The identity matrix.
$\mathbf{i}$	The identity vector.
$\mathcal{IF}$	The incorrect fraction of discriminator output space (see definition 5.13).
$\mathcal{IF}_{\text{mono}}$	The monotonic incorrect fraction of discriminator output space (see definition 5.17).
$\Phi \mathcal{IF}_{\text{mono}}(C)$	The monotonic incorrect fraction of discriminator output space associated with the objective function Φ , given a C -dimensional discriminator output space (i.e., a C -class learning task).

Symbol	Meaning
$\mathcal{IF}_{\text{mono}}$	The non-monotonic incorrect fraction of discriminator output space (see definition 5.16).
$L(\cdot)$	A log-likelihood function.
$\mathcal{L}$	The set of all learning strategies.
Λ	The general learning strategy.
Λ_{Δ}	The differential learning strategy (associated with the CFM objective function).
Λ_P	The probabilistic learning strategy (associated with error measure objective functions).
$\Lambda_{P\text{-MSE}}$	Probabilistic learning via the MSE objective function.
$\Lambda_{P\text{-CE}}$	Probabilistic learning via the CE objective function.
$\Lambda_{P\text{-LMS}}$	Probabilistic learning via the LMS objective function (this is identical to learning via the MSE objective function).
$\Lambda_{P\text{-ML}}$	Probabilistic learning via the method of maximum-likelihood.
MAE	The mean absolute error (MAE) objective function (also known as "least absolute error" and "least absolute deviation").
$\text{MAE}(S^n \theta)$	The MAE generated by the training sample S^n , given the discriminator $\mathcal{G}(X \theta)$ with parameterization θ .
$\mathcal{MF}$	The monotonic fraction of discriminator output space (see definition 5.18).
$\Phi \mathcal{MF}(C)$	The monotonic fraction of discriminator output space associated with the objective function Φ , given a C -dimensional discriminator output space (i.e., a C -class learning task).
$\text{MSDE}[\mathcal{G} n, \mathbf{G}(\Theta), \Lambda]$	The mean-squared discriminant error (MSDE) of the classifier generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ , given a training sample size of n (see definition 3.9).
$\widehat{\text{MSDE}}[\mathcal{G} \{n_1, \dots, n_K\}, \mathbf{G}(\Theta), \Lambda]$	The <i>estimated</i> MSDE of the classifier repeatedly generated from the hypothesis class $\mathbf{G}(\Theta)$ by the learning strategy Λ , given K training samples of sizes $\{n_1, \dots, n_K\}$.
MSE	The mean-squared error (MSE) objective function.
$\text{MSE}(S^n \theta)$	The MSE generated by the training sample S^n , given the discriminator $\mathcal{G}(X \theta)$ with parameterization θ .
μ	The mean of a Gaussian-distributed random vector. The notation μ_i generally refers to the mean of the random vector's i th class-conditional Gaussian probability density function.
n_p	The number of examples of the pattern X_p in a training sample of size n .
$n_{p,i}$	The number of examples of the pattern X_p representing class ω_i in a training sample of size n .
ς	A random noise variable.
$\mathbf{S}$	A random noise vector.
ω_i	The i th class (i.e., concept) that a random feature vector can represent.
$\neg \omega_i$	Read, "Not ω_i ."
$\omega_{\cdot}$	The classification assigned to X by the Bayes-optimal classifier (definition 2.1, page 17).
Ω	The domain of the class label $\mathcal{W}$. Sometimes called classification (or class label) space, that is, the set of all class labels with which a feature vector can be paired. For the C -class pattern recognition task, $\mathcal{W} \in \Omega = \{\omega_1, \dots, \omega_C\}$.

Symbol	Meaning
$P_z(\zeta)$	The probability that the random variable (or vector) z will take on the value ζ . This notation is equivalent to the notation $P(z = \zeta)$.
$\hat{P}_z(\zeta)$	An <i>estimate</i> of the probability that the random variable (or vector) z will take on the value ζ .
$P_W(\omega_i)$	The prior probability of class ω_i .
$P_{W X}(\omega_i \mathbf{X})$	The <i>a posteriori</i> probability of class ω_i , given the feature vector $\mathbf{X}$.
$P_e(\mathcal{G}, \theta)$	The error rate (i.e., probability of error) for the classifier with the discriminator $\mathcal{G}(\mathbf{X} \theta)$ (see definition 3.1).
$\hat{P}_e(\mathcal{G} \theta, \eta)$	An <i>estimate</i> of the error rate for the classifier with the discriminator $\mathcal{G}(\mathbf{X} \theta)$; the estimate is based on a test sample size of η (see definition 8.1).
$P_e(\mathcal{F}_{\text{Bayes}})$	The error rate exhibited by the Bayes-optimal classifier of $\mathbf{X}$ (see definition 3.2).
$\hat{P}_e(\mathcal{F}_{\text{Bayes}})$	An <i>estimate</i> of the error rate exhibited by the Bayes-optimal classifier of $\mathbf{X}$.
$\mathbf{R}$	A rotation matrix.
RE	Estimated relative efficiency (see definition 8.6).
$\mathbb{R}$	The set of all real numbers.
$\rho_{\mathbf{X}}(\mathbf{X})$	The probability density function (pdf) of the feature vector $\mathbf{X}$.

Note: When written $\rho_{\mathbf{X}}(\mathbf{X})$, the notation is meant to convey, "the pdf of $\mathbf{X}$ evaluated at some arbitrary value of $\mathbf{X}$;" when written $\rho_{\mathbf{X}}(\mathbf{Z})$, the notation is meant to convey, "the pdf of $\mathbf{X}$ evaluated at $\mathbf{X} = \mathbf{Z}$." We use this same notational convention for other probability measures of $\mathbf{X}$.

$\rho_{\mathbf{X} W}(\mathbf{X} \omega_i)$	The i th class-conditional pdf of $\mathbf{X}$, that is, the pdf of $\mathbf{X}$ when it represents class ω_i .
$\rho_{\mathbf{X}, W}(\mathbf{X}, \omega_i)$	The joint probability density of $\mathbf{X}$ and class ω_i .
s.t.	Read, "... such that ..."
SNR	Signal-to-noise ratio.
S^n	The training sample of size n , that is, the set of n randomly-drawn example/class label pairs $\{(\mathbf{X}^1, \mathcal{W}^1), \dots, (\mathbf{X}^n, \mathcal{W}^n)\}$ used to generate the classifier from its hypothesis class.
$\sigma[\delta, \psi]$	The CFM generated by a discriminant differential of δ when the CFM confidence parameter is ψ . Note: the somewhat smaller notation σ^2 denotes the variance parameter of a Gaussian-distributed random variable; the use is made clear in the text.
Σ	A covariance matrix. The notation Σ_i generally refers to the covariance matrix of the random vector's i th class-conditional probability density function.
τ	Denotes the transpose of a vector (e.g., $\mathbf{X}^T$).
$\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle)$	A target function: when $\mathcal{W}^j = \omega_i$, $\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) = 1$; otherwise $\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) = 0$.
$\boldsymbol{\tau}$	A target vector for the discriminator $\mathcal{G}(\mathbf{X} \theta)$, used when learning probabilistically via an error measure objective function.
$\boldsymbol{\tau}(\langle \mathbf{X}_p^j, \mathcal{W}_p^j \rangle)$	The target vector for the discriminator $\mathcal{G}(\mathbf{X}^j \theta)$, used when learning probabilistically via an error measure objective function. The target class is indicated by $\mathcal{W}^j$.

Symbol	Meaning
θ	A parameter associated with the classifier's discriminator.
θ	The parameter vector of the classifier's discriminator.
θ_0	The discriminator's initial parameterization (i.e., its parameterization prior to learning).
Θ	The domain of the parameter vector θ . Sometimes called parameter space, that is, the set of all parameter vectors that the discriminator can have: $\theta \in \Theta$.
θ^*	A discriminator's parameter vector that is optimal by some objective function.
θ_i^*	The i th element of the optimal parameter vector θ^* .
$\vartheta(\mathbf{X})$	Refer to (2.93), page 46.
$\neg\vartheta(\mathbf{X})$	Refer to (2.93), page 46.
$u(x)$	The Heaviside step function of x (e.g., [80, pg. 258])
$u^+(x)$	The modified Heaviside step function, which is equal to 1 for all $x > 0$, 0 for all $x < 0$, and $\frac{1}{2}$ for all $x = 0$.
$\text{Var}[x]$	The variance (i.e., second central moment) of the random variable x .
$\mathcal{W}$	The stochastic class label associated with the feature vector $\mathbf{X}$.
$\mathcal{W}^j$	The class label associated the j th example of $\mathbf{X}$.
$\mathcal{W}_p^j$	The class label associated with the j th example of the pattern $\mathbf{X}_p$. Note that $\mathcal{W}_p^j$ implies a specific value of $\mathbf{X}$ (i.e., it implies the pattern $\mathbf{X}_p$); $\mathcal{W}^j$ does not.
$\mathcal{X}$	The domain of the feature vector $\mathbf{X}$. Sometimes called feature vector space, that is, the set of all possible feature vectors such that $\mathbf{X} \in \mathcal{X}$.
$\mathbf{X}$	The feature vector (or attribute vector).
$\mathbf{X}^j$	The j th example of $\mathbf{X}$, that is, the j th <i>realization</i> of the random feature vector $\mathbf{X}$.
$\langle \mathbf{X}^j, \mathcal{W}^j \rangle$	The j th example of $\mathbf{X}$, along with its class label.
$\mathbf{X}_p$	A particular <i>pattern</i> (i.e., a particular value of $\mathbf{X}$).
$\mathbf{X}_p^j$	The j th example of $\mathbf{X}_p$, that is, the j th <i>realization</i> of the random feature vector $\mathbf{X}$ having the value $\mathbf{X}_p$. Note that $\mathbf{X}_p^j$ implies a specific value of $\mathbf{X}$ (i.e., it implies the pattern $\mathbf{X}_p$); $\mathbf{X}^j$ does not.
$\langle \mathbf{X}_p^j, \mathcal{W}_p^j \rangle$	The j th example of $\mathbf{X}_p$, along with its class label.
$\Xi(n)$	The number of misclassified examples in S^n , the training sample of size n .
$\Xi(\eta)$	The number of misclassified examples in the test sample of size η .
y_i	Short-hand notation for the i th discriminator output $g_i(\mathbf{X} \theta)$.
y_τ	Given the example/class label pair $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, y_τ is the discriminator output associated with the class $\mathcal{W}^j$ of the example $\mathbf{X}^j$ (see section 2.2.4).
$\bar{y}_\tau$	Given the example/class label pair $\langle \mathbf{X}^j, \mathcal{W}^j \rangle$, $\bar{y}_\tau$ is the largest discriminator output <i>not</i> associated with the class $\mathcal{W}^j$ of the example $\mathbf{X}^j$ (see section 2.2.4).
$\mathbf{Y}$	Short-hand notation for the output state of the classifier's discriminator $\mathcal{G}(\mathbf{X} \theta)$.
$\mathbf{Y}_{\text{correct}}$	The "correct" vertex of discriminator output space (see (5.3)).
$\mathbf{Y}_{\text{incorrect}}$	The "incorrect" vertex of discriminator output space (see (5.2)).
$\mathcal{Y}$	The domain of the discriminator's output $\mathbf{Y}$. Sometimes called discriminator output space (see section 2.2.1). Thus, $\mathbf{Y} \in \mathcal{Y}$.

Symbol	Meaning
$\mathcal{Y}_{correct}$	The correct region (or side) of discriminator output space (see definition 5.8).
$\mathcal{Y}_{incorrect}$	The correct region (or side) of discriminator output space (see definition 5.6).
ψ	The confidence parameter for the classification figure-of-merit (CFM objective function).
$\mathbb{Z}^+$	The set of all positive integers (i.e., all positive natural numbers).

Appendix B

Notes on Convergence

The proofs of chapters 2 and 3 rely on notions of convergence that require some explanation. *Ideally*, we would expect that the statistics of a training sample $S^n = \{ \langle X^1, W^1 \rangle, \dots, \langle X^n, W^n \rangle \}$ reflect the true nature of the random feature vector X (i.e., $\rho_X(X)$, $\{P_W(\omega_1), \dots, P_W(\omega_C)\}$, and $\{P_{W|X}(\omega_1 | X), \dots, P_{W|X}(\omega_C | X)\}$) in the limit that the training sample size grows infinitely large (i.e., $n \rightarrow \infty$). Moreover, we would expect that this convergence of the empirical probability measures to the true measures would, for each and every asymptotically large training sample, occur with certainty (i.e., convergence with probability one) and uniformly over all feature vector space $\mathcal{X}$ (i.e., convergence at some non-zero rate would be guaranteed for all $X \in \mathcal{X}$ at which $\rho_X(X)$ is defined).

In fact the empirical cumulative distribution function (cdf) of the arbitrary random variable x does, in general, uniformly converge to the true underlying cdf with probability one. The Glivenko-Cantelli Theorem (e.g., [105, sec. II.3]) proves this expression of the uniform strong law of large numbers.¹ Vapnik describes an extension of the theorem to the general N -dimensional random feature vector X in [136, ch. 6]; we assume that X exhibits uniform convergence with probability one accordingly.

¹ See [28, sec. 9.6] for an concise, readable summary of the Glivenko-Cantelli Theorem.

Appendix C

The Box Plot Statistical Summary

The box plot is a non-parametric statistical summary developed by John W. Tukey [131, ch. 2].¹ Given a sample $S^n = \{x_1, \dots, x_n\}$ of the random variable x , the box plot is a concise graphical summary of the empirical low-order moments of x — one that makes no assumptions about the probability density function (pdf) of x .

C.1 How to Read a Box Plot

Figure C.1 shows an annotated box plot for a hypothetical sample S^n of x . Note that all the examples of S^n fall well within the range of $30 \leq x \leq 100$. The box plot is formed by sorting all the examples and dividing them into “quartiles” (i.e., into four groups, each of which represents 25% of S^n). The box itself encompasses the middle 50% of S^n . The top 25% of S^n is depicted by the vertical line extending above the box, and the bottom 25% of S^n is depicted by the vertical line extending below the box. The box is divided by a horizontal line at the median of S^n . The inner and (if shown) outer “T”-shaped “fences” of each plot depict the nominal lower bound of the first quartile and nominal upper bound of the fourth quartile. Any extreme first/fourth quartile values falling beyond the outer fence(s) are plotted as dots. The box plot therefore displays all of the data, emphasizing the median and a quartile partitioning of the sample.

The box plot has a number of advantages as a statistical graphic:

- It is simple to compute and display.
- It makes no assumption about the pdf $\rho_x(x)$ of x .
- It is generally a more meaningful graphic than alternatives such as histograms, whisker plots, e.g., when the sample size (i.e., n) is small [131, ch. 2].
- One can easily infer the low order central/non-central moments of x from the box plot.

¹Tukey is well-known for the Cooley-Tukey fast fourier transform (FFT) algorithm and his work with Blackman in spectral estimation.

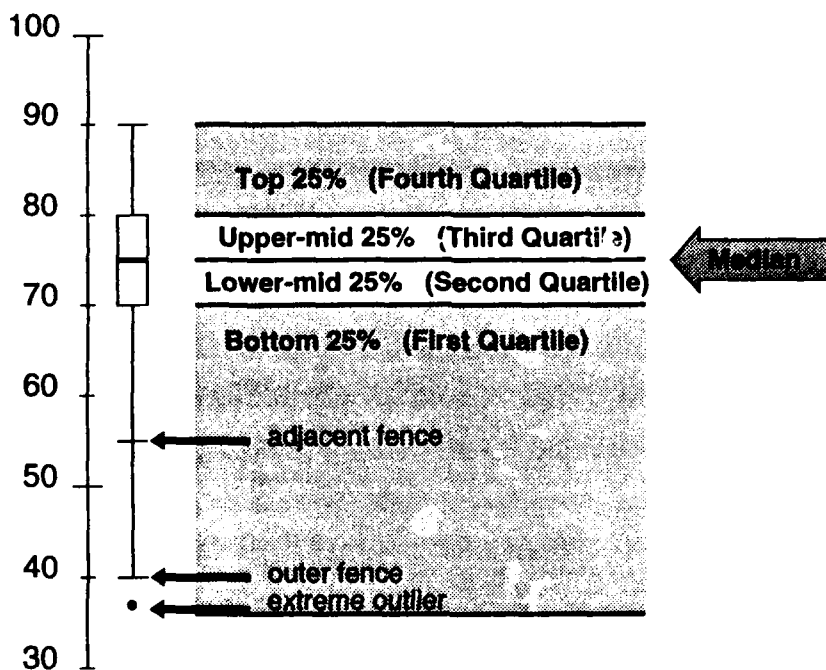


Figure C.1: A box plot for a sample of the random variable x .

From figure C.1 we can see that the median value for S^n is 75. The middle 50% of the sample is fairly tightly concentrated about the median on the interval [70, 80]. The bottom 25% of the sample (i.e., the first quartile) spans the interval [37, 70], and there is an extreme statistical outlier at 37. In contrast, the top 25% (i.e., the fourth quartile) is more tightly distributed on the interval [80, 90]. Thus, the box plot indicates that the empirical distribution of x is skewed towards higher values of x . It should be clear from the figure that the box plot gives the observer a concise non-parametric sketch of the median, variance, skewness, and kurtosis of x . Specifically, the height of the box and the length of its fences are an indication of the variance in the classifier's error rate over all trials; the symmetry of the box plot (or lack thereof) is an indication of the skewness; and the height of the box in comparison to the length of the fences is an indication of kurtosis (i.e., how abruptly the sample peaks about its median).

The computations by which the box plot is constructed from a data sample are detailed in [131, ch. 2]. We provide a summary of them in the following section. *We emphasize that it is not necessary to understand the following material in order to interpret a box plot; we provide it for the convenience of those who wish to know precisely how the box plot fences are constructed.*

C.2 How to Construct a Box Plot²

The first step in constructing a box plot is to sort the sample S^n . From this sorted version of S^n (which we will denote by S_x^n) we develop a "5-number summary": it comprises the lower extreme value (low), the first quartile boundary (Q1), the median (med), the third quartile boundary (Q3), and the upper extreme value (high) of the sample. If $x_{(i)}$ denotes the $(n - i + 1)$ th ranked example of S_x^n (i.e., $x_{(1)}$ denotes the lower extreme example and $x_{(n)}$ denotes the upper extreme example) then the indices of the examples that we use to compute the five number summary are obtained from the following five real numbers (note that the $\lfloor z \rfloor$ operator returns the largest integer not greater than z , and the $\lceil z \rceil$ operator returns the smallest integer not less than z):

$$\begin{aligned} i[\text{low}] &= 1 \\ i[\text{med}] &= \frac{n+1}{2} \\ i[\text{high}] &= n \\ i[\text{Q1}] &= \frac{1}{2} \cdot [\lfloor i[\text{med}] \rfloor + 1] \\ i[\text{Q3}] &= n + 1 - i[\text{Q1}] \end{aligned}$$

The resulting five number summary is given by

$$\begin{aligned} \text{low} &= x_{(i[\text{low}])} = x_{(1)} \\ \text{med} &= f(i[\text{med}]) \\ \text{high} &= x_{(i[\text{high}])} = x_{(n)} \\ \text{Q1} &= f(i[\text{Q1}]) \\ \text{Q3} &= f(i[\text{Q3}]) \end{aligned}$$

where Z^+ denotes the set of all positive integers, and

$$f(i[\text{number}]) = \begin{cases} x_{(i[\text{number}])}, & i[\text{number}] \in Z^+ \\ \frac{1}{2} \cdot [x_{(\lfloor i[\text{number} \rfloor)} + x_{(\lceil i[\text{number} \rceil)}], & \text{otherwise} \end{cases}$$

Table C.1 lists the indices of the sorted RVs that we would use to compute the five number summary for various sample sizes (i.e., various ns).

²Adapted from the original by Tukey [131, ch. 2].

Indices of $x_{(i)}$					
n	low	Q1	med	Q3	high
3	1	1	2	3	3
4	1	1,2	2,3	3,4	4
5	1	2	3	4	5
6	1	2	3,4	5	6
7	1	2,3	4	5,6	7
8	1	2,3	4,5	6,7	8
9	1	3	5	7	9
10	1	3	5,6	8	10
11	1	3,4	6	8,9	11
12	1	3,4	6,7	9,10	12
13	1	4	7	10	13
14	1	4	7,8	11	14
15	1	4,5	8	11,12	15
			.		
			.		
			.		

Table C.1: A listing of indices i for $x_{(i)}$ used to compute box plot 5-number summaries of S^n for various sample sizes (i.e., various ns).

Once we have computed the 5-number summary we have most of the box plot built (i.e., we know the location of the box and its median, as well as the upper and lower extreme values). The only remaining computations are those for the locations of the adjacent and outer fences for the first and fourth quartiles of S^n . In short, the adjacent fence locations are displaced from their quartile boundary by no more than 1.5 times the distance between the first and third quartiles. Likewise, the outer fence locations are displaced from their quartile boundary by no more than 3 times the distance between the first and third quartiles. The displacement of a fence from its respective quartile boundary *never* exceeds the location of the extreme example in the fence's quartile. This explains why some box plots appear to have missing fences (as is the case for the fourth quartile data of figure C.1); in reality the fences coincide with other fences or the quartile boundaries, so they do not appear in the plot.

Quantitatively, the first quartile fences are given by

$$\begin{aligned} \text{lower adjacent fence} &= \begin{cases} Q1 - 1.5[Q3 - Q1], & x_{(1)} < Q1 - 1.5[Q3 - Q1] \\ x_{(1)}, & \text{otherwise} \end{cases} \\ \text{lower outer fence} &= \begin{cases} Q1 - 3.0[Q3 - Q1], & x_{(1)} < Q1 - 3.0[Q3 - Q1] \\ x_{(1)}, & \text{otherwise} \end{cases} \end{aligned}$$

The fourth quartile quartile fences are given by

$$\begin{aligned}\text{upper adjacent fence} &= \begin{cases} Q3 + 1.5[Q3 - Q1], & x_{(n)} > Q3 + 1.5[Q3 - Q1] \\ x_{(n)}, & \text{otherwise} \end{cases} \\ \text{upper outer fence} &= \begin{cases} Q3 + 3.0[Q3 - Q1], & x_{(n)} > Q3 + 3.0[Q3 - Q1] \\ x_{(n)}, & \text{otherwise} \end{cases}\end{aligned}$$

(C.1)

For the case in which x is normally distributed (i.e., $x \sim N(\mu, \sigma^2)$), the adjacent and outer fences are displaced two and four standard deviations from the quartile boundary respectively³.

Again, all examples in the sample falling outside the upper and lower outer fences are extreme outliers, which are represented by individual "•" symbols.

³ Assuming that the sample size n is large so that S^n is representative of x .

Appendix D

A Synthetic Functional Form of the Classification Figure of Merit

This appendix describes in detail the synthetic asymmetric function we employ for the classification figure-of-merit. Our use of this functional form has a two-fold motivation:

- Chapter 2 shows that differential learning requires a sigmoidal CFM function with variable “steepness”. However, the logistic sigmoidal form originally described in [55] is symmetric: when it has a steep transition region its derivative is essentially zero outside of that region. As described in section D.3, this leads to very small gradients in the search algorithm used to find the optimal parameterization of the classifier. This in turn leads to unreasonably slow learning. In order to overcome the problem, we desire an asymmetric sigmoid that retains a significant non-zero first derivative for yet un-learned training examples (i.e., those with negative discriminant differentials δ) — even for the case in which the sigmoid is steep in its transition region.
- We require a mathematically simple synthetic form in order to minimize the number of floating point computations necessary to evaluate the function and its first and second derivatives.

In section D.1 we specify the synthetic form of the CFM objective function; in section D.2 we analyze the computational requirements posed by its evaluation and that of its first two derivatives; in section D.3 we analyze the convergence properties it engenders (i.e., how fast differential learning is, using synthetic CFM), and on this basis contrast it with the original logistic sigmoidal form of CFM; in section D.4 we derive an upper bound on the synthetic CFM confidence parameter ψ that guarantees Bayes-optimal discrimination, as described in in section 2.4; in section D.6 we list ANSI-C source code for the synthetic form and its first two derivatives.

We wish to emphasize that our development of the synthetic CFM objective function was motivated by palpable deficiencies in the original logistic sigmoidal form. The deficiencies relate primarily to the poor convergence properties and instability of differential learning via the original form of CFM, which we detail

in section D.3. Some readers might find this appendix — and section D.3 in particular — rather abstract and pedantic. We encourage such persons to recognize the cause-and-effect relationship here: the problems associated with differential learning via the original logistic sigmoidal CFM objective function led to the theory, rather than vice-versa. The details herein were (and remain) a necessary evil on the path to an implementation of differential learning that works in practice as well as it does in theory.

D.1 Specifications for the Synthetic CFM Objective Function

We create a piece-wise linear sigmoid by connecting three line segments with two arcs (we abuse notation by referring to these arcs in terms of their radii). This synthesis is depicted in figure D.1. The lower radius r_n is generated by a circle with a centroid (μ_m, μ_n) , which is constrained to lie on line segment A; the radius is also constrained to be tangent to line segment B. The upper radius r_p is generated by a circle with a centroid (μ_p, μ_n) , which is constrained to lie on line segment C; the radius is constrained to be tangent to the horizontal line of unit height.¹ A line drawn from point $(-1, 0)$ to point (x_{mn}, y_{mn}) (the latter of which is tangent to the lower radius) forms the lower “leg” of the sigmoid. A line drawn from point (x_m, y_m) to point (x_p, y_p) (points that are tangent to the lower and upper radii, respectively) forms the transition region of the sigmoid. A line drawn from point $(\mu_p, 1)$ to point $(1, 1)$ (the former of which is tangent to the upper radius) forms the upper leg of the sigmoid. This upper leg always has a value of one and a slope of zero. The steepness of the sigmoid is increased by moving the centroids of the two circles toward $\delta = 0$ along lines A and C. Conversely, the steepness is decreased by moving the centroids of the two circles away from $\delta = 0$ along lines A and C. Since the lower radius r_n is constrained always to be tangent to line segment B and the upper radius r_p is constrained always to be tangent to the horizontal line of unit height, the radii are proportional to their centroids’ horizontal distances from the vertical line at $\delta = 0$. In the limit that these centroid distances are zero (corresponding to a confidence parameter ψ of zero), $\sigma[\delta, \psi]$ is a Heaviside step function. In the limit that these centroid distances are their maximum values (corresponding to $\psi = 1$), $\sigma[\delta, \psi]$ is a nearly linear function of δ when $\delta \leq 1$, otherwise it assumes its maximum value of unity. Figure 2.6 shows $\sigma[\delta, \psi]$ for eight different values of its (single) confidence parameter ψ .

Recall from section 2.2.4 that the synthetic CFM objective function must satisfy the following constraints:

- I. The function must have finite lower and upper bounds l and h :

$$-\infty \ll l \leq \sigma[\delta, \psi] \leq h \ll \infty \quad (\text{D.1})$$

¹The parameters that specify line segments A, B, and C in figure D.1 were chosen by the author using a graphic tool designed for this express purpose. The qualitative design criterion for the synthetic function were 1) that it retain a significant non-zero slope, even when its transition region becomes steep, and 2) that the arcs connecting the three line segments of the function have reasonably large radii for all but very steep transition regions. This latter characteristic ensures relatively small higher-order derivatives for the synthetic function at the arc segments.

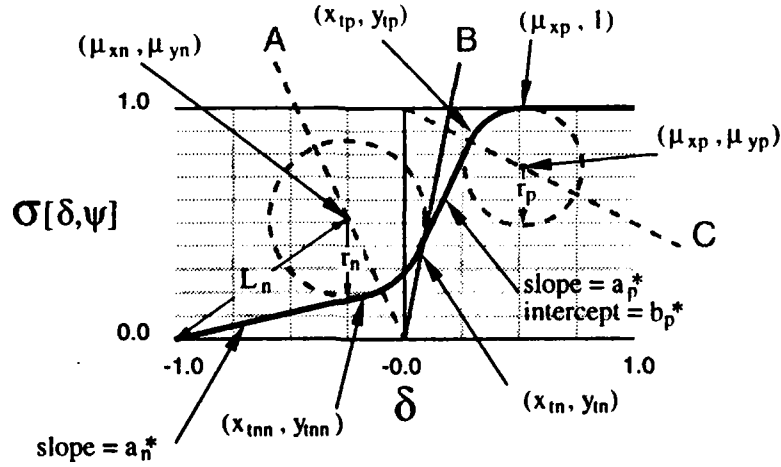


Figure D.1: Details of the synthetic asymmetric sigmoidal form of the classification figure-of-merit (CFM). This synthetic function is shown for various confidence parameter (ψ) values in figure 2.6.

The synthetic function is bounded on $[l = 0, h = 1]$ for $-1 \leq \delta \leq 1$, so it satisfies this constraint for classifiers with outputs bounded on $[0, 1]$. Since any classifier's output state can be normalized to the interval $[0, 1]$ by a simple affine transformation, this synthetic function can be used with any classifier.

2. The function must be a strictly non-decreasing sigmoidal function of δ :

$$\left\{ \begin{array}{ll} \frac{d}{d\delta} \sigma[\delta, \psi] > 0, & \text{for small } |\delta| \\ \frac{d}{d\delta} \sigma[\delta, \psi] \geq 0, & \text{otherwise} \end{array} \right\} \quad (\text{D.2})$$

Equation (D.8) and figures 2.6 and D.1 confirm that the synthetic function has this property.

3. The function must have a maximum slope occurring in its transition region. This transition slope should be inversely proportional to the confidence parameter ψ :

$$\max_{\delta} \frac{d}{d\delta} \sigma[\delta, \psi] \propto \psi^{-1}, \quad \psi \in (0, 1] \quad (\text{D.3})$$

By inspection of figure D.1, it is clear that $\max_{\delta} \frac{d}{d\delta} \sigma[\delta, \psi] = a_p^*$. The constraint that a_p^* be proportional to ψ^{-1} ensures that the function's derivative in the transition region is bounded for all non-zero values of ψ . Section D.3 confirms that the synthetic function has this property.

4. The lower leg of the sigmoidal function must have a positive slope, which should be linearly

proportional to ψ :

$$\min_{\delta < 0} \frac{d}{d\delta} \sigma[\delta, \psi] \propto \psi, \quad \psi \in (0, 1] \quad (D.4)$$

This constraint ensures that the derivative of the function retains a significant positive value for negative values of δ , as long as ψ is greater than zero. This, in turn, ensures that gradient-based searches used to optimize the parameters of the classifier by maximizing CFM do not exhibit exponentially long convergence times as the steepness of the sigmoidal function's transition region grows large. Section D.3 explains this property in more detail and confirms that the synthetic function has it.

5. The sigmoidal function should span a continuum between an approximately linear function of δ for $\psi = 1$ to a step function of δ for $\psi \rightarrow 0^+$:

$$\begin{aligned} \lim_{\psi \rightarrow 1} \sigma[\delta, \psi] &\approx a_0 \delta + b_0, \\ \lim_{\psi \rightarrow 0^+} \sigma[\delta, \psi] &= a_1 u^+(\delta) + b_1 \end{aligned} \quad (D.5)$$

where a_0 , b_0 , a_1 , and b_1 are constants and

$$u^+(\delta) = \begin{cases} 0, & \delta \leq 0 \\ 1, & \delta > 0 \end{cases} \quad (D.6)$$

Equations (D.7) and (D.10) – (D.46) and figures 2.6 and D.1 confirm that the synthetic function has this property.

Section D.6 lists the source code that implements this synthetic form of the CFM objective function. The precise mathematical expressions for $\sigma[\delta, \psi]$ and its first two derivatives are

$$\sigma[\delta, \psi] = \begin{cases} a_n^*(\delta + 1), & -1 \leq \delta \leq x_{mn} \\ \mu_{ym} - \sqrt{r_n^2 - (\delta - \mu_{xm})^2}, & x_{mn} < \delta < x_m \\ a_p^* \delta + b_p^*, & x_m \leq \delta \leq x_{mp} \\ \mu_{yp} + \sqrt{r_p^2 - (\delta - \mu_{xp})^2}, & x_{mp} < \delta < \mu_{xp} \\ 1, & \delta \geq \mu_{xp} \end{cases} \quad (D.7)$$

$$\frac{d}{d\delta} \sigma [\delta, \psi] = \begin{cases} \underbrace{a_n^*}_{\geq 0}, & -1 \leq \delta \leq x_{mn} \\ \underbrace{[r_n^2 - (\delta - \mu_{xm})^2]^{-1/2}}_{>0} \underbrace{(\delta - \mu_{xm})}_{>0}, & x_{mn} < \delta < x_{in} \\ \underbrace{a_p^*}_{>0}, & x_{in} \leq \delta \leq x_{ip} \\ - \underbrace{[r_p^2 - (\delta - \mu_{xp})^2]^{-1/2}}_{<0} \underbrace{(\delta - \mu_{xp})}_{<0}, & x_{ip} < \delta < \mu_{xp} \\ 0, & \delta \geq \mu_{xp} \end{cases} \quad (D.8)$$

$$\frac{d^2}{d\delta^2} \sigma [\delta, \psi] = \begin{cases} 0, & -1 \leq \delta \leq x_{mn} \\ [r_n^2 - (\delta - \mu_{xm})^2]^{-1/2} \cdot [(\delta - \mu_{xm})^2 \cdot [r_n^2 - (\delta - \mu_{xm})^2]^{-1} + 1], & x_{mn} < \delta < x_{in} \\ 0, & x_{in} \leq \delta \leq x_{ip} \\ - [r_p^2 - (\delta - \mu_{xp})^2]^{-1/2} \cdot [(\delta - \mu_{xp})^2 \cdot [r_p^2 - (\delta - \mu_{xp})^2]^{-1} + 1], & x_{ip} < \delta < \mu_{xp} \\ 0, & \delta \geq \mu_{xp} \end{cases} \quad (D.9)$$

Each time ψ is changed, the following computations (listed in regressive order) must be performed to update the synthetic function (all angles are in radians):

$$b_p^* = y_{ip} - a_p^* x_{ip} \quad (D.10)$$

$$x_{ip} = \mu_{xp} + r_p \cos(\angle_1) \quad (D.11)$$

$$y_{ip} = \mu_{yp} + r_p \sin(\angle_1) \quad (D.12)$$

$$x_{in} = \mu_{xm} - r_n \cos(\angle_1) \quad (D.13)$$

$$\angle_1 = \frac{\pi}{2} + \angle_p \quad (D.14)$$

$$\alpha_p^* = \tan(\angle_p) \quad (\text{D.15})$$

$$\angle_p = \tan^{-1} \left(\frac{\mu_{yp} - \mu_{yn}}{\mu_{xp} - \mu_{xm}} \right) + \sin^{-1} \left(\frac{r_p + r_n}{D_1} \right) \quad (\text{D.16})$$

$$D_1 = \sqrt{(\mu_{xp} - \mu_{xm})^2 + (\mu_{yp} - \mu_{yn})^2} \quad (\text{D.17})$$

$$\alpha_n^* = \tan(\angle_3) \quad (\text{D.18})$$

$$x_{mn} = D_3 \cos(\angle_3) - 1 \quad (\text{D.19})$$

$$\angle_3 = \tan^{-1} \left(\frac{\mu_{yn}}{\mu_{xm} + 1} \right) - \sin^{-1} \left(\frac{r_n}{L_n} \right) \quad (\text{D.20})$$

$$D_3 = \sqrt{L_n^2 - r_n^2} \quad (\text{D.21})$$

$$L_n = \sqrt{(\mu_{xm} + 1)^2 + \mu_{yn}^2} \quad (\text{D.22})$$

$$r_n = R_n \cdot \psi' \quad (\text{D.23})$$

$$\mu_{xm} = -\zeta_n \cos(\angle_4) \quad (\text{D.24})$$

$$\mu_{yn} = \zeta_n \sin(\angle_4) \quad (\text{D.25})$$

$$\mu_{xp} = -\frac{r_p}{a_0} = \psi' \quad (\text{D.26})$$

$$\mu_{yp} = 1 - r_p \quad (\text{D.27})$$

$$\zeta_n = R_n \cdot \psi' \quad (\text{D.28})$$

$$r_p = R_p \psi' \quad (\text{D.29})$$

The following quantities are constants (all angles are in radians):

$$\angle_4 = \tan^{-1} \left(\frac{\mu_{xm0}}{-\mu_{ym0}} \right) \quad (\text{D.30})$$

$$R_n = \sqrt{\mu_{xm0}^2 + \mu_{ym0}^2} \quad (\text{D.31})$$

$$\mu_{xm0} = x_1 - R_n \cos\left(\frac{\pi}{2} - \angle_6\right) \quad (\text{D.32})$$

$$\mu_{ym0} = y_1 + R_n \sin\left(\frac{\pi}{2} - \angle_6\right) \quad (\text{D.33})$$

$$x_1 = x_0 + L \sin(\angle_5) \quad (\text{D.34})$$

$$y_1 = y_0 + L \cos(\angle_5) \quad (\text{D.35})$$

$$x_0 = \frac{a_2}{a_1 - a_2} \quad (\text{D.36})$$

$$y_0 = \frac{a_1 a_2}{a_1 - a_2} \quad (\text{D.37})$$

$$L = \frac{R_n}{\tan \angle_7} \quad (\text{D.38})$$

$$\angle_5 = \tan^{-1}(a_2) \quad (\text{D.39})$$

$$\angle_6 = \tan^{-1}(a_1) \quad (\text{D.40})$$

$$\angle_7 = \frac{\pi}{2} - \frac{\tan^{-1}(a_1) - \tan^{-1}(a_2)}{2} \quad (\text{D.41})$$

$$R_n = 0.7 \quad (\text{i.e., } r_n | \psi^i = 1) \quad (\text{D.42})$$

$$a_2 = 0.5 \quad (\text{D.43})$$

$$a_1 = 5.0 \quad (\text{D.44})$$

$$R_p = -a_0 = 0.5 \quad (\text{i.e., } r_p | \psi^i = 1) \quad (\text{D.45})$$

$$a_0 = -0.5 \quad (\text{D.46})$$

D.2 The Computational Cost of the Synthetic CFM Objective Function

Since the steepness of the sigmoid is adjusted infrequently,² evaluation of this synthetic function and its first two derivatives involves few floating point computations, as indicated by (D.7) – (D.9). The function is evaluated by comparing its argument with the intervals on δ corresponding to the three line segments and two radii. In the case that the argument corresponds to a line segment, the function evaluation requires one multiplication and one addition, and its derivative evaluations require a constant look-up. In the case that the argument corresponds to a radius, the function evaluation requires one multiplication, three additions, and one square root computation; its first derivative's evaluation requires two multiplications, two additions, and one square root computation; its second derivative's evaluation requires four multiplications, three additions, and one square root computation. Thus, the computational cost of evaluating this synthetic function and its first two derivatives is comparable to the cost of evaluating the logistic sigmoidal form of CFM [55] (see section D.3.1) and its first two derivatives.

D.3 The Convergence Properties of Differential Learning via the CFM Objective Function

As described by definitions 2.8 and 2.10, the differentiable supervised classifier employing differential learning learns by maximizing the CFM objective function via a search (i.e., a numerical optimization procedure) on parameter space. Regardless of the search algorithm's specific characteristics (e.g., [106, ch. 10]), it uses the first derivative of the objective function in order to update the classifier's parameters iteratively. The magnitude of the parameter change induced by each iteration of the search — that is, the rate at which the classifier learns — is proportional to the objective function's first derivative (see section 5.5). The magnitude of this derivative, in turn, is sigmoidally related to the discriminant differential engendered by the training example. This leads us to define three classes of training examples on the basis of the discriminant differentials they engender. The following definitions are illustrated in figure D.2.

Definition D.1 Un-learned example: *This is a training example that exhibits a negative discriminant differential (i.e., one that the classifier does not classify correctly).*

Definition D.2 Learned example: *This is a training example that exhibits a positive discriminant differential (i.e., one that the classifier does classify correctly). The magnitude of the discriminant differential*

²Again, this adjustment is made by altering the confidence parameter $\psi^i \in (0, 1]$ of the function. Such an alteration requires the re-computation of the radii and their squares, the radius centroids, tangent points, and linear coefficients shown in figure D.1. Once these values are computed via equations (D.10) – (D.29), they need not be re-computed until and unless the confidence parameter is changed again.

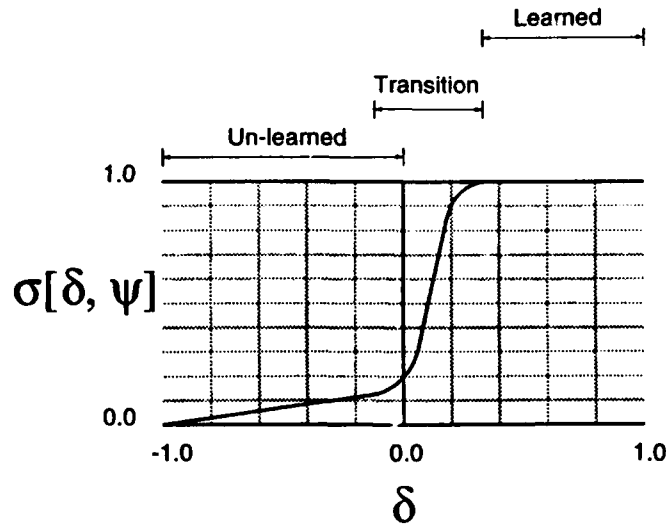


Figure D.2: Three types of training examples: un-learned examples exhibit negative discriminant differentials; transition examples exhibit discriminant differentials that correspond to the transition region of the synthetic CFM sigmoid (therefore, some un-learned examples are also transition examples); learned examples have positive differentials that correspond to the maximum CFM value of unity.

δ is large enough that the CFM it elicits is the maximum value of unity ($\sigma[\delta, \psi] = 1$). Thus, the minimum discriminant differential that a learned example can exhibit depends on the confidence parameter ψ of the CFM objective function; this minimum value of δ is μ_{ψ} (see figure D.1).

Definition D.3 Transition example: This is a training example that exhibits a discriminant differential δ (either positive or negative) for which $\sigma[\delta, \psi]$ is in the transition region of the sigmoidal function.

Remark: Note that an unlearned example may also be a transition example.

If we accept the differential notion of learning for statistical pattern recognition as detailed in chapter 2, then un-learned and transition training examples are the only ones that concern us. That is, a training example is either learned (by definition D.2) or it is not, and we are concerned only with those that are not. The convergence properties of differential learning via CFM follow from an analysis of the rate at which these yet un-learned examples are learned (i.e., the rate at which they are transformed to learned examples, as defined above). We wish this learning to proceed at a reasonable rate, so we must avoid unreasonably slow learning.

Definition D.4 Unreasonably slow learning strategy: Since the rate at which a training example is learned is proportional to $\frac{d}{d\delta}\sigma[\delta, \psi]$, where δ is the discriminant differential elicited by the example, transition examples have the highest learning rate. We denote the ratio of $\frac{d}{d\delta}\sigma[\delta, \psi]$ for transition examples to $\frac{d}{d\delta}\sigma[\delta, \psi]$ for unlearned examples by $\phi(\psi)$. If $\phi(\psi)$ increases exponentially with decreasing ψ ,

learning becomes dominated by the transition examples for small ψ : the classifier's parameters are updated to transform the transition examples into learned examples, while the un-learned examples are ignored (because the derivatives they elicit are so small in comparison to those of the transition examples). Under these circumstances, it takes an unreasonably long time (e.g., [58, pp. 155-158]) to learn the yet un-learned training examples, and we characterize the (differential) learning strategy as unreasonably slow.

Definition D.5 Reasonably fast learning strategy: A learning strategy that is not unreasonably slow by definition D.4 is reasonably fast.

D.3.1 The Convergence Properties Differential Learning via the Original Logistic Sigmoidal Form of CFM

Figure D.3 shows the original logistic sigmoid functional form used for the CFM objective function [55]:

$$\sigma[\delta, \beta] = \alpha [1 + \exp(-\beta\delta + \varsigma)]^{-1} \quad (\text{D.47})$$

The linear scaling parameter α is generally taken to be unity, the parameter ς sets the horizontal offset of the sigmoid's transition region, and the parameter $1 \leq \beta < \infty$ sets the steepness of the sigmoid's transition region. Note that β in this original functional form is proportional to the inverse of the synthetic function's confidence parameter:

$$\beta \propto \psi^{-1} \quad (\text{D.48})$$

— a relationship that validates definitions D.1 – D.5 for the logistic sigmoidal form as well as the synthetic form of CFM. From (D.47) it is straightforward to prove that the first two derivatives of $\sigma[\delta, \beta]$ with respect to δ are given by

$$\frac{d}{d\delta} \sigma[\delta, \beta] = \beta \sigma[\delta, \beta] \left(1 - \frac{\sigma[\delta, \beta]}{\alpha} \right) \quad (\text{D.49})$$

and

$$\frac{d^2}{d\delta^2} \sigma[\delta, \beta] = \beta^2 \sigma[\delta, \beta] \left(1 - \frac{\sigma[\delta, \beta]}{\alpha} \right) \left(1 - 2 \frac{\sigma[\delta, \beta]}{\alpha} \right) \quad (50)$$

Recall that a negative discriminant differential δ indicates a misclassified training example. An objective function with a non-zero first derivative for negative differentials is therefore essential to reasonably fast learning. We know from chapter 2 that the CFM objective function must sometimes approximate a step function in order to guarantee that the classifier approximates the error rate of the Bayesian discriminant function as closely as possible. The original logistic sigmoidal form of CFM shown in figure D.3 has a very

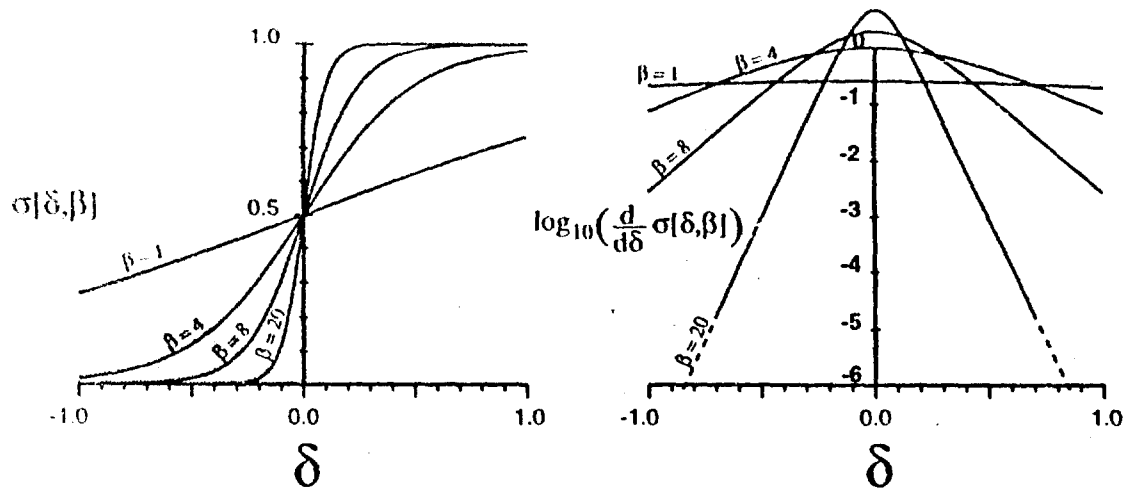


Figure D.3: Left: The original logistic sigmoidal form of the CFM objective function for four values of the steepness parameter β (figure adapted from [55]). The differential learning rate decreases exponentially with increasing β for all training examples that generate discriminant differentials (δ s) in the gray shaded region of the plot (see lemma D.1). This region varies with β : it comprises all values of δ to the left of the point at which the plot for a given value of β intersects with the shaded background. Right: The function's first derivative with respect to the discriminant differential δ for the same four values of β .

small first derivative (right side of the figure) for negative discriminant differentials when it approximates the step function.³ As implied in definition D.4, this prevents the search algorithm at the heart of the learning strategy from converging in time that is a polynomial function of the steepness parameter β . That is, the first derivative of the original logistic sigmoidal form of the CFM objective function decreases in exponential proportion to increasing β for $\delta < 0$. As a direct result, the learning rate of any search algorithm relying on this first derivative decreases exponentially with increasing β for $\delta < 0$.

Lemma D.1 *The rate of differential learning via the original logistic sigmoidal form of the CFM objective function generally decreases as $\mathcal{O}[\zeta^{-\beta}]$ ($\zeta > 1$, $\beta \in [1, \infty]$) for un-learned examples.*

Proof : We fix the parameters $\alpha = 1$ and $\varsigma = 0$ in (D.47) without loss of generality. By (D.47) and (D.49)

$$\frac{d}{d\delta} \sigma[\delta, \beta] = \frac{\beta}{\exp(\beta\delta) (1 + 2 \exp(-\beta\delta) + \exp(-2\beta\delta))}$$

³ Obviously, the logistic form of CFM also has a very small first derivative for positive discriminant differentials when it approximates the step function. However, positive discriminant differentials correspond to *learned* training examples when CFM approximates the step function. We are not particularly concerned with learned examples; rather we are concerned primarily with *un-learned* examples, which exhibit *negative* discriminant differentials. If the objective function has very small derivatives for negative differentials, it will take unreasonably long to learn the yet un-learned examples. An asymmetric sigmoidal form for CFM exhibits very small derivatives for relatively large positive differentials, so it effectively ignores training examples that have already been learned. At the same time, the asymmetric form retains sizable derivatives for all negative differentials, thereby focussing on learning the yet-unlearned examples.

$$= \frac{\beta}{2 + \exp(\beta\delta) + \exp(-\beta\delta)} \quad (\text{D.51})$$

If this derivative decreases exponentially with increasing β , then there must exist some constant $\zeta > 1$ for which

$$\frac{\beta}{2 + \exp(\beta\delta) + \exp(-\beta\delta)} \leq \zeta^{-\beta} \quad (\text{D.52})$$

or

$$\frac{\ln(2 + \exp(\beta\delta) + \exp(-\beta\delta)) - \ln(\beta)}{\beta} \geq \underbrace{\ln(\zeta)}_{>0} \quad (\text{D.53})$$

Since $\ln(2 + \exp(\beta\delta) + \exp(-\beta\delta)) > \underbrace{\ln(\exp(-\beta\delta))}_{-\beta\delta}$ (D.53) is satisfied if

$$\delta < -\frac{\ln(\beta)}{\beta} \quad (\text{D.54})$$

The bound is tight for large β , but loose when $|\beta\delta| \approx 1$. Thus, the first derivative of the logistic sigmoidal form of CFM decreases exponentially for increasing β when the discriminant differential is less than the upper bound given by the right-hand side of (D.54). This bound is plotted in figure D.4. The left side of figure D.3 also depicts the bound: it is the point at which $\sigma[\delta, \beta]$ intersects the gray shaded background. This leads us to conclude that the first derivative of the logistic sigmoidal form of CFM generally decreases exponentially with increasing β :

$$\frac{d}{d\delta} \sigma[\delta, \beta] = \mathcal{O}[\zeta^{-\beta}] \quad \forall \delta < -\frac{\ln(\beta)}{\beta}; \quad \zeta > 1 \quad (\text{D.55})$$

Note that the minimum of $-\frac{\ln(\beta)}{\beta}$ occurs at $\exp(-1) \approx -.368$, so (D.55) holds for all $\delta < -.368$, regardless of the value of β . Since the learning rate for yet un-learned examples is proportional to $\frac{d}{d\delta} \sigma[\delta, \beta]$, the theorem is proven. ■

Lemma D.2 *The rate of differential learning via the original logistic sigmoidal form of the CFM objective function generally increases as $\mathcal{O}[\beta]$ ($\beta \in [1, \infty]$) for transition examples.*

Proof : We fix the parameters $\alpha = 1$ and $\zeta = 0$ in (D.47) without loss of generality. By inspection of (D.50), we solve for the value of δ that yields a CFM second derivative of zero; this occurs at $\delta = 0$ for all choices of β :

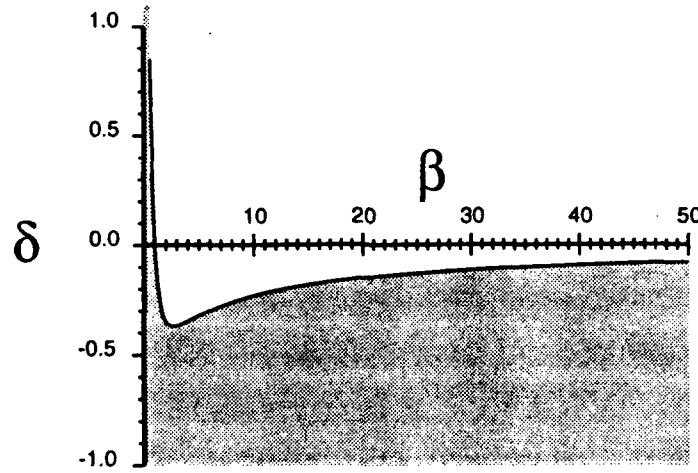


Figure D.4: The logistic sigmoidal form of the CFM objective function has a first derivative $\frac{d}{d\delta} \sigma[\delta, \beta]$ (see figure D.3, right) that decreases exponentially with increasing steepness parameter β when the discriminant differential δ of the classifier falls below the upper-bound value shown above. Note that the upper bound on δ varies with β : in no case is it less than $\exp^{-1} \approx .368$. Figure D.3 (left) plots this upper bound value of δ in light gray for all $\beta \geq 1$.

$$\frac{d^2}{d\delta^2} \sigma[\delta = 0, \beta] = 0 \quad \forall \beta \quad (\text{D.56})$$

By (D.51),

$$\frac{d^2}{d\delta^2} \sigma[\delta = 0, \beta] = \frac{\beta}{4} = \mathcal{O}[\beta] \quad \forall \beta \quad (\text{D.57})$$

■

Lemmas D.1 and D.2 lead us to the following theorem:

Theorem D.1 *Differential learning via the original logistic sigmoidal form of the CFM objective function is unreasonably slow for $\beta \gg 1$.*

Proof : We fix the parameters $\alpha = 1$ and $\zeta = 0$ in (D.47) without loss of generality. It is straightforward to prove that the derivative decreases with increasing $|\delta|$ on the tails of the sigmoid (i.e., for $-\beta\delta + \zeta \gg 1$): inspection of $\frac{d}{d\delta} \sigma[\delta, \beta]$ in (D.51) reveals that it is positive-definite, and lemma D.2 proves that it is maximum at $\delta = 0$. The ratio of the derivative in the transition region (lemma D.2) to the derivative in the lower tail (lemma D.1) gives us a ratio of the learning rate for transition examples (i.e., ones that have

small discriminant differentials) to the learning rate for yet un-learned examples. We refer to this ratio of derivatives as the *learning rate ratio* $\phi(\beta)$, which is

$$\begin{aligned}\phi(\beta) &\triangleq \frac{\frac{d}{d\delta}\sigma[0, \beta]}{\frac{d}{d\delta}\sigma[-\delta, \beta]} \\ &= \frac{2 + \exp(\delta\beta) + \exp(-\delta\beta)}{4} \\ &= \mathcal{O}[\exp(|\delta|\beta)], \quad \beta \gg 1, \quad \delta < 0\end{aligned}\tag{D.58}$$

Thus, by definition D.4, lemmas D.1 and D.2, and (D.58), differential learning via the logistic sigmoidal form of CFM is unreasonably slow when $\beta \gg 1$. ■

Remark: Theorem D.1 means that it takes a classifier employing differential learning via a gradient-based search and the logistic sigmoidal form of CFM an unreasonably long time to learn some training examples. Un-learnable examples are ones that require a steep CFM sigmoid to be learned (see section 2.4 and section D.4); the first derivative of the logistic sigmoid for these un-learnable examples (which, by definition, have a negative discriminant differential δ) is so small that it would take an unreasonably large number of search iterations to modify the classifier's parameters enough for the example to be correctly classified (i.e., learned). The gray curve in figure D.5 shows the learning rate ratio $\phi(\beta)$ for values of β from 2 to 30. The curve assumes a nominal discriminant differential value of $\delta = -0.7$ in (D.58) for the un-learned examples. In practice, values of β that exceed 10 result in unreasonably slow learning, owing to the dominance of the transition example learning rate (note that $\phi(\beta) = 315$ for $\beta = 10.5$). Numerical "tricks" such as increasing the step size of the learning algorithm to increase the learning rate for yet un-learned training examples do not compensate for the dominance of transition examples. In practice, they lead to unstable oscillations in the search algorithm (in [55] it was found that β had to be less than about 10 to prevent unstable learning). The net result is that β must be kept small to 1) prevent unreasonably slow learning of yet-unlearned training examples, and 2) prevent oscillations in the learning algorithm. Small values of β prevent (2.96) from being satisfied for all points on $\mathcal{X}$, so some training examples are un-learnable and the resulting classifier does not achieve as low an error rate as it might. This combination of deficiencies led us to develop the synthetic form of CFM.

D.3.2 The Convergence Properties of Differential Learning via the Synthetic Form of CFM

Differential learning via the synthetic form of the CFM objective function remains reasonably fast and free of unstable oscillations, even when the transition region of the sigmoid is quite steep.

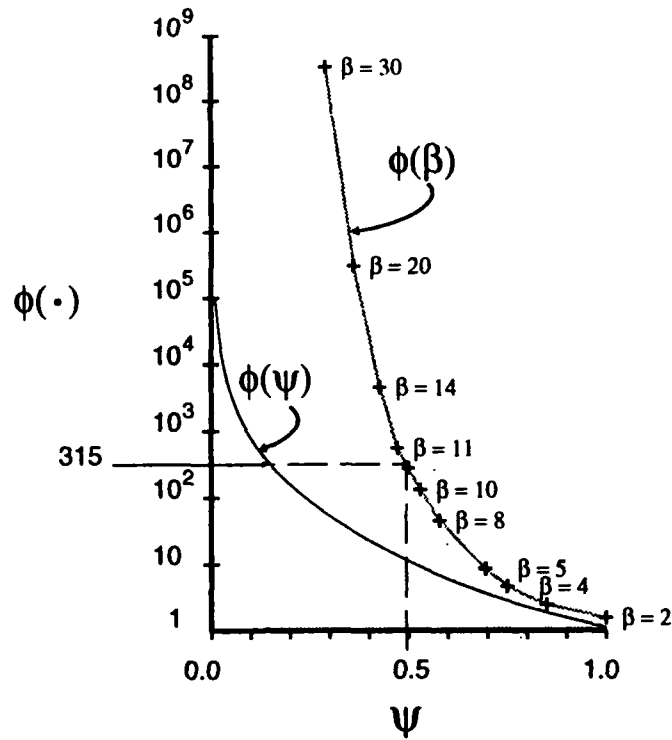


Figure D.5: The ratio $\phi(\cdot)$ of the differential learning rate for transition examples to that for un-learned examples with a nominal discriminant differential value of $\delta = -.7$. The ratio is plotted over the range of the confidence parameter that regulates the steepness of the CFM objective function's sigmoidal form. Ratios for the original logistic sigmoidal form ($\phi(\beta)$, gray) and the synthetic form ($\phi(\psi)$, black) are shown. Note that the β scale has been warped to match the ψ scale: values of β are shown along the gray curve. Light gray shading under the curves indicates the range of β and ψ values for which learning is reasonably fast and stable. The black curve shows that the synthetic form of CFM remains reasonably fast for un-learned examples as its transition region becomes steep ($\psi \rightarrow .15$); in contrast, the gray curve shows that the logistic sigmoidal form of CFM becomes unreasonably slow for un-learned examples while its transition region is still relatively shallow ($\beta = 10.5$). Figure D.8 (top) shows both forms of CFM for $\beta = 10.5$ (logistic sigmoidal) and the equivalent $\psi = .49$ (synthetic).

The rate of differential learning via the synthetic form of the CFM objective function generally decreases as $\mathcal{O}[\psi']$ ($\psi' \in (0,1)$) for un-learned training examples.

Proof : It can be shown that the first derivative of the synthetic CFM objective function described above is always greater in the lower radius and the transition region than it is in the lower leg — regardless of the value of $\psi' \in (0,1)$. Figure D.1 makes this plain, so we forego the rigorous proof. Given this relationship, the first derivative of the synthetic CFM objective function is bounded from below for all negative discriminant differentials (i.e., for all un-learned examples):

$$\frac{d}{d\delta} \sigma[\delta, \psi'] \geq a_n^* \quad \forall \delta < 0 \quad (\text{D.59})$$

By (D.18) – (D.29),

$$\begin{aligned}
 a_n^* &= \tan(\angle_3) \\
 &= \tan\left(\tan^{-1}\left(\frac{\mu_{yn}}{\mu_{xn} + 1}\right) - \sin^{-1}\left(\frac{r_n}{L_n}\right)\right)
 \end{aligned} \tag{D.60}$$

Since $\mu_{yn} = \psi^n R_n \sin(\angle_4)$ and $\mu_{xn} = -\psi^n R_n \cos(\angle_4)$,

$$\lim_{\psi^n \rightarrow 0^+} \tan^{-1}\left(\frac{\mu_{yn}}{\mu_{xn} + 1}\right) \cong \tan^{-1}(\psi^n R_n \sin(\angle_4)) \cong \psi^n R_n \sin(\angle_4) \tag{D.61}$$

Since $\lim_{\psi^n \rightarrow 0^+} L_n = 1$ (this is readily verifiable in figure D.1, so again we forego the rigorous proof),

$$\lim_{\psi^n \rightarrow 0^+} \sin^{-1}\left(\frac{r_n}{L_n}\right) \cong \sin^{-1}(\psi^n R_n) \cong \psi^n R_n \tag{D.62}$$

By (D.59) – (D.62),

$$\begin{aligned}
 \lim_{\psi^n \rightarrow 0^+} \frac{d}{d\delta} \sigma[\delta, \psi^n] &= \lim_{\psi^n \rightarrow 0^+} a_n^* \cong \tan(\psi^n (R_n \sin(\angle_4) - R_n)) \\
 &\cong \underbrace{\psi^n (R_n \sin(\angle_4) - R_n)}_{= k \text{ (a constant)}} \\
 &= \mathcal{O}[\psi^n] \quad \forall \delta < 0
 \end{aligned} \tag{D.63}$$

Thus, the first derivative of the synthetic CFM objective function is $\mathcal{O}[\psi^n]$ for all negative discriminant differentials when ψ^n is small. In fact, the relationship holds approximately for *all* values of ψ^n . Figure D.6 shows that the slope of the synthetic CFM objective function's lower leg is $\mathcal{O}[\psi^{1.15}]$ — or approximately linear with respect to ψ^n — for all $\psi^n \in (0, 1]$. Since the learning rate for un-learned examples is proportional to a_n^* , the theorem is proven. ■

Lemma D.4 *The rate of differential learning via the synthetic form of the CFM objective function generally increases as $\mathcal{O}[\psi^{1.1}]$ ($\psi^n \in (0, 1]$) for transition examples.*

Proof : By our specification of the synthetic CFM objective function in section D.1, it always attains its maximum derivative of a_n^* in the transition region (see figure D.1).

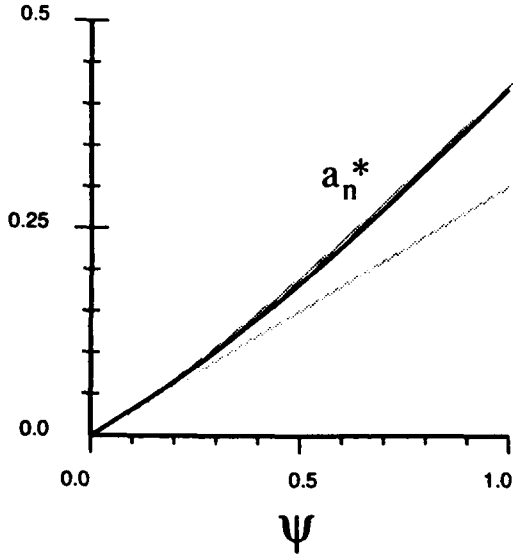


Figure D.6: The slope a_n^* (black) of the synthetic CFM objective function's lower leg, as a function of the confidence parameter ψ . Note that a_n^* is $\mathcal{O}[\psi]$ for $\psi < 0.15$ (see the proof of theorem D.3), and $\mathcal{O}[\psi^{1.15}]$ for $0.15 < \psi < 1$. The linear and $\mathcal{O}[\psi^{1.15}]$ asymptotes are shown in light gray.

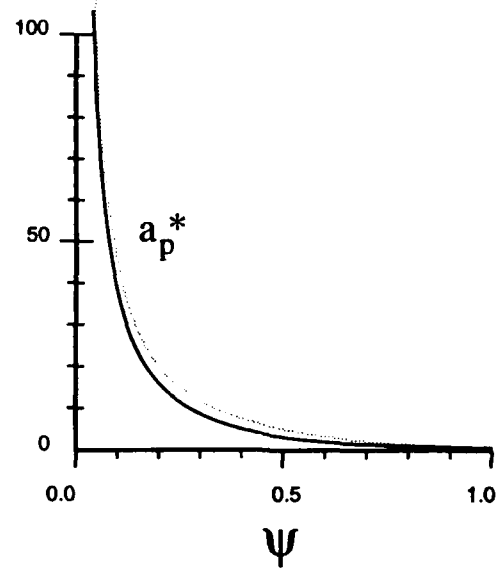


Figure D.7: The slope a_p^* (black) of the synthetic CFM objective function's transition region, as a function of the confidence parameter ψ . Note that a_p^* is $\mathcal{O}[\psi^{-1}]$ for all ψ (see the proof of lemma D.4), as indicated by the asymptote (which is a linear function of ψ^{-1}) shown in light gray.

$$\max_{\delta} \frac{d}{d\delta} \sigma[\delta, \psi] = a_p^* \quad \forall \psi \quad (\text{D.64})$$

By (D.15) – (D.29),

$$\begin{aligned} a_p^* &= \tan(\angle_p) \\ &= \tan\left(\tan^{-1}\left(\frac{\mu_{yp} - \mu_{ym}}{\mu_{xp} - \mu_{xm}}\right) - \sin^{-1}\left(\frac{r_p + r_n}{D_1}\right)\right) \end{aligned} \quad (\text{D.65})$$

Note that $\lim_{\psi \rightarrow 0^+} r_p = \lim_{\psi \rightarrow 0^+} r_n = 0$, and $\lim_{\psi \rightarrow 0^+} D_1 = 1$, so

$$\lim_{\psi \rightarrow 0^+} \sin^{-1}\left(\frac{r_p + r_n}{D_1}\right) = 0 \quad (\text{D.66})$$

Note also that $\lim_{\psi \rightarrow 0^+} \mu_{yp} - \mu_{ym} = 1$. Since $\mu_{yp} = \psi$ and $\mu_{xm} = -\psi \mathcal{R}_n \cos(\angle_4)$,

$$\begin{aligned}
\lim_{\psi' \rightarrow 0^+} \max_{\delta} \frac{d}{d\delta} \sigma[\delta, \psi'] &= \lim_{\psi' \rightarrow 0^+} a_p^* = \tan \left(\tan^{-1} \left(\frac{1}{\psi' (1 + \mathcal{R}_n \cos(\angle_4))} \right) \right) \\
&= \frac{1}{\psi' (1 + \mathcal{R}_n \cos(\angle_4))} \\
&= \mathcal{O}[\psi'^{-1}]
\end{aligned} \tag{D.67}$$

In fact, the relationship holds well for *all* values of $\psi' \in (0,1]$, as illustrated by figure D.7. Since the learning rate for transition examples is proportional to a_p^* , the theorem is proven. ■

Theorem D.2 *Differential learning via the synthetic form of the CFM objective function is reasonably fast.*

Proof : The ratio of the derivative in the transition region (lemma D.4) to the derivative in the lower tail (lemma D.3) gives us the learning rate ratio (i.e., the ratio of the learning rate for transition examples to the learning rate for yet un-learned examples):

$$\begin{aligned}
\phi(\psi') &\triangleq \frac{\max_{\delta} \frac{d}{d\delta} \sigma[\delta, \psi']}{\min_{\delta} \frac{d}{d\delta} \sigma[\delta, \psi']} \\
&= \frac{a_p^*}{a_n^*}
\end{aligned} \tag{D.68}$$

By (D.67) and (D.63)

$$\begin{aligned}
\lim_{\psi' \rightarrow 0^+} \phi(\psi') &= \left[\psi'^2 (1 + \mathcal{R}_n \cos(\angle_4)) (\mathcal{R}_n \sin(\angle_4) - \mathcal{R}_n) \right] \\
&= \mathcal{O}[\psi'^{-2}]
\end{aligned} \tag{D.69}$$

In fact, the learning rate ratio $\phi(\psi')$ remains $\mathcal{O}[\psi'^{-2}]$ for all $\psi' \in (0,1]$, so differential learning via synthetic CFM is reasonably fast. ■

Remark: From a practical viewpoint, differential learning via both forms of CFM becomes slow when the learning rate ratio exceeds about 315. Figure D.5 illustrates that for $\psi' > .15$ the synthetic CFM learning rate ratio is low enough to ensure reasonably fast learning. A comparison of the $\phi(\psi')$ curve with the $\phi(\beta)$ curve for the logistic sigmoidal form of CFM emphasizes that the first derivative of the steep synthetic CFM

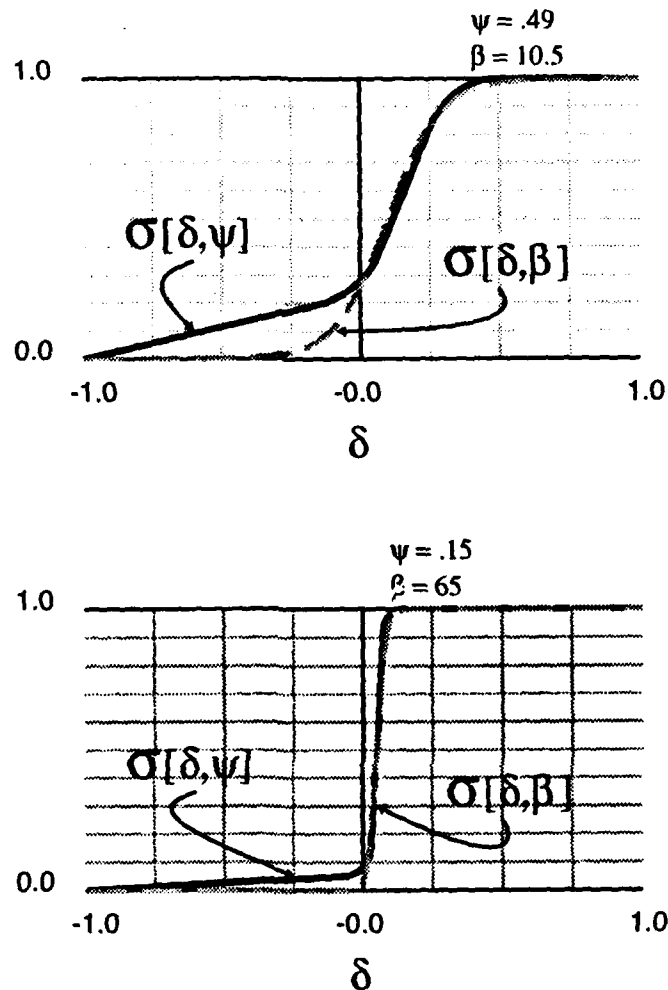


Figure D.8: Equivalent logistic (dashed gray) and synthetic (solid black) CFM functional forms. Note that the slope and shape of both functions is approximately the same in the transition region and upper leg. In the lower leg the synthetic function's slope is orders of magnitude larger than its logistic sigmoidal counterpart's. (top): The logistic sigmoidal form has a horizontal offset value of $\zeta = .1\beta$ (see (D.47)). Given the level of CFM steepness, the learning rate ratio is 11 for the synthetic function and 315 for its logistic sigmoidal counterpart; differential learning via the logistic sigmoidal form is unreasonably slow for un-learned examples. (bottom): The logistic sigmoidal form has a horizontal offset value of $\zeta = .05\beta$ (see (D.47)). Given this level of CFM steepness, the learning rate ratio is 315 for the synthetic function and 10^{19} for its logistic sigmoidal counterpart. Differential learning via the synthetic form of CFM remains reasonably fast and tenable for un-learned examples as long as its sigmoid is no steeper than this ($\psi \gtrsim .15$).

objective function is orders of magnitude larger than that of the comparable logistic sigmoidal form; as a result, the learning rate ratio for the synthetic form of CFM is orders of magnitude smaller than that for the comparable logistic sigmoidal form. Figure D.8 compares the logistic sigmoidal and synthetic forms of CFM for two cases. The top figure shows the two forms for the value of $\beta = 10.5$ at which differential learning

via the logistic sigmoidal form of CFM becomes slow. For this value of β , $\phi(\beta) \approx 315$; the comparable synthetic form of CFM has a confidence parameter of $\psi = .49$, for which $\phi(\psi) \approx 11$, so un-learned examples are learned approximately 30 times faster than they are using the logistic sigmoidal form of CFM. The bottom figure shows the two forms for the value of $\psi = .15$ at which differential learning via the synthetic form of CFM becomes slow. For this value of ψ , $\phi(\psi) \approx 315$; the logistic sigmoidal form of CFM has a confidence parameter of $\beta = 65$, for which $\phi(\beta) \approx 10^{19}$. The learning rates of the two functional forms differ by 17 orders of magnitude for un-learned examples, given this level of steepness in the sigmoidal function.

D.4 A Proof Relating to Synthetic CFM and Chapter 2

Recall from section 2.4 that for a given input $\mathbf{X}$ the classifier's discriminator generates C output activations $g_1(\mathbf{X}|\boldsymbol{\theta}), \dots, g_C(\mathbf{X}|\boldsymbol{\theta})$ and C corresponding discriminant differentials $\delta_1(\mathbf{X}|\boldsymbol{\theta}), \dots, \delta_C(\mathbf{X}|\boldsymbol{\theta})$. Since different training examples of $\mathbf{X}$ are assigned different empirical class labels according to the *a posteriori* probabilities $P_{W|\mathbf{X}}(\omega_1|\mathbf{X}), \dots, P_{W|\mathbf{X}}(\omega_C|\mathbf{X})$, the expected value of the CFM objective function for $\mathbf{X}$ is maximized when, by (2.94),

$$\begin{aligned} \vartheta(\mathbf{X})^* &= \sigma[\delta_+(\mathbf{X}|\boldsymbol{\theta}^*), \psi] - \sigma[0, \psi] \\ \neg\vartheta(\mathbf{X})^* &= \sigma[-\delta_+(\mathbf{X}|\boldsymbol{\theta}^*), \psi] - \sigma[0, \psi] \end{aligned}$$

$$\left(\underbrace{\left(\frac{\vartheta(\mathbf{X})^*}{-\neg\vartheta(\mathbf{X})^*} \geq \frac{1 - P_{W|\mathbf{X}}(\omega_+|\mathbf{X})}{P_{W|\mathbf{X}}(\omega_+|\mathbf{X})} \right)}_{\Delta\text{CFM}(\mathbf{X}|\boldsymbol{\theta}^*) > 0} \cup \underbrace{(\vartheta(\mathbf{X})^* = \neg\vartheta(\mathbf{X})^* = 0)}_{\Delta\text{CFM}(\mathbf{X}|\boldsymbol{\theta}^*) = 0} \right); \quad (\text{D.70})$$

$$\omega_{(1)} = \omega_+, \text{ s.t. } \delta_{(1)}(\mathbf{X}|\boldsymbol{\theta}) = \delta_+(\mathbf{X}|\boldsymbol{\theta}^*)$$

where ω_+ denotes the class with the largest *a posteriori* probability, and $\delta_+(\mathbf{X}|\boldsymbol{\theta}^*)$ is the corresponding discriminant differential. Recall that $\delta_{(1)}(\mathbf{X}|\boldsymbol{\theta})$ is the discriminant differential associated with the classifier's largest output; likewise, $\omega_{(1)}$ is the class associated with the classifier's largest output. CFM is maximized when the discriminator's largest output corresponds to the most likely class of $\mathbf{X}$: equivalently, CFM is maximized when the discriminant differential associated with the most likely class is positive (i.e., when $\delta_+(\mathbf{X}|\boldsymbol{\theta}^*) > 0$). Therefore, we simply wish to determine the upper bound on ψ , below which this condition is satisfied via (D.70). The following equations lead to such a bound on ψ . Although we would like to motivate them with a concise intuitive explanation, this proves rather difficult. We advise the reader to rely heavily on figure D.1 (taking δ in the figure to mean $\delta_+(\mathbf{X}|\boldsymbol{\theta}^*)$ in the present context); refer to (D.10) —

(D.46) when the figure fails to resolve the question.

If we assume that $\delta_*(X|\theta^*) = \mu_{\varphi} = \psi^i$ (i.e., it is the smallest value of $\delta_*(X|\theta)$ for which $\sigma[\delta_*(X|\theta), \psi^i]$ takes on its maximum value of unity — see figure D.1 and (D.26)), then $\sigma[\delta_*(X|\theta^*), \psi^i] = 1$ and $\frac{\vartheta(X)^*}{-\neg\vartheta(X)^*}$ in (D.70) simplifies to

$$\frac{\vartheta(X)^*}{-\neg\vartheta(X)^*} = \frac{1 - \sigma[0, \psi^i]}{\sigma[0, \psi^i] - \sigma[-\psi^i, \psi^i]} \quad (D.71)$$

Since $\sigma[0, \psi^i] < a_n^* + r_n = a_n^* + .7\psi^i$ and $\sigma[-\psi^i, \psi^i] \geq a_n^*(1 - \psi^i)$ ⁴

$$\frac{\vartheta(X)^*}{-\neg\vartheta(X)^*} \geq \frac{1 - a_n^* - .7\psi^i}{.7\psi^i + a_n^*\psi^i} = \frac{1 - (k + .7)\psi^i}{(k\psi^i + .7)\psi^i}, \quad (D.72)$$

where $k = \mathcal{R}_n \sin(\angle_4) - R_n \cong .3$. Therefore, $\frac{\vartheta(X)^*}{-\neg\vartheta(X)^*}$ is bounded from below:

$$\frac{\vartheta(X)^*}{-\neg\vartheta(X)^*} \geq \frac{1 - \psi^i}{.3\psi^i + .7\psi^i} = \frac{1 - \psi^i}{\psi^i(.3\psi^i + .7)} \quad (D.73)$$

This lower bound is tight for small ψ^i . It is loose for $\psi^i \rightarrow 1$, since $\lim_{\psi^i \rightarrow 1} \frac{\vartheta(X)^*}{-\neg\vartheta(X)^*} \cong 1$, whereas the lower bound yields

$$\lim_{\psi^i \rightarrow 1} \frac{1 - \psi^i}{.3\psi^i + .7\psi^i} = 0 \quad (D.74)$$

For smaller values of ψ^i the bound can be simplified to

$$\frac{\vartheta(X)^*}{-\neg\vartheta(X)^*} \geq \frac{1}{.7\psi^i} = 1.43\psi^{i-1} = \mathcal{O}[\psi^{i-1}] \quad (D.75)$$

so that (D.70) is satisfied if

$$\psi^{i-1} \geq \frac{1 - P_{W|X}(\omega_- | X)}{1.43 P_{W|X}(\omega_- | X)} \quad (D.76)$$

or

$$\psi^i \leq 1.43 \frac{P_{W|X}(\omega_- | X)}{1 - P_{W|X}(\omega_- | X)} \quad (D.77)$$

Thus, the requirement for learning the most probable class of X , stated in (2.94), is satisfied when (D.77) is satisfied.

⁴These bounds on the value of synthetic CFM are readily verified by a visual inspection of figure D.1.

D.5 Modifying Backpropagation for use with CFM

Any differentiable supervised classifier can use the CFM objective function to learn differentially. Since neural network classifiers employing the backpropagation learning procedure [119, 120] are a popular family of differentiable supervised classifiers, we show how to modify the backpropagation algorithm for use with CFM.

There are two fundamental differences between backpropagation with CFM and backpropagation with error measures such as MSE:

- For any given training example representing one of the C possible classes, CFM is a function of only two discriminator outputs; error measures are functions of all C discriminator outputs.
- CFM is *maximized*, whereas error measures are minimized.

Gradient computations — Recall from section 2.4, the CFM generated by a training sample S^n of n examples is given by

$$\text{CFM}(S^n | \theta) = \frac{1}{n} \sum_{j=1}^n (\sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] : \mathcal{W}^j = \omega_\tau), \quad (\text{D.78})$$

where $\mathbf{X}^j$ and $\mathcal{W}^j$ denote the j th of n training examples and its associated class label. The discriminant differential $\delta_\tau(\mathbf{X}^j | \theta)$ generated by the example $\mathbf{X}^j$ having the class label $\mathcal{W}^j = \omega_\tau$ ($\tau \in \{1, \dots, C\}$), is

$$\delta_\tau(\mathbf{X}^j | \theta) = \underbrace{g_\tau(\mathbf{X}^j | \theta)}_{y_\tau} - \underbrace{\max_{k \neq \tau} g_k(\mathbf{X}^j | \theta)}_{\bar{y}_\tau} \quad (\text{D.79})$$

Thus, the derivative of $\sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi]$ is non-zero with respect to only two outputs, y_τ and $\bar{y}_\tau$.⁵

$$\frac{\partial}{\partial y_i} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi] = \begin{cases} \frac{\partial}{\partial \delta_\tau} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi], & y_i = y_\tau \\ -\frac{\partial}{\partial \delta_\tau} \sigma[\delta_\tau(\mathbf{X}^j | \theta), \psi], & y_i = \bar{y}_\tau \\ 0, & \text{otherwise} \end{cases} \quad (\text{D.80})$$

Figure D.9 illustrates the significance of (D.80) for a hypothetical classifier that learns via backpropagation modified for use with CFM. The classifier has five discriminant functions, corresponding to the five classes that the feature vector can represent. The classifier's parameters are shown as black arrows pointing towards the discriminator's outputs, and the states of the classifier's nodes, given the example $\mathbf{X}^j$, are depicted

⁵The notation δ_τ is short-hand for $\delta_\tau(\mathbf{X}^j | \theta)$ throughout this section.

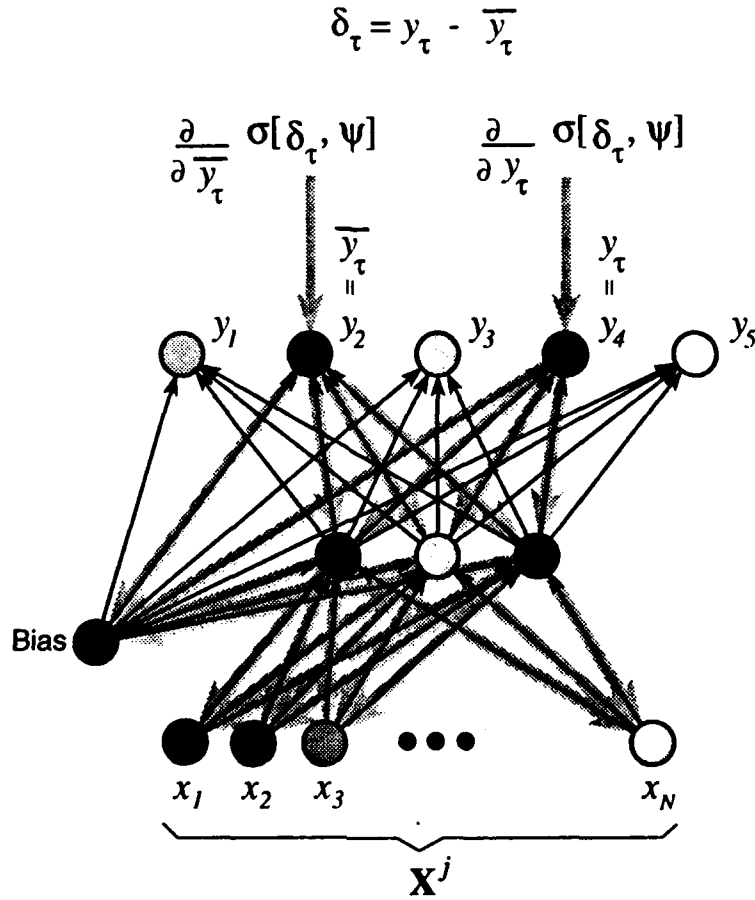


Figure D.9: A diagrammatic view of backpropagation with the CFM objective function. The classifier has $C = 5$ output nodes, which correspond to its five discriminant functions. The classifier's input is the N -dimensional feature vector, and the classifier has one hidden layer containing three nodes. The parameters (i.e., connections or weights) of the classifier are depicted by black arrows. The figure depicts the state of the classifier, given a particular training example $\mathbf{X}^j$ (darker nodes have larger values than lighter ones). The classifier's CFM $\sigma[\delta_\tau, \psi]$ is a function of the discriminant differential δ_τ , which is a function of only two outputs: the output $y_\tau = y_4$ corresponding to the input example's class label $\mathcal{W}^j$ (which is ω_4 in this case), and the largest *other* output $\bar{y}_\tau = y_2$. Thus, the derivative of $\sigma[\delta_\tau, \psi]$ is non-zero with respect to outputs y_τ and $\bar{y}_\tau$ only. The gray arrows pointing back through the classifier towards its input denote all the resulting non-zero derivatives of $\nabla_\theta (\sigma[\delta_\tau, \psi])$, the gradient of CFM with respect to the classifier's parameters, given the single training example/class label pair $(\mathbf{X}^j, \mathcal{W}^j)$.

in grayscale (darker nodes have larger values than lighter ones). Since $\mathbf{X}^j$ is an example of class ω_4 , $y_\tau = y_4$. Likewise, since y_2 is the most active of all other discriminator outputs, $\bar{y}_\tau = y_2$. The discriminant differential δ_τ is therefore $y_4 - y_2$. Note that $\bar{y}_\tau > y_\tau$, so δ_τ is negative and $\mathbf{X}^j$ is an un-learned example (definition D.1 — i.e., the classifier has not yet learned to classify $\mathbf{X}^j$ correctly).

We denote the gradient of $\sigma[\delta_\tau(\mathbf{X}^j|\theta), \psi]$ with respect to the classifier's parameters θ by

$\nabla_{\theta} (\sigma [\delta_{\tau}(\mathbf{X}^j | \theta), \psi^i])$. Since only the derivatives $\frac{\partial}{\partial v_2} \sigma [\delta_{\tau}(\mathbf{X}^j | \theta), \psi^i]$ and $\frac{\partial}{\partial v_4} \sigma [\delta_{\tau}(\mathbf{X}^j | \theta), \psi^i]$ are non-zero, only the parameters associated with the second and fourth discriminant functions affect the value of CFM for this example. Indeed, only those elements of $\nabla_{\theta} (\sigma [\delta_{\tau}(\mathbf{X}^j | \theta), \psi^i])$ corresponding to the parameters of the second and fourth discriminant functions are non-zero and need be computed. These gradient computations are depicted by the thick gray arrows of figure D.9, which point back towards the classifier's input. Note that once $\mathbf{X}^j$ becomes a learned example (definition D.2), $\delta_{\tau}(\mathbf{X}^j | \theta)$ exceeds μ_{τ} , and the derivative of CFM with respect to *all* outputs is zero. Mathematically,

$$\frac{\partial}{\partial y_i} \sigma [\delta_{\tau}(\mathbf{X}^j | \theta), \psi^i] = 0 \quad \forall i \quad \text{iff} \quad \delta_{\tau}(\mathbf{X}^j | \theta) > \mu_{\tau} \quad (\text{D.81})$$

When this is the case, no backpropagation computations have to be performed for $\mathbf{X}^j$. This characteristic of differential learning via synthetic CFM results in substantial computational savings as an increasingly large fraction of the training sample is learned (see section 7.5).

Steepest ascent search — Because CFM is maximized, we use a steepest *ascent* search for the optimal classifier parameters:

$$\theta[k+1] = \theta[k] + \left(\underbrace{\varepsilon \cdot \frac{\nabla_{\theta} (\text{CFM} (S^n | \theta[k]))}{\|\nabla_{\theta} (\text{CFM} (S^n | \theta[k]))\|}}_{\Delta \theta[k]} + \alpha \cdot \Delta \theta[k-1] \right), \quad (\text{D.82})$$

where k is an iteration index,

$$\nabla_{\theta} (\text{CFM} (S^n | \theta[k])) \triangleq \frac{1}{n} \sum_{j=1}^n \nabla_{\theta} (\sigma [\delta_{\tau}(\mathbf{X}^j | \theta[k]), \psi^i] : \mathcal{W}^i = \omega_{\tau}), \quad (\text{D.83})$$

$\|\nabla_{\theta} (\text{CFM} (S^n | \theta[k]))\|$ denotes the magnitude of the gradient $\nabla_{\theta} (\text{CFM} (S^n | \theta[k]))$, and $\alpha \cdot \Delta \theta[k-1]$ is the “momentum” term described in [119, eq. (9)]. Note that the steepest *ascent* algorithm of (D.82) differs from the conventional error measure-based steepest *descent* form of backpropagation in two ways

- The sign of the step-size parameter ε is positive for steepest ascent, but negative for steepest descent. Again, this is because CFM is *maximized*, whereas error measures are minimized.
- When $\alpha = 0$ in (D.82), the search step size $\Delta \theta[k]$ has a fixed magnitude of ε , since the equation employs a *normalized* gradient term. This feature is essential to stable convergence of the search because $\text{CFM} (S^n | \theta[k])$, the CFM generated by the training sample at iteration k , can have a large gradient *and* a large hessian on parameter space in the vicinity of its maximum. This occurs when

the CFM confidence parameter ψ is small (i.e., when the synthetic CFM sigmoid is steep). If a non-normalized gradient were used in (D.82), $\Delta\theta[k]$ could be very large, inducing a large search step precisely where the step should be small (i.e., where CFM ($S^* | \theta[k]$) has high curvature). See section 5.3.6, for a simple, hypothetical scenario in which CFM has a large gradient *and* a large hessian on parameter space in the vicinity of its maximum.

D.6 Source Code for the Synthetic CFM Objective Function

The following ANSI C source code implements the synthetic CFM function $\sigma[\delta, \psi]$ describe above, along with its first and second derivatives. The source code argument delta represents δ , and the argument conf represents ψ .

```

/*=====
NOTICE OF COPYRIGHT: Copyright 1992 by John Benjamin Hampshire II.
Individuals may compile, copy, distribute, and reuse this source code
with one restriction: this notice of copyright may NOT be removed.
The copyright holder disclaims any warranty of any kind,
expressed or implied, as to this code's fitness for any specific use.

Author: J. B. Hampshire II:
=====
Date: 3-14-92
=====
Purpose: Computes a synthetic asymmetric sigmoidal function cfm(delta, conf);
=====
-1 <= delta <= 1.
This synthetic function is used as the classification figure of merit (CFM)
in its "N-monotonic" form, described in J. B. Hampshire II's Ph.D. thesis
of 1993 and Hampshire & Waibel, IEEE Trans. Neural Networks, June, 1990. The
'discriminant differential' delta is the difference between the
classifier output representing the correct class and the largest other output.
In the case that the classifier is a single-output one (2-class case), it is
necessary to express delta as a function of the single classifier
output. This isn't hard, but it requires a little care...

First and second derivatives of cfm(delta, conf) are also computed.
The function has one "confidence" parameter (the variable "conf" in the
following code), in addition to its single argument.
The confidence parameter is on (0,1); low confidence corresponds to a steep
sigmoid (approaching a step function), whereas high confidence corresponds to
a nearly linear function of delta. Each call to cfm(), d_cfm(), and dd_cfm(),
checks to see if the confidence parameter has changed since the last call.
If it has, cfmSetup() is called, and cfm() and its derivatives are
synthesized for the new confidence. Following the re-synthesis,
cfm(delta, conf) or one of its derivatives is computed.
cfmSetup() is computationally expensive, but cfm(), d_cfm(), and dd_cfm()
are computationally cheaper than transcendental functions. Since the confidence
parameter is changed relatively infrequently, this synthetic function is
on average very cheap to evaluate. The advantages of the synthetic form
over closed-form functions described in the original CFM paper are described
in detail in the thesis.

Refs: JBH2 PhD thesis notes of 901114, 920314, and 920728.

Notes &
Latest

```

Revision: 3-14-92 by JBH2. Although cfm(delta, conf) is, strictly speaking, defined only on [-1,1] (corresponding to classifiers with outputs bounded on [0,1]), the code is written in such a way that it works (in practice) for any classifier with outputs on the real number line. Since the theoretical proofs pertaining to the optimality of CFM are restricted to a rather specifically bounded sigmoid, there are no explicit guarantees if you violate the corresponding constraints on the classifier outputs.

I've had no trouble with polynomial classifiers (as an example), but I can't be sure that there isn't some failure mode when delta is outside [-1,1].

7-28-92 by JBH2. Added second derivative function for use with modified

```

=====*/
#include <math.h>
#include <stdlib.h>
#include <stdio.h>

#define Pi          M_PI          /* 3.14158926... defined in /usr/include/math.h */
#define TWO_Pi      2.0 * Pi
#define HALF_Pi     M_PI_2       /* Pi/2... defined in /usr/include/math.h */
#define INFINITY    1.0e25

#define TRUE        1
#define FALSE       0

#define RN          0.7
#define A2          0.5
#define INV_A1      0.2
#define A0          -0.5
#define RP          (-1.0 * A0)

typedef struct _MyPoint {
    double x, y;
} MyPoint;

static double last_conf;
static double an, rn, xTnn, xTn, inv_ap, bp, xTp, yTp, rp;
static MyPoint Un, Up;

/*=====
 * void getCfmBreakpoints(): Returns the values of delta marking the lower and upper boundaries
 * of the synthetic CFM function's upper radius.
 *
 * Parameters:      l      lower bound of the synthetic CFM function's upper radius passed to
 *                      calling routine via
 *                      this pointer.
 *                      u      upper bound of the synthetic CFM function's upper radius passed to
 *                      calling routine via this pointer.
 *
 * Returns:         nothing
 *
 * Notes &
 * Latest
 * Revision: 1-20-93 by JBH2. Added. There's no point backprop'ing on deltas that exceed
 * the upper radius, since the synthetic function has zero slope beyond this
 * point. This function gets called every time the confidence parameter gets
 * changed, and it updates the bounds in the calling routine.
 *=====*/

void getCfmBreakpoints(l, u)
double *l, *u;
{
    *l = xTp;
    *u = Up.x;

```



```

    return;
}

/*=====
 *
 * void d_cfm():      Returns the synthetic CFM function's first derivative wrt delta, given
 *                   delta and conf.
 *
 * Parameters:      delta  the classifier's output differential.
 *                   conf   the CFM confidence parameter.
 *
 * Returns:         the synthetic CFM function's first derivative wrt delta.
 *
 * Notes &
 * Latest
 * Revision:        7-28-92 by JBH2.
 *=====*/

double d_cfm(delta, conf)
double delta, conf;
{
    double d_cfm, diff;
    void cfmSetup();

/* 1. don't allow confidence to go below .01 */
    if(conf < .01)
        conf = .01;

/* 2. if the present confidence isn't the same as the last one,
    set up the synthetic function anew.
*/
    if(conf != last_conf)
        cfmSetup(conf, &Un, &an, &rn, &xTnn, &xTn, &inv_ap, &bp, &xTp, &yTp, &Up, &rp);

/* 3. compute the value of the objective function */
    if(delta >= Up.x)
        d_cfm = 0.0;
    else if(delta > xTp) {
        diff = delta - Up.x;
        d_cfm = -diff / sqrt(rp*rp - diff*diff);
    }
    else if(delta >= xTn)
        d_cfm = 1.0/inv_ap;
    else if(delta > xTnn) {
        diff = delta - Un.x;
        d_cfm = diff / sqrt(rn*rn - diff*diff);
    }
    else
        d_cfm = an;

/* 4. stash the current confidence value, and return d_cfm */
    last_conf = conf;
    return(d_cfm);
}

/*=====
 *
 * void dd_cfm():    Returns the synthetic CFM function's second derivative wrt delta, given
 *                   delta and conf.
 *
 * Parameters:      delta  the classifier's output differential.
 *                   conf   the CFM confidence parameter.
 *=====*/

```

```

*
* Returns:          the synthetic CFM function's second derivative wrt delta.
*
* Notes &
* Latest
* Revision:        7-28-92 by JBH2.
*
*=====*/
double dd_cfm(delta, conf)
double delta, conf;
{
    double dd_cfm, diff, dtemp;
    void cfmSetup();

/* 1. don't allow confidence to go below .01 */
    if(conf < .01)
        conf = .01;

/* 2. if the present confidence isn't the same as the last one,
    set up the synthetic function anew.
*/
    if(conf != last_conf)
        cfmSetup(conf, &Un, &an, &rn, &xTnn, &xTn, &inv_ap, &bp, &xTp, &yTp, &Up, &rp);

/* 3. compute the value of the objective function */
    if(delta >= Up.x)
        dd_cfm = 0.0;
    else if(delta > xTp) {
        diff = delta - Up.x;
        dtemp = 1.0 / sqrt(rp*rp - diff*diff);
        dd_cfm = (-diff * diff * dtemp * dtemp - 1.0) * dtemp;
    }
    else if(delta >= xTn)
        dd_cfm = 0.0;
    else if(delta > xTnn) {
        diff = delta - Un.x;
        dtemp = 1.0 / sqrt(rn*rn - diff*diff);
        dd_cfm = (diff * diff * dtemp * dtemp + 1.0) * dtemp;
    }
    else
        dd_cfm = 0.0;

/* 4. stash the current confidence value, and return dd_cfm */
    last_conf = conf;
    return(dd_cfm);
}

/*=====
*
* void cfm():        Returns the synthetic CFM function's value, given delta and conf.
*
* Parameters:       delta  the classifier's output differential.
*                   conf   the CFM confidence parameter.
*
* Returns:          the synthetic CFM function's value.
*
* Notes &
* Latest
* Revision:        3-14-92 by JBH2.
*
*=====*/
double cfm(delta, conf)

```

```

double delta, conf;
{
double cfm, diff;
void cfmSetup();

/* 1. don't allow confidence to go below .01 */
if(conf < .01)
    conf = .01;

/* 2. if the present confidence isn't the same as the last one,
    set up the synthetic function anew.
*/
if(conf != last_conf)
    cfmSetup(conf, &Un, &an, &rn, &xTnn, &xTn, &inv_ap, &bp, &xTp, &yTp, &Up, &rp);

/* 3. compute the value of the objective function */
if(delta >= Up.x)
    cfm = 1.0;
else if(delta > xTp) {
    diff = delta - Up.x;
    cfm = Up.y + sqrt(rp*rp - diff*diff);
}
else if(delta >= xTn)
    cfm = delta/inv_ap + bp;
else if(delta > xTnn) {
    diff = delta - Un.x;
    cfm = Un.y - sqrt(rn*rn - diff*diff);
}
else
    cfm = an * delta + an;

/* 4. stash the current confidence value, and return cfm */
last_conf = conf;
return(cfm);
}

/*=====
*
* void cfmSetup():      Recomputes all the necessary metrics for the synthetic CFM function,
*                      given the new confidence parameter conf.
*
* Parameters:          conf    the new CFM confidence parameter.
*                      Un      the centroid of the function's lower radius.
*                      an      the slope of the function's lower leg.
*                      rn      the function's lower radius.
*                      xTnn    the value of delta at which the function's lower leg and lower
*                      radius are tangent.
*                      xTn     the value of delta at which the function's lower radius and
*                      (middle) transition leg are tangent.
*                      inv_ap  the inverse of the transition leg's slope.
*                      bp      the value of delta at which the transition leg intercepts the
*                      horizontal line      CFM = 0.
*                      xTp     the value of delta at which the function's (middle) transition
*                      leg and upper radius are tangent.
*                      yTp     the value of CFM at delta = xTp.
*                      Up      the centroid of the function's upper radius.
*                      rp      the function's upper radius.
*
* Returns:              nothing
*
* Notes &
* Latest
*/

```

```

* Revision:          3-14-92 by JBH2.
*
*-----*/
void cfmSetup(conf, Un, an, rn, xTnn, xTn, inv_ap, bp, xTp, yTp, Up, rp)
double conf;
double *an, *rn, *xTnn, *xTn, *inv_ap, *bp, *xTp, *yTp, *rp;
MyPoint *Un, *Up;
{
    static double          fconf, angle, RR, angle_1, angle_2, angle_3, angle_p;
    static double          zeta_n, l, D, a1, x0, y0, x1, y1, arg;
    static MyPoint         Tnn, Tn, U0;
    static int             virgin=TRUE;

/* phase 1 and 2 are computations are all constants, so do them only once. */
    if(virgin) {
/*****
* PHASE 1 *
*****/

        angle_2 = atan(A2);
        if(INV_A1 == 0.0)
            angle_1 = HALF_Pi;
        else {
            a1 = 1.0 / INV_A1;
            angle_1 = atan(a1);
        }

/* Notes of 920314, (1) */
        angle_3 = HALF_Pi - (angle_1 - angle_2) / 2.0;

/* Notes of 920314, (2) */
        l = RN / tan(angle_3);

/* Notes of 920314, (3) */
        if(INV_A1 == 0.0) {
            x0 = 0.0;
            y0 = A2;
        }
        else {
            x0 = A2 / (a1 - A2);
            y0 = a1 * x0;
        }

/* Notes of 920314, (6) */
        arg = HALF_Pi - angle_1;
        x1 = x0 + l * sin(arg);
        y1 = y0 + l * cos(arg);

/* Notes of 920314, (7) */
        U0.x = x1 - RN * cos(arg);
        U0.y = y1 + RN * sin(arg);
/*****
* PHASE 2 *
*****/

/* Notes of 920314, (8) */
        RR = sqrt(U0.x*U0.x + U0.y*U0.y);

/* Notes of 920314, (8a) */
        angle = atan(fabs(U0.y/U0.x));
        virgin = FALSE;
    }
/*****

```

```

* PHASE 3 *
...../

fconf = conf;

/* Notes of 920314, (11) */

*rn = RN * fconf;
zeta_n = RR * fconf;

Un->x = -zeta_n * cos(angle);
Un->y = zeta_n * sin(angle);

/* Notes of 920314, (12) */

*rp = RP * fconf;

Up->x = -*rp / A0;
Up->y = 1 - *rp;

/*****
* PHASE 4 *
...../

/* Notes of 920314, (13) */

l = sqrt((Un->x + 1.0) * (Un->x + 1.0) + Un->y * Un->y);
D = sqrt(l*l - *rn * *rn);

angle_3 = atan(Un->y/(Un->x + 1.0)) - asin(*rn / l);

/* Notes of 920314, (14) */

*an = tan(angle_3);

/* Notes of 920314, (15) */

*xTnn = D * cos(angle_3) - 1.0;
Tnn.y = D * sin(angle_3);

/*****
* PHASE 5 *
...../

/* Notes of 920314, preceeding (16) */

D = sqrt((Up->x - Un->x)*(Up->x - Un->x) + (Up->y - Un->y)*(Up->y - Un->y));
l = sqrt(D*D - (*rp + *rn)*(*rp + *rn));

angle_3 = acos(l/D);
angle_2 = atan((Up->y - Un->y)/(Up->x - Un->x));

/* Notes of 920314, (16) */

angle_p = angle_3 + angle_2;

if(angle_p == HALF_Pi)
    *inv_ap = 0.0;
else
    *inv_ap = 1.0 / tan(angle_p);

/*****
* PHASE 6 *
...../

/* Notes of 920314, (18) */

angle_1 = angle_p + HALF_Pi;

*xTp = Up->x + *rp * cos(angle_1);
*yTp = Up->y + *rp * sin(angle_1);

/* Notes of 920314, (19) */

*xTn = Un->x - *rn * cos(angle_1);
Tn.y = Un->y - *rn * sin(angle_1);

/* Notes of 920314, (19) */

if(angle_p == HALF_Pi)

```

```
    *bp = -INFINITY;
else
    *bp = *yTp - *xTp / *inv_ap;
return;
}
```

Appendix E

Differential Learning via CFM Viewed as a Generalization of Learning via Rosenblatt's Perceptron Criterion Function

In this appendix we explore the similarities between differential learning via the CFM objective function and learning via Rosenblatt's perceptron criterion function [116]. We prove that differential learning and perceptron learning are quite similar for the 2-class pattern recognition task in which the classifier has one linear discriminant function. We begin the proof with a differential learning formulation of the task; we then alter the form of the CFM objective function and complete the proof.

Since the pattern recognition task is a 2-class task, we need a discriminator with only one discriminant function $g_1(\mathbf{X}|\theta)$: if $g_1(\mathbf{X}|\theta)$ is positive, the classifier labels $\mathbf{X}$ as an example of class ω_1 ; if $g_1(\mathbf{X}|\theta)$ is negative, the classifier labels $\mathbf{X}$ as an example of class ω_2 . However, we assume a classifier of the form described in section 2.2.1, which obliges us to create a second *phantom* discriminant function $g_2(\mathbf{X}|\theta)$. This phantom discriminant function is related to $g_1(\mathbf{X}|\theta)$ by

$$g_2(\mathbf{X}|\theta) = -g_1(\mathbf{X}|\theta) \quad (\text{E.1})$$

so that the resulting classifier's operation is described by (2.6) and (2.7). The two discriminant functions are constrained to be linear functions of $\mathbf{X}$. If we adopt the convention of [29, ch. 5] and define the *augmented* feature vector $\mathbf{X}'$ as an $(N + 1)$ -element vector formed by preceding the original N -element vector with a single element of constant unit value

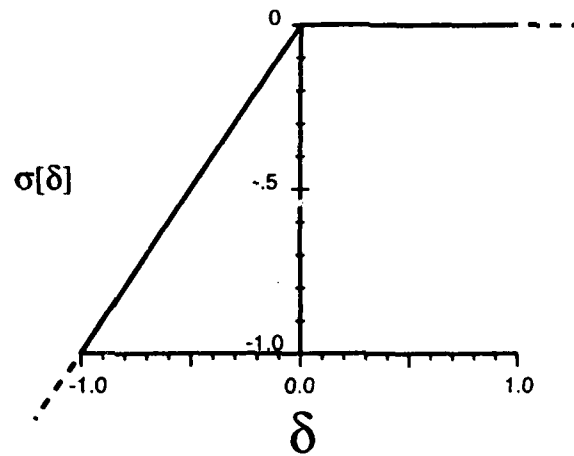


Figure E.1: A classifier comprising a single linear discriminant function is equivalent to Rosenblatt's perceptron when generated with this modified form of the CFM objective function.

$$\mathbf{X}' = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_N \end{bmatrix} \quad (\text{E.2})$$

and we give the parameter vector θ ($N + 1$) elements, then the two linear discriminant functions are described by

$$\begin{aligned} g_1(\mathbf{X}|\theta) &= \mathbf{X}'^T \theta \\ g_2(\mathbf{X}|\theta) &= -\mathbf{X}'^T \theta \end{aligned} \quad (\text{E.3})$$

(the notation $\mathbf{Z}^T$ denotes the transpose of the vector $\mathbf{Z}$). The discriminant differentials associated with $g_1(\mathbf{X}|\theta)$ and $g_2(\mathbf{X}|\theta)$ are therefore

$$\begin{aligned} \delta_1(\mathbf{X}|\theta) &= g_1(\mathbf{X}|\theta) - g_2(\mathbf{X}|\theta) = 2\mathbf{X}'^T \theta \\ \delta_2(\mathbf{X}|\theta) &= g_2(\mathbf{X}|\theta) - g_1(\mathbf{X}|\theta) = -2\mathbf{X}'^T \theta \end{aligned} \quad (\text{E.4})$$

Recall from section 2.2.4 that the argument of the CFM objective function associated with a training example is δ_τ when the class label of the training example is ω_τ . Let us change the functional form of

the CFM objective function from the sigmoidal function $\sigma[\delta, \psi]$ described in sections 2.2.4 and 2.4 and appendix D to the piece-wise linear function

$$\sigma[\delta] = \begin{cases} 0, & \delta \geq 0 \\ \delta, & \delta < 0 \end{cases} \quad (\text{E.5})$$

illustrated in figure E.1. Under these circumstances, the average CFM for the training sample S^n of size n is, by (2.81) and (2.82),

$$\begin{aligned} \text{CFM}(S^n | \theta) &= \frac{1}{n} \sum_{j=1}^n \sigma[\delta_\tau(\mathbf{X}^j | \theta)] ; \quad \delta_\tau(\mathbf{X}^j | \theta) = \begin{cases} \delta_1(\mathbf{X}^j | \theta), & \mathcal{W}^j = \omega_1 \\ \delta_2(\mathbf{X}^j | \theta), & \mathcal{W}^j = \omega_2 \end{cases} \\ &= \frac{2}{n} \sum_{j=1}^n \delta'_\tau(\mathbf{X}^j) ; \quad \delta'_\tau(\mathbf{X}^j) = \begin{cases} \mathbf{X}^{jT} \theta, & \mathcal{W}^j = \omega_1 \text{ \& } \mathbf{X}^{jT} \theta < 0 \\ -\mathbf{X}^{jT} \theta, & \mathcal{W}^j = \omega_2 \text{ \& } \mathbf{X}^{jT} \theta > 0 \\ 0, & \text{otherwise} \end{cases} \quad (\text{E.6}) \end{aligned}$$

Maximizing $\text{CFM}(S^n | \theta)$ over θ is equivalent to minimizing $-\text{CFM}(S^n | \theta)$ over θ

$$\therefore \max_{\theta} \text{CFM}(S^n | \theta) \equiv$$

$$\min_{\theta} \underbrace{\frac{2}{n} \sum_{j=1}^n \delta''_\tau(\mathbf{X}^j) ; \quad \delta''_\tau(\mathbf{X}^j) = \begin{cases} -\mathbf{X}^{jT} \theta, & \mathcal{W}^j = \omega_1 \text{ \& } \mathbf{X}^{jT} \theta < 0 \\ \mathbf{X}^{jT} \theta, & \mathcal{W}^j = \omega_2 \text{ \& } \mathbf{X}^{jT} \theta > 0 \\ 0, & \text{otherwise} \end{cases}}_{\text{Rosenblatt's Perceptron Criterion Function}} \quad (\text{E.7})$$

Equation (E.7) is — but for a constant — identical to Rosenblatt's perceptron criterion function (cf. (12) of [29, ch. 5]). Thus, differential learning via CFM can be viewed as a generalization of the perceptron approach to discriminative learning. The generalization is four-part:

1. Learning is extended from the 2-class pattern recognition task to the general $C \geq 2$ -class task.
2. The functional form of the classifier's discriminant functions need not be linear. The removal of this restriction allows differential learning to be applied to any differentiable supervised classifier (see table 2.1 for some examples).

3. The functional form of the CFM objective function, described in sections 2.2.4 and 2.4 and appendix D, guarantees that the classifier will be asymptotically efficient (the focus of the entire text). By the proof of section 2.4, the functional form of Rosenblatt's perceptron criterion function (represented as a CFM objective function in figure E.1) lacks the sigmoidal shape necessary for engendering minimum-error discrimination. When the number of classes is greater than two and/or the class-conditional densities of $\mathbf{X}$ overlap (i.e., the concepts to be learned are stochastic), differential learning via the perceptron criterion function of figure E.1 is provably *not* asymptotically efficient; differential learning via CFM is.
4. If the class-conditional densities of $\mathbf{X}$ ($C = 2$) are linearly separable, the linear discriminant generated with the perceptron criterion function is guaranteed to separate the two classes of $\mathbf{X}$. The differential learning guarantee associated with the ($C \geq 2$)-class $\mathbf{X}$ having potentially overlapping class-conditional densities is analogous, albeit considerably stronger: the differentially-generated classifier is guaranteed to yield the lowest error rate possible,¹ given an asymptotically large training sample.

¹The lower bound on the classifier's error rate is determined by how well its discriminator can approximate the Bayesian discriminant function. Thus, when we state, "the lowest error rate possible," we mean the lowest possible given our particular choice of discriminator, *not* the lowest possible given *any* choice of discriminator.

Appendix F

Proper Parametric Models of the Homoscedastic Gaussian Feature Vector¹

In this appendix we replicate two proofs: the normal-based linear discriminant analysis paradigm is the *fully-parametric* proper model for the feature vector $\mathbf{X}$ with homoscedastic Gaussian class-conditional pdfs; the logistic regression paradigm (a.k.a. logistic discriminant analysis) is the *partially-parametric* proper model for the feature vector $\mathbf{X}$ with equal class prior probabilities and homoscedastic Gaussian class-conditional pdfs. The first proof can be found in any introductory textbook on probability and statistics, since the fully-parametric model learns by computing the maximum-likelihood estimates for the means and covariance matrices of the feature vector's class-conditional pdfs. The second proof has been worked by Akaike and White [2, 140, 142, 141] and by Hjort.²

The proofs require that the class-conditional covariance matrices are all of the form $\sigma^2 \cdot \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix. Under a simple linear transformation, a feature vector with homoscedastic Gaussian class-conditional pdfs that are *not* of this form will be transformed to one with class-conditional pdfs that *are* of this form. We assume such a transformation has been performed without loss of generality. Given this assumption, the homoscedastic restriction on the class-conditional pdfs of $\mathbf{X}$ ensures that class boundaries on $\mathbf{X}$ are piece-wise linear because all the class-conditional covariance matrices have orthonormal eigenvectors and eigenvalues that all equal σ^2 .

¹ A homoscedastic feature vector's class-conditional probability density functions (pdfs) all have the same covariance matrix.

² Hjort's proof is contained in [65], which we have not obtained; the reference is mentioned in [68, pg. 169].

F.1 The Fully-Parametric Proper Model

Consider the N -dimensional feature vector $\mathbf{X}$ with Gaussian class-conditional pdfs:

$$\rho_{\mathbf{X}|\mathcal{W}}(\mathbf{X}|\omega_i, \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{N}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (\mathbf{X} - \mu_i)^T \Sigma_i^{-1} (\mathbf{X} - \mu_i) \right] \quad (\text{F.1})$$

As described above, we assume that $\mathbf{X}$ is homoscedastic, and — without further loss of generality — that it has undergone a linear transformation such that all its class-conditional covariance matrices are given by $\sigma^2 \mathbf{I}$. Under these conditions, (F.1) reduces to

$$\rho_{\mathbf{X}|\mathcal{W}}(\mathbf{X}|\omega_i, \mu_i, \sigma^2) = \frac{1}{(2\pi)^{\frac{N}{2}} \sigma^N} \exp \left[-\frac{1}{2\sigma^2} (\mathbf{X} - \mu_i)^T (\mathbf{X} - \mu_i) \right] \quad (\text{F.2})$$

By Bayes' rule, the *a posteriori* class probabilities of $\mathbf{X}$ are given by

$$P_{\mathcal{W}|\mathbf{X}}(\omega_i|\mathbf{X}, \mu_i, \sigma^2) = \frac{\rho_{\mathbf{X}|\mathcal{W}}(\mathbf{X}|\omega_i, \mu_i, \sigma^2) \cdot P_{\mathcal{W}}(\omega_i)}{\sum_{k=1}^C \rho_{\mathbf{X}|\mathcal{W}}(\mathbf{X}|\omega_k, \mu_k, \sigma^2) \cdot P_{\mathcal{W}}(\omega_k)} \quad (\text{F.3})$$

$$\propto \rho_{\mathbf{X}|\mathcal{W}}(\mathbf{X}|\omega_i, \mu_i, \sigma^2) \cdot P_{\mathcal{W}}(\omega_i) \quad (\text{F.4})$$

In the form of (2.27), let

$$\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) = \begin{cases} 1, & \mathcal{W}^j = \omega_i \\ 0, & \text{otherwise} \end{cases} \quad (\text{F.5})$$

If we view the right-hand side of (F.4) as the basis for the likelihood equation, given the training sample of n independently-drawn training example/class label pairs $\mathcal{S}^n = \{(\mathbf{X}^1, \mathcal{W}^1), \dots, (\mathbf{X}^n, \mathcal{W}^n)\}$, the log-likelihood equation is

$$\begin{aligned} L(\tilde{\mu}_1, \dots, \tilde{\mu}_C, \tilde{\sigma}^2) &= \\ &= \ln \left(\prod_{j=1}^n \prod_{i=1}^C \left(\rho_{\mathbf{X}|\mathcal{W}}(\mathbf{X}^j|\omega_i, \tilde{\mu}_i, \tilde{\sigma}^2) \cdot P_{\mathcal{W}}(\omega_i) \right)^{\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle)} \right) \\ &= \sum_{j=1}^n \sum_{i=1}^C \left[\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) \cdot \left(\ln(P_{\mathcal{W}}(\omega_i)) + \ln(\rho_{\mathbf{X}|\mathcal{W}}(\mathbf{X}^j|\omega_i, \tilde{\mu}_i, \tilde{\sigma}^2)) \right) \right] \\ &= \sum_{j=1}^n \sum_{i=1}^C \left[\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) \cdot \left(\ln(P_{\mathcal{W}}(\omega_i)) - \ln((2\pi)^{\frac{N}{2}}) - N \cdot \ln(\sqrt{\tilde{\sigma}^2}) \right. \right. \\ &\quad \left. \left. - \frac{1}{2\tilde{\sigma}^2} \cdot (\mathbf{X}^j - \tilde{\mu}_i)^T (\mathbf{X}^j - \tilde{\mu}_i) \right) \right] \quad (\text{F.6}) \end{aligned}$$

where $\mathbf{X}^j$ denotes the j th example of $\mathbf{X}$ in the training sample, $\mathcal{W}^j$ denotes the class label of $\mathbf{X}^j$, $\tilde{\mu}_i$ is the maximum-likelihood estimate of the class-conditional mean μ_i , and $\tilde{\sigma}^2$ is the maximum-likelihood estimate of the variance parameter σ^2 .

Setting the gradient of (F.6) with respect to $\{\tilde{\mu}_1, \dots, \tilde{\mu}_c, \tilde{\sigma}^2\}$ equal to zero and solving the resulting normal equations gives us the maximum-likelihood parameter estimates

$$\begin{aligned}\tilde{\mu}_i &= \frac{1}{n_i} \sum_{j=1}^n \tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) \cdot \mathbf{X}^j \\ \tilde{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^c \sum_{j=1}^n \tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) \cdot (\mathbf{X}^j - \tilde{\mu}_i)^T (\mathbf{X}^j - \tilde{\mu}_i),\end{aligned}\tag{F.7}$$

where n_i denotes the number of examples in S^n with the class label ω_i . These estimates are used in (F.4), along with sample-based estimates of the class prior probabilities $\{P_{\mathcal{W}}(\omega_1), \dots, P_{\mathcal{W}}(\omega_c)\}$, to produce the fully-parametric proper model of $\mathbf{X}$. The process is commonly called normal-based linear discriminant analysis (e.g., [91]).

F.2 The Partially-Parametric Proper Model

The *a posteriori* class probabilities in (F.4) can be expressed as follows:

$$\begin{aligned}P_{\mathcal{W}|\mathbf{X}}(\omega_i | \mathbf{X}, \mu_i, \sigma^2) \\ = \left[1 + \sum_{\substack{k=1 \\ k \neq i}}^c \frac{P_{\mathcal{W}}(\omega_k)}{P_{\mathcal{W}}(\omega_i)} \exp \left[-\frac{1}{\sigma^2} (\mu_i - \mu_k)^T \mathbf{X} + \frac{1}{2\sigma^2} (\mu_i^T \mu_i - \mu_k^T \mu_k) \right] \right]^{-1}\end{aligned}\tag{F.8}$$

F.2.1 $C = 2$: Logistic Regression

When $C = 2$, $\mathbf{X}$ represents one of two classes. If both classes have prior probabilities of $\frac{1}{2}$, the *a posteriori* class probabilities of $\mathbf{X}$ are

$$\begin{aligned}P_{\mathcal{W}|\mathbf{X}}(\omega_1 | \mathbf{X}, \mu_1, \sigma^2) \\ = \left[1 + \exp \left[\underbrace{\frac{1}{\sigma^2} (\mu_2 - \mu_1)^T \mathbf{X}}_{\alpha^T} + \underbrace{\frac{1}{2\sigma^2} (\mu_1^T \mu_1 - \mu_2^T \mu_2)}_{\beta} \right] \right]^{-1}\end{aligned}\tag{F.9}$$

and

$$\begin{aligned}
 P_{1|X}(\omega_2 | X, \mu_2, \sigma^2) &= 1 - P_{1|X}(\omega_1 | X, \mu_1, \sigma^2) \\
 &= \left[1 + \exp \left[- \underbrace{\frac{1}{\sigma^2} (\mu_2 - \mu_1)^T X}_{\alpha^T} - \underbrace{\frac{1}{2\sigma^2} (\mu_1^T \mu_1 - \mu_2^T \mu_2)}_{\beta} \right] \right]^{-1}
 \end{aligned} \tag{F.10}$$

Note that the N -dimensional vector α and the scalar β can be viewed as the ultimate parameters of this partially-parametric model, so we can make the following equations:

$$\begin{aligned}
 P_{W|X}(\omega_1 | X, \mu_1, \sigma^2) &\equiv P_{W|X}(\omega_1 | X, \alpha, \beta) = [1 + \exp[\alpha^T X + \beta]]^{-1} \\
 P_{W|X}(\omega_2 | X, \mu_2, \sigma^2) &\equiv P_{W|X}(\omega_2 | X, \alpha, \beta) = [1 + \exp[-\alpha^T X - \beta]]^{-1}
 \end{aligned} \tag{F.11}$$

When the method of maximum-likelihood is used to estimate the parameters α and β , it is modified to maximize a product of independent *a posteriori* class probabilities rather than a product of independent class-conditional pdf terms. Maximizing the logarithm of this product is equivalent — a procedure called maximizing the logit of risk [83, pp. 80-82]. The model of (F.11) is, by definition 3.13, a partially-parametric proper model of X . Assuming n independently drawn training example/class label pairs $S^n = \{ \langle X^1, W^1 \rangle, \dots, \langle X^n, W^n \rangle \}$, the logit risk function is (e.g., [68, pg. 9])

$$\begin{aligned}
 L(\tilde{\alpha}, \tilde{\beta}) &= \ln \left(\prod_{j=1}^n \left(P_{1|X}(\omega_1 | X^j, \tilde{\alpha}, \tilde{\beta}) \right)^{\tau_1(\langle X^j, W^j \rangle)} \cdot \left(P_{W|X}(\omega_2 | X^j, \tilde{\alpha}, \tilde{\beta}) \right)^{\tau_2(\langle X^j, W^j \rangle)} \right) \\
 &= \sum_{j=1}^n \left[\tau_1(\langle X^j, W^j \rangle) \cdot \ln \left(P_{W|X}(\omega_1 | X^j, \tilde{\alpha}, \tilde{\beta}) \right) \right. \\
 &\quad \left. + \tau_2(\langle X^j, W^j \rangle) \cdot \ln \left(P_{W|X}(\omega_2 | X^j, \tilde{\alpha}, \tilde{\beta}) \right) \right],
 \end{aligned} \tag{F.12}$$

where $\tau_i(\langle X^j, W^j \rangle)$ is given in (F.6), and $\tilde{\alpha}$ and $\tilde{\beta}$ denote *estimates* of α and β . Equation (F.12) can be differentiated with respect to its $N+1$ parameters in order to generate normal equations. The resulting equations are non-linear with respect to the parameters, so they must be solved iteratively. We omit the normal equations because they are not essential to our argument; see [68] as an example of such details.

If we view the estimated *a posteriori* class probabilities in (F.12) as discriminant functions, and we use the notation $\theta = \{ \tilde{\alpha}, \tilde{\beta} \}$, then

$$g_i(\mathbf{X}|\boldsymbol{\theta}) = P_{W|\mathbf{X}}(\omega_i|\mathbf{X}^j, \tilde{\alpha}, \tilde{\beta}) \quad (\text{F.13})$$

and (F.12) can be re-stated as

$$L(\tilde{\alpha}, \tilde{\beta}) = \sum_{j=1}^n [\tau_1(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) \cdot \ln(g_1(\mathbf{X}|\boldsymbol{\theta})) + \tau_2(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) \cdot \ln(g_2(\mathbf{X}|\boldsymbol{\theta}))] \quad (\text{F.14})$$

The reader should recognize that this form of $L(\tilde{\alpha}, \tilde{\beta})$ is (but for a constant factor, cf. section 2.3.2) the Kullback-Leibler information distance [82, 81] of the training sample, given the discriminant functions $g_1(\mathbf{X}|\boldsymbol{\theta})$ and $g_2(\mathbf{X}|\boldsymbol{\theta})$. Thus, the maximum-likelihood parameters of the logistic regression model are obtained by minimizing the Kullback-Leibler information distance between the training sample and the discriminator $\mathcal{G}(\mathbf{X}|\boldsymbol{\theta}) = \{g_1(\mathbf{X}|\boldsymbol{\theta}), g_2(\mathbf{X}|\boldsymbol{\theta})\}$, where $g_i(\mathbf{X}|\boldsymbol{\theta})$ is given by (F.13). By the proof of section 2.3.2, this learning strategy leads to the following parameterization for large training sample sizes:

$$\begin{aligned} \tilde{\alpha} &= \alpha = \frac{1}{\sigma^2} (\mu_2 - \mu_1) \\ \lim_{n \rightarrow \infty} \tilde{\beta} &= \beta = \frac{1}{2\sigma^2} (\mu_1^\top \mu_1 - \mu_2^\top \mu_2) \end{aligned} \quad (\text{F.15})$$

F.2.2 $C > 2$: Logistic Discriminant Analysis

If the class prior probabilities are all equal, then (F.8) simplifies to

$$\begin{aligned} P_{W|\mathbf{X}}(\omega_i|\mathbf{X}, \mu_i, \sigma^2) \\ = \left[1 + \sum_{\substack{k=1 \\ k \neq i}}^C \exp \left[-\frac{1}{\sigma^2} (\mu_i - \mu_k)^\top \mathbf{X} + \frac{1}{2\sigma^2} (\mu_i^\top \mu_i - \mu_k^\top \mu_k) \right] \right]^{-1} \end{aligned} \quad (\text{F.16})$$

When $C > 2$, (F.12) assumes the more general form

$$\begin{aligned} L(\tilde{\alpha}_1, \dots, \tilde{\alpha}_C, \tilde{\beta}_1, \dots, \tilde{\beta}_C) &= \\ \ln \left(\prod_{j=1}^n \prod_{i=1}^C (P_{W|\mathbf{X}}(\omega_i|\mathbf{X}^j, \tilde{\alpha}_i, \tilde{\beta}_i))^{\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle)} \right) \\ &= \sum_{j=1}^n \sum_{i=1}^C [\tau_i(\langle \mathbf{X}^j, \mathcal{W}^j \rangle) \cdot \ln(P_{W|\mathbf{X}}(\omega_i|\mathbf{X}^j, \tilde{\alpha}_i, \tilde{\beta}_i))] \end{aligned} \quad (\text{F.17})$$

A single logistic discriminant function of the form

$$g_i(\mathbf{X}|\theta) = \left[1 + \exp \left[\tilde{\alpha}_i^T \mathbf{X} + \tilde{\beta}_i \right] \right]^{-1} \quad (\text{F.18})$$

is used to model each of the C *a posteriori* probabilities of $\mathbf{X}$. In order for $g_i(\mathbf{X}|\theta)$ to be a reasonably good approximation of the *a posteriori* probabilities given in (F.11), each class conditional mean μ_i must have only one neighboring mean (which we will denote by $\mu_{i'}$) closer than about 3σ . Under this condition, each class-conditional pdf of the feature vector has only one close neighbor. That is, each class is confusable with only one other,³ and the i th *a posteriori* probability of (F.11) is reasonably well approximated by the logistic function

$$P_{Y|\mathbf{X}}(\omega_i | \mathbf{X}, \mu_i, \sigma^2) \approx \left[1 + \exp \left[\underbrace{\frac{1}{\sigma^2} (\mu_{i'} - \mu_i)^T \mathbf{X}}_{\alpha_i^T} + \underbrace{\frac{1}{2\sigma^2} (\mu_i^T \mu_i - \mu_{i'}^T \mu_{i'})}_{\beta_i} \right] \right]^{-1} \quad (\text{F.19})$$

By the same arguments as those of the preceding section, $\mathcal{G}(\mathbf{X}|\theta) = \{g_1(\mathbf{X}|\theta), \dots, g_C(\mathbf{X}|\theta)\}$ will be a reasonable approximation to the proper parametric model of $\mathbf{X}$. If the model learns by minimizing its Kullback-Leibler information distance with the training sample, the resulting parameters $\{\tilde{\alpha}_1, \dots, \tilde{\alpha}_C\}$ and $\{\tilde{\beta}_1, \dots, \tilde{\beta}_C\}$ will be maximum-likelihood estimates of their true values:

$$\begin{aligned} \tilde{\alpha}_i &= \alpha_i = \frac{1}{\sigma^2} (\mu_{i'} - \mu_i) \\ \lim_{n \rightarrow \infty} \tilde{\beta}_i &= \beta_i = \frac{1}{2\sigma^2} (\mu_i^T \mu_i - \mu_{i'}^T \mu_{i'}) \end{aligned} \quad (\text{F.20})$$

F.3 The Asymptotic Relative Efficiency of Logistic Discriminant Analysis Versus Normal-Based Linear Discriminant Analysis

Efron studies the asymptotic relative efficiency (ARE) (see section 3.6.1) of the fully- and partially-parametric proper models for the homoscedastic Gaussian feature vector in [30]. We remind the reader that normal-based linear discriminant analysis is the fully-parametric model and logistic discriminant analysis is the partially-parametric model. Efron's definition of ARE is based on the ratio of the fully-parametric model's error rate to that of the partially-parametric model. Our definition of ARE (definition 3.18) is based on the ratio of

³We subjectively characterize two class-conditional pdfs as "close" neighbors if their means are separated by less than three standard deviations. If we were to set a more rigorous standard for closeness — say, five standard deviations — the approximation of (F.19) would be so much the better.

one classifier's MSDE to another classifier's MSDE: we generally assume the two classifiers differ only in terms of the learning strategy they employ, although a simple notational generalization of definition 3.18 allows for a comparison of classifiers with different hypothesis classes. Our definition therefore has a similar philosophical motivation to Efron's, but it focuses on a comparison of MSDE (*squared discriminant bias plus discriminant variance*), whereas Efron's definition focuses only on a comparison of discriminant bias.

These differences notwithstanding, Efron's work proves for the $C = 2$ -class case, (fully-parametric) normal-based linear discriminant analysis is more efficient than (partially parametric) logistic discriminant analysis. A similar analysis of the $C > 2$ -class case can be found in [19]. The reason, stated in intuitive terms, is that the fully-parametric paradigm is a more constrained model of the data, despite its having more parameters. The class-conditional means are explicitly modeled in the fully-parametric paradigm, whereas only the difference between these means is modeled in the partially-parametric paradigm. The higher degree of specificity in the fully-parametric proper model makes it more efficient by Efron's definition; again, by our definition of discriminant efficiency, Efron's work proves that the fully-parametric model exhibits lower discriminant bias than its partially-parametric counterpart: no statement is made concerning discriminant variance.

The greater specificity of the fully-parametric model is an advantage when it is indeed proper (i.e., when the assumptions regarding the probabilistic nature of $\mathbf{X}$ are valid), but it is a disadvantage when the model is improper. Specifically, the fully-parametric model described in this appendix is proper only for homoscedastic Gaussian-distributed feature vectors, whereas the partially-parametric model described herein is proper for a much broader family of homoscedastic *exponentially*-distributed feature vectors (e.g., [83]). Thus, if the feature vector is exponentially-distributed rather than Gaussian-distributed, the partially-parametric model will be proper (and efficient), whereas the fully-parametric model will be neither proper nor efficient. This phenomenon leads us full-circle to the arguments of chapter 3: if we assume that the feature vector is *arbitrarily-distributed*, then neither the fully- nor the partially-parametric model is proper, so the most efficient classifier possible, given the improper logistic linear hypothesis class implied by both parametric models described herein, will be generated by the differential learning strategy.

F.4 The Proper Parametric Model Constraints are Severe

Given the preceding insights, the choice of learning strategy hinges on whether the model of the data is proper. In order for the (fully-parametric) normal-based linear discriminant analysis paradigm to be a proper parametric model of the $C \geq 2$ -class Gaussian feature vector $\mathbf{X}$, the class-conditional pdfs of $\mathbf{X}$ must be homoscedastic.

In order for the (partially-parametric) logistic discriminant analysis paradigm to be a proper parametric model of the 2-class exponentially-distributed feature vector $\mathbf{X}$, the two class-conditional pdfs of $\mathbf{X}$ must be homoscedastic and the class prior probabilities must be equal. In order for the logistic discriminant analysis

paradigm to be a proper parametric model of the $C > 2$ -class exponentially-distributed feature vector $\mathbf{X}$, all the class-conditional pdfs of $\mathbf{X}$ must be homoscedastic; furthermore, no class-conditional pdf can be a close neighbor in $\mathcal{X}$ of more than one other pdf, and the class prior probabilities must be equal.

These constraints on the form of $\mathbf{X}$ are strong, and in reality they rarely hold. When they do hold, it is usually in the context of a deterministic feature vector that is corrupted by independent additive Gaussian noise with these nice properties. The resulting "random" feature vector can be modeled quite well by either of the proper models described herein. Classical hypothesis testing procedures (e.g., see [140]) can be employed to verify whether or not the models are indeed proper. Unless the proper hypothesis is confirmed, both of the parametric models described herein will, by the proofs of chapter 3, be both improper models and inefficient classifiers of $\mathbf{X}$.

Appendix G

Error Rate Computations for the Classifiers of Chapter 4

Recall from definition 3.1 (page 55) that the *true* error rate $P_e(\mathcal{G}|\theta)$ for the classifier of x is given by

$$P_e(\mathcal{G}|\theta) \triangleq E_x[P_e(\mathcal{G}(x|\theta))] = \int_{\mathcal{X}} P_e(\mathcal{G}(x|\theta)) \rho_x(x) dx, \quad (\text{G.1})$$

where

$$\begin{aligned} P_e(\mathcal{G}(X|\theta)) &\triangleq 1 - P_{W|x}(\mathcal{D}(x|\theta) | x) \\ &= 1 - P_{W|x}(\Gamma(\mathcal{G}(x|\theta)) | x), \end{aligned} \quad (\text{G.2})$$

and $\mathcal{D}(x|\theta) = \Gamma(\mathcal{G}(x|\theta))$ denotes the class label that the classifier assigns to its input x . The error rates that we quote in chapter 4 — for both the proper and improper parametric models — are computed according to (G.1) because we play the role of an oracle and we know the probabilistic nature of the feature x . The following two sections outline the procedures we use to do the computations.

G.1 Error Rate Computations for the Proper Parametric Model

The fully-parametric and partially-parametric models of section 4.2 form one and only one class boundary on the domain of the homoscedastic Gaussian feature x . The boundary for the fully-parametric model is, by (4.8),

$$B_{1,2}^{\text{Fully-Parametric}} = \frac{\tilde{\mu}_1 + \tilde{\mu}_2}{2}, \quad (\text{G.3})$$

where $\widetilde{\mu}_1$ and $\widetilde{\mu}_1$ are two of the three model parameters. The boundary for the partially-parametric model is, by (4.12),

$$B_{1,2}^{\text{Partially-Parametric}} = \frac{-\theta_{1,0}}{\theta_{1,1}}, \quad (\text{G.4})$$

where $\theta_{1,0}$ and $\theta_{1,1}$ are the two model parameters.

The error rate of both models can be expressed in terms of the class boundary $B_{1,2}$. All examples of class ω_1 having values of x greater than the boundary are misclassified, as are all examples of class ω_2 having values of x less than the boundary: mathematically,

$$\begin{aligned} P_e(\mathcal{G}|\theta) &= P_W(\omega_1) \cdot \int_{B_{1,2}}^{\infty} \underbrace{\left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x - \mu_1)^2\right] \right]}_{\rho_{x|W}(x|\omega_1)} dx \\ &\quad + P_W(\omega_2) \cdot \int_{-\infty}^{B_{1,2}} \underbrace{\left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x - \mu_2)^2\right] \right]}_{\rho_{x|W}(x|\omega_2)} dx, \end{aligned} \quad (\text{G.5})$$

The integrals in (G.5) are easily computed via the Chebyshev approximation to the error function (erf) [106, sec. 6.2].

G.2 Error Rate Computations for the Improper Parametric Model

The error rates of the polynomial classifiers in section 4.3 are evaluated in a crude but computationally simplistic fashion. Since the high-complexity classifier can, in principle, form many class boundaries on the domain of x , we compute the integral of (G.1) numerically, using a successive approximation technique. This saves the trouble of computing the class boundaries — essentially a polynomial root-finding task — and then evaluating the integral as in (G.5). Given the *specific* probabilistic nature of x in section 4.3, the classifier's error rate is expressed as

$$\begin{aligned} P_e(\mathcal{G}|\theta) &= \int_{-5.8}^{-3.8} \neg\tau_1(\mathcal{D}(x|\theta)) \underbrace{\rho_{x|W}(x|\omega_1) P_W(\omega_1)}_{0.05} dx \\ &\quad + \int_{-4}^4 \neg\tau_2(\mathcal{D}(x|\theta)) \underbrace{\rho_{x|W}(x|\omega_2) P_W(\omega_2)}_{0.1} dx \end{aligned}$$

$$+ \int_{3.8}^{5.8} \neg \tau_3 (\mathcal{D}(x|\theta)) \underbrace{\rho_{x|W}(x|\omega_3) P_W(\omega_3)}_{0.05} dx, \quad (G.6)$$

where

$$\neg \tau_i (\mathcal{D}(x|\theta)) = \begin{cases} 0, & \mathcal{D}(x|\theta) = \omega_i \\ 1, & \text{otherwise} \end{cases} \quad (G.7)$$

Again, $\mathcal{D}(x|\theta)$ denotes the class label that the classifier assigns to its input x . We use the following numerical quadrature approximation for each of the three integrals in (G.6):

$$\begin{aligned} \int_a^b \neg \tau_i (\mathcal{D}(x|\theta)) \rho_{x|W}(x|\omega_i) P_W(\omega_i) dx &\cong \\ \frac{b-a}{M} \sum_{j=0}^{M-1} \neg \tau_i \left(\mathcal{D} \left(a + \left(j + \frac{1}{2} \right) \cdot \frac{b-a}{M} \middle| \theta \right) \right) &\rho_{x|W} \left(a + \left(j + \frac{1}{2} \right) \cdot \frac{b-a}{M} \middle| \omega_i \right) P_W(\omega_i) \end{aligned} \quad (G.8)$$

We begin with $M = 30$ intervals, and double M until subsequent approximations differ by less than 10^{-4} . This numerical integration technique is equivalent to the trapezoid rule (e.g., [20, sec. 4.3]) as M grows large. It is admittedly crude, but it is trivial to implement and provides sufficient precision for our purposes.

Appendix H

Asymptotic Parameterizations for the Probabilistically-Generated Improper Parametric Models of Chapter 4

We generate improper parametric models in section 4.3 by minimizing the mean-squared error (MSE) between the discriminator output vector $\mathbf{Y}$ and a corresponding target vector denoting the class of the training example (see section 2.3); the minimization is done for all examples in the training sample, and generally takes the form of an iterative search procedure. We employ backpropagation, a well-known probabilistic learning paradigm; its iterative search procedure is gradient descent, and the gradient of the classifier's MSE with respect to the parameter vector θ is computed by the chain-rule [119, 120].¹

We denote the training sample of size n by S^n , and we denote a particular unique value (or *pattern*) of x by x_p . If there are P unique patterns in S^n , and for each of these patterns there are $n_{p,i}$ examples belonging to class ω_i , the sample MSE of the classifier $\mathcal{G}(x|\theta) = \{g_1(x|\theta), \dots, g_C(x|\theta)\}$ with parameterization θ is given by

$$\text{MSE}(S^n|\theta) = \sum_{i=1}^C \sum_{p=1}^P \frac{n_p}{n} \left[(g_i(x_p|\theta) - 1)^2 \cdot \frac{n_{p,i}}{n_p} + (g_i(x_p|\theta))^2 \cdot \frac{n_p - n_{p,i}}{n_p} \right]; \quad (\text{H.1})$$
$$\sum_{p=1}^P n_{p,i} = n$$

Recall that C denotes the total number of classes that x can represent. In the case of the random variable x in section 4.3, $C = 3$. It is straightforward to prove that the classifier's MSE can be expressed by the following expectation as the training sample size grows asymptotically large (see section 2.3.2):

¹Backpropagation generally employs MSE, although other objective functions can be used. We employ only the MSE objective function for probabilistic learning. The CE objective function, for example, cannot be used because the polynomial classifier's outputs are unbounded; this violates the conditions necessary for using CE (see section 2.3.2).

$$\lim_{n \rightarrow \infty} \text{MSE}(\mathcal{S}^n | \theta) = E_x [\text{MSE}(x | \theta)] = \sum_{i=1}^c \int_{-\infty}^{\infty} \left[(g_i(x | \theta) - 1)^2 \cdot P_{W|x}(\omega_i | x) + (g_i(x | \theta))^2 \cdot P_{W|x}(\neg \omega_i | x) \right] \rho_x(x) dx \quad (\text{H.2})$$

where the notation $E_x[\cdot]$ denotes the expectation over the domain of x , and

$$P_{W|x}(\neg \omega_i | x) \triangleq 1 - P_{W|x}(\omega_i | x) \quad (\text{H.3})$$

The parameterization θ^* that minimizes the classifier's MSE can be found by substituting the expressions for the classifier's discriminant functions into (H.2), deriving the expression for the gradient $\nabla_{\theta} (E_x [\text{MSE}(x | \mathcal{G}(x | \theta^*))])$, setting this gradient equal to the zero vector, and solving the resulting normal equations for θ^* . Barnard and Casasent use this technique for deriving the minimum-MSE parameterization of a linear classifier, given a 2-class Gaussian feature [6]. We derive distribution-independent expressions for the asymptotic minimum-MSE parameterization of the i th discriminant function $g_i(x | \theta)$ in (H.2); expressions are given for constant, linear, and quadratic discriminant functions. Distribution-independent expressions for the minimum-MSE parameterizations of higher-order polynomial discriminant functions become cumbersome, so we derive the minimum-MSE parameterization of the high-complexity classifier (i.e., the MSE-generated "10-10-10" model) in distribution-dependent form. We use the probabilistic nature of the feature x , described by (4.28) – (4.29).

The polynomial discriminant functions of the improper parametric model are described by (4.32). Since no polynomial discriminant function in (4.32) shares parameters with another, we can minimize the MSE for each discriminant function independent of the other discriminant functions. The operative equation therefore becomes

$$\lim_{n \rightarrow \infty} \text{MSE}(\mathcal{S}^n | g_i(x | \theta)) = E_x [\text{MSE}(x | g_i(x | \theta))] = \int_{-\infty}^{\infty} \left[(g_i(x | \theta) - 1)^2 \cdot P_{W|x}(\omega_i | x) + (g_i(x | \theta))^2 \cdot P_{W|x}(\neg \omega_i | x) \right] \rho_x(x) dx \quad (\text{H.4})$$

If the i th discriminant function has K_i parameters, we derive that many normal equations of the form

$$\frac{d}{d\theta_{i,k}} E_x [\text{MSE}(x | g_i(x | \theta^*))] = \frac{d}{d\theta_{i,k}} \int_{-\infty}^{\infty} \left[(g_i(x | \theta^*) - 1)^2 \cdot P_{W|x}(\omega_i | x) + (g_i(x | \theta^*))^2 \cdot P_{W|x}(\neg \omega_i | x) \right] \rho_x(x) dx$$

$$\begin{aligned}
&= \int_{-\infty}^{\infty} \left[2 (g_i(x|\theta^*) - 1) \cdot \frac{d}{d\theta_{i,k}} g_i(x|\theta^*) \cdot P_{W|x}(\omega_i|x) \right. \\
&\quad \left. + 2 g_i(x|\theta^*) \cdot \frac{d}{d\theta_{i,k}} g_i(x|\theta^*) \cdot P_{W|x}(\neg\omega_i|x) \right] \rho_x(x) dx \\
&= 0
\end{aligned} \tag{H.5}$$

(where $\theta_{i,k}^*$ denotes the k th element of the parameter vector θ^* that minimizes the MSE of the i th discriminant function) in order to solve for the minimum-MSE parameterization θ^* .

H.1 Distribution-Independent Expressions for the Parameterization of Low-Order Polynomial Discriminant Functions

When the discriminant function in (H.4) is a constant, i.e.,

$$g_i(x|\theta) = \theta_{i,0}, \tag{H.6}$$

we denote the value of the parameter that minimizes $E_x [\text{MSE}(x|g_i(x|\theta))]$ in (H.4) by $\theta_{i,0}^*$. Substituting (H.6) into (H.5), we obtain the single normal equation:

$$\begin{aligned}
&\frac{d}{d\theta_{i,0}} E_x [\text{MSE}(x|g_i(x|\theta^*))] = \\
&\quad \frac{d}{d\theta_{i,0}} \int_{-\infty}^{\infty} \left[(\theta_{i,0}^* - 1)^2 \cdot P_{W|x}(\omega_i|x) + (\theta_{i,0}^*)^2 \cdot P_{W|x}(\neg\omega_i|x) \right] \rho_x(x) dx \\
&= \int_{-\infty}^{\infty} \left[2 (\theta_{i,0}^* - 1) \cdot P_{W|x}(\omega_i|x) + 2 \theta_{i,0}^* \cdot P_{W|x}(\neg\omega_i|x) \right] \rho_x(x) dx \\
&= 2 (\theta_{i,0}^* - P_W(\omega_i)) \\
&= 0
\end{aligned} \tag{H.7}$$

Thus

$$\theta_{i,0}^* = P_W(\omega_i) \tag{H.8}$$

When the discriminant function in (H.4) is a linear function of x , i.e.,

$$g_i(x|\theta) = \theta_{i,1} x + \theta_{i,0}, \tag{H.9}$$

the two normal equations are given by

$$\begin{aligned}
\frac{d}{d\theta_{i,k}} E_x [\text{MSE}(x|g_i(x|\theta^*))] &= \\
\frac{d}{d\theta_{i,0}} \int_{-\infty}^{\infty} \left[(\theta_{i,1}^* x + \theta_{i,0}^* - 1)^2 \cdot P_{W|x}(\omega_i|x) + (\theta_{i,1}^* x + \theta_{i,0}^*)^2 \cdot P_{W|x}(-\omega_i|x) \right] \rho_x(x) dx \\
&= 0,
\end{aligned} \tag{H.10}$$

($k = 0, 1$). Expanding and solving them in the manner leading to (H.7) yields the following MSE-minimizing parameters:²

$$\begin{aligned}
\theta_{i,1}^* &= [m_1(x, \omega_i) \cdot P_W(-\omega_i) - m_1(x, -\omega_i) \cdot P_W(\omega_i)] / \zeta_{i,1} \\
\theta_{i,0}^* &= [m_2(x) \cdot P_W(\omega_i) - m_1(x, \omega_i) \cdot m_1(x)] / \zeta_{i,1}
\end{aligned} \tag{H.11}$$

where

$$\begin{aligned}
\zeta_{i,1} &= m_2(x) - (m_1(x))^2 \\
m_j(x, \omega_i) &\triangleq E_x[(x)^j | \omega_i] \cdot P_W(\omega_i) \\
m_j(x, -\omega_i) &\triangleq \sum_{\substack{k=1 \\ k \neq i}}^c m_j(x, \omega_k) \\
m_j(x) &\triangleq E_x[(x)^j] = \sum_{k=1}^c E_x[(x)^j | \omega_k] \cdot P_W(\omega_k)
\end{aligned} \tag{H.12}$$

When the discriminant function in (H.4) is a quadratic function of x , i.e.,

$$g_i(x|\theta) = \theta_{i,2}(x)^2 + \theta_{i,1}x + \theta_{i,0}, \tag{H.13}$$

the three normal equations are given by

$$\begin{aligned}
\frac{d}{d\theta_{i,k}} E_x [\text{MSE}(x|g_i(x|\theta^*))] &= \\
\frac{d}{d\theta_{i,k}} \int_{-\infty}^{\infty} \left[(\theta_{i,2}(x)^2 + \theta_{i,1}x + \theta_{i,0} - 1)^2 \cdot P_{W|x}(\omega_i|x) \right. \\
&\quad \left. + (\theta_{i,2}(x)^2 + \theta_{i,1}x + \theta_{i,0})^2 \cdot P_{W|x}(-\omega_i|x) \right] \rho_x(x) dx \\
&= 0,
\end{aligned} \tag{H.14}$$

²Equation (H.5) represents a distribution-independent multi-class generalization of a result for the 2-class Gaussian feature in [6].

($k = 0, 1, 2$). Expanding and solving them in the manner leading to (H.7) yields the following MSE-minimizing parameters:

$$\begin{aligned}
 \theta_{i,2}^* &= \left[P_{1V}(\omega_i) [(m_2(x))^2 - m_1(x) \cdot m_3(x)] \right. \\
 &\quad + m_2(x, \omega_i) \cdot [(m_1(x))^2 - m_2(x)] \\
 &\quad \left. + m_1(x, \omega_i) \cdot [m_3(x) - m_1(x) \cdot m_2(x)] \right] / \zeta_{i,2} \\
 \theta_{i,1}^* &= \left[P_{1V}(\omega_i) [m_1(x) \cdot m_4(x) - m_2(x) \cdot m_3(x)] \right. \\
 &\quad + m_2(x, \omega_i) [m_3(x) - m_1(x) \cdot m_2(x)] \\
 &\quad \left. + m_1(x, \omega_i) [(m_2(x))^2 - m_4(x)] \right] / \zeta_{i,2} \quad (H.15) \\
 \theta_{i,0}^* &= \left[P_{1V}(\omega_i) [(m_3(x))^2 - m_2(x) \cdot m_4(x)] \right. \\
 &\quad + m_2(x, \omega_i) [(m_2(x))^2 - m_1(x) \cdot m_3(x)] \\
 &\quad \left. + m_1(x, \omega_i) [m_1(x) \cdot m_4(x) - m_2(x) \cdot m_3(x)] \right] / \zeta_{i,2}
 \end{aligned}$$

where

$$\zeta_{i,2} = (m_2(x))^3 + (m_3(x))^2 + m_4(x) \cdot [(m_1(x))^2 - m_2(x)] - 2 m_1(x) \cdot m_2(x) \cdot m_3(x) \quad (H.16)$$

The j th moment of a uniformly-distributed random variable with lower and upper bounds of l and u is

$$m_j(x) = \frac{1}{u-l} \frac{1}{j+1} [(u)^{j+1} - (l)^{j+1}] \quad (H.17)$$

Using (H.17) and (4.28) – (4.29), Equations (H.8) – (H.16) can be evaluated. The resulting values for the minimum- and low-complexity polynomial classifiers of the homoscedastic uniformly-distributed random variable in section 4.28 are given in the top and middle entries of table 4.1 (page 104).

H.2 Distribution-Dependent Expressions for the Parameterization of Polynomial Discriminant Functions

The preceding distribution-independent, minimum-MSE parameter expressions are cumbersome. Fortunately, the piece-wise constant nature of the class-conditional pdfs and *a posteriori* probabilities of x (see figure 4.9) allow a straightforward expansion of (H.5), by which compact expressions for the minimum-MSE polynomial discriminant function parameters can be obtained. Equation (H.5) can be re-stated as follows:

$$\begin{aligned}
& \frac{d}{d\theta_{i,k}} E_x [\text{MSE}(x|g_i(x|\theta^*))] = \\
& = 2 \int_{-\infty}^{\infty} (g_i(x|\theta^*) - 1) \cdot \frac{d}{d\theta_{i,k}} g_i(x|\theta^*) \cdot P_W(\omega_i) \cdot \rho_{x|W}(x|\omega_i) dx \\
& \quad + 2 \sum_{\substack{j=1 \\ j \neq i}}^c \int_{-\infty}^{\infty} g_i(x|\theta^*) \cdot \frac{d}{d\theta_{i,k}} g_i(x|\theta^*) \cdot P_W(\omega_j) \cdot \rho_{x|W}(x|\omega_j) dx \\
& = 0
\end{aligned} \tag{H.18}$$

Using (4.28) and (4.29), we can express the k th normal equation for the three polynomial discriminant functions thus (dropping the factor of 2):

$$\begin{aligned}
& \frac{d}{d\theta_{1,k}} E_x [\text{MSE}(x|g_1(x|\theta^*))] = \\
& \quad .05 \int_{-5.8}^{-3.8} (g_1(x|\theta^*) - 1) \cdot \frac{d}{d\theta_{1,k}} g_1(x|\theta^*) dx \\
& \quad + .1 \int_{-4}^4 g_1(x|\theta^*) \cdot \frac{d}{d\theta_{1,k}} g_1(x|\theta^*) dx \\
& \quad + .05 \int_{3.8}^{5.8} g_1(x|\theta^*) \cdot \frac{d}{d\theta_{1,k}} g_1(x|\theta^*) dx \\
& = 0
\end{aligned} \tag{H.19}$$

$$\begin{aligned}
& \frac{d}{d\theta_{2,k}} E_x [\text{MSE}(x|g_2(x|\theta^*))] = \\
& \quad .05 \int_{-5.8}^{-3.8} g_2(x|\theta^*) \cdot \frac{d}{d\theta_{2,k}} g_2(x|\theta^*) dx \\
& \quad + .1 \int_{-4}^4 (g_2(x|\theta^*) - 1) \cdot \frac{d}{d\theta_{2,k}} g_2(x|\theta^*) dx \\
& \quad + .05 \int_{3.8}^{5.8} g_2(x|\theta^*) \cdot \frac{d}{d\theta_{2,k}} g_2(x|\theta^*) dx \\
& = 0
\end{aligned} \tag{H.20}$$

$$\begin{aligned}
\frac{d}{d\theta_{3,k}} E_x [\text{MSE}(x|g_3(x|\theta^*))] = & \\
.05 \int_{-5.8}^{-3.8} g_3(x|\theta^*) \cdot \frac{d}{d\theta_{3,k}} g_3(x|\theta^*) dx & \\
+ .1 \int_{-4}^4 g_3(x|\theta^*) \cdot \frac{d}{d\theta_{3,k}} g_3(x|\theta^*) dx & \\
+ .05 \int_{3.8}^{5.8} (g_3(x|\theta^*) - 1) \cdot \frac{d}{d\theta_{3,k}} g_3(x|\theta^*) dx & \\
= 0 & \quad \text{(H.21)}
\end{aligned}$$

We use the normal equations of (H.19) – (H.21) to solve for the minimum-MSE parameterization of the high-complexity polynomial classifier. Since there are three 10th-order polynomial discriminant functions, there are three sets of ten equations with ten unknowns. The resulting 10th-order parameters are listed at the bottom of table 4.1, page 104; they were computed from the normal equations with Mathematica.³

³Mathematica is a registered trademark of Wolfram Research, Inc.

Appendix I

Monotonic Fractions Generated by Three Error Measures

This appendix supports section 5.3; it contains derivations for the monotonic fractions of discriminator output space generated by the mean absolute error (MAE), mean-squared error (MSE), and Kullback-Leibler information distance (CE) objective functions. All derivations are performed for the discriminator output space $\mathcal{Y} = [0, 1]^C$, rather than the more general space $\mathcal{Y} = [l, h]^C$. This is done to simplify the notation. In the case of the MAE and MSE error measures, the derivations for $\mathcal{Y} = [0, 1]^C$ yield results that are identical to those for the more general space, owing to the scalable properties of the MAE and MSE objective functions. In the case of the CE error measure, the derivations for $\mathcal{Y} = [0, 1]^C$ yield results that are *not* identical to those for the more general space, owing to the non-scalable properties of the CE objective function. Unfortunately, the CE derivations for $\mathcal{Y} = [l, h]^C$ are tedious, so we limit ourselves to treatment of the space $\mathcal{Y} = [0, 1]^C$. These specific results are qualitatively representative of those for the more general space, as long as l and h are finite — a constraint that is consistent with (2.60).

As in section 5.3, we assume that y_{τ} is always y_1 in order to simplify notation further.

I.1 MAE Monotonic Fractions

Assuming $\neg D = l = 0$ and $D = h = 1$, (5.37) and (5.38) ensure that an example of ω_1 is correctly classified if $\text{MAE} < 1$, such that

$$y_1 > \sum_{j=2}^C y_j \quad (\text{I.1})$$

Thus, we can compute the monotonic correct fraction of discriminator output space $\text{MAECF}_{\text{mono}}(C)$ by using (I.1) to set the limits of integration for the C -tuple cardinality (i.e., volume) integral thus:

$$\begin{aligned}
\text{MAECF}_{\text{mono}}(C) &= \underbrace{\int_0^1 \int_0^{y_1} \int_0^{y_1 - y_2} \cdots \int_0^{y_1 - \sum_{j=2}^{C-1} y_j} dy_C \cdots dy_2 dy_1}_{C-1 \text{ integral terms}} \\
&= \int_0^1 \cdots \int_0^{y_1 - \sum_{j=2}^{C-2} y_j} \left[y_1 - \sum_{j=2}^{C-1} y_j \right] dy_{C-1} \cdots dy_1 \\
&= \int_0^1 \cdots \int_0^{y_1 - \sum_{j=2}^{C-2} y_j} \left[y_1 - \sum_{j=2}^{C-2} y_j - y_{C-1} \right] dy_{C-1} \cdots dy_1 \\
&= \int_0^1 \cdots \int_0^{y_1 - \sum_{j=2}^{C-3} y_j} -\frac{1}{2} \cdot \left[y_1 - \sum_{j=2}^{C-2} y_j - y_{C-1} \right]^2 \Big|_0^{y_1 - \sum_{j=2}^{C-2} y_j} dy_{C-2} \cdots dy_1 \\
&= \int_0^1 \cdots \int_0^{y_1 - \sum_{j=2}^{C-3} y_j} \frac{1}{2} \cdot \left[y_1 - \sum_{j=2}^{C-2} y_j \right]^2 dy_{C-2} \cdots dy_1 \\
&= \int_0^1 \cdots \int_0^{y_1 - \sum_{j=2}^{C-4} y_j} \frac{1}{2 \cdot 3} \cdot \left[y_1 - \sum_{j=2}^{C-3} y_j \right]^3 dy_{C-3} \cdots dy_1 \\
&\vdots \\
&= \frac{1}{C!} = \frac{1}{\Gamma(C+1)} \tag{I.2}
\end{aligned}$$

Assuming $\neg D = l = 0$ and $D = h = 1$, (5.33)–(5.36) ensure that an example of ω_1 is incorrectly classified if $\text{MAE} \geq C - 1$:

$$1 - y_1 + \sum_{j=2}^C y_j \geq C - 1 \tag{I.3}$$

Equivalently,

$$y_1 \leq \sum_{j=2}^C y_j - (C - 2) \tag{I.4}$$

If we let $y'_j = 1 - y_j$, (I.4) can be expressed as

$$y'_1 \geq \sum_{j=2}^C y'_j \quad (I.5)$$

Thus, we can compute the monotonic incorrect fraction of discriminator output space $\text{MAE} \mathcal{IF}_{\text{mono}}(C)$ by using (I.5) to set the limits of integration for the C -tuple cardinality (i.e., volume) integral in precisely the same manner we use (I.1) to set the limits of (I.2):

$$\begin{aligned} \text{MAE} \mathcal{IF}_{\text{mono}}(C) &= \int_0^1 \underbrace{\int_0^{y'_1} \int_0^{y'_1 - y'_2} \cdots \int_0^{y'_1 - \sum_{j=2}^{C-1} y'_j} dy'_C \cdots dy'_2 dy'_1}_{C-1 \text{ integral terms}} \\ &= \frac{1}{C!} = \frac{1}{\Gamma(C+1)} \end{aligned} \quad (I.6)$$

Recall from (5.30)

$$\mathcal{MF} = \mathcal{IF}_{\text{mono}} + \mathcal{CF}_{\text{mono}}, \quad (I.7)$$

so, by (I.2), (I.6), and (I.7),

$$\begin{aligned} \text{MAE} \mathcal{IF}_{\text{mono}}(C) &= \frac{1}{C!} = \frac{1}{\Gamma(C+1)} \\ \text{MAE} \mathcal{CF}_{\text{mono}}(C) &= \frac{1}{C!} = \frac{1}{\Gamma(C+1)} \quad ; \quad C \geq 2 \\ \therefore \text{MAE} \mathcal{MF}(C) &= \frac{2}{C!} = \frac{2}{\Gamma(C+1)} \end{aligned} \quad (I.8)$$

Thus, (5.40) – (5.42) are derived.

I.2 MSE Monotonic Fractions

Assuming $\neg D = l = 0$ and $D = h = 1$, (5.52) and (5.53) ensure that an example of ω_1 is correctly classified if $\text{MSE} < \frac{1}{4}$; one point in $\mathcal{Y}$ generating this value of MSE occurs at

$$y_1 = y_2 = \frac{1}{2}; \quad y_j = 0 \quad \forall k > 2 \quad (I.9)$$

Thus, the lower bound on y_1 in (5.54) necessary to ensure that an example of ω_1 is correctly classified reduces to

$$y_1 > 1 - \left[\frac{1}{2} - \sum_{j=2}^c (y_j)^2 \right]^{\frac{1}{2}} \quad (\text{I.10})$$

It can be shown that (I.10) yields an expression for $\text{MSECF}_{mono}(C)$ that is equal to 2^{-c} times the volume of the C -dimensional sphere with radius $\frac{1}{\sqrt{2}}$, centered at the point $\mathbf{Y}_{correct}$ (recall from (5.3) that $\mathbf{Y}_{correct}$ is, in the case of an example of ω_1 , the point at which $y_1 = 1$ and $y_j = 0 \forall j \neq 1$).

Given the volume of C -dimensional sphere with radius $\frac{1}{\sqrt{2}}$ [4, pg.411],

$$\begin{aligned} \text{MSECF}_{mono}(C) &= 2^{-c} \frac{\pi^{\frac{C}{2}}}{\Gamma(\frac{C}{2} + 1)} \left(\frac{1}{\sqrt{2}} \right)^C \\ &= \frac{\left(\frac{\pi}{8} \right)^{\frac{C}{2}}}{\Gamma(\frac{C}{2} + 1)} \end{aligned} \quad (\text{I.11})$$

Assuming $\neg D = l = 0$ and $D = h = 1$, (5.47)–(5.50) ensure that an example of ω_1 is incorrectly classified if $\text{MSE} \geq \frac{C-1}{2}$:

$$(1 - y_1)^2 + \sum_{j=2}^c (y_j)^2 \geq C - 1 \quad (\text{I.12})$$

Equivalently,

$$y_1 \leq 1 - \left[(C - 1) - \sum_{j=2}^c (y_j)^2 \right]^{\frac{1}{2}} \quad (\text{I.13})$$

Equation (I.13) leads to a relatively complicated C -tuple cardinality (i.e., volume) integral for $\text{MSEIF}_{mono}(C)$. We spare the effort of evaluating the integral explicitly by bounding it from above as follows. Compare the condition for a surely MSE-misclassified example in (I.12) with that for a surely MAE-misclassified example in (I.3). Equation (I.12) defines the inner boundary — as measured from $\mathbf{Y}_{correct}$ — of the monotonic region of incorrect space, given the MSE objective function. Likewise, (I.3) defines the inner boundary — as measured from $\mathbf{Y}_{correct}$ — of the monotonic region of incorrect space, given the MAE objective function. In fact, the set of discriminator output states that satisfy (I.12) describes a convex hypersurface. Each point on this hypersurface has a Euclidean distance from $\mathbf{Y}_{correct}$ that is at least as great as that of any point on the hyperplane described by (I.3). As a result, the monotonic region of incorrect space generated by the MSE objective function is always fully enclosed by the monotonic region of incorrect

space generated by the MAE objective function. It follows immediately that the monotonic incorrect fraction of discriminator output space generated by MSE is bounded from above by its MAE-generated counterpart:

$$\begin{aligned} \text{MSEIF}_{\text{mono}}(C) &< \text{MAEIF}_{\text{mono}}(C) \\ \therefore \text{MSEIF}_{\text{mono}}(C) &< \frac{1}{\Gamma(C+1)} \end{aligned} \quad (1.14)$$

Since the upper bound of (1.14) decreases super-exponentially with C , so does $\text{MSEIF}_{\text{mono}}(C)$. By (1.7), (1.11), and (1.14),

$$\begin{aligned} \text{MSEIF}_{\text{mono}}(C) &< \frac{1}{\Gamma(C+1)} \\ \text{MSECF}_{\text{mono}}(C) &= \frac{\left(\frac{\pi}{8}\right)^{\frac{C}{2}}}{\Gamma\left(\frac{C}{2}+1\right)} \quad ; \quad C \geq 2 \quad (1.15) \\ \therefore \text{MSEMF}(C) &< \frac{\left(\frac{\pi}{8}\right)^{\frac{C}{2}}}{\Gamma\left(\frac{C}{2}+1\right)} + \frac{1}{\Gamma(C+1)} < \frac{2}{\Gamma\left(\frac{C}{2}+1\right)} \end{aligned}$$

Thus, (5.55) – (5.57) are derived.

1.3 Kullback-Leibler Monotonic Fractions

Assuming $\neg D = l = 0$ and $D = h = 1$, the CE expression of (5.62) reduces to

$$\text{CE} = -\log(y_1) - \sum_{j=2}^C \log(1 - y_j) \quad (1.16)$$

The minimum value of CE generated by an incorrectly classified example of ω_1 is, by (5.68), $-\log(\lambda)$, where $\lambda = \frac{1}{4}$. Thus, the monotonic correct fraction of discriminator output space is described by the set of points in $\mathcal{Y}$ satisfying the following equation:

$$\log(y_1) + \sum_{j=2}^C \log(1 - y_j) < \log(\lambda) \quad (1.17)$$

Equivalently,

$$y_1 \cdot \prod_{j=2}^C (1 - y_j) < \lambda \quad (1.18)$$

Thus, we can compute the monotonic correct fraction of discriminator output space $\text{CECF}_{\text{mono}}(C)$ by using (I.18) to set the limits of integration for the C -tuple cardinality (i.e., volume) integral thus:

$$\text{CECF}_{\text{mono}}(C) = \underbrace{\int_{\lambda}^1 \int_0^{1-\lambda/y_1} \int_0^{1-\lambda[y_1 \cdot (1-y_2)]^{-1}} \cdots \int_0^{1-\lambda[y_1 \cdot \prod_{j=2}^{C-1} (1-y_j)]^{-1}} dy_C \cdots dy_2 dy_1}_{C-1 \text{ integral terms}} \quad (I.19)$$

Using the short-hand notation

$$B_{C-k} = \lambda \left[y_1 \cdot \prod_{j=2}^{C-k} (1-y_j) \right]^{-1} \quad (I.20)$$

we restate (I.19) thus:

$$\begin{aligned} \text{CECF}_{\text{mono}}(C) &= \int_{\lambda}^1 \cdots \int_0^{1-B_{C-2}} 1 - B_{C-1} dy_{C-1} \cdots dy_1 \\ &= \int_{\lambda}^1 \cdots \int_0^{1-B_{C-2}} 1 - \frac{1}{1-y_{C-1}} B_{C-2} dy_{C-1} \cdots dy_1 \\ &= \int_{\lambda}^1 \cdots \int_0^{1-B_{C-2}} y_{C-1} + B_{C-2} \cdot \ln(1-y_{C-1}) \Big|_0^{1-B_{C-2}} dy_{C-2} \cdots dy_1 \\ &= \int_{\lambda}^1 \cdots \int_0^{1-B_{C-2}} 1 - B_{C-2} + B_{C-2} \cdot \ln(B_{C-2}) dy_{C-2} \cdots dy_1 \\ &= \int_{\lambda}^1 \cdots \int_0^{1-B_{C-3}} 1 - B_{C-3} + B_{C-3} \cdot \ln(B_{C-3}) - \frac{B_{C-3}}{2} \cdot [\ln(B_{C-3})]^2 dy_{C-3} \cdots dy_1 \\ &\vdots \\ &= 1 - \lambda \cdot \sum_{j=0}^{C-1} \frac{(-1)^j}{j!} \cdot [\ln(\lambda)]^j ; \\ \mathcal{Y} &= [l=0, h=1]^C, \quad \lambda = \frac{1}{4} \end{aligned} \quad (I.21)$$

Thus, (5.70) is derived.

Assuming the more general $\neg D = l$ and $D = h$, the CE expression of (5.62) is

$$CE = -\log(y_1 - l) - \sum_{j=2}^c \log(h - y_j) \quad (I.22)$$

The minimum value of CE generated by an incorrectly classified example of ω_1 is, by (5.68), $-C \cdot \log(h - l) - \log(\lambda)$, where $\lambda = \frac{1}{4}$. Thus, the monotonic correct fraction of discriminator output space is described by the set of points in $\mathcal{Y}$ satisfying the following equation:

$$\log(y_1 - l) + \sum_{j=2}^c \log(h - y_j) < C \cdot \log(h - l) + \log(\lambda) \quad (I.23)$$

Equivalently,

$$(y_1 - l) \cdot \prod_{j=2}^c (h - y_j) < \lambda \cdot (h - l)^C \quad (I.24)$$

Thus, the more general version of (I.21) is given by

$$CECF_{mono}(C) = \frac{1}{(h - l)^C} \cdot \int_{\zeta}^h \underbrace{\dots \int_l^{h - [\lambda \cdot (h - l)^C]} [(y_1 - l) \cdot \prod_{j=2}^{C-1} (h - y_j)]^{-1}}_{C-1 \text{ integral terms}} dy_C \dots dy_2 dy_1, \quad (I.25)$$

where

$$\zeta = \lambda \cdot (h + 3l) \quad (I.26)$$

Evaluating (I.25) becomes a non-trivial exercise in bookkeeping, which we omit for the sake of brevity. As mentioned previously, the derivations for $\mathcal{Y} = [0, 1]^C$ yield results that are qualitatively representative of those for the more general space, as long as l and h are finite — a constraint that is consistent with (2.60).

Appendix J

Tabulated Die Casting Bounds

Section 6.4.1 derives a greatest lower bound $\sim n_{\Delta}$ on the number of die casts n_{Δ} necessary for the most likely face $\omega_{(1)}$ of an unfair die to become empirically evident with probability at least $\alpha = 1 - d = .95$:

$$\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05] \geq \underbrace{20}_{\zeta} \frac{P(\omega_{(1)}) \cdot (1 - P(\omega_{(1)})) + P(\omega_{(2)}) \cdot (1 - P(\omega_{(2)}))}{(P(\omega_{(1)}) - P(\omega_{(2)}))^2} \quad (\text{J.1})$$

Recall that $P(\omega_{(i)})$ is short-hand notation for $P_{W|X}(\omega_{(i)} | X)$. Through the Monte Carlo simulations tabulated below, we have found that the most likely die face remains empirically evident with empirical probability not less than .95 if ζ above is reduced from the Chebyshev-imposed value of 20 to 9.

The following table compares $\sim n_{\Delta}$ ($\zeta = 9$ in (J.1) above) with empirical estimates of n_{Δ} , which we denote by $\hat{n}_{\Delta}$. The empirical estimates were obtained by simulating 1,000 independent die casting sequences, each having up to 10,000 casts, for each tabulated value of $P(\omega_{(1)})$ and $P(\omega_{(2)})$. The value of n above which $\omega_{(1)}$ became empirically evident (i.e., above which $\hat{P}(\omega_{(1)})$ remained maximal for all subsequent casts of the die) was recorded for each trial. The values of n for all 1,000 trials were sorted, and $\hat{n}_{\Delta}$ was taken to be the value of n at the 950th position from the bottom of the sorted list (i.e., the 95th percentile of the 1,000 trials).

The number of faces C_{min} on the die for each set of 1,000 trials was chosen to be the smallest number for which the lesser ranked face probabilities did not exceed $P(\omega_{(2)})$:

$$\text{choose } C_{min} \text{ s.t. } P(\omega_{(j)}) \leq P(\omega_{(2)}) \quad \forall j > 2 \quad (\text{J.2})$$

The lesser ranked probabilities $P(\omega_{(3)}), \dots, P(\omega_{(C-1)})$ were set to the value of $P(\omega_{(2)})$; the remaining probability $P(\omega_{(C)})$ was set to the value $1 - \sum_{i=1}^{C-1} P(\omega_{(i)})$. This choice of C and the lesser ranked face probabilities is approximately worst-case, in that it ensures that the lesser ranked face probabilities are

as large as possible. This, in turn, tends to maximize the the number of die casts necessary for the most likely die face to become empirically evident, given a particular choice of top ranked face probabilities¹ $\langle P(\omega_{(1)}), P(\omega_{(2)}) \rangle$.

The values of $\hat{n}_\Delta$ listed below are not rounded up for a *a posteriori* class differential values greater than .2 (i.e., for $\Delta_{W|X}(\omega_{(1)} | X) = P(\omega_{(1)}) - P(\omega_{(2)}) > .2$); they are rounded up to the nearest value divisible by 5 for a *a posteriori* class differential values greater than .1; they are rounded up to the nearest value divisible by 10 for a *a posteriori* class differential values not greater than .1. The bound $\sim n_\Delta$ is a relatively tight one on $\hat{n}_\Delta$, not over-estimating $\hat{n}_\Delta$ by more than about 100%. Likewise, $\sim n_\Delta$ under-estimates $\hat{n}_\Delta$ only for values of $\hat{n}_\Delta \lesssim 10$.

$P(\omega_{(1)})$	$P(\omega_{(2)})$	C_{min}	Empirical Number of Casts $\hat{n}_\Delta[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_\Delta[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.1	0.02	46	140	155
0.1	0.04	24	260	321
0.1	0.06	16	640	824
0.1	0.08	13	2800	3681
0.12	0.02	45	100	113
0.12	0.04	23	170	203
0.12	0.06	16	320	405
0.12	0.08	12	730	1008
0.12	0.1	10	3190	4401
0.14	0.02	44	75	88
0.14	0.04	23	130	143
0.14	0.06	16	190	249
0.14	0.08	12	370	485
0.14	0.1	10	860	1184
0.14	0.12	9	3460	5085
0.16	0.02	43	60	71
0.16	0.04	22	90	109
0.16	0.06	15	140	172
0.16	0.08	12	230	293
0.16	0.1	10	390	562
0.16	0.12	8	990	1351
0.16	0.14	7	4050	5734
0.18	0.02	42	50	59
0.18	0.04	22	75	86
0.18	0.06	15	100	128
0.18	0.08	12	160	200

¹A rigorous proof of this assertion is beyond our interest and stamina; we choose this protocol as a reasonable means of approximating the number and values of the lesser-ranked face probabilities that would result in the largest number of die casts required for the most likely face to become empirically evident, given the probabilities for the two most likely die faces.

$P(\omega_{(1)})$	$P(\omega_{(2)})$	C_{min}	Empirical Number of Casts $\hat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.18	0.1	10	250	335
0.18	0.12	8	440	634
0.18	0.14	7	1050	1508
0.18	0.16	7	4420	6346
0.2	0.02	41	45	50
0.2	0.04	21	55	70
0.2	0.06	15	80	100
0.2	0.08	11	120	147
0.2	0.1	9	160	226
0.2	0.12	8	270	374
0.2	0.14	7	520	702
0.2	0.16	6	1160	1657
0.2	0.18	6	4250	6922
0.22	0.02	40	40	44
0.22	0.04	21	50	59
0.22	0.06	15	65	81
0.22	0.08	11	90	113
0.22	0.1	9	110	164
0.22	0.12	8	180	250
0.22	0.14	7	290	411
0.22	0.16	6	530	766
0.22	0.18	6	1100	1796
0.22	0.2	5	4890	7462
0.24	0.02	39	33	38
0.24	0.04	20	45	50
0.24	0.06	14	55	67
0.24	0.08	11	70	91
0.24	0.1	9	90	126
0.24	0.12	8	135	181
0.24	0.14	7	180	273
0.24	0.16	6	290	446
0.24	0.18	6	570	826
0.24	0.2	5	1120	1927
0.24	0.22	5	4820	7966
0.26	0.02	38	29	34
0.26	0.04	20	34	43
0.26	0.06	14	50	56
0.26	0.08	11	60	74
0.26	0.1	9	75	100
0.26	0.12	8	105	137
0.26	0.14	7	135	196
0.26	0.16	6	210	295
0.26	0.18	6	350	479
0.26	0.2	5	560	882
0.26	0.22	5	1290	2048
0.26	0.24	5	5540	8434
0.28	0.02	37	25	30

$P(\omega_{(1)})$	$P(\omega_{(2)})$	C_{min}	Empirical Number of Casts $\hat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.28	0.04	19	31	38
0.28	0.06	14	39	48
0.28	0.08	10	55	62
0.28	0.1	9	60	82
0.28	0.12	7	80	109
0.28	0.14	7	105	148
0.28	0.16	6	145	211
0.28	0.18	6	230	315
0.28	0.2	5	310	509
0.28	0.22	5	630	933
0.28	0.24	5	1320	2160
0.28	0.26	4	5260	8865
0.3	0.02	36	22	27
0.3	0.04	19	26	34
0.3	0.06	13	34	42
0.3	0.08	10	38	53
0.3	0.1	8	55	68
0.3	0.12	7	65	88
0.3	0.14	6	80	117
0.3	0.16	6	110	159
0.3	0.18	5	140	224
0.3	0.2	5	210	333
0.3	0.22	5	360	537
0.3	0.24	4	570	981
0.3	0.26	4	1350	2264
0.3	0.28	4	5870	9261
0.32	0.02	35	20	24
0.32	0.04	18	26	30
0.32	0.06	13	31	37
0.32	0.08	10	37	46
0.32	0.1	8	47	58
0.32	0.12	7	60	73
0.32	0.14	6	75	94
0.32	0.16	6	80	124
0.32	0.18	5	115	168
0.32	0.2	5	170	236
0.32	0.22	5	250	351
0.32	0.24	4	290	563
0.32	0.26	4	660	1025
0.32	0.28	4	1510	2358
0.32	0.3	4	5770	9621
0.34	0.02	34	19	22
0.34	0.04	18	23	27
0.34	0.06	12	30	33
0.34	0.08	10	33	40
0.34	0.1	8	38	50
0.34	0.12	7	45	62

$P(\omega_{(1)})$	$P(\omega_{(2)})$	C_{min}	Empirical Number of Casts $\hat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.34	0.14	6	65	78
0.34	0.16	6	75	100
0.34	0.18	5	90	131
0.34	0.2	5	115	177
0.34	0.22	5	170	248
0.34	0.24	4	240	367
0.34	0.26	4	340	587
0.34	0.28	4	680	1065
0.34	0.3	4	1600	2444
0.34	0.32	4	6360	9945
0.36	0.02	33	16	20
0.36	0.04	17	22	24
0.36	0.06	12	25	29
0.36	0.08	9	30	35
0.36	0.1	8	31	43
0.36	0.12	7	41	53
0.36	0.14	6	47	66
0.36	0.16	5	63	83
0.36	0.18	5	80	105
0.36	0.2	5	95	138
0.36	0.22	4	110	185
0.36	0.24	4	165	258
0.36	0.26	4	250	381
0.36	0.28	4	390	608
0.36	0.3	4	680	1101
0.36	0.32	3	1530	2520
0.36	0.34	3	5360	10233
0.38	0.02	32	16	18
0.38	0.04	17	19	22
0.38	0.06	12	24	26
0.38	0.08	9	25	31
0.38	0.1	8	29	38
0.38	0.12	7	37	46
0.38	0.14	6	42	56
0.38	0.16	5	50	69
0.38	0.18	5	62	87
0.38	0.2	5	85	110
0.38	0.22	4	105	144
0.38	0.24	4	120	192
0.38	0.26	4	175	268
0.38	0.28	4	245	394
0.38	0.3	4	390	627
0.38	0.32	3	570	1133
0.38	0.34	3	1370	2588
0.38	0.36	3	4950	10485
0.4	0.02	31	14	17
0.4	0.04	16	17	20

$P(\omega_{(1)})$ $P(\omega_{(2)})$ C_{min} Empirical Number of Casts Bound
 $\hat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$ $=$ $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$

0.4	0.06	11	20	24
0.4	0.08	9	23	28
0.4	0.1	7	25	33
0.4	0.12	6	33	40
0.4	0.14	6	37	48
0.4	0.16	5	46	59
0.4	0.18	5	53	73
0.4	0.2	4	71	90
0.4	0.22	4	75	115
0.4	0.24	4	95	149
0.4	0.26	4	110	199
0.4	0.28	4	185	276
0.4	0.3	3	245	405
0.4	0.32	3	340	644
0.4	0.34	3	540	1161
0.4	0.36	3	1340	2646
0.4	0.38	3	5790	10701
0.42	0.02	30	14	15
0.42	0.04	16	15	18
0.42	0.06	11	17	21
0.42	0.08	9	21	25
0.42	0.1	7	23	30
0.42	0.12	6	28	35
0.42	0.14	6	36	42
0.42	0.16	5	36	51
0.42	0.18	5	43	62
0.42	0.2	4	53	76
0.42	0.22	4	61	94
0.42	0.24	4	80	119
0.42	0.26	4	95	154
0.42	0.28	4	140	205
0.42	0.3	3	160	284
0.42	0.32	3	200	416
0.42	0.34	3	330	659
0.42	0.36	3	550	1185
0.42	0.38	3	1590	2696
0.42	0.4	3	5630	10881
0.44	0.02	29	12	14
0.44	0.04	15	15	17
0.44	0.06	11	17	19
0.44	0.08	8	19	23
0.44	0.1	7	20	27
0.44	0.12	6	26	31
0.44	0.14	5	29	37
0.44	0.16	5	33	44
0.44	0.18	5	44	53
0.44	0.2	4	46	64

$P(\omega_{(1)})$ $P(\omega_{(2)})$ C_{min} $\widehat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d]$ $=$ $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$ Bound
 $.05]$

0.44	0.22	4	52	78
0.44	0.24	4	67	97
0.44	0.26	4	85	122
0.44	0.28	3	105	158
0.44	0.3	3	125	210
0.44	0.32	3	150	290
0.44	0.34	3	230	424
0.44	0.36	3	370	671
0.44	0.38	3	650	1205
0.44	0.4	3	1600	2736
0.44	0.42	3	6030	11025
0.46	0.02	28	12	13
0.46	0.04	15	14	15
0.46	0.06	10	16	18
0.46	0.08	8	19	21
0.46	0.1	7	20	24
0.46	0.12	6	22	28
0.46	0.14	5	28	33
0.46	0.16	5	29	39
0.46	0.18	4	34	46
0.46	0.2	4	37	55
0.46	0.22	4	44	66
0.46	0.24	4	53	81
0.46	0.26	4	70	100
0.46	0.28	3	80	125
0.46	0.3	3	90	162
0.46	0.32	3	130	214
0.46	0.34	3	155	296
0.46	0.36	3	230	431
0.46	0.38	3	370	681
0.46	0.4	3	610	1221
0.46	0.42	3	1400	2768
0.46	0.44	3	5770	11133
0.48	0.02	27	11	12
0.48	0.04	14	14	14
0.48	0.06	10	15	16
0.48	0.08	8	16	19
0.48	0.1	7	19	22
0.48	0.12	6	20	25
0.48	0.14	5	23	29
0.48	0.16	5	27	34
0.48	0.18	4	31	40
0.48	0.2	4	31	48
0.48	0.22	4	41	57
0.48	0.24	4	47	68
0.48	0.26	3	56	83
0.48	0.28	3	64	102

$P(\omega_{(1)})$	$P(\omega_{(2)})$	C_{min}	Empirical Number of Casts $\hat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.48	0.3	3	70	128
0.48	0.32	3	105	165
0.48	0.34	3	130	218
0.48	0.36	3	150	300
0.48	0.38	3	235	437
0.48	0.4	3	390	689
0.48	0.42	3	690	1233
0.48	0.44	3	1590	2790
0.48	0.46	3	5800	11205
0.5	0.02	26	10	11
0.5	0.04	14	12	13
0.5	0.06	10	14	15
0.5	0.08	8	15	17
0.5	0.1	6	17	20
0.5	0.12	6	20	23
0.5	0.14	5	22	26
0.5	0.16	5	22	30
0.5	0.18	4	25	35
0.5	0.2	4	27	41
0.5	0.22	4	37	49
0.5	0.24	4	45	58
0.5	0.26	3	48	70
0.5	0.28	3	48	84
0.5	0.3	3	59	104
0.5	0.32	3	70	130
0.5	0.34	3	85	167
0.5	0.36	3	130	221
0.5	0.38	3	165	304
0.5	0.4	3	265	441
0.5	0.42	3	380	695
0.5	0.44	3	760	1242
0.5	0.46	3	1710	2804
0.5	0.48	3	7180	11242
0.52	0.02	25	10	10
0.52	0.04	13	12	12
0.52	0.06	9	14	14
0.52	0.08	7	14	16
0.52	0.1	6	15	18
0.52	0.12	5	18	20
0.52	0.14	5	20	24
0.52	0.16	4	21	27
0.52	0.18	4	24	31
0.52	0.2	4	27	36
0.52	0.22	4	30	43
0.52	0.24	3	34	50
0.52	0.26	3	38	59
0.52	0.28	3	44	71

$P(\omega_{(1)})$ $P(\omega_{(2)})$ C_{min} $\widehat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d, .05]$ $=$ $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$ Bound

0.52	0.3	3	51	86
0.52	0.32	3	70	106
0.52	0.34	3	75	132
0.52	0.36	3	105	169
0.52	0.38	3	145	223
0.52	0.4	3	200	306
0.52	0.42	3	315	444
0.52	0.44	3	500	698
0.52	0.46	3	810	1245
0.54	0.02	24	9	9
0.54	0.04	13	10	11
0.54	0.06	9	11	12
0.54	0.08	7	13	14
0.54	0.1	6	15	16
0.54	0.12	5	16	19
0.54	0.14	5	18	21
0.54	0.16	4	21	24
0.54	0.18	4	21	28
0.54	0.2	4	25	32
0.54	0.22	4	25	37
0.54	0.24	3	31	44
0.54	0.26	3	29	51
0.54	0.28	3	37	60
0.54	0.3	3	49	72
0.54	0.32	3	53	87
0.54	0.34	3	73	107
0.54	0.36	3	90	133
0.54	0.38	3	105	171
0.54	0.4	3	140	225
0.54	0.42	3	175	308
0.54	0.44	3	285	446
0.56	0.02	23	9	9
0.56	0.04	12	10	10
0.56	0.06	9	11	11
0.56	0.08	7	11	13
0.56	0.1	6	15	15
0.56	0.12	5	15	17
0.56	0.14	5	15	19
0.56	0.16	4	17	22
0.56	0.18	4	20	25
0.56	0.2	4	21	29
0.56	0.22	3	25	33
0.56	0.24	3	26	38
0.56	0.26	3	31	44
0.56	0.28	3	30	52
0.56	0.3	3	37	61
0.56	0.32	3	47	73

$P(\omega_{(1)})$	$P(\omega_{(2)})$	C_{min}	Empirical Number of Casts $\widehat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.56	0.34	3	54	88
0.56	0.36	3	64	108
0.56	0.38	3	95	134
0.56	0.4	3	100	171
0.56	0.42	3	140	225
0.58	0.02	22	8	8
0.58	0.04	12	9	9
0.58	0.06	8	10	10
0.58	0.08	7	11	12
0.58	0.1	6	12	14
0.58	0.12	5	13	15
0.58	0.14	4	14	17
0.58	0.16	4	16	20
0.58	0.18	4	18	23
0.58	0.2	4	21	26
0.58	0.22	3	20	29
0.58	0.24	3	23	34
0.58	0.26	3	26	39
0.58	0.28	3	28	45
0.58	0.3	3	30	53
0.58	0.32	3	35	62
0.58	0.34	3	51	74
0.58	0.36	3	52	89
0.58	0.38	3	69	108
0.58	0.4	3	95	135
0.6	0.02	21	8	7
0.6	0.04	11	9	8
0.6	0.06	8	9	10
0.6	0.08	6	11	11
0.6	0.1	5	13	12
0.6	0.12	5	12	14
0.6	0.14	4	14	16
0.6	0.16	4	16	18
0.6	0.18	4	17	20
0.6	0.2	3	19	23
0.6	0.22	3	20	26
0.6	0.24	3	24	30
0.6	0.26	3	22	34
0.6	0.28	3	27	39
0.6	0.3	3	31	45
0.6	0.32	3	33	53
0.6	0.34	3	44	62
0.6	0.36	3	44	74
0.6	0.38	3	61	89
0.62	0.02	20	8	7
0.62	0.04	11	8	8
0.62	0.06	8	9	9

$P(\omega_{(1)})$ $P(\omega_{(2)})$ C_{min} $\widehat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d]$ $=$ $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$ Bound
 .05]

0.62	0.08	6	10	10
0.62	0.1	5	11	11
0.62	0.12	5	10	13
0.62	0.14	4	13	14
0.62	0.16	4	15	16
0.62	0.18	4	13	18
0.62	0.2	3	16	21
0.62	0.22	3	16	23
0.62	0.24	3	18	27
0.62	0.26	3	21	30
0.62	0.28	3	25	35
0.62	0.3	3	27	40
0.62	0.32	3	33	46
0.62	0.34	3	36	53
0.62	0.36	3	44	63
0.64	0.02	19	7	6
0.64	0.04	10	8	7
0.64	0.06	7	9	8
0.64	0.08	6	8	9
0.64	0.1	5	10	10
0.64	0.12	4	12	12
0.64	0.14	4	11	13
0.64	0.16	4	14	15
0.64	0.18	3	13	17
0.64	0.2	3	13	19
0.64	0.22	3	15	21
0.64	0.24	3	16	24
0.64	0.26	3	19	27
0.64	0.28	3	19	30
0.64	0.3	3	27	35
0.64	0.32	3	31	40
0.64	0.34	3	35	46
0.66	0.02	18	6	6
0.66	0.04	10	8	7
0.66	0.06	7	8	8
0.66	0.08	6	8	8
0.66	0.1	5	9	10
0.66	0.12	4	10	11
0.66	0.14	4	10	12
0.66	0.16	4	11	13
0.66	0.18	3	13	15
0.66	0.2	3	14	17
0.66	0.22	3	14	19
0.66	0.24	3	17	21
0.66	0.26	3	16	24
0.66	0.28	3	18	27
0.66	0.3	3	22	31

$P(\omega_{(1)})$	$P(\omega_{(2)})$	C_{min}	Empirical Number of Casts $\widehat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.66	0.32	3	25	35
0.68	0.02	17	6	5
0.68	0.04	9	7	6
0.68	0.06	7	7	7
0.68	0.08	5	8	8
0.68	0.1	5	9	9
0.68	0.12	4	8	10
0.68	0.14	4	10	11
0.68	0.16	3	12	12
0.68	0.18	3	11	14
0.68	0.2	3	12	15
0.68	0.22	3	12	17
0.68	0.24	3	13	19
0.68	0.26	3	15	21
0.68	0.28	3	18	24
0.68	0.3	3	17	27
0.7	0.02	16	6	5
0.7	0.04	9	6	6
0.7	0.06	6	7	6
0.7	0.08	5	7	7
0.7	0.1	4	8	8
0.7	0.12	4	9	9
0.7	0.14	4	9	10
0.7	0.16	3	9	11
0.7	0.18	3	10	12
0.7	0.2	3	12	14
0.7	0.22	3	11	15
0.7	0.24	3	13	17
0.7	0.26	3	15	19
0.7	0.28	3	15	21
0.72	0.02	15	6	5
0.72	0.04	8	6	5
0.72	0.06	6	7	6
0.72	0.08	5	7	7
0.72	0.1	4	7	7
0.72	0.12	4	8	8
0.72	0.14	3	9	9
0.72	0.16	3	10	10
0.72	0.18	3	10	11
0.72	0.2	3	9	13
0.72	0.22	3	10	14
0.72	0.24	3	12	15
0.72	0.26	3	15	17
0.74	0.02	14	5	4
0.74	0.04	8	5	5
0.74	0.06	6	6	5
0.74	0.08	5	7	6

$P(\omega_{(1)})$	$P(\omega_{(2)})$	c_{min}	Empirical Number of Casts $\hat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.74	0.1	4	7	7
0.74	0.12	4	7	7
0.74	0.14	3	8	8
0.74	0.16	3	9	9
0.74	0.18	3	9	10
0.74	0.2	3	9	11
0.74	0.22	3	11	13
0.74	0.24	3	11	14
0.76	0.02	13	5	4
0.76	0.04	7	5	4
0.76	0.06	5	6	5
0.76	0.08	4	7	5
0.76	0.1	4	6	6
0.76	0.12	3	7	7
0.76	0.14	3	8	8
0.76	0.16	3	8	8
0.76	0.18	3	8	9
0.76	0.2	3	7	10
0.76	0.22	3	11	11
0.78	0.02	12	4	3
0.78	0.04	7	5	4
0.78	0.06	5	6	4
0.78	0.08	4	6	5
0.78	0.1	4	6	6
0.78	0.12	3	7	6
0.78	0.14	3	6	7
0.78	0.16	3	7	8
0.78	0.18	3	7	8
0.78	0.2	3	9	9
0.8	0.02	11	5	3
0.8	0.04	6	5	4
0.8	0.06	5	5	4
0.8	0.08	4	6	5
0.8	0.1	3	6	5
0.8	0.12	3	6	6
0.8	0.14	3	7	6
0.8	0.16	3	7	7
0.8	0.18	3	7	8
0.82	0.02	10	4	3
0.82	0.04	6	4	3
0.82	0.06	4	5	4
0.82	0.08	4	5	4
0.82	0.1	3	5	5
0.82	0.12	3	5	5
0.82	0.14	3	5	6
0.82	0.16	3	5	6
0.84	0.02	9	4	3

$P(\omega_{(1)})$	$P(\omega_{(2)})$	C_{min}	Empirical Number of Casts $\hat{n}_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$	Bound $\sim n_{\Delta}[P(\omega_{(1)}), P(\omega_{(2)}), d = .05]$
0.84	0.04	5	4	3
0.84	0.06	4	5	3
0.84	0.08	3	5	4
0.84	0.1	3	5	4
0.84	0.12	3	5	5
0.84	0.14	3	5	5
0.86	0.02	8	3	2
0.86	0.04	5	3	3
0.86	0.06	4	3	3
0.86	0.08	3	5	3
0.86	0.1	3	5	4
0.86	0.12	3	5	4
0.88	0.02	7	3	2
0.88	0.04	4	3	2
0.88	0.06	3	3	3
0.88	0.08	3	3	3
0.88	0.1	3	3	3
0.9	0.02	6	3	2
0.9	0.04	4	3	2
0.9	0.06	3	3	2
0.9	0.08	3	3	3
0.92	0.02	5	3	2
0.92	0.04	3	3	2
0.92	0.06	3	3	2
0.94	0.02	4	3	1
0.94	0.04	3	3	2
0.96	0.02	3	3	1

Appendix K

A Modified Radial Basis Function Classifier

The Gaussian Radial Basis Function (RBF) neural network architecture (e.g., [18, 95, 104, 92]) employs the following non-linear input-to-output mapping, where $\mathbf{x}$ denotes the input to the RBF unit (or *node*), and $f(\mathbf{x})$ denotes the node's output:

$$f(\mathbf{x}) = \underbrace{\frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma|^{\frac{1}{2}}}}_{\zeta} \cdot \exp \left[-\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right] \quad (\text{K.1})$$

The notation $\mathbf{z}^T$ denotes the transpose of vector $\mathbf{z}$, and m denotes the dimensionality of the input vector $\mathbf{x}$.

The modified radial basis function node is identical to the standard RBF node with two exceptions.

- The covariance matrix Σ associated with each RBF node is diagonal, and all of its diagonal elements have the same value (i.e., the covariance matrix has orthonormal eigenvectors and all of its eigenvalues are identical). The matrix is described by the equation

$$\Sigma = \sigma \mathbf{I}, \quad (\text{K.2})$$

where $\mathbf{I}$ denotes the identity matrix and σ denotes the node's single variance parameter. For this reason, the modified RBF node has $m^2 - 1$ fewer parameters than its standard counterpart. In the case of an 11-element input vector, the modified RBF node has 12 parameters compared to the standard RBF node's 132. Equation (K.2) constrains the RBF node to have hyperspherical (rather than hyperellipsoidal) contours of constant value on the domain of $\mathbf{x}$

- The standard RBF's ζ term in (K.1) is eliminated. This has the effect of bounding the modified RBF's output $f(\mathbf{x})$ on the interval $[0,1]$ — much like the logistic non-linearity. The standard RBF node's

output, by contrast, is bounded on $[0, \infty]$ and has unit area over the domain of x .

Thus, the non-linear input-to-output mapping for the modified RBF unit is given by

$$f(x) = \exp \left[-\frac{1}{2\sigma^2} (x - \mu)^T \mathbf{I} (x - \mu) \right], \quad (\text{K.3})$$

where $\mathbf{I}$ denotes the identity matrix and σ^2 denotes the node's single variance parameter. By having $m^2 - 1$ fewer parameters than its standard counterpart, the modified RBF node has significantly lower functional complexity. Despite their reduced complexity, networks of modified RBF nodes are still capable of forming a classifier with complex non-linear decision boundaries. As a result, such networks are well suited for differential learning (e.g., see chapter 10). We refer to the differentially-generated variants as Differential Radial Basis Function (DRBF) classifiers.

Appendix L

Anderson & Fisher's Iris Data

L.1 Original Iris Data¹

The following data describing three varieties of Iris (*Iris virginica*, *Iris versicolor*, and *Iris setosa*) was originally collected by E. Anderson[3], and subsequently used by R. A. Fisher in his seminal paper on linear discriminants [34]. The feature vector $\mathbf{X}$ has four elements: x_1 denotes sepal length, x_2 denotes sepal width, x_3 denotes petal length, and x_4 denotes petal width. There are three classes: ω_1 denotes *Iris setosa*, ω_2 denotes *Iris versicolor*, and ω_3 denotes *Iris virginica*. The examples are listed below:

Example	Class	x_1	x_2	x_3	x_4
0	1	5.1	3.5	1.4	0.2
1	1	4.9	3.0	1.4	0.2
2	1	4.7	3.2	1.3	0.2
3	1	4.6	3.1	1.5	0.2
4	1	5.0	3.6	1.4	0.2
5	1	5.4	3.9	1.7	0.4
6	1	4.6	3.4	1.4	0.3
7	1	5.0	3.4	1.5	0.2
8	1	4.4	2.9	1.4	0.2
9	1	4.9	3.1	1.5	0.1
10	1	5.4	3.7	1.5	0.2
11	1	4.8	3.4	1.6	0.2
12	1	4.8	3.0	1.4	0.1
13	1	4.3	3.0	1.1	0.1
14	1	5.8	4.0	1.2	0.2
15	1	5.7	4.4	1.5	0.4
16	1	5.4	3.9	1.3	0.4
17	1	5.1	3.5	1.4	0.3
18	1	5.7	3.8	1.7	0.3
19	1	5.1	3.8	1.5	0.3
20	1	5.4	3.4	1.7	0.2
21	1	5.1	3.7	1.5	0.4
22	1	4.6	3.6	1.0	0.2
23	1	5.1	3.3	1.7	0.5
24	1	4.8	3.4	1.9	0.2
25	1	5.0	3.0	1.6	0.2
26	1	5.0	3.4	1.6	0.4
27	1	5.2	3.5	1.5	0.2
28	1	5.2	3.4	1.4	0.2

¹We thank Professor Casimir Kulikowski of Rutgers University for providing us with an electronic version of Anderson/Fisher's original data.

29	1	4.7	3.2	1.6	0.2
30	1	4.8	3.1	1.6	0.2
31	1	5.4	3.4	1.5	0.4
32	1	5.2	4.1	1.5	0.1
33	1	5.5	4.2	1.4	0.2
34	1	4.9	3.1	1.5	0.2
35	1	5.0	3.2	1.2	0.2
36	1	5.5	3.5	1.3	0.2
37	1	4.9	3.6	1.4	0.1
38	1	4.4	3.0	1.3	0.2
39	1	5.1	3.4	1.5	0.2
40	1	5.0	3.5	1.3	0.3
41	1	4.5	2.3	1.3	0.3
42	1	4.4	3.2	1.3	0.2
43	1	5.0	3.5	1.6	0.6
44	1	5.1	3.8	1.9	0.4
45	1	4.8	3.0	1.4	0.3
46	1	5.1	3.8	1.6	0.2
47	1	4.6	3.2	1.4	0.2
48	1	5.3	3.7	1.5	0.2
49	1	5.0	3.3	1.4	0.2
50	2	7.0	3.2	4.7	1.4
51	2	6.4	3.2	4.5	1.5
52	2	6.9	3.1	4.9	1.5
53	2	5.5	2.3	4.0	1.3
54	2	6.5	2.8	4.6	1.5
55	2	5.7	2.8	4.5	1.3
56	2	6.3	3.3	4.7	1.6
57	2	4.9	2.4	3.3	1.0
58	2	6.6	2.9	4.6	1.3
59	2	5.2	2.7	3.9	1.4
60	2	5.0	2.0	3.5	1.0
61	2	5.9	3.0	4.2	1.5
62	2	6.0	2.2	4.0	1.0
63	2	6.1	2.9	4.7	1.4
64	2	5.6	2.9	3.6	1.3
65	2	6.7	3.1	4.4	1.4
66	2	5.6	3.0	4.5	1.5
67	2	5.8	2.7	4.1	1.0
68	2	6.2	2.2	4.5	1.5
69	2	5.6	2.5	3.9	1.1
70	2	5.9	3.2	4.8	1.8
71	2	6.1	2.8	4.0	1.3
72	2	6.3	2.5	4.9	1.5
73	2	6.1	2.8	4.7	1.2
74	2	6.4	2.9	4.3	1.3
75	2	6.6	3.0	4.4	1.4
76	2	6.8	2.8	4.8	1.4
77	2	6.7	3.0	5.0	1.7
78	2	6.0	2.9	4.5	1.5
79	2	5.7	2.6	3.5	1.0
80	2	5.5	2.4	3.8	1.1
81	2	5.5	2.4	3.7	1.0
82	2	5.8	2.7	3.9	1.2
83	2	6.0	2.7	5.1	1.6
84	2	5.4	3.0	4.5	1.5
85	2	6.0	3.4	4.5	1.6
86	2	6.7	3.1	4.7	1.5
87	2	6.3	2.3	4.4	1.3
88	2	5.6	3.0	4.1	1.3
89	2	5.5	2.5	4.0	1.3
90	2	5.5	2.6	4.4	1.2
91	2	6.1	3.0	4.6	1.4
92	2	5.8	2.6	4.0	1.2
93	2	5.0	2.3	3.3	1.0
94	2	5.6	2.7	4.2	1.3
95	2	5.7	3.0	4.2	1.2
96	2	5.7	2.9	4.2	1.3
97	2	6.2	2.9	4.3	1.3
98	2	5.1	2.5	3.0	1.1

99	2	5.7	2.8	4.1	1.3
100	3	6.3	3.3	6.0	2.5
101	3	5.8	2.7	5.1	1.9
102	3	7.1	3.0	5.9	2.1
103	3	6.3	2.9	5.6	1.8
104	3	6.5	3.0	5.8	2.2
105	3	7.6	3.0	6.6	2.1
106	3	4.9	2.5	4.5	1.7
107	3	7.3	2.9	6.3	1.8
108	3	6.7	2.5	5.8	1.8
109	3	7.2	3.6	6.1	2.5
110	3	6.5	3.2	5.1	2.0
111	3	6.4	2.7	5.3	1.9
112	3	6.8	3.0	5.5	2.1
113	3	5.7	2.5	5.0	2.0
114	3	5.8	2.8	5.1	2.4
115	3	6.4	3.2	5.3	2.3
116	3	6.5	3.0	5.5	1.8
117	3	7.7	3.8	6.7	2.2
118	3	7.7	2.6	6.9	2.3
119	3	6.0	2.2	5.0	1.5
120	3	6.9	3.2	5.7	2.3
121	3	5.6	2.8	4.9	2.0
122	3	7.7	2.8	6.7	2.0
123	3	6.3	2.7	4.9	1.8
124	3	6.7	3.3	5.7	2.1
125	3	7.2	3.2	6.0	1.8
126	3	6.2	2.8	4.8	1.8
127	3	6.1	3.0	4.9	1.8
128	3	6.4	2.8	5.6	2.1
129	3	7.2	3.0	5.8	1.6
130	3	7.4	2.8	6.1	1.9
131	3	7.9	3.8	6.4	2.0
132	3	6.4	2.8	5.6	2.2
133	3	6.3	2.8	5.1	1.5
134	3	6.1	2.6	5.6	1.4
135	3	7.7	3.0	6.1	2.3
136	3	6.3	3.4	5.6	2.4
137	3	6.4	3.1	5.5	1.8
138	3	6.0	3.0	4.8	1.8
139	3	6.9	3.1	5.4	2.1
140	3	6.7	3.1	5.6	2.4
141	3	6.9	3.1	5.1	2.3
142	3	5.8	2.7	5.1	1.9
143	3	6.8	3.2	5.9	2.3
144	3	6.7	3.3	5.7	2.5
145	3	6.7	3.0	5.2	2.3
146	3	6.3	2.5	5.0	1.9
147	3	6.5	3.0	5.2	2.0
148	3	6.2	3.4	5.4	2.3
149	3	5.9	3.0	5.1	1.8

L.2 Normalized Iris Data

The following data was computed via an affine transformation of the original data. We determined the lower and upper bound on each of the four feature vector elements $\{\langle l_1, u_1 \rangle, \dots, \langle l_4, u_4 \rangle\}$; we then transformed each element of the 150 vectors as follows

$$x_i^{j'} = 2 \frac{(x_i^j - l_i)}{(u_i - l_i)} - 1, \quad (\text{L.1})$$

where x_i^j denotes the i th element of example $\mathbf{X}^j$ and $x_i^{j'}$ denotes the post-transformation value of x_i^j . This

affine transformation normalizes each element of the $N = 4$ -dimensional feature vector $\mathbf{X}$ to the closed interval $[-1,1]$. That is, the affine transformation projects the original feature vector $\mathbf{X} \in \mathbb{R}^4$ onto the normalized vector $\mathbf{X}' \in [-1,1]^4$. The normalized data are given below:

Example	Class	x1	x2	x3	x4
0	1	-0.555556	0.250000	-0.864407	-0.916667
1	1	-0.666667	-0.166667	-0.864407	-0.916667
2	1	-0.777778	0.000000	-0.898305	-0.916667
3	1	-0.833333	-0.083333	-0.830508	-0.916667
4	1	-0.611111	0.333333	-0.864407	-0.916667
5	1	-0.388889	0.583333	-0.762712	-0.750000
6	1	-0.833333	0.166667	-0.864407	-0.833333
7	1	-0.611111	0.166667	-0.830508	-0.916667
8	1	-0.944444	-0.250000	-0.864407	-0.916667
9	1	-0.666667	-0.083333	-0.830508	-1.000000
10	1	-0.388889	0.416667	-0.830508	-0.916667
11	1	-0.722222	0.166667	-0.796610	-0.916667
12	1	-0.722222	-0.166667	-0.864407	-1.000000
13	1	-1.000000	-0.166667	-0.966102	-1.000000
14	1	-0.166667	0.666667	-0.932203	-0.916667
15	1	-0.222222	1.000000	-0.830508	-0.750000
16	1	-0.388889	0.583333	-0.898305	-0.750000
17	1	-0.555556	0.250000	-0.864407	-0.833333
18	1	-0.222222	0.500000	-0.762712	-0.833333
19	1	-0.555556	0.500000	-0.830508	-0.833333
20	1	-0.388889	0.166667	-0.762712	-0.916667
21	1	-0.555556	0.416667	-0.830508	-0.750000
22	1	-0.833333	0.333333	-1.000000	-0.916667
23	1	-0.555556	0.083333	-0.762712	-0.666667
24	1	-0.722222	0.166667	-0.694915	-0.916667
25	1	-0.611111	-0.166667	-0.796610	-0.916667
26	1	-0.611111	0.166667	-0.796610	-0.750000
27	1	-0.500000	0.250000	-0.830508	-0.916667
28	1	-0.500000	0.166667	-0.864407	-0.916667
29	1	-0.777778	0.000000	-0.796610	-0.916667
30	1	-0.722222	-0.083333	-0.796610	-0.916667
31	1	-0.388889	0.166667	-0.830508	-0.750000
32	1	-0.500000	0.750000	-0.830508	-1.000000
33	1	-0.333333	0.833333	-0.864407	-0.916667
34	1	-0.666667	-0.083333	-0.830508	-0.916667
35	1	-0.611111	0.000000	-0.932203	-0.916667
36	1	-0.333333	0.250000	-0.898305	-0.916667
37	1	-0.666667	0.333333	-0.864407	-1.000000
38	1	-0.944444	-0.166667	-0.898305	-0.916667
39	1	-0.555556	0.166667	-0.830508	-0.916667
40	1	-0.611111	0.250000	-0.898305	-0.833333
41	1	-0.888889	-0.750000	-0.898305	-0.833333
42	1	-0.944444	0.000000	-0.898305	-0.916667
43	1	-0.611111	0.250000	-0.796610	-0.583333
44	1	-0.555556	0.500000	-0.694915	-0.750000
45	1	-0.722222	-0.166667	-0.864407	-0.833333
46	1	-0.555556	0.500000	-0.796610	-0.916667
47	1	-0.833333	0.000000	-0.864407	-0.916667
48	1	-0.444444	0.416667	-0.830508	-0.916667
49	1	-0.611111	0.083333	-0.864407	-0.916667
50	2	0.500000	0.000000	0.254237	0.083333
51	2	0.166667	0.000000	0.186441	0.166667
52	2	0.444444	-0.083333	0.322034	0.166667
53	2	-0.333333	-0.750000	0.016949	0.000000
54	2	0.222222	-0.333333	0.220339	0.166667
55	2	-0.222222	-0.333333	0.186441	0.000000
56	2	0.111111	0.083333	0.254237	0.250000
57	2	-0.666667	-0.666667	-0.220339	-0.250000
58	2	0.277778	-0.250000	0.220339	0.000000
59	2	-0.500000	-0.416667	-0.016949	0.083333
60	2	-0.611111	-1.000000	-0.152542	-0.250000
61	2	-0.111111	-0.166667	0.084746	0.166667

62	2	-0.055556	-0.833333	0.016949	-0.250000
63	2	0.000000	-0.250000	0.254237	0.083333
64	2	-0.277778	-0.250000	-0.118644	0.000000
65	2	0.333333	-0.083333	0.152542	0.083333
66	2	-0.277778	-0.166667	0.186441	0.166667
67	2	-0.166667	-0.416667	0.050847	-0.250000
68	2	0.055556	-0.833333	0.186441	0.166667
69	2	-0.277778	-0.583333	-0.016949	-0.166667
70	2	-0.111111	0.000000	0.288136	0.416667
71	2	0.000000	-0.333333	0.016949	0.000000
72	2	0.111111	-0.583333	0.322034	0.166667
73	2	0.000000	-0.333333	0.254237	-0.083333
74	2	0.166667	-0.250000	0.118644	0.000000
75	2	0.277778	-0.166667	0.152542	0.083333
76	2	0.388889	-0.333333	0.288136	0.083333
77	2	0.333333	-0.166667	0.355932	0.333333
78	2	-0.055556	-0.250000	0.186441	0.166667
79	2	-0.222222	-0.500000	-0.152542	-0.250000
80	2	-0.333333	-0.666667	-0.050847	-0.166667
81	2	-0.333333	-0.666667	-0.084746	-0.250000
82	2	-0.166667	-0.416667	-0.016949	-0.083333
83	2	-0.055556	-0.416667	0.389831	0.250000
84	2	-0.388889	-0.166667	0.186441	0.166667
85	2	-0.055556	0.166667	0.186441	0.250000
86	2	0.333333	-0.083333	0.254237	0.166667
87	2	0.111111	-0.750000	0.152542	0.000000
88	2	-0.277778	-0.166667	0.050847	0.000000
89	2	-0.333333	-0.583333	0.016949	0.000000
90	2	-0.333333	-0.500000	0.152542	-0.083333
91	2	0.000000	-0.166667	0.220339	0.083333
92	2	-0.166667	-0.500000	0.016949	-0.083333
93	2	-0.611111	-0.750000	-0.220339	-0.250000
94	2	-0.277778	-0.416667	0.084746	0.000000
95	2	-0.222222	-0.166667	0.084746	-0.083333
96	2	-0.222222	-0.250000	0.084746	0.000000
97	2	0.055556	-0.250000	0.118644	0.000000
98	2	-0.555556	-0.583333	-0.322034	-0.166667
99	2	-0.222222	-0.333333	0.050847	0.000000
100	3	0.111111	0.083333	0.694915	1.000000
101	3	-0.166667	-0.416667	0.389831	0.500000
102	3	0.555556	-0.166667	0.661017	0.666667
103	3	0.111111	-0.250000	0.559322	0.416667
104	3	0.222222	-0.166667	0.627119	0.750000
105	3	0.833333	-0.166667	0.898305	0.666667
106	3	-0.666667	-0.583333	0.186441	0.333333
107	3	0.666667	-0.250000	0.796610	0.416667
108	3	0.333333	-0.583333	0.627119	0.416667
109	3	0.611111	0.333333	0.728814	1.000000
110	3	0.222222	0.000000	0.389831	0.583333
111	3	0.166667	-0.416667	0.457627	0.500000
112	3	0.388889	-0.166667	0.525424	0.666667
113	3	-0.222222	-0.583333	0.355932	0.583333
114	3	-0.166667	-0.333333	0.389831	0.916667
115	3	0.166667	0.000000	0.457627	0.833333
116	3	0.222222	-0.166667	0.525424	0.416667
117	3	0.888889	0.500000	0.932203	0.750000
118	3	0.888889	-0.500000	1.000000	0.833333
119	3	-0.055556	-0.833333	0.355932	0.166667
120	3	0.444444	0.000000	0.593220	0.833333
121	3	-0.277778	-0.333333	0.322034	0.583333
122	3	0.888889	-0.333333	0.932203	0.583333
123	3	0.111111	-0.416667	0.322034	0.416667
124	3	0.333333	0.083333	0.593220	0.666667
125	3	0.611111	0.000000	0.694915	0.416667
126	3	0.055556	-0.333333	0.288136	0.416667
127	3	0.000000	-0.166667	0.322034	0.416667
128	3	0.166667	-0.333333	0.559322	0.666667
129	3	0.611111	-0.166667	0.627119	0.250000
130	3	0.722222	-0.333333	0.728814	0.500000
131	3	1.000000	0.500000	0.830508	0.583333

132	3	0.166667	-0.333333	0.559322	0.750000
133	3	0.111111	-0.333333	0.389831	0.166667
134	3	0.000000	-0.500000	0.559322	0.093333
135	3	0.888889	-0.166667	0.728814	0.833333
136	3	0.111111	0.166667	0.559322	0.916667
137	3	0.166667	-0.083333	0.525424	0.416667
138	3	-0.055556	-0.166667	0.288136	0.416667
139	3	0.444444	-0.083333	0.491525	0.666667
140	3	0.333333	-0.083333	0.559322	0.916667
141	3	0.444444	-0.083333	0.389831	0.833333
142	3	-0.166667	-0.416667	0.389831	0.500000
143	3	0.388889	0.000000	0.661017	0.833333
144	3	0.333333	0.083333	0.593220	1.000000
145	3	0.333333	-0.166667	0.423729	0.833333
146	3	0.111111	-0.583333	0.355932	0.500000
147	3	0.222222	-0.166667	0.423729	0.583333
148	3	0.055556	0.166667	0.491525	0.833333
149	3	-0.111111	-0.166667	0.389831	0.416667

Appendix M

Complexity Reduction Techniques

We describe our implementation of three techniques for reducing the classifier's complexity:

- Weight decay.
- Weight smoothing.
- Linear non-invertible feature vector compression

The number of parameters is the implicit measure of discriminator complexity in all three techniques. The first two techniques work by reducing what Moody calls the *effective* number of parameters [97] in the discriminator; the third technique reduces the actual number of parameters. By reducing the number of parameters (both actual and effective), we reduce the classifier's discriminant variance.

Although these techniques are the only ones we use in the experiments of part II, many other useful ones exist.

M.1 Weight Decay

We employ the weight-decay formalism described by Hanson and Pratt [57], in which parameters (i.e., weights) decay to a value of zero in the absence of learning influences. This form of complexity reduction can be used with any type of parameter vector for which setting an element to zero is equivalent to removing the element from the vector.¹ Assuming a learning procedure that updates each parameter θ iteratively, the notation $\theta[\eta]$ denotes the parameter value at the η th update. By this notation, the value of the parameter at the beginning of learning is $\theta[0]$. At the η th update, $\theta[\eta]$ is equal to a fraction of its value after the previous update plus the parameter change $\Delta\theta[\eta]$ provided by the iterative learning procedure:

¹ Hanson and Pratt's formalism can be generalized to one in which the parameters decay to a potentially non-zero *null* value in the absence of learning influences. Such a generalization would make the technique applicable to radial basis function (RBF) classifiers, as an example, for which zero-valued parameters are generally not null, but have an effect on the discriminator's mappings.

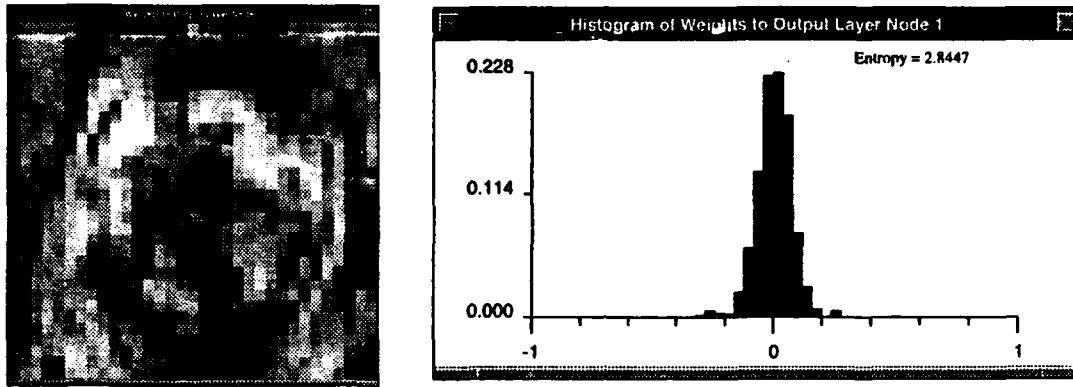


Figure M.1: Left: The parameters of a 1024-pixel differentially-generated logistic linear classifier described in chapter 9, generated by differential learning without weight smoothing or weight decay. Light parameters, or *weights* are positive; dark weights are negative; the gray tone in (all but the vertically centered pixel of) the display's left edge represents the value zero. Right: A histogram of the weights in the left figure. Note the entropy of the weights is 2.8.

$$\theta[\eta] = (1 - \zeta)\theta[\eta - 1] + \Delta\theta[\eta] \quad (\text{M.1})$$

The decay rate $\zeta \in [0, 1)$ determines how fast the parameter decays to zero. Equation (M.1) is a first-order difference equation with the forcing function $\Delta\theta[\eta]$. If we set the initial condition $\theta[-1] = 0$, then $\theta[0] = \Delta\theta[0]$, and the solution to the difference equation is given by

$$\theta[\eta] = (1 - \zeta)^\eta \theta[0] + \sum_{k=1}^{\eta} (1 - \zeta)^{\eta-k} \Delta\theta[k] \quad \forall \eta > 0 \quad (\text{M.2})$$

Thus, the parameter's dependence on its initial value decreases exponentially as learning progresses (i.e., as η increases). Likewise, the parameter's dependence on prior updates decreases exponentially. As $\zeta \rightarrow 1$, decay is very rapid; as $\zeta \rightarrow 0$, decay is very slow. After a large number of iterations (i.e., $\eta \gg 1$) the parameter's value effectively depends solely on the sequence of past learning updates $(\Delta\theta[\eta], \Delta\theta[\eta - 1], \dots)$. Specifically,

$$\theta[\eta] \approx \sum_{k=1}^{\eta} (1 - \zeta)^{\eta-k} \Delta\theta[k]; \quad \eta \gg 0, \zeta > 0, \quad (\text{M.3})$$

which means that the parameter value can be viewed as the output of a first-order auto-regressive (AR) filter operating on the learning procedure's parameter update sequence.

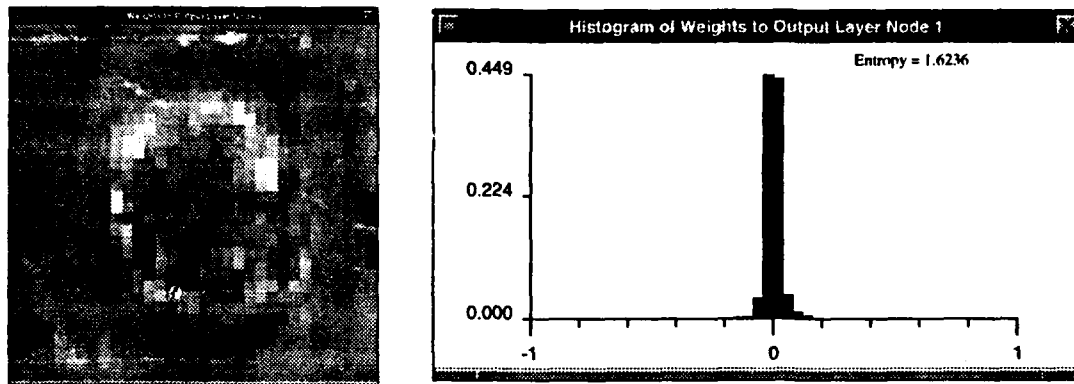


Figure M.2: **Left:** The parameters of the logistic linear classifier shown in figure M.1, generated by differential learning with weight decay. **Right:** A histogram of the weights in the left figure. Note the entropy of the weights is now 1.6, reflecting the lower variance in their distribution engendered by weight decay. This lower variance/entropy accounts for the low-contrast in the weight display on the left: many of the weights have decayed to zero.

M.1.1 Parametric Entropy

Figure M.1 (left) illustrates the weights of a linear classifier with a single output unit, described in chapter 9. The classifier is used to diagnose a joint disorder in magnetic resonance images, so its input is retinotopic (i.e., image-like) and its weights can themselves be displayed as an image. The weights are generated by differential learning without weight decay or weight smoothing. The lighter weights have positive values; the darker weights have negative values; the gray tone in (all but the vertically centered pixel of) the display's left edge represents the value zero. Figure M.1 (right) shows a histogram of the weight values.

If we view all the parameters as realizations of a single random variable (rv), their histogram can be loosely interpreted as an indicator of the rv's information content — if all the parameters have the same value, they don't contain any information about the patterns that the discriminator classifies; if they have widely varying (e.g., uniformly-distributed) values, they probably do contain information.

Definition M.1 Parametric Entropy: *The entropy of a parameter vector θ is based on a 50-bin histogram of its values, and is simply*

$$-\sum_{i=1}^{50} p_i \log_2(p_i), \quad (\text{M.4})$$

where p_i denotes the fraction of parameter vector elements with values that fall within the i th histogram bin. The histogram spans the range of the largest integer not greater than the most negative parameter to the smallest integer not less than the most positive parameter: $\text{bin}[1] = \lfloor \min_k \theta_k \rfloor$ & $\text{bin}[50] = \lceil \max_k \theta_k \rceil$.

Remark: We stress that this definition is an intuitive and rather arbitrary one, neither rigorously substantiated nor generally applicable. It is restricted to parameter vectors associated with retinotopic feature vectors (e.g., images, speech spectrograms, etc.) because it assumes all parameters are realizations of the same single random parameter variable. This assumption is plausible for image-like feature vectors because they tend to have a large number of spatially correlated elements. As a result, discriminator parameter vectors associated with the feature vector tend to have an equally large number of spatially correlated elements. Where the correlation is low between parameters, there tend to be details in the feature vector that are critical to the classification process. Of course, uncorrelated parameters tend to have higher variance, so their associated parametric entropy is higher. Thus, parametric entropy is a convenient albeit *ad-hoc* measure of the parameter vector's information content.

The parametric entropy of the weight vector in figure M.1, generated without weight decay or weight smoothing, is 2.8. The parametric entropy of the weight vector in figure M.1, generated with a weight decay rate of $\zeta = .005$, is 1.6. The lower parametric entropy is evident when one compares the two weight displays and their associated histograms: the decayed weight distribution has less variance and more kurtosis (i.e., the histogram peaks more sharply about zero) — both related to lower parametric entropy. There is visibly less structure in the decayed weights.

M.2 Weight Smoothing

We employ a simple form of weight smoothing developed by Pomerleau² in which the parameter vector is filtered after each update. This form of complexity reduction is restricted to weight vectors associated with retinotopic feature vectors because it relies on the assumption that “neighboring” parameters (i.e., those corresponding to neighboring pixels in the feature vector) can be highly correlated without increasing discriminant error.

We arrange the parameter vector in a manner that reflects the feature vector pixel map with which it is associated (the weight displays of figures M.1 and M.2 exemplify such an arrangement). We convolve a simple moving average (MA) filter with this parameter map. The filter perturbs each parameter towards the average value of it and its eight neighboring parameters in the map. Figure M.3 illustrates the filter's kernel function. The gray box at the center represents θ_0 , the parameter being processed at a given point in the filtering convolution; the surrounding boxes represent this parameter's neighbors $\{\theta_1, \dots, \theta_8\}$ in the parameter map. The kernel omits the appropriate parameters (with a commensurate modification to (M.5) below) when they do not exist (e.g., for θ_0 parameters on the edge of the map). We denote the output of the filter by θ'_0 when the kernel is centered on θ_0 :

²Personal communication.

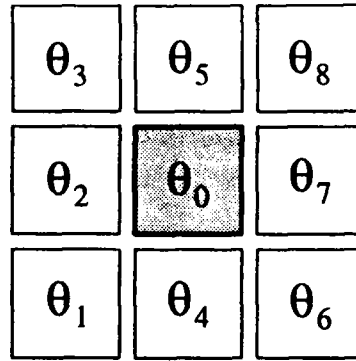


Figure M.3: The moving average filter kernel used for weight smoothing. The gray parameter θ_0 is the principal input to the filter; its neighboring parameters comprise the other terms in the moving average of (M.5).

$$\begin{aligned}
 \theta'_0 &= \theta_0 - \kappa \left(\theta_0 - \frac{1}{9} \sum_{j=0}^8 \theta_j \right) \\
 &= \left(1 - \frac{8}{9} \kappa \right) \theta_0 + \frac{\kappa}{9} \sum_{j=1}^8 \theta_j
 \end{aligned}
 \tag{M.5}$$

The parameter $\kappa \in [0,1)$ adjusts the level of smoothing. The filter is convolved with the parameter map after each update of the parameter vector.

Figure M.4 (left) illustrates weights generated by differential learning with weight smoothing ($\kappa = 0.05$). The parametric entropy of the weight vector in figure M.1, generated without weight decay or weight smoothing, is 2.8. The parametric entropy of the weight vector in figure M.4 is 2.2. The lower parametric entropy is evident when one compares the two weight displays and their associated histograms: the smoothed weight distribution has less variance and more kurtosis. There is visibly less structure in the smoothed weights, and they appear blurred as a result of the iterative filtering operation.

M.3 Linear Non-Invertible Feature Vector Compression

We employ a simple form of lossy (i.e., non-invertible) data compression as a third approach to complexity reduction. Like weight smoothing, it is restricted to weight vectors associated with retinotopic feature vectors because it relies on the assumption that “neighboring” elements (i.e., those corresponding to neighboring pixels in the feature vector map) are highly correlated and, as a result, redundant.

The compression ratio (CR) is the ratio of the number of elements in the original feature vector to the number of elements in the compressed feature vector. Figure M.5 illustrates compression when the ratio CR

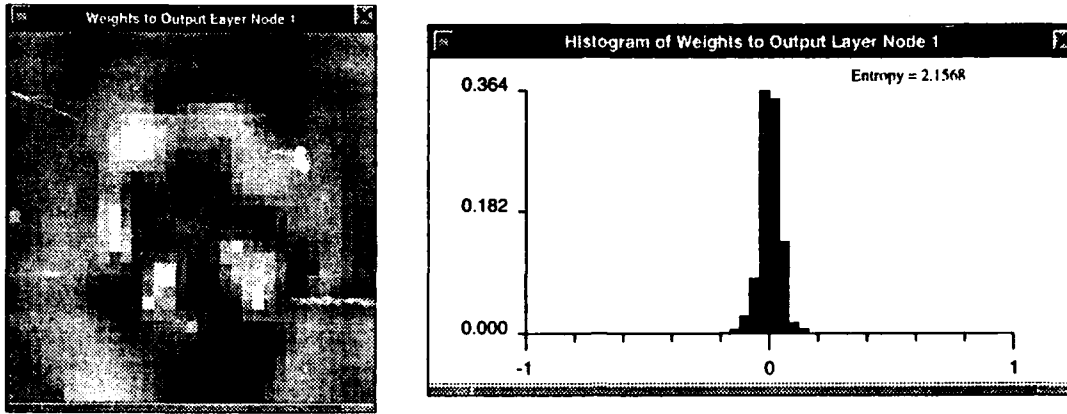


Figure M.4: **Left:** The parameters of the logistic linear classifier shown in figure M.1, generated by differential learning with weight smoothing. **Right:** A histogram of the weights in the left figure. Note the entropy of the weights is 2.2, compared with 2.8 for the weights generated without smoothing. The lower entropy reflects the lower variance in the distribution of weights caused by weight smoothing, which accounts for the lower contrast in this weight display compared to the one in figure M.2. Note the blurred appearance of the weights due to the filtering effect of weight smoothing.

is an integer (top), and when the it is not an integer (bottom). For the case in which the compression ratio is an integer, the compressed element of $\mathbf{X}$, which we denote by x' , is equal to the average value of the elements in $\mathbf{X}$ from which it is formed; we denote these elements by $\{x_1, \dots, x_{CR}\}$:

$$x' = \frac{1}{CR} \sum_{i=1}^{CR} x_i \quad (\text{M.6})$$

In figure M.5 (top) $CR = 4$, and each x' in the compressed image is the average of four pixels in the original image (i.e., the average of four elements in the original feature vector).

For the case in which the compression ratio is not an integer, x' is proportional to the value of the elements in $\mathbf{X}$ from which it is formed. Figure M.5 (bottom) illustrates compression when $CR = 2.25$. In this case, every element of the compressed feature vector is formed from four elements in the original vector. As an example, the lower right element of the compressed vector, which we will denote by x' , is given by

$$x' = \frac{1}{CR = 2.25} \left[\frac{1}{2} x_1 + \frac{1}{4} x_2 + \frac{1}{2} x_3 + x_4 \right] \quad (\text{M.7})$$

The subscripts of $\{x_1, \dots, x_4\}$ in (M.7) are shown in the left side of the figure, which depicts the original feature vector elements.

Figures 8.1 and 8.5 in chapter 8 and figures 9.1 and 9.3 in chapter 9 illustrate the effects of linear non-invertible compression for two different retinotopic feature vectors. The compression ratio in both tasks

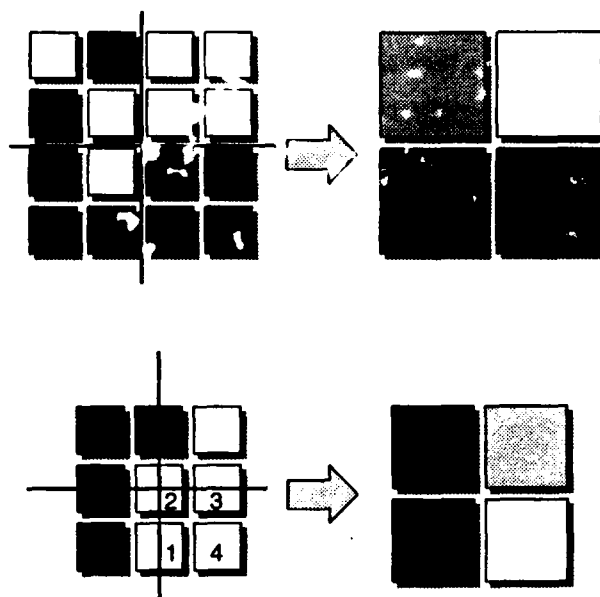


Figure M.5: **Top:** Linear non-invertible compression with a compression ratio of 4:1. The value of each pixel in the compressed image is equal to the average value of the four constituent pixels in the original image. **Bottom:** Linear non-invertible compression with a compression ratio of 2.25:1. The constituent pixels in the original image contribute to the compressed image in proportion to the fraction of their area that falls within the bounds of the compressed pixel (see equation (M.7)).

is 4:1. We characterize the compression as non-invertible because the original feature vector cannot be derived from the compressed vector.

M.3.1 A Brief Argument Against Principal Components Analysis

Readers familiar with the method of principal components analysis (PCA) might wonder why we do not employ this technique. In its details the reason is rather long-winded, so we give only a brief explanation. Principal components analysis relies on the following assumptions:

- A feature vector $\mathbf{X}$'s first and second moments are assumed to be sufficient statistics for the pattern recognition task, to the extent that the following assumption holds:
- The feature vector's covariance matrix can be expressed in terms of its eigenvectors and eigenvalues. The eigenvectors associated with the largest eigenvalues (i.e., $\mathbf{X}$'s principal components) contain the bulk of $\mathbf{X}$'s variance, and as a result, *they are assumed to contain the bulk of the information necessary for separating the class-conditional densities of $\mathbf{X}$ in feature vector space $\mathcal{X}$.*

Fukunaga has written extensively on this topic; we refer the reader to [40, ch's. 9-10]. The second assumption above is one that is frequently violated, as eloquently and succinctly described in [40, pp. 442-443]. In short, under certain circumstances the *minor* components of $\mathbf{X}$ (i.e., the eigenvectors associated with the *smallest* eigenvalues) will contain the bulk of the information necessary for separating the class-conditional densities of $\mathbf{X}$: principal components analysis would discard these very components. Under similar circumstances, the information necessary for robust discrimination of $\mathbf{X}$ is distributed across *all* its elements, so eliminating any of them would result in higher discriminant bias. As a result, we eschew all but the crudest and most general form of feature vector dimensionality reduction: the linear non-invertible compression described above. The circumstances under which it is applicable are obvious (the feature vector must be retinotopic in nature), so there is little danger of applying the technique when it is inappropriate. The only danger is using a compression ratio that is so high, information essential to robust discrimination is lost (see, for example, chapter 9).

Bibliography

- [1] Y. S. Abu-Mostafa. The Vapnik-Chervonenkis Dimension: Information versus Complexity in Learning. *Neural Computation*, 1:312--317, 1989.
- [2] H. Akaike. Information Theory and an Extension of the Likelihood Principle. In B. N. Petrov and F. Csadki, editors, *Proceedings of the Second International Symposium on Information Theory*, page , Budapest, Hungary, 1973. Akademiai Kiado.
- [3] E. Anderson. The irises of the gaspe peninsula. *Bulletin of the American Iris Society*, 59:2--5, 1935.
- [4] T. M. Apostol. *Calculus*, volume 2. Xerox College Publishing, Waltham, MA, 2nd edition, 1969.
- [5] E. Barnard. Performance and Generalization of the CFM Criterion Function. *IEEE Transactions on Neural Networks*, 2(2):322--325, March 1991.
- [6] E. Barnard and D. Casasent. A Comparison between Criterion Functions for Linear Classifiers, with an Application to Neural Nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1030--1041, September 1989.
- [7] A. R. Barron and R. L. Barron. Statistical Learning Networks: A Unifying View. In G. Wegmen, editor, *Computing Science and Statistics*, pages 192--203. American Statistical Association, 1988.
- [8] A. R. Barron and T. M. Cover. Minimum Complexity Density Estimation. *IEEE Transactions on Information Theory*, 37(4):1034--1054, July 1991.
- [9] G. Bassett, Jr. and R. Koenker. Asymptotic Theory of Least Absolute Error Regression. *Journal of the American Statistical Association*, 73(363):618--622, September 1978.
- [10] E. B. Baum and D. Haussler. What Size Net Gives Valid Generalization? *Neural Computation*, 1:151--160, 1989.
- [11] E. B. Baum and F. Wilczek. Supervised Learning of Probability Distributions by Neural Networks. In D. Anderson, editor, *Neural Information Processing Systems*, pages 52--61. American Institute of Physics, New York, NY, 1988.

- [12] R. E. Bellman. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ, 1961.
- [13] E. J. Borowski and J. M. Borwein. *The Harper Collins Dictionary of Mathematics*. Harper Collins Publishers, New York, 1991.
- [14] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory (COLT-92)*, pages 144--152, New York, NY, 1992. ACM Press.
- [15] H. Bourlard and C. Wellekens. Links Between Markov Models and Multilayer Perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1167--1178, December 1990.
- [16] D. S. Broomhead and D. Lowe. Multivariable Function Interpolation and Adaptive Networks. *Complex Systems*, 2:321--355, 1988.
- [17] S. B. Bull and A. Donner. The Efficiency of Multinomial Logistic Regression Compared with Multiple Group Discriminant Analysis. *Journal of the American Statistical Association*, 82:1118--1122, 1987.
- [18] R. L. Burden and J. D. Faires. *Numerical Analysis*. Prindle, Weber, and Schmidt, Boston, 3rd edition, 1985.
- [19] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. John Wiley & Sons, Inc., New York, NY, 1991.
- [20] H. Cramer. *Mathematical Methods of Statistics*. Princeton University Press, Princeton, NJ, 1946.
- [21] G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 2:303--314, 1989.
- [22] W. B. Davenport, Jr. and W. L. Root. *An Introduction to the Theory of Random Signals and Noise*. McGraw Hill, New York, NY, 1958. Re-printed by the IEEE press in 1987, ISBN 0-87942-235-1.
- [23] R. M. Dawes. The Robust Beauty of Improper Linear Models in Decision Making. *American Psychologist*, 34:571--582, 1979. Reprinted as chapter 28 in, *Judgement Under Uncertainty: Heuristics and Biases*, Kahneman, Slovic, and Tversky, eds., Cambridge, UK: Cambridge University Press, 1982.
- [24] R. M. Dawes. *Rational Choice in an Uncertain World*. Harcourt, Brace, Jovanovich, New York, NY, 1988.
- [25] M. H. DeGroot. *Probability and Statistics*. Addison-Wesley, Reading, MA, 2nd edition, 1986.
- [26] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.

- [27] B. Efron. The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis. *Journal of the American Statistical Association*, 70(352):892--898, 1975.
- [28] A. El-Jaroudi and J. Makhoul. A New Error Criterion for Posterior Probability Estimation with Neural Nets. In *IEEE Proceedings of the 1990 International Joint Conference on Neural Networks*, Vol. 3, pages 185--192, San Diego, June 1990.
- [29] S. E. Fahlman and C. LeBiere. The cascade correlation learning architecture. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, vol. 2, pages 524--532. Morgan Kauffman, San Mateo, CA, 1990.
- [30] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley & Sons, Inc., New York, 2nd edition, 1957.
- [31] R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7:179--188, 1936.
- [32] S. J. Ford, J. B. Hampshire II, and D. M. McKeown, Jr. Performance evaluation of multispectral analysis for surface material classification. In *Proceedings of the DARPA Image Understanding Workshop*, San Mateo, CA, April 1993. Morgan Kaufmann Publishers.
- [33] S. J. Ford and D. M. McKeown, Jr. Utilization of Multispectral Imagery for Cartographic Feature Extraction. In *Proceedings of the DARPA Image Understanding Workshop*, pages 805--820, San Mateo, CA, January 1992. Morgan Kaufmann Publishers.
- [34] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1972.
- [35] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 2nd edition, 1990.
- [36] S. Geman, E. Bienenstock, and R. Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1--58, January 1992.
- [37] H. Gish. A Probabilistic Approach to the Understanding and Training of Neural Network Classifiers. In *Proceedings of the 1990 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pages 1361--1364, April 1990.
- [38] H. Gish. A Minimum Classification Error, Maximum likelihood, Neural Network. In *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing* Vol. 2, pages 289--292, 1991.
- [39] R. M. Gray and L. D. Davisson. *Random Processes: A Mathematical Approach for Engineers*. Prentice-Hall, Englewood Cliffs, NJ, 1986.

- [40] I. Guyon, I. Poujaud, L. Personnaz, G. Dreyfus, J. Denker, and Y. LeCun. Comparing Different Neural Network Architectures for Classifying Handwritten Digits. In *IEEE Proceedings of the 1989 International Joint Conference on Neural Networks, Vol. II*, pages 127--132, 1989.
- [41] I. Guyon, V. Vapnik, B. Boser, L. Bottou, and S. Solla. Structural risk minimization for character recognition. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems, vol. 4*, pages 471--479, San Mateo, CA, 1992. Morgan Kauffman.
- [42] P. Haffner, A. Waibel, H. Sawai, and K. Shikano. Fast back-propagation methods for neural networks in speech. technical report TR-I-0058, ATR Interpreting Telephony Research Laboratories, November 1988.
- [43] J. Hajek and Z. Sidak. *Theory of Rank Tests*. Academic Press, New York, NY, 1967.
- [44] J. B. Hampshire II and B. V. K. Vijaya Kumar. Shooting Craps in Search of an Optimal Strategy for Training Connectionist Pattern Classifiers. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems, vol. 4*, pages 1125--1132, San Mateo, CA, 1992. Morgan Kauffman.
- [45] J. B. Hampshire II and B. V. K. Vijaya Kumar. Why Error Measures are Sub-Optimal for Training Neural Network Pattern Classifiers. In *IEEE Proceedings of the 1992 International Joint Conference on Neural Networks, Vol. IV*, pages 220--227, June 1992.
- [46] J. B. Hampshire II and B. V. K. Vijaya Kumar. Differentially Generated Neural Network Classifiers are Efficient. In C. A. Kamm, G. M. Kuhn, B. Yoon, R. Chellappa, and S. Y. Kung, editors, *Neural Networks for Signal Processing III: Proceedings of the 1993 IEEE Workshop*, pages 151--160, New York, September 1993. The Institute of Electrical and Electronic Engineers, Inc.
- [47] J. B. Hampshire II and B. A. Pearlmutter. Equivalence Proofs for Multi-Layer Perceptron Classifiers and the Bayesian Discriminant Function. In Touretzky, Elman, Sejnowski, and Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School*, pages 159--172, San Mateo, CA, 1991. Morgan Kaufmann. Announced and published electronically in the Ohio-State University pub/neuroprose archive, September 23, 1990: available via anonymous ftp from archive.cis.ohio-state.edu, in file pub/neuroprose/hampshire.bayes90.ps.Z.
- [48] J. B. Hampshire II and A. H. Waibel. A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Neural Networks. *IEEE Transactions on Neural Networks*, 1(2):216--228, June 1990. A revised and extended version of work published earlier in 1) Carnegie Mellon University, School of Computer Science Technical Report CMU-CS-89-118, March 31, 1989, and 2) the IEEE

- Proceedings of the 1989 International Joint Conference on Neural Networks, vol. I, pp. 235-241, June, 1989.
- [49] S. J. Hanson and L. Y. Pratt. Comparing Biases for Minimal Network Construction with Back-Propagation. In Dave Touretzky, editor, *Advances in Neural Information Processing Systems*, vol. 1, pages 177--185. Morgan Kaufmann, San Diego, CA, 1989.
- [50] D. Harel. *Algorithms: The Spirit of Computing*. Addison-Wesley, Reading, MA, 1987.
- [51] D. Haussler. Decision Theoretic Generalizations of the PAC Model for Neural Net and Other Learning Applications. Technical Report UCSC-CRL-91-02, University of California, Santa Cruz, January 1991. A revised and extended version of UCSC-CRL-89-30.
- [52] D. Haussler, M. Kearns, and R. Schapire. Bounds on the Sample Complexity of Bayesian Learning Using Information Theory and the VC Dimension. In M. K. Warmuth and L. G. Valiant, editors, *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 61--74, San Mateo, CA, August 1991. Morgan Kaufmann, Inc.
- [53] W. H. Highleyman. The Design and Analysis of Pattern Recognition Experiments. *Bell Systems Technical Journal*, 41:723--744, March 1962.
- [54] W. W. Hines and D. C. Montgomery. *Probability and Statistics in Engineering and Management Science*. John Wiley and Sons, New York, NY, 2nd edition, 1980.
- [55] G. E. Hinton. Connectionist Learning Procedures. In J. G. Carbonell, editor, *Machine Learning: Paradigms and Methods*, pages 185--234. MIT Press, Cambridge, MA, 1990. Based on the 1987 Carnegie Mellon University technical report (CMU-CS-87-115) of the same title.
- [56] N. L. Hjort. Estimating the Logistic Regression Equation when the Model is Incorrect. technical report, Norwegian Computing Center, Oslo, Norway, 1988.
- [57] J. J. Hopfield. Learning Algorithms and Probability Distributions in Feed-Forward and Feed-Back Networks. *Proceedings of the National Academy of Science, U.S.A.*, 84:8429--8433, 1987.
- [58] D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, New York, NY, 1989.
- [59] D. Kahneman, P. Slovic, and A. Tversky. *Judgement Under Uncertainty: Heuristics and Biases*. Cambridge University Press, Cambridge, England, 1982.
- [60] F. Kanaya and S. Miyake. Bayes Statistical Behavior and Valid Generalization of Pattern Classifying Neural Networks. *IEEE Transactions on Neural Networks*, 2:471--475, July 1991.

- [61] S. Karlin. Admissibility for Estimation with Quadratic Loss. *Annals of Mathematical Statistics*, 29:406--436, 1958.
- [62] M. J. Kearns and R. E. Schapire. Efficient Distribution-free Learning of Probabilistic Concepts. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 382--391, Los Alamitos, CA, October 1990. IEEE Computer Society Press.
- [63] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward Efficient Agnostic Learning. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 341--352, New York, NY, July 1992. ACM Press.
- [64] J. P. Keating, R. L. Mason, and P. K. Sen. *Pitman's Measure of Closeness: A Comparison of Statistical Estimators*. SIAM, Philadelphia, 1993.
- [65] D. E. Knuth. *The Art of Computer Programming*, volume 1. Addison-Wesley, Reading, MA, 2 edition, 1973.
- [66] T. Kohonen, G. Barna, and R. Chrisley. Statistical Pattern Recognition with Neural Networks: Benchmark Studies. In *Proceedings of the 1988 IEEE International Conference on Neural Networks*, pages I--61 -- I--68. IEEE, July 1988.
- [67] A. N. Kolmogorov. Three Approaches to the Quantitative Definition of Information. *Problems of Information Transmission*, 1(1):1--7, Jan. - Mar. 1965. Faraday Press translation of Problemy Peredachi Informatsii.
- [68] A. N. Kolmogorov and S. V. Fomin. *Introductory Real Analysis*. Dover Publications, Inc., New York, NY, 1970. Translated and edited by R. A. Silverman.
- [69] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, New York, NY, 6th edition, 1988.
- [70] S. Kullback. *Information Theory and Statistics*. Wiley, New York, NY, 1959.
- [71] S. Kullback and A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22:79--86, 1951.
- [72] P. A. Lachenbruch. *Discriminant Analysis*. Hafner Press, New York, NY, 1975.
- [73] P. A. Lachenbruch and M. R. Mickey. Estimation of Error Rates in Discriminant Analysis. *Technometrics*, 10:1--11, 1968.
- [74] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, vol. 2, pages 598--605. Morgan Kauffman, San Mateo, CA, 1990.

- [75] R. Lippmann. Pattern Classification Using Neural Networks. *IEEE Communications Magazine*, 27(11), 1989.
- [76] D. J. C. MacKay. The Evidence Framework Applied to Classification Networks. *Neural Computation*, 4(5):720--736, September 1992.
- [77] J. I. Makhoul, S. Roucos, and H. Gish. Vector quantization in speech coding. *Proceedings of the IEEE*, 73(11):1551--1588, November 1985.
- [78] A. Manduca, P. Christy, and R. Ehman. Neural Network Diagnosis of Avascular Necrosis from Magnetic Resonance Images. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems*, vol. 4, pages 645--650. San Mateo, CA, 1992. Morgan Kauffman.
- [79] G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley and Sons, New York, 1992.
- [80] P. Medgassy. *Decomposition of Superposition of Distribution Functions*. Publishing House of the Hungarian Academy of Sciences, Budapest, 1961.
- [81] J. Miller, R. Goodman, and P. Smyth. On loss functions which minimize to conditional expected values and posteriori probabilities. *IEEE Transactions on Information Theory*, 39(4), July 1993.
- [82] S. Miyake and F. Kanaya. A Neural Network Approach to a Bayesian Statistical Decision Problem. *IEEE Transactions on Neural Networks*, 2:538--540, September 1991.
- [83] J. Moody and C. Darken. Learning with Localised Receptive Fields. In Touretzky, Hinton, and Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, San Mateo, CA, 1988. Morgan Kaufmann.
- [84] J. E. Moody. Note on Generalization, Regularization, and Architecture Selection. In B. H. Juang, S. Y. Kung, and C. A. Kamm, editors, *Neural Networks for Signal Processing*, Piscataway, NJ, 1991. IEEE Press.
- [85] J. E. Moody. The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems*, vol. 4, pages 847--854, San Mateo, CA, 1992. Morgan Kauffman.
- [86] J. R. Movellan. Error Functions to Improve Noise Resistance and Generalization in Backpropagation Networks. In *IEEE Proceedings of the 1990 International Joint Conference on Neural Networks*, Vol. 1, pages 557--560, Washington, DC, January 1990.
- [87] B. K. Natarajan. *Machine Learning: A Theoretical Approach*. Morgan Kaufmann, San Mateo, CA, 1991.

- [88] J. D. Patterson and B. F. Womack. An Adaptive Pattern Classification System. *IEEE Transactions on Systems, Man, and Cybernetics*, SSC-2:62--67, August 1966.
- [89] T. Poggio and F. Girosi. A Theory of Networks for Approximation and Learning. AI Memo 1140, MIT, 1989.
- [90] D. Pollard. *Convergence of Stochastic Processes*. Springer Verlag, New York, NY, 1984.
- [91] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1986. Based on the original *Numerical Recipes* (in FORTRAN).
- [92] C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37:81--91, 1945.
- [93] C. R. Rao. Some Comments on the Minimum Mean-Square Error as a Criterion of Estimation. In M. Csörgö, D. A. Dawson, J. N. K. Rao, and A. K. Md. E. Saleh, editors, *Statistics and Related Topics*, pages 123--143. North-Holland Publishing Co., Amsterdam, 1981.
- [94] M. D. Richard and R. P. Lippmann. Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities. *Neural Computation*, 3(4):461--483, 1991.
- [95] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465--471, 1978.
- [96] J. Rissanen. A Universal Prior for Integers and Estimation By Minimum Description Length. *The Annals of Statistics*, 11(2):416--431, 1983.
- [97] J. Rissanen. Stochastic Complexity. *Journal of the Royal Statistical Society, B*, 49(3):252--265, 1987.
- [98] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, Teaneck, N.J., 1989.
- [99] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the theory of brain mechanisms*. Spartan Books, Washington, D.C., 1962.
- [100] H. L. Royden. *Real Analysis*. Macmillan, New York, 3rd edition, 1988.
- [101] D. W. Ruck, S. K. Rogers, M. Kabrinsky, M. E. Oxley, and B. W. Sutter. The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function. *IEEE Transactions on Neural Networks*, 1(4):296--298, December 1990.
- [102] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Representations by Backpropagation Errors. *Nature*, 323:533--536, October 1986.
- [103] D. E. Rumelhart, J. L. McClelland, et al. *Parallel Distributed Processing*, volume 1. MIT Press, 1987.

- [104] P. A. Shoemaker. A Note on Least-Squares Learning Procedures and Classification by Neural Network Models. *IEEE Transactions on Neural Networks*, 2(1):158--160, January 1991.
- [105] E. Singer and R. P. Lippmann. Improved Hidden Markov Model Speech Recognition Using Radial Basis Function Neural Networks. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems*, vol. 4, pages 159--166, San Mateo, CA, 1992. Morgan Kauffman.
- [106] S. A. Solla, E. Levin, and M. Fleisher. Accelerated Learning in Layered Neural Networks. *Complex Systems*, 2:625--640, 1988.
- [107] S. C. Tormay. *Ockham: Studies and Selections*. Open Court Publishers, La Salle, IL, 1938.
- [108] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977.
- [109] L. G. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11):1134--1142, November 1984.
- [110] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part 1*. John Wiley & Sons, New York, NY, 1968.
- [111] V. Vapnik. Principles of risk minimization for learning theory. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems*, vol. 4, pages 831--838, San Mateo, CA, 1992. Morgan Kauffman.
- [112] V. N. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, NY, 1982. The publisher lists the title in the following alternate form: "Estimation of Dependencies Based on Empirical Data Dependencies", ISBN 0-387-90733-5.
- [113] V. N. Vapnik and A. YA. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to their Probabilities. *Theory of Probability and its Applications*, XVI(2):264--280, 1971.
- [114] E. A. Wan. Neural Network Classification: A Bayesian Interpretation. *IEEE Transactions on Neural Networks*, 1(4):303--305, December 1990.
- [115] S. M. Weiss and C. A. Kulikowski. *Computer Systems that Learn*. Morgan Kaufmann, San Mateo, CA, 1991.
- [116] H. White. Maximum Likelihood Estimation of Misspecified Models. *Econometrica*, 50:1--25, 1982.
- [117] H. White. *Estimation, Inference, and Specification Analysis*. Cambridge University Press, Cambridge, UK, 1989.

- [118] H. White. Learning in Artificial Neural Networks: A Statistical Perspective. *Neural Computation*, 1(4):425--464, Winter 1989.
- [119] H. White. Connectionist Nonparametric Regression: Multilayer Feedforward Networks Can Learn Arbitrary Mappings. *Neural Networks*, 3(5):535--549, 1990.
- [120] K. Yamanishi. A learning criterion for stochastic rules. *Machine Learning*, 9:165--203, 1992.

Index

- a posteriori* class differential 21
- a posteriori* class probability 15
- Anderson, E. 187, 407
 - Iris 187, 407
- ARE, see *relative efficiency, asymptotic*
- asymptotic relative efficiency, see *relative efficiency, asymptotic*
- attribute vector, see *feature vector*
- AT&T DB1 database 219
 - benchmark training/test samples 220, 233
- avascular necrosis (AVN) of the femoral head, diagnosing 271
- backpropagation 92, 102, 287, 375
 - with CFM 348
- Bayes error rate 55
 - estimated 256
- Bayes-optimal classifier 17
- Bayesian discriminant function (BDF) 18
 - differential form 21
 - probabilistic form 19
 - strictly differential form 20
 - strictly probabilistic form 19
- BDF, see *Bayesian discriminant function*
- Bernoulli trials 252
- bias/variance tradeoff 75, 306
- binomial distribution, DeMoivre — Laplace approximation 252
- boundary value of feature vector 21
- box plot 224, 321
- cardinality 123
- CFM 29, 327
 - confidence parameter ψ 50, 148
 - controlling ... 191
 - convergence properties of differential learning via ... 334
 - convergence properties of differential learning via the original logistic sigmoidal form 336
 - convergence properties of differential learning via the synthetic form 340
 - functional form 47
 - monotonic nature of 142
 - original forms 155
 - synthetic form 327
 - ANSI C source code 351
 - confidence parameter bounds 346
 - learning rate via ... 153
 - low-order derivatives 330
 - specifications 328
- class boundaries, Bayes-optimal 3, 305
- class label space, see *classification, space*
- class prior probability 15
- class-conditional probability density function (pdf) 15
- classification 16
 - rejecting 202, 246
 - differential 202, 246
 - probabilistic 206
 - space 14
- classification figure-of-merit, see *CFM*
- classifier 16
 - asymptotically unbiased 59
 - Bayes-optimal 17

- classifier (cont'd)
 - consistent 59, 69
 - differentiable supervised 25
 - efficient 4, 60
 - error rate 55
 - minimum-complexity 72
 - polynomial 100
 - relatively efficient 4, 64
- complexity
 - general measure 74
 - minimizing 215
 - minimum necessary for Bayesian discrimination 72
 - reduction of 413
- compression, feature vector 274, 417
 - effect on generalization 225
 - ratio 417
- confidence 148
- convergence, training sample 319
- correct space 119
- Cramer, H. 62
- Cramer-Rao bound 62, 77
- Cross Entropy objective function, see *Kullback-Leibler information distance*
- cross validation 222
- Dawes, R. M. 81, 83
- decision tree 27
- density, see *probability density function*
- differential learning, see *learning, differential*
- differential, discriminant, see *discriminant differential*
- differential, a posteriori class ... 21
- discriminant bias 58
 - estimated 223
- discriminant boundary 117
 - reduced 117
- discriminant continuum 115
 - reduced 116
- discriminant differential 22, 29
- discriminant error 58, 71
- discriminant function, Bayesian, see *Bayesian discriminant function*
- discriminant variance 59
 - estimated 223
- discrimination 16
- discriminator 16
- discriminator output space 16
 - correct fraction of 123
 - correct side of 118
 - incorrect fraction of 124
 - incorrect side of 118
 - monotonic correct fraction of 124
 - monotonic fraction of 126
 - monotonic incorrect fraction of 125
 - non-monotonic correct fraction of 124
 - non-monotonic incorrect fraction of 125
 - reduced 116
 - correct side of 118
 - incorrect side of 118
- DRBF 187, 287, 406
- efficiency, relative, see *relative efficiency*
- efficient classifier, see *classifier, efficient*
- efficient learning, see *learning, efficient*
- Efron, B. 95, 368
- error
 - discriminant, see *discriminant error functional* 71
- error measure 28
 - general 32
 - general, inefficiency of learning via ... 72, 140
 - non-monotonic nature of 140
 - strictly probabilistic 36
- error rate 55, 92, 371
 - average estimated 223
 - Bayes 55
 - estimated 256
 - estimated 221
 - of improper parametric model 372
 - of polynomial classifier 372
 - of proper parametric model 371
 - test sample
 - average 223
 - empirical 221

- estimated relative efficiency, *see relative efficiency, estimated* 255
- example
 - easy 149, 191
 - hard 149, 191
 - learned 148, 334, 350
 - training
 - types 148
 - unlearned, focussing on ... 193
 - transition 148, 335
 - un-learned 148, 334, 349
- feature vector 14
 - boundary value of 21
 - heteroscedastic uniformly-distributed 99
 - homoscedastic Gaussian-distributed 85, 261
- feature vector space 14
- Fisher, R. A. 62
 - Iris 187, 407
- Fisher information 95
- functional error 71
- Glivenko-Cantelli Theorem 319
- glossary of notation 311
- heteroscedastic uniformly-distributed feature vector 99
- Highleyman, W. H. 222
- homoscedastic Gaussian-distributed feature vector 85, 250, 261
- human recognition
 - AVN diagnosis 284
 - noisy OCR images 258, 263
- hypothesis
 - alternative 277
 - null 277
- hypothesis class 56
 - DRBF 187, 287, 406
 - linear 186, 239
 - logistic linear 186, 232
 - modified Gaussian RBF 187, 242, 405
 - types 186
- incorrect space 118
- information, Fisher, *see Fisher information*
- Iris 187, 407
- k nearest neighbors 27
- Kolmogorov's Theorem 75
- Kullback-Leibler information distance 37, 187, 367
 - non-monotonic nature of 136
- learning 55
 - agnostic 70
 - asymptotically efficient strategy 65
 - differential 4, 26, 43
 - asymptotic efficiency proof 66
 - controlling CFM confidence parameter ψ 191
 - convergence properties 334
 - convergence properties via the original logistic sigmoidal form of CFM 336
 - convergence properties via the synthetic form of CFM 340
 - focussing on un-learned examples 193
 - generates consistent classifier 69
 - learning rate 150, 155, 334
 - relationship to learning via perceptron criterion function 359
 - discriminative 3
 - efficient 4, 65
 - general strategy 56
 - probabilistic 3, 26, 31
 - case for ... 76
 - reasonably fast 336
 - unreasonably slow 335
- least absolute deviation, *see mean absolute error*
- least absolute error, *see mean absolute error*
- log-likelihood, equation for fully-parametric proper model 364
- logistic discriminant analysis 286, 367
- logistic regression 26, 365, 79

- logit risk function 366
- Mahalanobis distance 97
- Manduca, Christy, and Ehman 271
- maximum-likelihood, learning for fully-parametric proper model 364
- mean absolute error (MAE) 42
 - non-monotonic nature of 127
- mean-squared discriminant error (MSDE) 59
 - estimated 223
- mean-squared error (MSE) 39, 102
 - inefficiency of learning via ... 71
 - minimum-MSE parameters of polynomial classifier 375
 - non-monotonic nature of 132
- medical diagnosis 271
- minimum description length (MDL) 160
- Minkowski- r power metric 41
- monotonic fraction 126, 383
 - of Kullback-Leibler information distance 136, 387
 - of mean absolute error 127, 383
 - of mean-squared error 132, 385
- monotonicity 113, 122, 383
- multi-layer perceptron (MLP) 27
- noise 248
- non-parametric model 3
- normal-based linear discriminant analysis 79, 286, 365
- numerical optimization, *see search*
- objective function 28, 113
 - monotonic 113, 122
 - non-monotonic 122
- optical character recognition (OCR) 219
- optimization, *see search*
- parametric entropy 227, 415
- parametric model 3
 - improper 3, 63, 99
 - proper 3, 63, 77, 85
 - fully parametric 63, 87
 - fully parametric variant for homoscedastic Gaussian feature vector 364
 - partially parametric 63, 88, 26188
 - partially-parametric variant for homoscedastic Gaussian feature vector 365
- Parzen windows 27
- pattern 33
- pattern recognition 16
- perceptron criterion function 359
- polynomial classifier, minimum-MSE parameters of 375
- principal components analysis 419
- probabilistic learning, *see learning, probabilistic*
- probability density function (pdf), class-conditional 15
- proper (parametric) model 81
- prototype 33
- radial basis function (RBF) 27
 - modified Gaussian RBF 405
- Rao, C. R. 62
- RBF, *see radial basis function*
- RE, *see relative efficiency, estimated*
- receiver operator characteristic (ROC) 277
- reduced discriminant boundary 117
- reduced discriminant continuum 116
- reduced discriminator output space 116
 - correct side of 118
 - incorrect side of 118
- reject region 202
 - minimum discriminant differential (δ_{reject}) defining ... 206
- reject threshold δ_{reject} 206
- rejecting classifications 202, 246
- relative efficiency
 - asymptotic (ARE) 79, 97
 - of logistic discriminant analysis versus normal-based linear discriminant analysis 368
- relative efficiency

- estimated (RE) 255
- relatively efficient classifier, see *classifier, relatively efficient*
- relatively efficient learning, see *learning, relatively efficient*
- remote sensing 285
- representation 186, 207, 307
- risk, weighting 307
- Rissanen, J. 160
- ROC, see *receiver operator characteristic*
- Rosenblatt, F. 359
- search 2, 13, 25, 31, 55, 67, 92, 102, 327, 55
- sensitivity 277
- signal-to-noise ratio (SNR) 248
- specificity 277
- stochastic complexity 160
- training sample 56
- Tukey, J. W. 321
- Vapnik, V. 319
- Vapnik-Chervonenkis (VC) dimension 74
- vector quantization 27
- weight decay 413
 - rate 414
- weight smoothing 227, 274, 416
 - parameter κ 417
- whisker plot 224, 321