

Nonparametric Bayesian Context Learning for Buried Threat Detection

by

Christopher Ralph Ratto

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

Leslie M. Collins, Supervisor

Loren W. Nolte

Jeffrey L. Krolik

Qing H. Liu

David L. Banks

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Electrical and Computer Engineering
in the Graduate School of Duke University

2012

Report Documentation Page		Form Approved OMB No. 0704-0188
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.		
1. REPORT DATE 2012	2. REPORT TYPE	3. DATES COVERED 00-00-2012 to 00-00-2012
4. TITLE AND SUBTITLE Nonparametric Bayesian Context Learning for Buried Threat Detection		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER	
	5e. TASK NUMBER	
	5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Duke University, Department of Electrical and Computer Engineering, Durham, NC, 27708		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

This dissertation addresses the problem of detecting buried explosive threats (i.e. landmines and improvised explosive devices) with ground-penetrating radar (GPR) and hyperspectral imaging (HSI) across widely-varying environmental conditions. Automated detection of buried objects with GPR and HSI is particularly difficult due to the sensitivity of sensor phenomenology to variations in local environmental conditions. Past approaches have attempted to mitigate the effects of ambient factors by designing statistical detection and classification algorithms to be invariant to such conditions. These methods have generally taken the approach of extracting features that exploit the physics of a particular sensor to provide a low-dimensional representation of the raw data for characterizing targets from non-targets. A statistical classification rule is then usually applied to the features. However, it may be difficult for feature extraction techniques to adapt to the highly nonlinear effects of near-surface environmental conditions on sensor phenomenology, as well as to retrain the classifier for use under new conditions. Furthermore, the search for an invariant set of features ignores that possibility that one approach may yield best performance under one set of terrain conditions (e.g., dry"), and another might be better for another set of conditions (e.g., wet"). An alternative approach to improving detection performance is to consider exploiting differences in sensor behavior across environments rather than mitigating them, and treat changes in the background data as a possible source of supplemental information for the task of classifying targets and non-targets. This approach is referred to as context-dependent learning. Although past researchers have proposed context-based approaches to detection and decision fusion, the definition of context used in this work differs from those used in the past. In this work, context is motivated by the physical state of the world from which an observation is made, and not from properties of the observation itself. The proposed context-dependent learning technique therefore utilized additional features that characterize soil properties from the sensor background, and a variety of nonparametric models were proposed for clustering these features into individual contexts. The number of contexts was assumed to be unknown a priori and was learned via Bayesian inference using Dirichlet process priors. The learned contextual information was then exploited by an ensemble of classifiers trained for classifying targets in each of the learned contexts. For GPR applications the classifiers were trained for performing algorithm fusion. For HSI applications the classifiers were trained for performing band selection. The detection

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT

unclassified

b. ABSTRACT

unclassified

c. THIS PAGE

unclassified17. LIMITATION OF
ABSTRACT**Same as
Report (SAR)**18. NUMBER
OF PAGES**331**19a. NAME OF
RESPONSIBLE PERSON

ABSTRACT

Nonparametric Bayesian Context Learning for Buried Threat Detection

by

Christopher Ralph Ratto

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

Leslie M. Collins, Supervisor

Loren W. Nolte

Jeffrey L. Krolik

Qing H. Liu

David L. Banks

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in the Department of Electrical and Computer
Engineering
in the Graduate School of Duke University
2012

Abstract

This dissertation addresses the problem of detecting buried explosive threats (i.e., landmines and improvised explosive devices) with ground-penetrating radar (GPR) and hyperspectral imaging (HSI) across widely-varying environmental conditions. Automated detection of buried objects with GPR and HSI is particularly difficult due to the sensitivity of sensor phenomenology to variations in local environmental conditions. Past approaches have attempted to mitigate the effects of ambient factors by designing statistical detection and classification algorithms to be invariant to such conditions. These methods have generally taken the approach of extracting features that exploit the physics of a particular sensor to provide a low-dimensional representation of the raw data for characterizing targets from non-targets. A statistical classification rule is then usually applied to the features. However, it may be difficult for feature extraction techniques to adapt to the highly nonlinear effects of near-surface environmental conditions on sensor phenomenology, as well as to re-train the classifier for use under new conditions. Furthermore, the search for an invariant set of features ignores that possibility that one approach may yield best performance under one set of terrain conditions (e.g., “dry”), and another might be better for another set of conditions (e.g., “wet”).

An alternative approach to improving detection performance is to consider *exploiting* differences in sensor behavior across environments rather than mitigating them, and treat changes in the background data as a possible source of supplemen-

tal information for the task of classifying targets and non-targets. This approach is referred to as *context-dependent learning*.

Although past researchers have proposed context-based approaches to detection and decision fusion, the definition of context used in this work differs from those used in the past. In this work, context is motivated by the physical state of the world from which an observation is made, and not from properties of the observation itself. The proposed context-dependent learning technique therefore utilized additional features that characterize soil properties from the sensor background, and a variety of nonparametric models were proposed for clustering these features into individual *contexts*. The number of contexts was assumed to be unknown *a priori*, and was learned via Bayesian inference using Dirichlet process priors.

The learned contextual information was then exploited by an ensemble of classifiers trained for classifying targets in each of the learned contexts. For GPR applications, the classifiers were trained for performing *algorithm fusion*. For HSI applications, the classifiers were trained for performing *band selection*. The detection performance of all proposed methods were evaluated on data from U.S. government test sites. Performance was compared to several algorithms from the recent literature, several which have been deployed in fielded systems. Experimental results illustrate the potential for context-dependent learning to improve detection performance of GPR and HSI across varying environments.

Contents

Abstract	iv
List of Tables	xv
List of Figures	xvi
List of Abbreviations and Symbols	xxiii
Acknowledgements	xxvii
1 Introduction	1
1.1 Landmine and IED Detection	1
1.2 Ground-Penetrating Radar	3
1.2.1 Background	3
1.2.2 Environmental Effects on GPR Sensing	8
1.2.3 Buried Threat Detection with GPR in Changing Environmental Conditions	12
1.3 Context-Dependent Learning	16
1.4 Novel Contributions	20
2 Extracting Contextual Information from GPR Data	24
2.1 Transmission Line Model for GPR	25
2.2 GPR Contextual Features	26
2.2.1 Feature consolidation	34
2.3 Evaluating GPR Contextual Features: Simulated Data Experiment	36

2.3.1	Simulated Data Set	36
2.3.2	Feature Extraction	40
2.3.3	Correlation Analysis	41
2.3.4	Classification Results	41
2.3.5	Regression Results	43
2.4	Evaluating GPR Context Features: Field Data Experiment	45
2.4.1	Field-collected data set	45
2.4.2	Feature Extraction	46
2.4.3	Correlation Analysis	48
2.4.4	Regression Results	48
2.5	Discussion	50
3	Basic Context Learning Techniques	51
3.1	Supervised Context Learning	52
3.2	Unsupervised Context Learning	53
3.3	Within-Context Target Classification	54
3.4	GPR Data for Evaluating Landmine/IED Detection Performance . .	57
3.5	Experimental Results	58
3.5.1	Supervised Context Learning	58
3.5.2	Unsupervised Context Learning	59
3.5.3	Context-Dependent Fusion Results	60
3.5.4	Detection Performance	63
3.6	Discussion	64
4	Generative Nonparametric Context Learning	67
4.1	Bayesian Inference and Variational Learning	68
4.1.1	Point Estimation of Model Parameters	68

4.1.2	Bayesian Inference	69
4.1.3	Variational Bayesian Inference	72
4.2	Dirichlet Process	75
4.3	Dirichlet Process Gaussian Mixture Model	78
4.4	Dirichlet Process Mixture of Factor Analyzers	81
4.5	Experimental Results	88
4.5.1	Context Learning with the DPGMM	88
4.5.2	Context Learning with the DPMFA	90
4.5.3	Context-Dependent Fusion Results	94
4.5.4	Detection Performance	96
4.6	Conclusions	99
5	Discriminative Nonparametric Context Learning	100
5.1	Generative vs. Discriminative Learning	101
5.2	Mixture-of-Experts Models	101
5.3	Discriminative Context Models	104
5.4	Synthetic Data Examples	108
5.5	Experimental Results with GPR Data	124
5.5.1	Context Identification Performance	125
5.5.2	Context-Dependent Fusion Results	129
5.5.3	Detection Performance	129
5.6	Conclusion	132
6	Nonparametric Spatial Context Models	135
6.1	Spatial Context Sampling	136
6.2	DPGMM Spatial Context Model	139
6.3	SBHMM Spatial Context Model	140

6.4	Experimental Results	148
6.4.1	Context Modeling Performance	149
6.4.2	Context-Dependent Fusion Results	156
6.4.3	Detection Performance	158
6.5	Conclusion	162
7	Applications to Hyperspectral Sensing	164
7.1	Hyperspectral Imagery	165
7.1.1	Background	165
7.1.2	Environmental Effects on HSI Sensing	166
7.1.3	Buried Threat Detection with HSI in Changing Conditions . .	168
7.2	HSI Data Set	169
7.3	Feature Extraction from HSI Data	170
7.3.1	Context Learning Based on Background Spectra	171
7.3.2	Context Learning Based on Spectral Unmixing	172
7.3.3	Linear Spectral Mixing Model	173
7.3.4	Endmember Extraction	174
7.4	Experimental Results	177
7.4.1	Context Learning Results	178
7.4.2	Context-Dependent Band Selection Results	182
7.4.3	Detection Performance	186
7.5	Conclusions	188
8	Conclusions and Future Work	191
8.1	Summary of Contributions	191
8.2	Considerations for Fielded Systems	198
8.3	Future Work	199

8.4	Broader Applications	202
A	Probability Distributions	205
A.1	Bernoulli Distribution	205
A.1.1	Parameters	205
A.1.2	Probability Density Function	206
A.1.3	Moments	206
A.1.4	Kullback-Leibler Divergence	206
A.2	Binomial Distribution	206
A.2.1	Parameters	206
A.2.2	Probability Density Function	207
A.2.3	Moments	207
A.2.4	Kullback-Leibler Divergence	207
A.3	Multinomial Distribution	207
A.3.1	Parameters	207
A.3.2	Probability Density Function	208
A.3.3	Moments	208
A.3.4	Kullback-Leibler Divergence	208
A.4	Beta Distribution	208
A.4.1	Parameters	209
A.4.2	Probability Density Function	209
A.4.3	Moments	209
A.4.4	Kullback-Leibler Divergence	209
A.5	Dirichlet Distribution	210
A.5.1	Parameters	210
A.5.2	Probability Density Function	210

A.5.3	Moments	210
A.5.4	Kullback-Leibler Divergence	211
A.6	Gamma Distribution	211
A.6.1	Parameters	211
A.6.2	Probability Density Function	211
A.6.3	Moments	212
A.6.4	Kullback-Leibler Divergence	212
A.7	Normal (Gaussian) Distribution	212
A.7.1	Parameters	212
A.7.2	Probability Density Function	213
A.7.3	Moments	213
A.7.4	Kullback-Leibler Divergence	213
A.8	Multivariate Normal (Gaussian) Distribution	213
A.8.1	Parameters	213
A.8.2	Probability Density Function	214
A.8.3	Moments	214
A.8.4	Kullback-Leibler Divergence	214
A.9	Wishart Distribution	214
A.9.1	Parameters	215
A.9.2	Probability Density Function	215
A.9.3	Moments	215
A.9.4	Kullback-Leibler Divergence	216
A.10	Normal-Wishart Distribution	216
A.10.1	Parameters	216
A.10.2	Probability Density Function	217

A.10.3 Moments	217
A.10.4 Kullback-Leibler Divergence	217
B Relevance Vector Machines	219
B.1 RVM Regression	220
B.1.1 Generative Model and Variable Definitions	220
B.1.2 Priors	220
B.1.3 Variational Posterior on $\mathbf{w}$	220
B.1.4 Variational Posterior on α	222
B.1.5 Variational Posterior on τ	223
B.1.6 Negative Free Energy	224
B.2 RVM Classification	225
B.2.1 Generative Model and Variable Definitions	225
B.2.2 Priors	225
B.2.3 Approximate Likelihood	225
B.2.4 Variational Posterior on $\mathbf{w}$	226
B.2.5 Updating ξ	227
B.2.6 Variational Posterior on α	228
B.2.7 Negative Free Energy	229
B.3 Mixture of RVM Classifiers	230
B.3.1 Generative Model and Variable Definitions	230
B.3.2 Priors	231
B.3.3 Variational Posterior on $\mathbf{w}$	231
B.3.4 Updating ξ	233
B.3.5 Variational Posterior on α	233
B.3.6 Treatment of $\mathbf{c}$	235

B.3.7	Negative Free Energy	235
C	Dirichlet Process Gaussian Mixture Models	237
C.1	Generative Model and Variable Definitions	237
C.2	Priors	238
C.3	Model likelihood	238
C.4	Variational Posterior on $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$	238
C.5	Variational Posterior on $\mathbf{v}$	242
C.6	Variational Posterior on α	243
C.7	Variational Posterior on $\mathbf{C}$	244
C.8	Negative Free Energy	245
D	Dirichlet Process Mixture of Factor Analyzers	247
D.1	Model and Variable Definitions	247
D.2	Priors	248
D.3	Model Likelihood	249
D.4	Variational Posterior on $\mathbf{A}$	252
D.5	Variational posterior on $\mathbf{S}$	254
D.6	Variational Posterior on $\mathbf{z}$	256
D.7	Variational Posterior on $\boldsymbol{\mu}$	258
D.8	Variational Posterior on $\boldsymbol{\psi}$	260
D.9	Variational Posterior on $\boldsymbol{\pi}$	263
D.10	Variational Posterior on $\boldsymbol{\gamma}$	264
D.11	Variational Posterior on $\mathbf{C}$	265
D.12	Variational Posterior on $\mathbf{v}$	265
D.13	Variational Posterior on α	266
D.14	Variational Posterior on δ	268

D.15 Negative Free Energy	268
E Discriminative DPGMM-RVM	271
E.1 Generative Model and Variable Definitions	271
E.2 Priors	272
E.3 Model Likelihood	272
E.4 Variational Posterior on μ and Λ	273
E.5 Variational Posterior on w	277
E.6 Variational Posterior on ξ	279
E.7 Variational Posterior on β	280
E.8 Variational Posterior on V	281
E.9 Variational Posterior on α	282
E.10 Variational Posterior on C	283
E.11 Negative Free Energy	284
Bibliography	286
Biography	301

List of Tables

1.1	Performance of Landmine Detection Algorithms as Compared by Wilson et al., [41].	16
2.1	Features ($\mathbf{x}^{(C)}$) for classification and regression of environmental parameters.	35
2.2	Feature extraction parameters for simulated data experiment	41
2.3	Feature extraction parameters for field data experiment	47
3.1	Alarm Distribution by Soil Type and Ground Truth	57
6.1	Alarm Distribution by Soil Type and Ground Truth (Smaller Data Set)	149
7.1	AMI of HSI Context Models Trained on PCA of Background Spectra	180
7.2	AMI of HSI Context Models Trained on Endmember Abundances . .	182

List of Figures

1.1	The NIITEK Husky Mounted Detection System (HMDS), which consists of 4 GPR antenna array panels (each with 12 channels) mounted in front of a Husky route clearance vehicle [14].	5
1.2	An example of a GPR A-scan collected over an anti-tank landmine buried under a paved road.	5
1.3	An example of a GPR B-scan collected over an anti-tank landmine buried under a paved road.	6
1.4	Example GPR B-scans illustrating the signatures of different anti-tank landmine types.	7
1.5	GPR B-scans of a low-metal, anti-tank landmine buried at 3 inches under different moisture conditions.	10
1.6	GPR B-scans of a low-metal, anti-tank landmine buried in four different types of road construction: dirt (left), gravel (center-left), asphalt (center-right), and concrete (right).	12
1.7	Flowchart illustrating a basic context-dependent classification technique.	21
2.1	A diagram of transmission line model for GPR A-scans [39].	26
2.2	Examples of energy and reflection coefficient features for FDTD-simulated B-scans.	30
2.3	Example of MP histogram extracted from B-scans over soils.	32
2.4	Example of LP prediction-error power extracted from aligned B-scans over simulated soils.	35
2.5	Examples of computational domain and FDTD-simulated GPR data.	38
2.6	Plot of correlations between features (horizontal axis) and soil labels (line color) for the simulated GPR experiment.	42

2.7	Confusion matrices illustrating results of RVM classification of $\epsilon_r^{(soil)}$ (top-left), $\sigma^{(soil)}$ (top-right), $l^{(soil)}/\lambda_c$ (bottom-left), and $N^{(scats)}$ (bottom-right) for the simulated data experiment.	43
2.8	Results of RVM regression for predicting $\epsilon_r^{(soil)}$ (top-left), $\sigma^{(soil)}$ (top-right), $l^{(soil)}$ (bottom-left), and $N^{(scats)}$ (bottom-right) for the simulated data experiment.	44
2.9	Photograph of the meteorological station located at the Eastern US test site.	46
2.10	Example B-scans of field-collected GPR data collected on dirt (top) and gravel (bottom) lanes at an Eastern US test site.	47
2.11	Plot of correlations between features (horizontal axis) and measured soil properties (line color) for the simulated GPR experiment.	49
2.12	Results of RVM regression to predict dirt temperature (left), gravel temperature (center), and soil moisture (right) from contextual features extracted from field data.	49
3.1	Scatter plots and confusion matrix illustrating performance of basic supervised context learning.	58
3.2	Scatter plot and similarity matrix illustrating performance of basic unsupervised context learning.	61
3.3	RVM discriminant weights learned for algorithm fusion in each supervised context.	62
3.4	RVM discriminant weights learned for algorithm fusion in both 3 and 8 unsupervised contexts.	63
3.5	ROC curves for basic context-dependent fusion techniques, compared to non-context-dependent RVM fusion and the individual fused algorithms.	65
4.1	Example of the DPGMM learned on mixture of 9 Gaussian distributions	80
4.2	An example factor analysis problem to illustrate DPMFA model performance.	86
4.3	<i>A posteriori</i> expected values of the DPMFA model parameters learned from the example data shown in Figure 4.2.	87
4.4	Results of denoising the example data shown in Figure 4.2 with DPMFA.	88

4.5	Scatterplot comparing results of context learning using the DPGMM on the GPR contextual features to the known soil labels.	89
4.6	Similarity matrix comparing DPGMM clustering results to the known soil labels.	90
4.7	Means of clusters learned by the DPGMM context model.	91
4.8	Covariance matrices of clusters learned by the DPGMM context model.	92
4.9	Similarity matrix comparing DPMFA clustering results to the known soil labels.	93
4.10	<i>A posteriori</i> expected values of the DPMFA model parameters learned from the GPR contextual features.	94
4.11	Means of clusters learned by the DPMFA context model.	95
4.12	Covariance matrices of clusters learned by the DPMFA context model.	96
4.13	RVM discriminant weights learned for algorithm fusion in each DPGMM context.	97
4.14	RVM discriminant weights learned for algorithm fusion in each DPMFA context.	97
4.15	ROC curves for context-dependent fusion, using either the DPGMM or DPMFA context models, compared to non-context-dependent RVM fusion and the individual fused algorithms.	98
5.1	Scatterplot of target and context features for the first synthetic data example to illustrate discriminative context-dependent learning. . . .	109
5.2	Results of context identification using the discriminative DPGMM-RVM model for the first synthetic data example.	110
5.3	Results of context identification using the IQGME model for the first synthetic data example.	111
5.4	Component classifiers learned by the discriminative DPGMM-RVM model for the first synthetic data example.	112
5.5	Component classifiers learned by the IQGME model for the first synthetic data example.	113
5.6	Discriminant weights learned by the DPGMM-RVM, IQGME, and the context oracle for the first synthetic data example.	114

5.7	ROC curves comparing discriminative context-dependent learning on the first synthetic data example.	115
5.8	Scatterplot of target and context features for the second synthetic data example.	115
5.9	Results of context identification using the discriminative DPGMM-RVM model for the second synthetic data example.	116
5.10	Results of context identification using the IQGME model for the second synthetic data example.	117
5.11	Component classifiers learned by the discriminative DPGMM-RVM model for the second synthetic data example.	118
5.12	Component classifiers learned by the IQGME model for the second synthetic data example.	119
5.13	Discriminant weights learned by the DPGMM-RVM, IQGME, and the context oracle for the second synthetic data example.	120
5.14	ROC curves comparing discriminative context-dependent learning on the second synthetic data example.	121
5.15	Results of context identification using the discriminative DPGMM-RVM model for the third synthetic data example.	122
5.16	Results of context identification using the IQGME model for the third synthetic data example.	122
5.17	Discriminant weights learned by the DPGMM-RVM, IQGME, and the context oracle for the third synthetic data example.	123
5.18	ROC curves comparing discriminative context-dependent learning on the third synthetic data example.	124
5.19	Scatterplot comparing results of context learning using the discriminative DPGMM-RVM on the GPR contextual features to the known soil labels.	126
5.20	Similarity matrix comparing DPGMM-RVM clustering results to the known soil labels.	126
5.21	Scatterplot comparing results of context learning using IQGME on the GPR contextual features to the known soil labels.	127

5.22	Similarity matrix comparing IQGME clustering results to the known soil labels.	127
5.23	Similarity matrix comparing IQGME context identification to DPGMM-RVM context identification.	128
5.24	Discriminant weights learned by the DPGMM-RVM and IQGME for algorithm fusion on the GPR data set.	130
5.25	ROC curves for discriminative context-dependent fusion (IQGME and DPGMM-RVM) compared to generative context-dependent fusion, non-context-dependent RVM fusion, and the individual fused algorithms.	131
6.1	Example of GPR data collected on a concrete lane and apparent spatial context regions.	137
6.2	GPR data from Figure 6.1, zoomed in to illustrate background sampling near a contextual shift.	139
6.3	True parameters for the SBHMM synthetic data example.	145
6.4	Illustration of state pruning from converged SBHMM.	146
6.5	Learned parameters for the SBHMM synthetic data example.	147
6.6	Learned context means for the spatial DPGMM and SBHMM context models on GPR data.	150
6.7	Covariance matrices of clusters learned by the spatial DPGMM context model.	150
6.8	Covariance matrices of clusters learned by the SBHMM context model.	151
6.9	Initial state probabilities learned by the SBHMM context model.	151
6.10	State transition probabilities learned by the SBHMM context model.	152
6.11	Example GPR data from the dirt lane and associated state posteriors from SBHMM and DPGMM context models.	153
6.12	Example GPR data from the gravel lane and associated state posteriors from SBHMM and DPGMM context models.	154
6.13	Example GPR data from the asphalt lane and associated state posteriors from SBHMM and DPGMM context models.	155

6.14	Example GPR data from the concrete lane and associated state posteriors from SBHMM and DPGMM context models.	157
6.15	RVM discriminant weights learned for algorithm fusion in each spatial DPGMM context.	159
6.16	RVM discriminant weights learned for algorithm fusion in each SBHMM context.	159
6.17	ROC curves for context-dependent fusion, using SBHMM and DPGMM spatial context models, compared to alarm-based context-dependent fusion, global RVM fusion, and the individual fused algorithms. . . .	161
7.1	Example HSI image chips corresponding to antitank landmines recorded by the RX detector over a minefield located at an arid Western US test site.	166
7.2	Example HSI image chips corresponding to false alarms recorded by the RX detector over a minefield located at an arid Western US test site.	167
7.3	Example target and false alarm spectra from HSI collected in morning (solid lines), afternoon (dotted lines), and night (dashed lines). Target spectra are provided in the top panel, and false alarm spectra in the bottom panel.	168
7.4	Illustration of the context and target feature extraction regions for a typical HSI image chip.	170
7.5	Scatterplot of the 3-D PCA projection of the averaged background pixels of each image chip, colored by time of day.	172
7.6	Results of endmember extraction using ICE on 3-dimensional toy data.	176
7.7	Results of endmember extraction using ICE on a synthetic mixture of endmember spectra from the USGS spectral library.	177
7.8	Endmember spectra extracted from background regions of AHI image chips using ICE with $\mu = 0.001$	178
7.9	Scatterplots illustrating results of supervised (top-left), generative DPGMM (top-right), and discriminative (bottom) context learning from the PCA-projected background spectra.	179
7.10	Scatterplots illustrating results of supervised (top-left), generative DPGMM (top-right), and discriminative (bottom) context learning from the endmember abundances learned by ICE.	181

7.11	RVM discriminant weights corresponding to supervised (top), DPGMM (center), and discriminative (bottom) contexts learned from background spectra.	184
7.12	RVM discriminant weights corresponding to supervised (top), DPGMM (center), and discriminative (bottom) contexts learned from endmember abundances.	185
7.13	ROC curves for context-dependent classification of HSI data using background context features.	187
7.14	ROC curves for context-dependent classification of HSI data using endmember context features.	188

List of Abbreviations and Symbols

Mathematical Notation

x	Scalar quantity
$\mathbf{x}$	Vector quantity
$\mathbf{X}$	Matrix quantity
$\mathbf{x}^T$ or $\mathbf{X}^T$	Vector/matrix transpose
$\log(\cdot)$	Natural logarithm
$\exp(\cdot)$	Exponential, i.e. $\exp(x) = e^x$.
$\text{Tr}(\mathbf{X})$	Trace of $\mathbf{X}$
$p(x)$	Probability density of x (if x is a continuous random variable), or probability mass of x (if x is a discrete random variable)
$\mathbb{E}(x)$	Expected value of x
$\langle x \rangle$	Variational expectation of x
$\text{Var}(x)$	Variance of x

Acronyms

AHI	Airborne Hyperspectral Imager
ALS	Amyotrophic Lateral Sclerosis
AMI	Adjusted Mutual Information
AP	Anti-personnel
AR	Autoregressive
AT	Anti-tank

AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
BCI	Brain-Computer Interface
CBRNE	Chemical, Biological, Radiological, Nuclear, and Explosive
CELF	Context Extraction for Local Fusion
DP	Dirichlet Process
DPGMM	Dirichlet Process Gaussian Mixture Model
DPMFA	Dirichlet Process Mixture of Factor Analyzers
EEG	Electroencephalogram
EHD	Edge Histogram Descriptor
EM	Expectation-Maximization
EMI	Electromagnetic Induction
FAR	False Alarm Rate
FDTD	Finite-Difference Time Domain
GEOM	Geometric Features
GMM	Gaussian Mixture Model
GPR	Ground-Penetrating Radar
GPS	Global Positioning System
HMDS	Husky Mounted Detection System
HSI	Hyperspectral Imagery
ICE	Iterative Constrained Endmembers
IR	Infrared
KLD	Kullback-Leibler Divergence
KNN	k -Nearest Neighbors
HMM	Hidden Markov Model
IED	Improvised Explosive Device
IQGME	Infinite Quadratically-Gated Mixture of Experts

LIBS	Laser-Induced Breakdown Spectroscopy
LP	Linear Prediction
MAP	Maximum <i>a posteriori</i>
MCMC	Markov Chain Monte Carlo
MFA	Mixture of Factor Analyzers
ML	Maximum Likelihood
MP	Matching Pursuits
NIITEK	Non-Intrusive Inspection Technology, Inc.
NFE	Negative Free Energy
PCA	Principal Components Analysis
PD	Probability of Detection
PDF	Probability Density Function
QGME	Quadratically-Gated Mixture of Experts
RMS	Root Mean Square
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RSF	Random Set Framework
RVM	Relevance Vector Machine
RX	Reed-Xu
SB	Stick-Breaking
SBHMM	Stick-Breaking Hidden Markov Model
SCF	Spectral Correlation Features
SPICE	Sparseness-Promoting Iterative Constrained Endmembers
SPSCF	Subspace-Projected Spectral Correlation Features
SVM	Support-Vector Machine
US	United States

USDA	United States Department of Agriculture
USGS	United States Geological Survey
VB	Variational Bayes
WAAMD	Wide Area Airborne Minefield Detection

Acknowledgements

It has truly been a blessing to have worked with and learned from so many great people over the past five years. I would first like to thank my advisor, Leslie Collins, for her guidance and support throughout my graduate studies. Leslie, thank you for allowing me to run with an idea that turned out to be a successful one, but as long as I was willing to live up to the many failures along the way. I may have taken for granted all the work you've done to acquire funding for me to attend graduate school and write this thesis - but after seeing how much I've accomplished, I realize how none of my success would have been possible without your efforts on that end. Thank you, Leslie, for a great five years.

I also want to thank two very talented individuals that were crucial to my success, and without their guidance (and at times, their pessimism) I would not have learned nearly as much in graduate school as I did. Pete Torrione was on the way out of grad school when I arrived at Duke, and I probably bothered him a lot while he was trying to finish up his dissertation. But from day one, Pete always had his office door open and quickly became the unofficial steward of my research. Out of everything that he has done for me, I am most grateful for his patience all these years. Kenny Morton was still a junior grad student when I arrived. My first interactions with him mostly involved Duke basketball and all things that come with it (beer, cornhole, and sleeping in a tent), but he eventually became my go-to guy on all things Bayes. Thank you Kenny, for always being there to explain the inexplicable in a way I can

understand.

I must also thank my remaining SSPACISS lab colleagues for their support and camaraderie. These include Stacy Tantum and Sandy Throckmorton, as well as my fellow students: Sara Duran, Ken Colwell, Jill Desmond, Achut Manandhar, Rayn Sakaguchi, Jordan Malof, Patrick Wang, and Boyla Mainsah. Stacy and Sandy, thanks for never being afraid to call shenanigans on my lab meeting presentations, even when I thought I had the rest of the lab convinced. Sara, I'm looking forward to your successful Ph.D. defense, as well as the day you finally learn to love America. Ken, Jill, Achut, Rayn, Jordan, Patrick, and Boyla - it's been a pleasure working alongside you, and I'm sure we'll have the "fun lab" designation locked up for years to come. I also want to thank three brave souls who entered the PhD program with me in 2007 - Kyle Bradbury, Jason Yu, and Phil Brown. If I didn't have you guys around to commiserate with, I don't think I would have made it past year one.

I am grateful for the financial support of the US Army Research Laboratory and the Night Vision and Electronic Sensors Directorate, which were possible through the efforts of Dick Weaver, Pete Howard, and Russ Harmon. I am also glad to have had the opportunity to collaborate with several outstanding researchers from across the country. Special thanks go to Paul Gader and Joe Wilson from the University of Florida, Dominic Ho from the University of Missouri, Hichem Frigui from the University of Louisville, and the staff of NIITEK, Inc. Without their efforts in sharing code, features, and data, exploring context-dependent fusion would have been impossible.

I also must acknowledge the endless support from my family through all these years. Mom and Dad, thanks for encouraging me to leave the nest, always work hard, and not take any crap from anybody. Danny and Cathy, thanks for always being around when I'm home, always making time to visit, and making sure I don't get old too fast. Grandma and Grammie, I always look forward to your phone calls, and

love every minute I get to spend with you when I'm home. And Grandpa, thanks for letting me steal your VIC-20 when I was a kid so I could learn to program in BASIC. Bet you didn't think it would eventually lead to this!

Finally, my ultimate thanks go to my better half. It was through Allie's "encouragement" that I decided to go to graduate school in the first place. Between moving to Durham together, getting married, moving again, looking for jobs from coast to coast, and writing our respective dissertations, this experience has brought us closer together than anything else could have. I love you because you revel in my successes and don't sweat it when I fall short, and I admire how much you are personally vested in your work. Part of me wishes that my Ph.D. could be awarded to you, since in comparison you've done enough work for two of them, but nah - I think I'll keep it.

Introduction

1.1 Landmine and IED Detection

Detection and remediation of buried explosives is a serious problem faced by military and civilian personnel around the world. Historically, this threat has taken the form of anti-tank (AT) and anti-personnel (AP) landmines, which are typically emplaced *en masse* over a wide area as a strategic barrier to prevent enemy advances. The use of landmines in armed conflict often results in a severe humanitarian problem once fighting has ended, as the majority of casualties of landmine detonations in post-conflict regions tend to be civilians. According to the International Campaign to Ban Landmines, civilians made up approximately 70% of the 3,531 worldwide casualties due to landmines and unexploded ordnance in 2009, and children made up almost a third of all casualties for whom the age was known [1].

Over the past decade, a new threat has emerged with the proliferation of improvised explosive devices (IEDs), which the United States Department of Defense reports as the leading cause of casualties to American soldiers in Iraq and Afghanistan [2]. Unlike landmines, IEDs by definition are not systematically manufactured and

vary widely in the explosive compounds, containers, and detonation mechanisms used in their construction. Often, the main charge of an IED is composed of a fertilizer such as ammonium nitrate and a solid fuel such as aluminum or sugar, and containers tend to be common items such as plastic jugs, buckets, or metal cooking pots [3]. A recent study has found that although the total number of worldwide casualties from victim-activated explosives (including landmines, IEDs, and unexploded ordnance) has decreased from 5,426 in 2007 to 3,956 in 2009, victim-activated IED casualties have increased in absolute terms (80 in 2008 to 549 in 2009) and percentage of all attacks (3% in 2008 to 18% in 2009) [4]. Over half of these casualties have occurred in Afghanistan (accounting for 20% of total casualties in that country), with other countries reporting anti-personnel IED casualties including Cambodia, the Democratic Republic of the Congo, India, Iraq, Nepal, Pakistan, Peru, Colombia, Burma, and Turkey.

In landmine and IED detection, as in many other detection problems, the ultimate goal is to robustly and accurately identify objects of interest with as few false alarms as possible. This trade-off can be expressed in terms of probability of detection (PD) and either probability of false alarm (PF) or false alarm rate (FAR), with the later usually measured in units of false alarms per square meter (FA/m²). The obvious risks faced by humanitarian deminers or military route clearance patrols make landmine and IED remediation very costly and time-consuming. It has been estimated that while it may only cost a few dollars to manufacture and emplace a single landmine, the cost of safely removing and neutralizing it can run from several hundred to one thousand dollars [5]. Therefore, the trade-off between detection and false alarm rate can also be seen as a trade-off between safety and cost. Humanitarian deminers may require a PD of 1 at the lowest FAR possible [6], while military route clearance patrols may stress the importance of maintaining a constant rate of advance through a potentially-threatening area and may be content with a PD as low as

0.90 [7].

Currently, a major focus of military research is improving the detection robustness of counter-mine/IED platforms used in Afghanistan. Afghanistan is notorious for its difficult terrain, with the South and West characterized by the Registan Desert and Sistan Basin (one of the driest places on Earth), and the North and East include the Hindu Kush and Pamir mountains [9]. Within the desert and mountainous regions, the geology is highly variable, even within single provinces [8]. The climate of Afghanistan varies regionally, with the Southwest portion of the country being considerably drier than the Northeast, where mountain snowfall contributes to wetter conditions at lower elevations.

The impact of varying terrain and weather conditions on the performance of counter-mine/IED sensors is tremendous. This dissertation primarily focuses on algorithms for detecting buried threats with ground-penetrating radar (GPR). Although GPR has long been used in a variety of applications, its effectiveness in landmine detection has been highlighted in much of the research literature over the past decade. However, the unique signal processing challenges presented by varying environmental factors must be considered. The following section introduces the phenomenology of GPR, its sensitivity to various environmental factors, and past approaches to improve detection performance.

1.2 Ground-Penetrating Radar

1.2.1 Background

GPR operates by transmitting an electromagnetic signal (e.g., a differentiated Gaussian pulse) into the ground and measuring the reflections of the signal at subsurface dielectric interfaces in either the temporal or frequency domain. The versatility of GPR is best illustrated by its wide range of applications, which include geophysics, forensics, utilities, and archeology [10]. Over the past two decades, GPR

has emerged as a complementary alternative to electromagnetic induction (EMI) sensors (i.e., “metal detectors”) as the next generation of landmine detection systems [11–13]. Metal detectors have historically performed very poorly in detecting nonmetal targets because they rely on inducing currents in buried conductors and sensing the resulting magnetic field. Therefore, it may be difficult to detect targets such as plastic, ceramic, or wood landmines or IEDs using an EMI sensor. GPR can potentially be used to detect any type of buried object, as long as the its dielectric properties contrast with the surrounding soil to reflect the transmitted signal.

GPRs used in buried threat detection tend to be wide-band systems with a frequency range and spatio-temporal sampling rates much higher than those used in most geophysical applications. For example, the GPR used in the Husky Mounted Detection System (HMDS) manufactured by NIITEK, Inc. (shown in Figure 1.1) transmits a differentiated Gaussian GPR signal with a bandwidth of 200 MHz - 7 GHz, and time-gates the received reflections at 6.6 ns (which corresponds to 1 m ranging in air) [14]. The received time-domain signal is referred to as an *A-scan*. An example of a GPR A-scan collected over an anti-tank landmine is shown in Figure 1.2. The first received pulse is the reflection from the ground surface, referred to as *ground-bounce*, and is typically of high magnitude. After the ground bounce, the reflection from the target is received and generally is of lesser magnitude and may be embedded in clutter corresponding to reflections between subsurface layers.

In vehicular GPR systems such as the HMDS, A-scans may be collected at multiple spatial locations to form a two-dimensional “image” of the subsurface that is referred to as a *B-scan*. A B-scan may illustrate the signals received from each channel across the array (the *crosstrack* direction), or at locations corresponding to the direction of vehicle motion (the *downtrack* direction). An example of a GPR B-scan collected over the same anti-tank landmine is shown in Figure 1.3. The B-scan allows for visual interpretation of the relative locations of the ground, subsurface layer, and



FIGURE 1.1: The NIITEK Husky Mounted Detection System (HMDS), which consists of 4 GPR antenna array panels (each with 12 channels) mounted in front of a Husky route clearance vehicle [14].

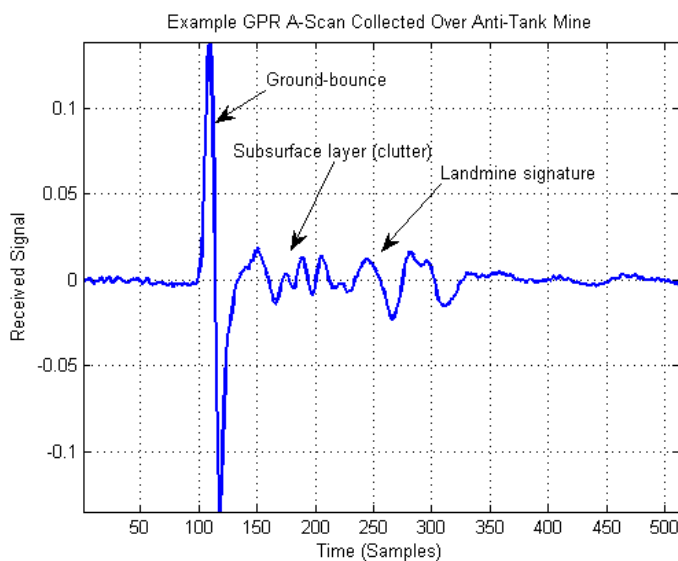


FIGURE 1.2: An example of a GPR A-scan collected over an anti-tank landmine buried under a paved road. The horizontal axis represents time (in samples) and the vertical axis represents the amplitude of the received signal. Received pulses corresponding to the ground-bounce, subsurface layering, and the target itself are marked.

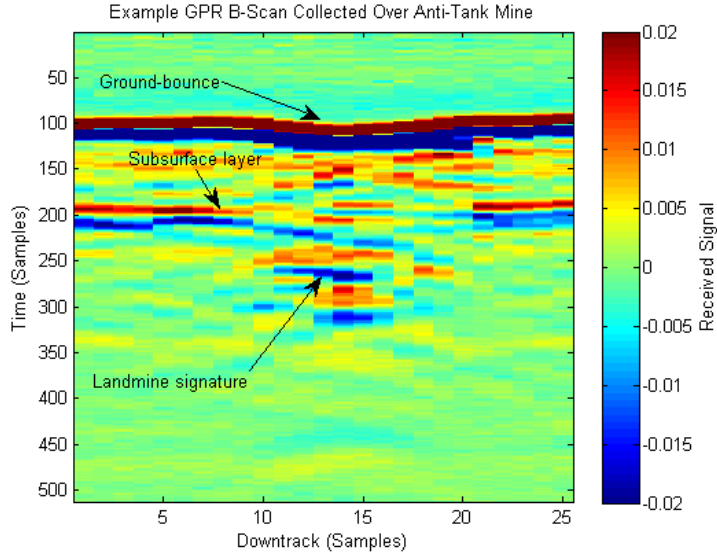


FIGURE 1.3: An example of a GPR B-scan collected over an anti-tank landmine buried under a paved road. The horizontal axis represents downtrack position (in samples), the vertical axis represents time (in samples), and the amplitude of the received signal corresponds to pixel color. Received pulses corresponding to the ground-bounce, subsurface layering, and the target itself are marked.

target over a given area. Note that the landmine signature has a distinctive hyperbolic shape as the sensor approaches and passes over the target. This distinctive property of GPR phenomenology is exploited by many statistical pattern recognition algorithms which will be discussed later.

The frequency range, lack of significant self-signature artifacts, and high spatial and temporal sampling rates of the NIITEK GPR has made it an attractive choice for high-resolution subsurface imaging. The great amount of detail in a target's GPR signature can potentially allow for inference of its geometry, composition, and inner structure [15]. Figure 1.4 illustrates the GPR signatures of four different anti-tank landmines, two high-metal and two low-metal types, buried at the same depth in a dirt road. The signatures of the metallic targets are higher in energy, since the metal casings reflect the incident GPR pulse almost perfectly. When several A-

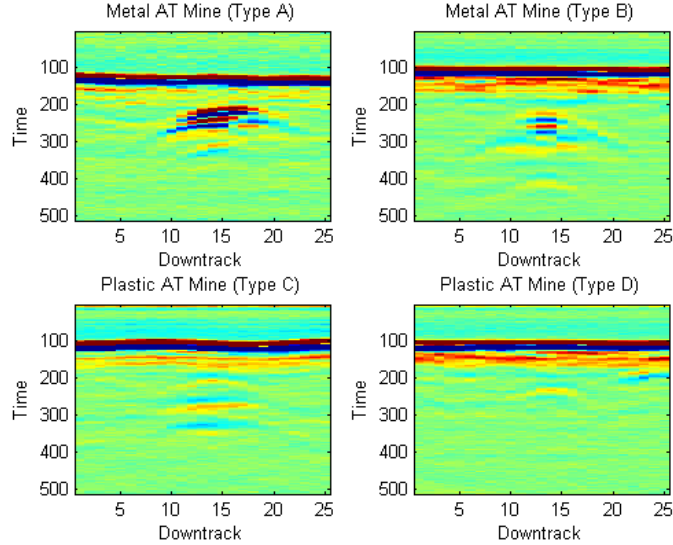


FIGURE 1.4: Example GPR B-scans illustrating the signatures of different anti-tank landmine types. The top two B-scans illustrate signatures of landmines with high metal content, and the bottom two B-scans illustrate signatures of landmines with low metal content.

scans are collected over the target, the resulting B-scan illustrates a single hyperbolic target signature. While the plastic targets' signatures are lower in energy, they are characterized by multiple reflections that occur within the landmine itself. Therefore, the signatures of plastic targets are made up of multiple hyperbolas decreasing in energy with time.

GPR signatures are rich in information about shape, size, and composition of a buried target. Therefore, GPR data has shown to be applicable for statistical pattern recognition algorithms to differentiate between responses from targets and non-threatening clutter, including natural and artificial debris, rocks, roots, and empty holes. However, a significant challenge is encountered when classifying GPR signatures collected across widely-varying environmental conditions, such as different soil types or moisture levels. The effects of these environmental factors on GPR have been studied extensively, and the body of research in this area is summarized in the

following subsection.

1.2.2 *Environmental Effects on GPR Sensing*

The signals generated and sensed by GPR are very sensitive to fluctuations in environmental conditions because unlike metal detectors, GPR signals interact with virtually everything present in the local environment. A large body of research has investigated the effects of various environmental factors on the performance of GPR in landmine detection applications. In particular, researchers have focused on the effects of soil dielectric properties (i.e. electrical permittivity and conductivity), heterogeneity, and surface texture.

Permittivity is an property of soil that partially governs the speed at which electromagnetic waves propagate through it. It is a factor of various physical properties of the soil, including grain size and composition as well as moisture content [10]. Often, a material’s permittivity is expressed in terms of its value relative to that of free space ($\epsilon_0 = 8.85 \times 10^{-12}$ F/m) through its *relative permittivity* or *dielectric constant*, ϵ_r . A seminal paper by Topp et al. focused on the effect of increased moisture on the dielectric constant of soils, and illustrated that a polynomial relationship exists between dielectric constant and volumetric soil water content [16]. Later investigations by Miller et al. also illustrated that the effect of soil moisture on conductivity is also nonlinear, exhibiting a logarithmic relationship in which increasing moisture generally increases conductivity to a saturation level [17, 18].

Permittivity and conductivity affect GPR signals in many ways. The greatest effect is due to dielectric contrast between the target and surrounding soil. If the contrast between the two materials’ dielectric properties is large, waves will reflect off of the target with greater magnitude than if their dielectric properties were similar. Borchers et al. illustrated that in many cases, increasing soil moisture also increases this dielectric contrast to yield target signatures with higher magnitude [19]. Fur-

thermore, soils with higher dielectric constants will force GPR pulses to propagate more slowly through them. Miller et al. demonstrated this effect, in which the GPR response of a target appeared later in time in soils with high dielectric constant, and can easily be confused with the response of a deeper target buried in a soil with low dielectric constant [17, 18]. Electrical conductivity governs the rate at which propagating electromagnetic waves are attenuated due to heat dissipation. Increased conductivity will dissipate propagating waves faster than soils with low conductivity, and will greatly diminish the amplitudes of GPR responses. Takahashi et al. suggested that the effects of increased conductivity on the fidelity of target signatures are only noticeable for high values, measured on the order of 0.1 S/m [20].

Figure 1.5 illustrates the effect of increasing soil moisture on the GPR signature of another low-metal, anti-tank landmine. Each of the three B-scans corresponds to a different moisture scenario; the left plot corresponds to dry conditions (more than 5 days since the last rainfall), the center plot corresponds to moderate conditions (3-5 days since the last rainfall), and the right plot corresponds to wet conditions (less than 3 days since the last rainfall). Note how the target’s hyperbolic signature both decreases in energy and appears later in time as moisture increases. This is due to combined effects of moisture on soil permittivity and conductivity. Increased moisture decreases dielectric contrast between the target and surrounding soil, while also increasing attenuation. As a result, and forces the target’s GPR response decreases in magnitude. Increased moisture also decreases the propagation speed, causing the response to appear later in time.

Subsurface heterogeneity is another major factor impacting the performance of GPR sensors. Soils are naturally heterogeneous, composed of a mixture of organic and non-organic matter, and reflections of GPR pulses from heterogeneities can yield significant amounts of clutter in GPR signals. Types of natural heterogeneity include buried rocks, roots, animal burrows, as well as stratifications in soil moisture,

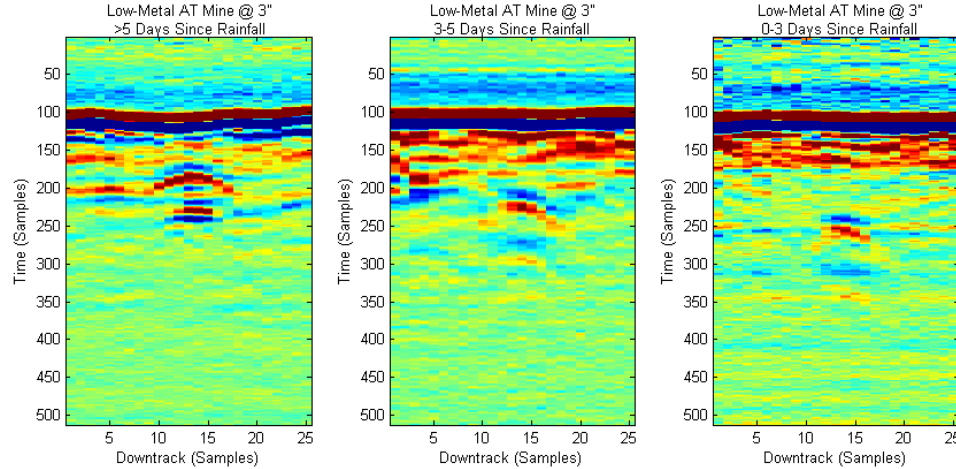


FIGURE 1.5: GPR B-scans of a low-metal, anti-tank landmine buried at 3 inches under different moisture conditions. Left: dry conditions, i.e. greater than 5 days since the last rainfall; Center: moderate conditions, i.e. between 3-5 days since the last rainfall; Right: wet conditions, i.e. less than 3 days since the last rainfall. [21]

density, or composition. The effects of heterogeneity on the performance of GPR in subsurface target detection have generally been studied in experiments controlled by electromagnetic simulations. In a study by Gürel and Oğuz [22], heterogeneities were approximated by random subsurface scatterers and were varied in quantity, size, and shape. These experiments demonstrate that in very heterogeneous soils, scattering from the individual heterogeneities can severely mitigate the GPR signature of the primary target via destructive interference. In these scenarios, visual target detection becomes increasingly difficult and automated techniques yield high false alarm rates.

In landmine detection applications which concern primary and secondary roads, it is also important to consider the effects of road construction. The presence of bumps, potholes, or obstructions in a road can cause the GPR array to bounce vertically, and depending on the displacement of the antenna significant propagation losses can be induced along with distortion of the hyperbolic shape that characterized a target signature, as presented by Milner [23]. Furthermore, elements of the road surface such as gravel, asphalt, and concrete layers of can also yield significant

clutter in a manner similar to soil inhomogeneities. These effects become even more pronounced when the surfaces are rough. A variety of simulated GPR experiments have been performed to determine the feasibility of subsurface target detection in the presence of rough surface. [24–27] and subsurface [28] interfaces. In these studies, rough surfaces were generally simulated by a stochastic process with a Gaussian spectrum, parameterized by its variance and correlation length. It has generally been found that variations of both parameters impact the GPR responses of targets, with variance dominating the overall effect on arrival time and correlation length impacting distortion of the signature’s hyperbolic shape. Inclusion of rough subsurface layers (e.g., the asphalt/concrete or concrete/soil interfaces) in the detection scenario further compounds these effects.

Figure 1.6 illustrates B-scans containing the GPR signature of the same low-metal, anti-tank landmine buried at the same depth in four different types of road construction: dirt, gravel, asphalt, and concrete. It can be seen in the dirt and gravel B-scans that the target’s signature is surrounded by responses from other subsurface heterogeneities. These could be rocks or local differences in soil density or moisture. The ground bounce also illustrates the effects rough surface scattering, with several “blobs” of high-energy reflections occurring immediately after the primary ground reflection. The asphalt lane exhibits an intermediate layer, which is characterized by a reflection at its top and bottom interfaces that appears to be of similar magnitude to the target response. The asphalt surface is also smoother than the dirt and gravel, as illustrated by the ground bounce. Finally, concrete appears to be the most homogeneous type of lane. The target signature stands out, and is not surrounded by any secondary signatures from subsurface clutter. The ground bounce is like that for the asphalt lane, since the surface is paved and therefore smoother than dirt or gravel. The concrete layer also appears either to have little dielectric contrast with the soil below it, or has caused the GPR pulse to propagate so slowly that it

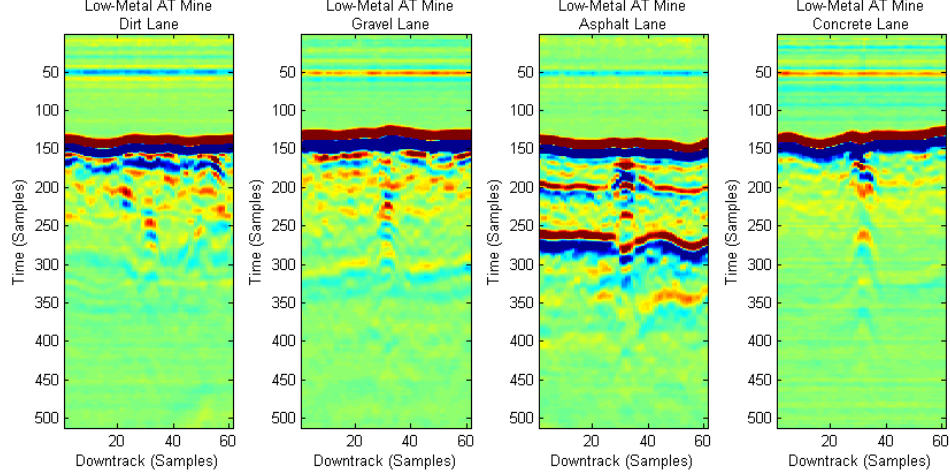


FIGURE 1.6: GPR B-scans of a low-metal, anti-tank landmine buried in four different types of road construction: dirt (left), gravel (center-left), asphalt (center-right), and concrete (right).

did not reach the soil layer in the allotted time, since a distinct reflection from the concrete/soil interface is not visible at the same scale as other reflections.

1.2.3 Buried Threat Detection with GPR in Changing Environmental Conditions

Due to the tremendous impact that varying environmental conditions have on GPR signatures of buried targets, much research has focused on the task of robust automated detection and discrimination. These approaches mostly fall under two general categories. The first group of techniques that will be discussed includes techniques based upon electromagnetic theory, which utilize model inversion strategies to decouple the interactions of GPR signals with the target from environmental artifacts. In contrast, the second category consists of statistical methods, which are based upon adaptive signal processing, pattern recognition, and machine learning theory.

Inversion Approaches to Target Detection with GPR

The first major category of approaches to target detection with GPR involve inverse solutions to Maxwell's equations via rigorous scattering models. The aim of these

approaches is to explicitly model the environment’s response to GPR and decouple it from the response of the target. After recovering the basic GPR signature of the target, visual confirmation or a simple detector can be used to determine whether a target is present.

Several electromagnetic model inversion techniques have been proposed for mitigating the effects of antenna reverberation [29–31], rough surface scattering [27], and lossy/moist soils [32–34]. Prototype GPR signatures, either collected in a laboratory or in a controlled field campaign are usually employed as a target model. When data is collected in the field, a deconvolution technique is applied to the GPR signals for isolating the target signature from the environmental artifacts. Inversion techniques have been shown to quantitatively estimate various environmental parameters (e.g., soil permittivity and conductivity) in addition to several aspects of the target’s geometry, including its location and burial depth.

However, applying closed-form model inversions to GPR data pose several implementation difficulties that must be considered. The greatest shortfall of inverse modeling lies in the time needed to compute these solutions; subsurface threat detection is already an arduous and time-consuming task, and improvements in technology should not impose any additional time expense onto deminers. Furthermore, vehicular route clearance platforms are required to operate at a constant rate of advance, and therefore all on-board algorithms must operate in real-time [7]. Finally, closed-form models are difficult to obtain for GPR responses from non-canonical or oddly-shaped targets. Even if numerically-simulated or laboratory-measured prototype signals can be obtained, it will be difficult to keep up with the threat of IEDs that are constantly evolving with changes in countermeasures, available material, training of bomb-makers, and the sophistication of production facilities. Alternatively, statistical techniques may be a more robust approach to accounting for these aspects of potential targets, as well as the ever-changing subsurface environment, in

subsurface threat detection algorithms.

Statistical Approaches to Target Detection with GPR

The category of statistical techniques for target detection in GPR can be further divided into two sub-categories: prescreeners and classifiers. Prescreeners are computationally inexpensive anomaly detectors that must detect a wide variety of potential threats and adapt to changing background statistics. Although template matching techniques based on correlation filters can perform well in detecting specific target types in a static environment, as demonstrated by Brunzell [35], they may fail when faced with a diverse target population and multiple environments. Instead, adaptive filtering approaches have shown promise as prescreeters that model the GPR background and detect anomalies that statistically differ from the background. Examples include linear prediction as proposed by Ho et al. [36] and Yoldemir and Sezgin [37], least-mean-square (LMS) prediction proposed by Torrione et al. [38,39], and particle filters proposed by Ng et al. [40]. The goal of prescreening is to detect *all* of the anomalies present in the data, whether they are associated with true landmine signatures or not. The leading prescreeners do succeed at this, but also mistake many clutter anomalies for potential targets. Therefore, prescreeners generally perform at a high PD, but at the expense of a moderate FAR.

A larger body of research has been focused on the development of feature-based classifiers based on statistical pattern recognition and machine learning theory. After the prescreeener finds locations in the raw data where an anomaly is present (referred to as *alarms*), features are extracted to provide a low-dimensional representation of the GPR data collected at that location. Features are generally physics-based and/or morphological, and aim to be invariant with respect to the environment. The classifier then applies a statistical decision rule to the feature space, and classifies the anomalies as targets or clutter. The approach of prescreening followed by feature-

based classification has shown to be effective in maintaining high PD while reducing PF/FAR to levels appropriate for fielded systems [39, 41–47].

Feature extraction approaches are generally motivated by the underlying phenomenology of a particular sensor, so as to exploit the physical characteristics of target responses. In GPR, feature extraction is used to characterize the hyperbolic shape and reverberation properties of target responses. A wide variety of feature extraction approaches have been proposed in the recent literature, including edge-based [42–44, 48], spectral [49, 50], geometric [45, 46, 51], and texture [47] features. The decision rules are learned from the features using statistical models. These include hidden Markov models [42, 48], self-organizing maps and fuzzy k -nearest neighbors (KNN) [43, 44], relevance vector machines [47], and neural networks [45, 46]. GPR features have also been combined with features extracted from other sensor data, such as EMI or seismic sensors [44, 52–54], as a feature-level form of sensor fusion.

Until recently, the performance of leading feature-based landmine detection algorithms were not compared with respect to environmental context. Wilson et al. made a large-scale comparison between four leading classification algorithms on a large GPR data set that was collected at four environmentally distinct test sites [41]. The following algorithms were compared: hidden Markov model (HMM) algorithm proposed by Gader et al. [42, 48], the edge histogram descriptor (EHD) algorithm proposed by Frigui et al. [44, 55], the algorithm based on geometric features (GEOM) proposed by Gader et al. [45], and the spectral correlation feature (SCF) algorithm proposed by Ho et al. [49].

Table 1.1 summarizes the results of the experiment, in which the algorithms were ranked based on benchmark PDs and FARs. The table illustrates that although EHD and HMM were the best-performing algorithms on the aggregate of all sites, certain algorithms performed better than others on specific sites and for specific performance metrics. In other words, the comparisons by Wilson et al. showed

Table 1.1: Performance of Landmine Detection Algorithms as Compared by Wilson et al., [41].

Metric	PD =.95	PD =0.90	PD =0.85	FAR =0	FAR =0.0007	FAR =0.00007
Site A	EHD	HMM	SCF	GEOM	SCF	GEOM
Site B	EHD	EHD	EHD	EHD	EHD	EHD
Site C	SCF	GEOM	GEOM/SCF	EHD	EHD	GEOM
Site D	EHD	HMM	HMM	EHD	EHD	EHD
All Sites	EHD	HMM	HMM	EHD	EHD	EHD

that there is currently no “silver bullet” classifier for GPR-based landmine detection across all environments. Furthermore, since the four algorithms exploit complementary features of GPR signatures, it was suggested that *algorithm fusion* may provide additional performance benefits. Experimental results illustrated that fusing the confidences of each algorithm, weighted according to their relative performance in each environment, could yield significant performance improvements.

1.3 Context-Dependent Learning

The impact of underlying *contextual factors* on how observations can be interpreted is not unique to landmine signatures in GPR data. Such effects, known as *context-dependency*, have been investigated much earlier in the field of semantic memory [56]. Words have virtually an infinite number of properties (e.g. “hospital” is both a “building” and “a place where food is served”). However, certain properties may be emphasized by how the word appears in certain semantic context (context-dependent properties), while others are always evident (context-independent properties). Referring to the hospital example, it is clearly evident that a hospital is a building, making it a context-independent property. However, the property of hospitals being a place of food service may only become evident in a discussion with patients and their dietitians, therefore making it a context-dependent property.

In statistical learning, the manifestation of context-dependent properties may

come in the form of changes in the distribution of a class or variable of interest (i.e., the *target concept*) with respect to underlying contextual factors. This problem is often referred to as *concept drift* [57, 58]. For learning in the presence of concept drift, it may be beneficial to utilize a context-dependent model. Speech recognition is a field that embraced this notion early on, where it was shown that context-dependent phonetic models (i.e., modeling phones as statistically-dependent on the phones immediately preceding and following it) yielded substantial improvements in the word recognition performance [59–61].

In remote sensing applications, it can be useful to exploit the dependency of sensor phenomenology on ambient environmental factors. Although the contextual factors being exploited are often sensor-specific, the common thread is that local similarities in sensor data can be exploited to improve overall robustness in detection performance. For example, in airborne remote sensing imagery, segmentation algorithms aim to find several locally homogeneous regions in a macroscopically heterogeneous image. These areas could correspond to buildings, different types of planted crops and vegetation cover, roads, or areas affected by natural disasters. While all pixels covering these types of areas should appear similar at a macroscopic scale, pixel-based segmentation generally leads to significant misclassification error within these regions. Incorporating spatial context has therefore been proposed for “smoothing out” these errors to yield more homogeneous segmentation regions [62, 63].

Anomaly detection is another problem that can benefit from a context-dependent learning approach, since ambient conditions can significantly affect the statistical distributions of anomalous sensor data, features extracted from such anomalies, or the confidence values of anomaly detection algorithms that exploit complementary information. In any of these spaces, similar observations can potentially be clustered into discrete contexts that are representative of unique environmental conditions. This process is referred to as *context identification*. After context identification is per-

formed, context-specific models can be trained for performing anomaly classification within each context. The decision rule for each of these classifiers may be unique for each of the contexts that were learned.

Several techniques have been proposed for context-dependent learning to assist with landmine detection in GPR data as well as in hyperspectral imagery (HSI). One method, known as Context Extraction for Local Fusion (CELF) [64], was proposed by Frigui et al. for multi-sensor fusion (e.g. GPR/EMI) in autonomous landmine detection systems. The CELF algorithm is motivated by the assumption that different subsets of the threat population will respond differently to different sensors. For example, shallow AP landmines are more easily detected with an EMI sensor than with GPR, because their GPR signature is often lost in the ground bounce. Therefore, the EMI sensor should be relied upon more heavily when those types of targets are encountered. Conversely, low-metal AT landmines are more easily detected by GPR than EMI, so GPR should be relied upon more heavily for these targets.

In CELF, a fuzzy clustering scheme was proposed for grouping together observations with similar responses from each sensor, and these clusters describe the underlying contexts. Learning the contexts is performed *discriminatively* by optimizing an objective function that accounts for both cluster size as well as discriminability of observations in each cluster by a linear decision rule. Experimental results showed that CELF was able to partition large data sets into observations with similar GPR/EMI responses, and it achieved better classification performance than either individual sensor as well as a conventional linear fusion incorporating no contextual information. It was also shown that CELF can be applied to fusion of multiple classification algorithms for the same sensor type [55]. For example, the four GPR algorithms that were originally compared by Wilson et al. [41] can be fused differently based on the underlying context, yielding significant improvements in performance over conventional algorithm fusion.

Another context-dependent classification technique, originally proposed for HSI, is the random set framework (RSF) proposed by Bolton and Gader [65]. The RSF treats observation *populations*, rather than individual observations, as random sets. The random sets of spectra that constitute the individual contexts were represented by a germ-and-grain model [66], which allowed for tractable modeling of irregular orientations of the observation space. A unique GMM classifier (based on the likelihood ratio test) was then trained via maximum-likelihood for each of the learned contexts.

The RSF differs significantly from CELF in how training is performed; the context model is trained in a supervised manner, with each context corresponding to the distinct environmental conditions in which data was collected, and the classifiers are learned independently from the context model. The germ-and-grain model is learned by minimizing the misclassification error between contexts, and the classifiers are trained by expectation-maximization of the GMM parameters for each class using the observations found in each context. Experimental results illustrated that the RSF achieved better classification performance than GMM classifiers incorporating no contextual information, including several baseline algorithms from the literature.

Both CELF and RSF have illustrated the potential that context-dependent learning has in improving overall performance. However, the approach on which these techniques are based can be improved upon further. First, in both CELF and RSF contextual factors are learned from similarities and differences in target responses. However, it may be desirable in some applications to be able to infer the context from a background, since it can be generally assumed that most data collected in the field will be target-free. For example, vehicular route-clearance systems that may travel and collect data for many kilometers may be able to obtain valuable contextual information from the background before encountering a target.

Furthermore, both CELF and RSF require specification of the number of con-

texts to be learned *a priori*. This caveat could be especially problematic in situations where the number of contexts that can potentially be encountered is unknown. Because each of these approaches essentially uses a mixture model to partition a high-dimensional data set into discrete contexts, the context model can easily be overtrained by specifying too many contexts, or undertrained by specifying too few. It may be more desirable to use a model that facilitates learning of the number of contexts that best explain the training data, while also facilitate the learning of new contexts as field data becomes available.

1.4 Novel Contributions

In contrast to the past literature, this dissertation is based on a different interpretation of context for anomaly detection applications. While past techniques by Frigui et al. and Bolton et al. have focused on context being a property of individual sensor observations, the algorithms developed in this work interpret context as the state of the world at a given location in space and time. Contextual information was extracted from raw background data through a set of physically-motivated features, which were developed for characterizing various environmental properties. Using these features, a variety of nonparametric context models were trained via Bayesian methods to learn a distinct number of contexts. Then, unique algorithm fusion weights were learned for each of the contexts. The overall classification performance of context-dependent fusion was compared to the leading target detection algorithms from the literature, as well as conventional algorithm fusion approaches.

A flowchart outlining the general procedure for context-dependent learning, as proposed in this dissertation, is shown in Figure 1.7. Given a set of observations $\mathbf{x}$, the underlying context of each observation is first identified probabilistically from the *contextual features* $\mathbf{x}^{(C)}$. The resulting *context posteriors*, $p(c_n = m | \mathbf{x}_n^{(C)})$, indicate the probability that $\mathbf{x}_n$ was observed under context m , for $m = 1, 2, \dots, M$. After

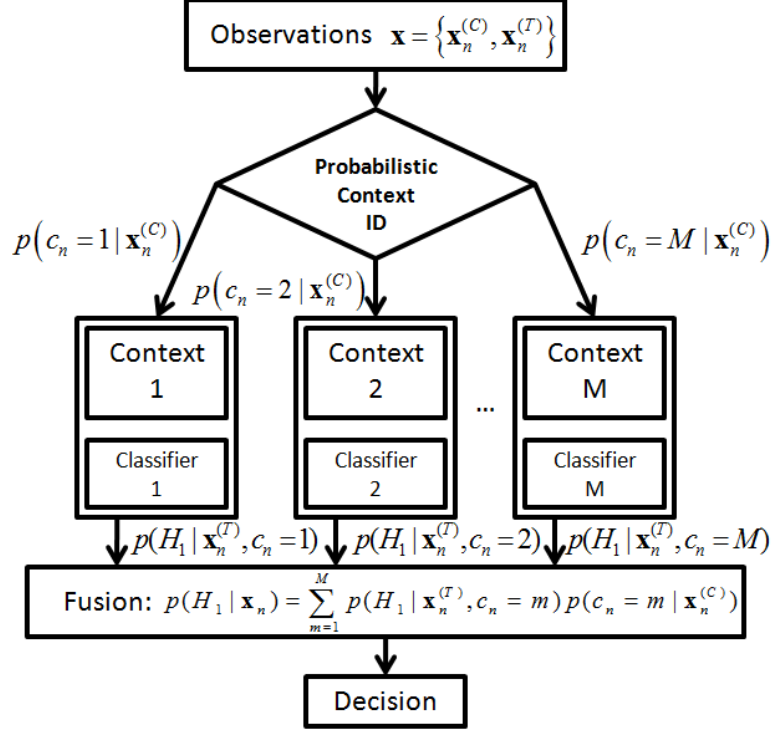


FIGURE 1.7: Flowchart illustrating a basic context-dependent classification technique.

partitioning the training data into M contexts, an ensemble of M binary classifiers are trained on the *target features* $\mathbf{x}^{(T)}$ of observations from each context. The resulting *within-context target posteriors*, $p(H_1 | \mathbf{x}_n^{(T)}, c_n = m)$, represent the probabilities that $\mathbf{x}_n$ belongs to the H_1 class, given that it was observed under context c_n . Finally, *target posteriors* $p(H_1 | \mathbf{x}_n)$ are calculated by integrating over uncertainty in context:

$$p(H_1 | \mathbf{x}_n) = \sum_{m=1}^M p(H_1 | \mathbf{x}_n^{(T)}, c_n = m) p(c_n = m | \mathbf{x}_n^{(C)}) \quad (1.1)$$

Context learning was performed using features, motivated by GPR phenomenology, that provide a low-dimensional characterization of local environmental conditions. These features were considered separately from the features used to characterize targets from non-targets, thereby facilitating learning of the context model's parameters

independently. Experiments were performed with real and simulated sensor data to illustrate that the context features are indicative of quantitative environmental properties which represent contextually-relevant factors in subsurface sensing.

The context models proposed in this dissertation are based on nonparametric Bayesian inference. The statistics literature has proposed several nonparametric Bayesian techniques that are useful in learning models of uncertain order, and these models facilitate an approach to learning the effective model order. In context learning, this amounts to learning not only the parameters that characterize each context’s distribution in feature space, but also the number of contexts present in the training data.

Several distinct context models are proposed in this dissertation. Although they all are essentially mixture densities that will partition the data into M components, they differ in the information used to partition the data. First, approaches that assume independence of observations are proposed. These include a Gaussian mixture model and a mixture of factor analysis models, each incorporating a Dirichlet process prior to facilitate learning of the number of contexts [67, 68]. A context model that incorporates spatial information is also presented, and is based upon an HMM with a Dirichlet process prior to facilitate learning of the number of states [69]. Comparisons are made between the different types of context models, and the advantages and disadvantages of using each are discussed. Furthermore, the merits of incorporating spatial information are also highlighted.

Two general techniques for learning the proposed context models are used. First, several *generative* context learning approaches are presented that consider the training of the context model as an independent task from training the binary target classifiers. A generative approach will learn the model that best explains the training data by maximizing the posterior probability of the model parameters. Such a learning approach can be useful in scenarios in which contextually-diverse training

data is available, and can potentially avoid overtraining to the given target/clutter population. The other approach is *discriminative* context learning, which will learn contexts that allow for the best discrimination of targets from non-targets. This is achieved by maximizing the posterior probability of the class labels, given the training data and the context model parameters.

Experimental results are presented for using context-dependent learning as a means for improving decision fusion of several detection algorithms used in fielded GPR systems. Performance is compared to the individual algorithms as well as to global fusion. In addition, results are presented illustrating the performance of context-dependent learning for improving anomaly classification in HSI. In both types of problems, context-dependent learning is shown to achieve higher PD and lower FAR than conventional machine learning approaches, emphasizing that valuable contextual information can be exploited from the background data to improve sensing robustness in the presence of changing environmental conditions.

Extracting Contextual Information from GPR Data

One of the goals of this work is to statistically model the distinct contexts present in large bodies of GPR data collected over varying environmental conditions. However, the dimensionality of raw sensor data can be very high. For example, data collected with the NIITEK GPR has a temporal resolution of 512 samples and a 5 cm spatial sampling rate, and a B-scan covering the entire signature of a target could be as many as 25 downtrack samples long. Therefore, vectorizing the B-scan would result in a 64,000-dimensional observation. Furthermore, many dimensions (i.e., pixels) of raw data could be highly correlated (e.g., neighboring pixels), while others could be non-informative (e.g., pixels above the ground bounce). High-dimensional data is very difficult to model statistically due to the oft-cited *curse of dimensionality* [70–72], which suggests that the number of required training samples increases exponentially with the number of dimensions. Therefore, in order to effectively model the distribution of various contextual factors in GPR data, it may be desirable to utilize low-dimensional features that characterize such factors and are amenable to clustering.

In this chapter, physics-based techniques for extracting contextual features from

raw GPR data are described¹. GPR phenomenology suggests that a single B-scan may contain an abundance of information about various contextual factors, including surface roughness, soil electromagnetic properties, the presence of multiple layers, and the subsurface heterogeneity. Direct estimation of these subsurface environmental properties may be achieved via inverse numerical modeling or deconvolution [27, 29–34]. However, the computational complexity of inversion and deconvolution makes real-time implementation of these approaches infeasible.

This chapter proposes an alternative technique for extracting contextual information from GPR background data using several features that were developed based upon a transmission line model [10, 39]. Statistical classification and regression models were trained on the features to predict multiple environmental properties from real and simulated GPR data. Experimental results illustrate that the proposed features are indicative of several quantitative factors that can be used to facilitate context learning in buried threat detection applications.

2.1 Transmission Line Model for GPR

A simple phenomenological model for GPR A-scans can be motivated by electrical transmission lines [10, 39]. In a similar manner to a signal transmitted down a transmission line with several impedance mismatches, a GPR signal consists of several reflections of the transmitted pulse at various amplitudes and delays. According to this model, each received pulse therefore corresponds to a subsurface interface. Figure 2.1 provides a basic illustration of the transmission line model as an approximation of a heterogeneous soil environment. Note the similarity of the signal derived from such a model to a typical GPR A-scan.

¹ This chapter is derivative of previously published work, © 2012 IEEE. Reprinted, with permission, from Ratto et al., “Characterization of the subsurface environment with GPR using feature-based statistical learning,” *IEEE Transactions on Geoscience and Remote Sensing*, in review as of Feb. 2012.

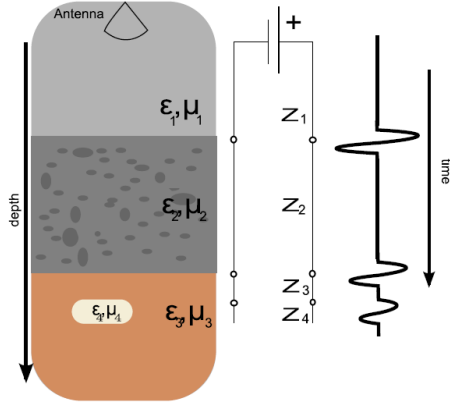


FIGURE 2.1: A diagram of transmission line model for GPR A-scans [39]. Left: an example of unique dielectric layers in subsurface. Center: the corresponding transmission line with three characteristic impedances. Right: An A-scan generated under this model.

Several broad assumptions are made by modeling GPR A-scans as the signal received from a mismatched transmission line. Multipath effects are ignored, propagating waves are assumed to be planar, all interfaces are assumed planar and infinite in extent, and that the respective transmission media are assumed to be homogeneous, lossless, and non-dispersive. However, any deviations of real signals from the model assumptions may be accounted for by a statistical model. In the remainder of this chapter, the features derived from the transmission line model are described, and experimental results illustrate that these features are indicative of quantitative environmental properties via statistical inference.

2.2 GPR Contextual Features

A variety of features are proposed in this chapter for extracting contextual information from GPR B-scans. The following notation is used in describing the features: columns of the B-scan are A-scans denoted as $a(t)$, where t is a temporal sample index ($t = 1, 2, \dots, T$); rows are the time-slices denoted as $b(n)$, where n is a spatial sample index ($n = 1, 2, \dots, N$). For each feature that is described in this section, sample

feature vectors extracted from simulated B-scans are shown. The simulated B-scans were generated using the publicly-available finite-difference time-domain (FDTD) modeling software GprMax [73, 74].

Energy features

The total energy of an A-scan is a basic feature that is calculated by summing the time samples of a squared A-scan:

$$e = \sum_{t=1}^T a^2(t) \quad (2.1)$$

The energy feature provides information regarding several properties of the subsurface environment. In scenarios where the GPR antenna is close to the ground, there is high dielectric contrast between the air and the ground, or the soil is very heterogeneous, the energy feature should have a high value. Furthermore, scenarios in which the GPR antenna is high above the ground, the soil has little dielectric contrast with the air, or the subsurface is largely free of inhomogeneities, the energy feature should yield a low value.

Reflection coefficient features

In transmission lines, the degree of impedance mismatch is often expressed in terms of reflection coefficients, i.e. the ratio of reflected to transmitted power. In GPR, the reflection coefficient at the air/ground interface is of particular interest; because the dielectric properties of air are usually assumed to be equal to those of free space, the air/ground reflection coefficient may characterize subsurface dielectric properties. Accurate estimation of the air/ground reflection coefficient must take into account propagation losses, rather than simply compare the ground bounce magnitude to the transmitted power, or else estimates may be inaccurate [75]. The free-space loss

(L_{FS}) of a line-of-sight path follows the power law given by

$$L_{FS} = \left(\frac{4\pi d}{\lambda} \right)^2 = \left(\frac{4\pi df}{c} \right)^2, \quad (2.2)$$

where distance is denoted by d , λ is the signal's wavelength, and c is the free-space propagation speed. For a given distance, transmit and receive antennas with respective gains P_T and P_R , and a reflector with cross-sectional area A , and transmitted power P_T , the received power P_R can be expressed as a function of the reflection coefficient Γ :

$$P_R = \frac{P_T G_T G_R A}{4\pi d^2 L_{FS}} \Gamma^2 = \frac{P_T G_T G_R c^2}{(4\pi)^2 f^2 d^4} \Gamma^2 \quad (2.3)$$

Solving for Γ , and consolidating P_T , G_T , G_R , A , f , and c into a single constant that characterizes the radar system, the reflection coefficient can be expressed as proportional to a function of distance and received power:

$$\Gamma \propto d^2 \sqrt{P_R}. \quad (2.4)$$

In GPR data, the reflection coefficient can be approximated by applying basic radar ranging to the approximate ground bounce. First, d must be calculated by dividing the ground bounce arrival time by the system's range resolution (expressed in samples/m) T :

$$d = t_{GB}/S \quad (2.5)$$

where it is assumed that

$$t_{GB} = \underset{t}{\operatorname{argmax}} a(t) \quad (2.6)$$

The received power is calculated by windowing out the ground bounce from an A-scan using the Gaussian function $w(t)$. The Gaussian window is centered on the midpoint between the A-scan's global maximum and minimum, and its width is specified by

the constant σ_w :

$$P_R = \sum_{t=1}^T w(t)a^2(t) \quad (2.7)$$

$$w(t) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp \left[-\frac{(x - \mu_w)^2}{2\sigma_w^2} \right] \quad (2.8)$$

$$\mu_w = t_{GB} + (t_{min} - t_{GB})/2 \quad (2.9)$$

$$\sigma_w = \text{const.} \quad (2.10)$$

$$t_{min} = \underset{t}{\operatorname{argmin}} a(t) \quad (2.11)$$

Figure 2.2 illustrates a comparison of the energy and reflection coefficient features extracted from simulated B-scans generated over simulated soils with different dielectric properties. The top panel illustrates a soil characterized by a low dielectric constant ($\epsilon_r = 3$), and the bottom panel illustrates a soil characterized by a high dielectric constant ($\epsilon_r = 10$), and the electrical conductivities of both soils were equal. The plots of the feature values illustrate that the energy and reflection coefficient values are higher for the soil with high dielectric constant. Note that the values of the reflection coefficient feature do not reflect valid reflection coefficient values (i.e., between 0 and 1) because the scaling constants in (2.3) are ignored.

Matching pursuits features

GPR data can appear very cluttered when collected over heterogeneous soils due to reflections from multiple subsurface interfaces, and it may be useful to determine when heterogeneous soils are encountered. One technique for measuring soil heterogeneity based on the transmission line model is to determine how many unique pulses can be used to replicate an A-scan. This is based on the hypothesis that A-scans collected over heterogeneous soils would consist of more pulses than A-scans collected over homogeneous soils. In this work, the matching pursuits (MP) algorithm

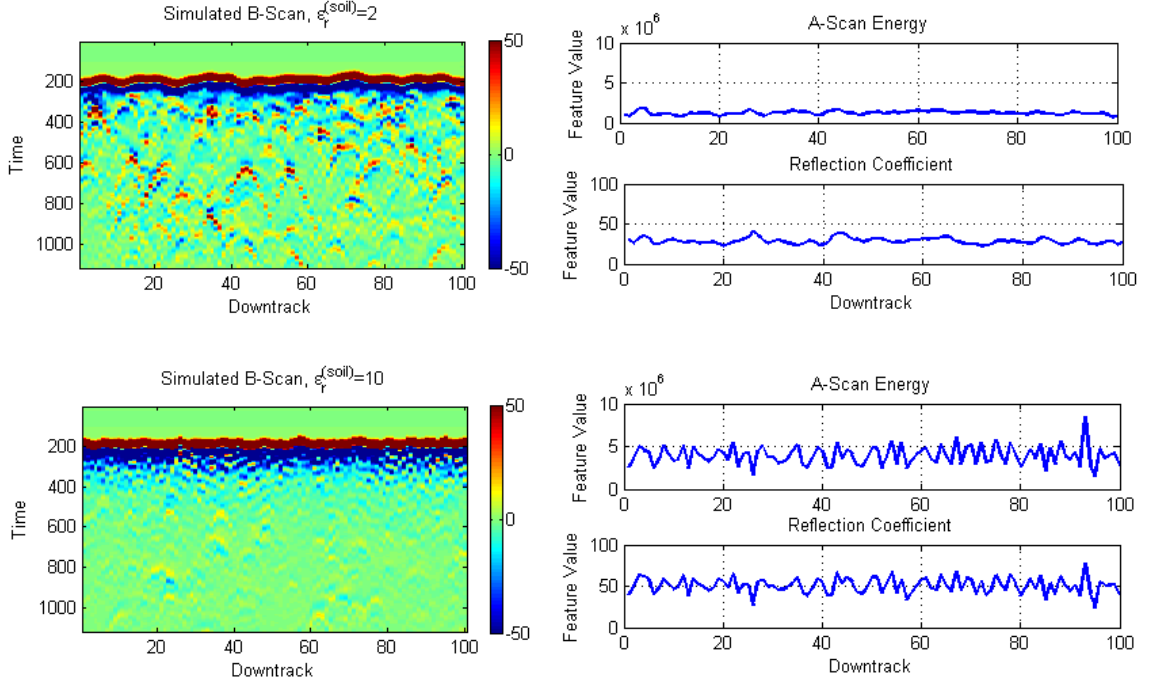


FIGURE 2.2: Examples of energy and reflection coefficient features for FDTD-simulated B-scans. Top: soil with dielectric constant of 3. Bottom: soil with dielectric constant of 10. The simulated B-scans are shown at left, and plots of the energy and reflection coefficient features values are shown at right. © 2012 IEEE.

proposed by Mallat and Zhang [76]) is used to approximate an A-scan as a sum of unique pulses, which are selected from a dictionary $\mathbf{D} = \{\mathbf{d}(\omega, t_0)\}$ of differentiated Gaussian elements with varying widths ω and temporal positions t_0 :

$$\mathbf{d}(\omega, t_0) = -\frac{(t - t_0)}{\omega} \exp\left(\frac{-(t - t_0)^2}{2\omega}\right), \quad t_0 = 1, 2, \dots, T \quad (2.12)$$

The basic MP algorithm first correlates each dictionary element with the original signal, $\mathbf{a} = \{a(t)\}$, then subtracts from the signal the most-correlated element weighted by its correlation. The process is then repeated using the residual signals, and continues until the change in energy falls below a specified threshold (δ_0). Algorithm 1 summarizes the application of MP to GPR A-scans.

From the set of selected dictionary elements, features can be extracted to characterize subsurface heterogeneity. One feature is the number of iterations for MP to

converge (n_{MP}), which is analogous to the number of unique pulses that make up an A-scan. From the transmission line model, each pulse corresponds to a unique reflection at a dielectric interface. Therefore, the number of MP iterations may characterize the amount of subsurface heterogeneity. Soils with low levels of heterogeneity should yield lower values of n_{MP} than soils with high levels of heterogeneity.

Algorithm 1 Basic matching pursuits [76]

```

input  $\mathbf{a}, \mathbf{d}, \delta_0, \omega$ 
 $n = 0$ 
 $\mathbf{r} = \mathbf{a}$ 
while  $\delta E \leq \delta_0$  do
   $n = n + 1$ 
  for  $t_0 = 1, 2, \dots, T$  do
     $\rho(\omega, t_0) = \mathbf{d}(\omega, t_0)^T \mathbf{r} / \|\mathbf{d}(\omega, t_0)\|^2$ 
  end for
   $t'_{0n} = \underset{t_0}{\operatorname{argmax}} \rho(\omega, t_0)$ 
   $\mathbf{r} = \mathbf{r} - \rho(\omega, t'_{0n}) \mathbf{d}(\omega, t'_{0n})$ 
   $E_n = \|\mathbf{r}\|^2$ 
   $\delta E = E_{n-1} - E_n$ 
end while
 $n_{MP} = n$ 
 $\hat{\mathbf{a}} = \mathbf{a} - \mathbf{r}$ 
return  $n_{MP}, \hat{\mathbf{a}}, \mathbf{t}_0'$ 

```

Another feature derived from MP is the temporal histogram of the selected dictionary elements, denoted by $\mathbf{h}_{MP}$. The histogram bins correspond to the temporal centers t' , and are experimentally determined. The goal of using the MP histogram is to differentiate between soils of varying heterogeneity, using the hypothesis that as heterogeneity increases so will the number of late-time reflections. The number of late-time reflections would be reflected in the late-time values of the histogram.

Figure 2.3 illustrates the MP features for two B-scans. The top plots illustrate a simulated B-scan over soil with low heterogeneity and corresponding MP histogram, while the bottom plots corresponds to a highly heterogeneous soil. Note the differences in the number of MP iterations and late-time histogram bins. The more heterogeneous soil yields a higher number of total reflections, with a greater propor-

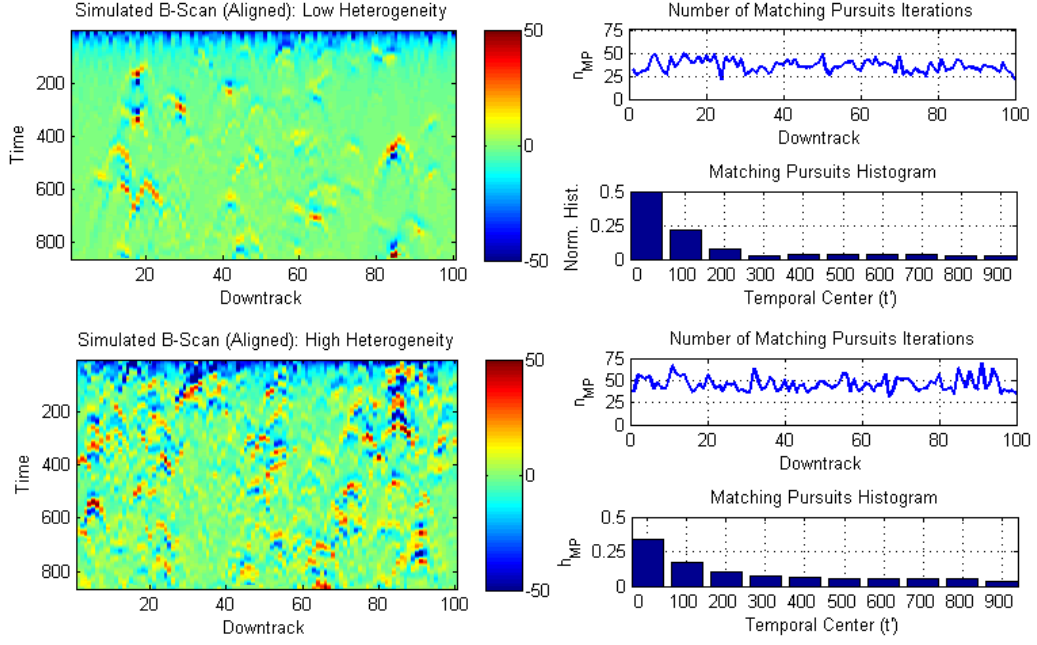


FIGURE 2.3: Example of MP histogram extracted from B-scans over soils. Top-left: Simulated B-scan of low-heterogeneity soil; Top-right: number of MP iterations until convergence and MP histogram for low-heterogeneity soil; Bottom-left: Simulated B-scan of high-heterogeneity soil; Bottom-right: number of MP iterations until convergence and MP histogram for high-heterogeneity soil. © 2012 IEEE.

tion of them occurring late in time. Therefore, MP places more dictionary elements in the later portion of the signal. Since the ground bounce is the portion of the A-scan with the highest local energy, the majority of dictionary elements would be selected to describe that portion of the signal. To prevent this from occurring, and to obtain more information regarding subsurface reflections, MP was only run on A-scan samples that occur after t_{GB} .

Linear prediction features

Linear prediction (LP) filters provide causal estimates of the power spectrum of a signal [77]. In GPR applications, LP filters have been found to be particularly useful as anomaly detectors [36–40]. Linear predictors can be applied to time-slices (rows)

of GPR data, so that anomalies could be characterized by high prediction error. This result is because the LP filter is based on assumptions similar to the transmission line model - planar interfaces infinite in extent should yield the same response for all locations, and deviations from that planar assumption are considered as random noise. Therefore, the behavior of LP filters may characterize “steady-state” environmental properties such as soil dielectric constant [78], as well as stochastic properties such as surface roughness and heterogeneity [79].

The equation for a LP filter takes the form of an autoregressive model of order K , characterized by weights $\boldsymbol{\alpha} = [\alpha(1), \alpha(2), \dots, \alpha(K)]^T$, which are applied to B-scan time-slices $\mathbf{b}(n) = [b(n), b(n-1), \dots, b(n-K)]$. The filter outputs a zero-mean, white noise process $e(n)$ with variance ν :

$$\sum_{k=1}^K \alpha(k)b(n-k) = e(n). \quad (2.13)$$

The weights can be determined from the normal equations,

$$\boldsymbol{\alpha} = \mathbf{R}^{-1}\mathbf{p}, \quad (2.14)$$

where the correlation matrix ($\mathbf{R}$) and cross-correlation vector ($\mathbf{p}$) are given by

$$\mathbf{R} = \mathbb{E} [\mathbf{b}(n-1)\mathbf{b}^H(n-1)] \quad (2.15)$$

$$\mathbf{p} = \mathbb{E} [\mathbf{b}(n-1)b(n)]. \quad (2.16)$$

Given the weights, the prediction-error power (ν) can be found by calculating the mean-square error of the filter applied to the data:

$$\begin{aligned} \nu &= \mathbb{E} [|b(n) - \boldsymbol{\alpha}^T \mathbf{b}(n-1)|^2] \\ &= \sigma_b^2 - 2\boldsymbol{\alpha}^T \mathbf{p} + \boldsymbol{\alpha}^T \mathbf{R} \boldsymbol{\alpha} \end{aligned} \quad (2.17)$$

To extract the LP features from GPR data, a B-scan is first aligned according to each column's t_{GB} , and all data up to and including t_{GB} are discarded. LP filters of

order K are then trained and evaluated on aligned B-scan rows $\mathbf{b}_{t'}$, where t' are 10 experimentally-determined row indices. The calculated values of $\nu_{t'}$ are concatenated into a feature vector of length equal to the number of rows used. In line with past investigations [78, 79], models of order $M = 4$ were used. Although different values of M were considered, the overall effect of changing the model order on prediction error was not significant.

Figure 2.4 illustrates two aligned B-scans corresponding to soils with different dielectric and surface roughness properties (shown at left), and compares the differences in corresponding prediction-error power (shown at right). In the top panel, the features corresponding to a soil with low dielectric constant and high “roughness” (characterized by a surface with low correlation length) are shown, and the corresponding prediction-error power tends to be high. As shown in the bottom panel, a soil with high dielectric constant and low “roughness” (characterized by a highly-correlated surface) yields more predictable data, and therefore the prediction-error power is much lower and is more constant with respect to time slice index.

2.2.1 Feature consolidation

Several of the proposed contextual features (e , Γ , n_{MP}) are extracted from individual A-scans, and it is important to eliminate redundancy and spatial dependence in feature extraction. Therefore, these features were averaged across the columns of the B-scan from which they were extracted. Table 2.2.1 summarizes the elements of the 23-dimensional feature vector, $\mathbf{x}^{(C)} = [\bar{e}, \bar{\Gamma}, \bar{n}_{MP}, \mathbf{h}_{MP}, \boldsymbol{\nu}]$, to provide a low-dimensional representation of the B-scan’s contextual information. In all experiments performed on these features, the dimensions of $\mathbf{x}^{(C)}$ were normalized to be zero mean, unit variance prior to further processing.

The following sections illustrate the efficacy of the proposed features in characterizing multiple subsurface environmental properties. Experiments were performed

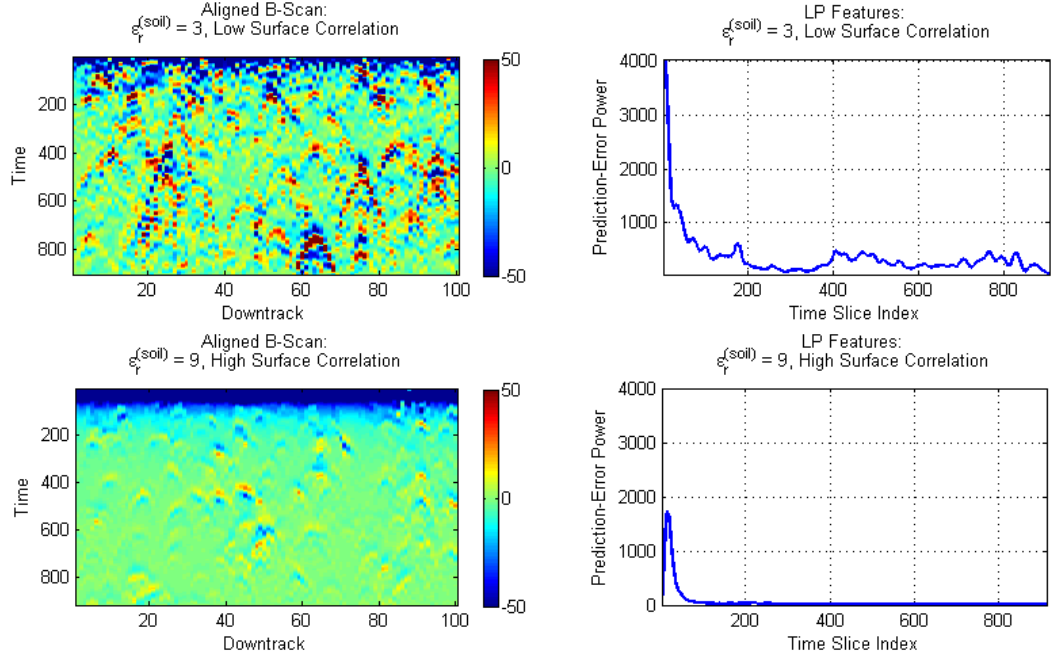


FIGURE 2.4: Example of LP prediction-error power extracted from aligned B-scans over simulated soils. Top-left: simulated aligned B-scan of soil with low dielectric constant and low surface correlation length. Top-right: LP prediction-error power, measured as a function of temporal index, for soil with low dielectric constant and low surface correlation length. Bottom-left: simulated aligned B-scan of soil with high dielectric constant and high surface correlation length. Bottom-right: LP prediction-error power, measured as a function of temporal index, for soil with high dielectric constant and high surface correlation length. © 2012 IEEE.

Table 2.1: Features ($\mathbf{x}^{(C)}$) for classification and regression of environmental parameters.

Element	Description	# of Dimensions
$\bar{e}$	Average A-scan energy	1
$\bar{\Gamma}$	Average reflection coefficient	1
$\bar{n}_{MP}$	Average # MP iterations	1
$\mathbf{h}_{MP}$	MP Temporal Histogram	10
$\boldsymbol{\nu}$	LP filter prediction-error power	10

using both simulated and field-collected GPR data, and preliminary analysis was originally presented in [80]. Descriptions of the data sets are also provided, including the settings of feature extraction parameters for each experiment.

2.3 Evaluating GPR Contextual Features: Simulated Data Experiment

The first experiment to test the efficacy of GPR contextual features was performed on simulated GPR data with known environmental properties. Simulated B-scans were generated using the publicly-available GprMax software [73, 74], which is based on the finite-difference time domain (FDTD) modeling technique [81, 82]. Simulated B-scans were constructed by displaying the measured electric field as a function of time at a series of fixed locations corresponding to the receiving antenna’s position.

GPR data was simulated over many realizations of a soil environment with several parameters, some random and others deterministic. The soil environment consisted of a random rough surface, homogeneous soil background, and random subsurface scatterers, and is characterized by four environmental model parameters. B-scans collected over the simulated soil were meant to approximate target-free background data. Contextual features were extracted from the simulated B-scans, and the simulation parameters were predicted from the features via relevance vector machine (RVM) regression and classification [83, 84]. Details regarding RVM implementation can be found in Appendix B.

2.3.1 *Simulated Data Set*

The two-dimensional computational domain $5\text{ m} \times 60\text{ cm}$ with a spatial resolution of 2.5 mm. The computational domain was surrounded by a perfectly matched layer (PML) boundary condition, which is necessary to absorb any extraneous reflections of the electromagnetic fields off the edges of the domain. A-scans were measured

as the received electric field, collected at spatial intervals of 5 cm, yielding a total of 100 A-scans per simulation. The transmit and receive elements were modeled as co-located infinite line sources, polarized in the perpendicular location, and located 10 cm above the mean surface elevation. The transmitter was excited by a Gaussian current pulse with center frequency $f_c = 2$ GHz, therefore yielding a differentiated Gaussian pulse in the electric field. FDTD was run with a time gate of 6.6 ns, i.e. the round-trip travel time for a propagation distance of 1 m in air, with a temporal resolution of $S = 1120$ time samples per A-scan. Therefore, each FDTD simulation yielded an 1120×100 B-scan consisting of received electric field as a function of time and receiver location.

The computational domain included a soil half-space characterized by a variety of model parameters specified *a priori*. The soil surface was stochastically generated from a Gaussian power spectrum, characterized by the correlation length parameter ($l^{(surf)}$). The homogeneous soil background was characterized by a range of values of dielectric constant ($\epsilon_r^{(soil)}$) and conductivity ($\sigma^{(soil)}$). Finally, the subsurface heterogeneities were of random quantity, characterized by a binomial distribution with mean $N^{(scats)}$. Several examples of the computational domain are shown in Figure 2.5 alongside the corresponding simulated B-scans. Each example illustrates a unique combination of model parameters. In the following subsections, the generation of the various elements of the simulated soil are described in detail.

Rough soil surface

A common technique for modeling rough surfaces in scattering experiments is to model the surface as a stochastic process $f(n)$ (where n denotes spatial index) with a Gaussian spectrum [24–28]. The surface profile is realized by passing white noise

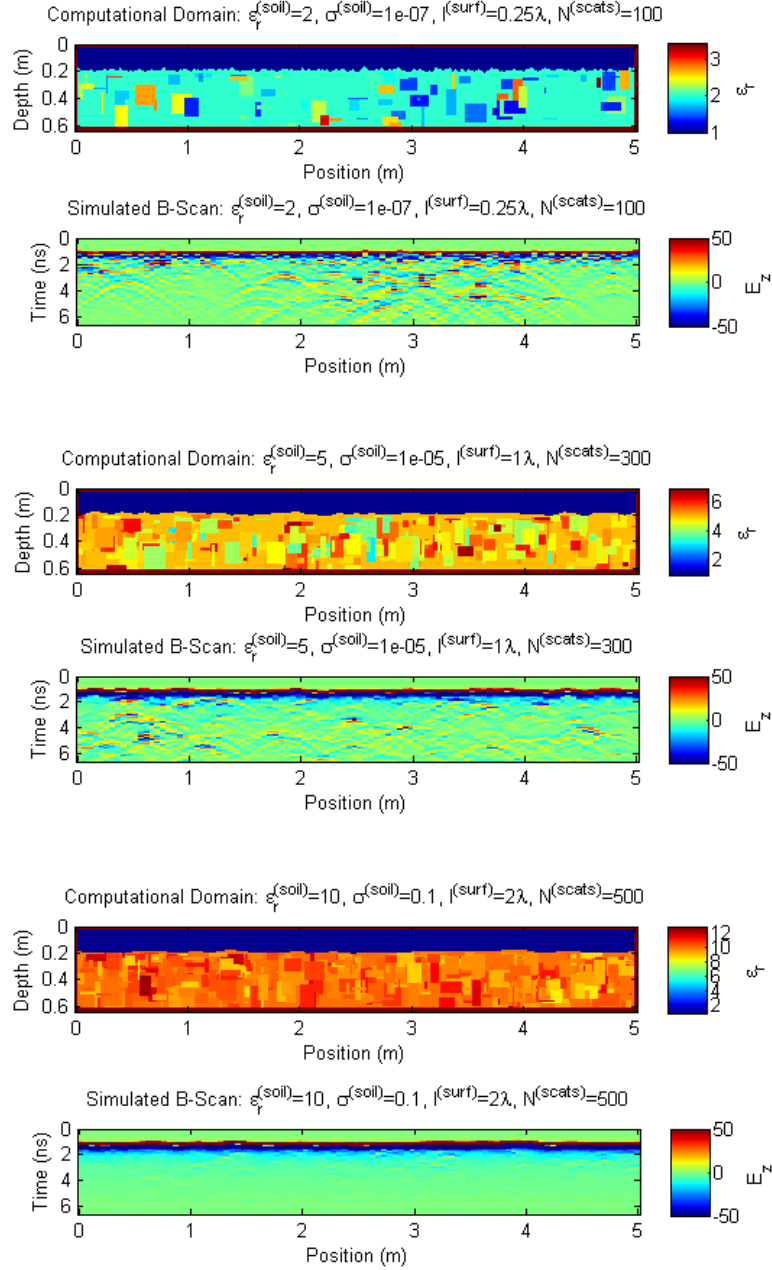


FIGURE 2.5: Examples of computational domain and FDTD-simulated GPR data. Top:) $\epsilon_r^{(soil)} = 2$, $\sigma^{(soil)} = 10^{-7}$, $l^{(surf)} = 0.25\lambda$, $N^{(scats)} = 100$. Center: $\epsilon_r^{(soil)} = 5$, $\sigma^{(soil)} = 10^{-5}$, $l^{(surf)} = 1\lambda$, $N^{(scats)} = 300$. Bottom: $\epsilon_r^{(soil)} = 10$, $\sigma^{(soil)} = 10^{-1}$, $l^{(surf)} = 2\lambda$, $N^{(scats)} = 500$. The top plot of each panel illustrates the computational domain, where the horizontal axis represents position, the vertical axis represents depth, and color represents dielectric constant. The bottom plot illustrates the corresponding simulated B-scan, where the horizontal axis represents position, the vertical axis represents time, and color represents the received electric field amplitude. © 2012 IEEE.

through a filter with spatial frequency response $H(k)$:

$$H(k) = \frac{l^{(surf)} h^{(surf)^2}}{2\sqrt{\pi}} \exp\left(\frac{-k^2 l^{(surf)^2}}{4}\right), \quad (2.18)$$

where $l^{(surf)}$ and $h^{(surf)^2}$ are parameters for correlation length and variance, respectively. In this experiment, rough surfaces were generated using parameter values suggested in [28]. The value of $h^{(surf)}$ was fixed at $\lambda_c/20$, where λ_c is the wavelength corresponding to the center frequency of the GPR pulse. The value of $l^{(surf)}$ was variable, with potential values of $\{\lambda_c/4, \lambda_c/2, \lambda_c, 2\lambda_c\}$.

Soil background

The soil half-space was characterized by a homogeneous background with spatially-invariant values of dielectric constant $\epsilon_r^{(soil)}$, conductivity $\sigma^{(soil)}$, and permeability $\mu_r^{(soil)} = \mu_0$. Dispersion effects were ignored, so all electromagnetic parameters were also assumed constant with respect to frequency. Both $\epsilon_r^{(soil)}$ and $\sigma^{(soil)}$ were variable, with potential values to characterize a wide range of soils as tabulated in [10]: $\epsilon_r^{(soil)} = \{2, 3, \dots, 10\}$ and $\sigma^{(soil)} = \{10^{-7}, 10^{-6}, \dots, 10^{-1}\}$.

Random Scatterers

Heterogeneity in the soil half-space was modeled by overlaying many random box-shaped scatterers onto the background medium, in a manner similar to that used in [22]. The lower left-hand coordinates of the scatterers were uniformly-distributed: $X^{(scat)} \sim \mathcal{U}(0, 5m)$, $Y^{(scat)} \sim \mathcal{U}(0, \max f(n) - 2.5\text{cm})$. The dimensions of the scatterers were also uniformly-distributed: $x \sim \mathcal{U}(d, 20\text{cm})$, $y \sim \mathcal{U}(d, 20\text{cm})$. The dielectric constant of each scatterer ($\epsilon_r^{(scat)}$) was also random, but drawn from a distribution that allowed them to appear as perturbations from the background:

$$\epsilon_r^{(scat)} = 1 + \tilde{\epsilon}_r \quad (2.19)$$

$$\tilde{\epsilon}_r \sim \log\mathcal{N}(\tilde{\mu}, \tilde{\sigma}), \quad (2.20)$$

where

$$\tilde{\mu} = \log\left(\frac{m^2}{\sqrt{v + m^2}}\right) \quad (2.21)$$

$$\tilde{\sigma} = \sqrt{\log(1 + v/m^2)} \quad (2.22)$$

$$m = \epsilon_r^{(soil)} - 1 \quad (2.23)$$

$$v = 0.5. \quad (2.24)$$

Drawing values from this distribution ensures $\epsilon_r^{(scat)} \geq 1$ and $\mathbb{E}[\epsilon_r^{(scat)}] = \epsilon_r^{(soil)}$. The conductivity of all scatterers was fixed at $\sigma^{(soil)}$. Finally, the number of scatterers present in the soil half-space was drawn from a binomial distribution, $n \sim \text{bin}(2N^{(scats)}, p = 0.5)$, where $N^{(scats)}$ is a variable parameter indicating the expected number of subsurface scatterers. Three potential values of $N^{(scats)}$ were used: $\{100, 300, 500\}$.

Size of Data Set

In total, there were 756 possible combinations of the variable soil parameters (9 different values of $\epsilon_r^{(soil)}$, 7 values of $\sigma^{(soil)}$, 4 values of $l^{(surf)}/\lambda$, 3 values of $N^{(scats)}$). Two unique simulations were performed for each combination of these parameters, yielding a total of 1512 B-scans from which features were extracted.

2.3.2 Feature Extraction

All of the features described in Section 2.2 were extracted from the 1512 simulated B-scans. Table 2.2 lists the parameters were used in order to extract features from this data. The values of σ_w , δ_0 , t' , and K were determined experimentally to yield the best overall performance, the value of C is arbitrary, the value of ω was determined by inspection of the transmitted GPR waveform, and the values of N and T are artifacts of the data simulation.

Table 2.2: Feature extraction parameters for simulated data experiment

Parameter	Description	Value
N	Number of A-scans per B-scan	100
T	1m temporal sampling rate	1120
σ_w	Window width parameter	30
δ_0	MP convergence threshold	0.01
ω	Width of MP dictionary elements	200
t'	B-scan row indices	0,70,...,700
K	Linear predictor filter order	4

2.3.3 Correlation Analysis

Pairwise correlations between the features and the labels $[\epsilon_r^{(soil)}, \sigma^{(soil)}, l^{(surf)}, N^{(scats)}]$ were calculated to illustrate the efficacy of each feature in characterizing one or more soil properties. Figure 2.6 illustrates the correlation of each feature with each of the four labels. The values of $\epsilon_r^{(soil)}$ were correlated (or inversely correlated) with most of the features. This is because $\epsilon_r^{(soil)}$ greatly affects the signal amplitude, both at the air/ground interface and within the soil itself, and most of the proposed features are functions of signal amplitude. Furthermore, $\sigma^{(soil)}$ was correlated with the matching pursuits histogram, suggesting that this feature may be indicative of the attenuation of signals as a function of time. The values of $l^{(surf)}$ are most correlated with the early-time measurements of LP power, suggesting that the most unpredictable rows of the B-scan may be due to rough surface scattering. Finally, $N^{(scats)}$ did not correlate as highly with the matching pursuits histogram as originally hypothesized. This could be due to insufficient binning of the matching pursuits histogram, or not enough variation in the values of $N^{(scats)}$ considered in this experiment.

2.3.4 Classification Results

A RVM was used to classify features extracted from the simulated GPR data according to the known soil properties. For each multi-class problem, the RVM was trained using a one-against-all approach, and test observations were assigned to the

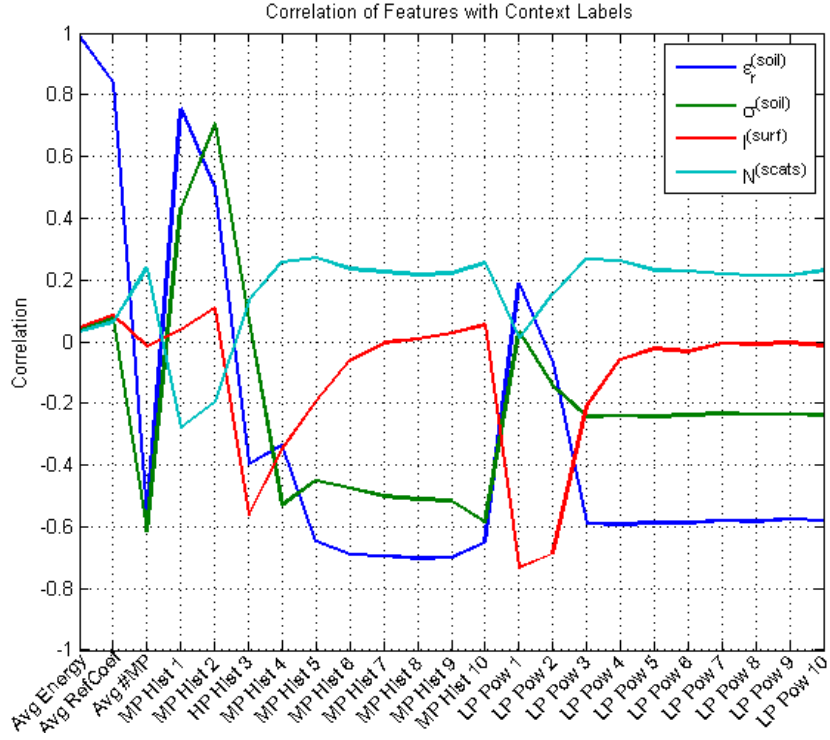


FIGURE 2.6: Plot of correlations between features (horizontal axis) and soil labels (line color) for the simulated GPR experiment. © 2012 IEEE.

maximum *a posteriori* (MAP) class. Classification of B-scans according to the soil labels with the RVM was evaluated via 10-folds cross-validation. Results are shown in Figure 2.7. Each confusion matrix illustrates overall classification performance, with truth listed on the vertical axis and classification result on the horizontal. The percent of B-scans classified correctly is shown at the top of each confusion matrix. For some of the labels, classification was very good - classifying the simulated GPR data by $\epsilon_r^{(soil)}$ yielded an overall accuracy of 97.24% (compared to 11.1% chance accuracy), and classification by $l^{(soil)}/\lambda_c$ yielded an accuracy of 90.41% (compared to 25% chance accuracy). The result of classification by $N^{(scats)}$ was still relatively good, achieving an correct classification rate of 76.5% (compared to 33.3% chance accuracy). Classification by $\sigma^{(soil)}$ yielded good performance in identifying conditions with very high values of conductivity, while lower conductivities were often confused.

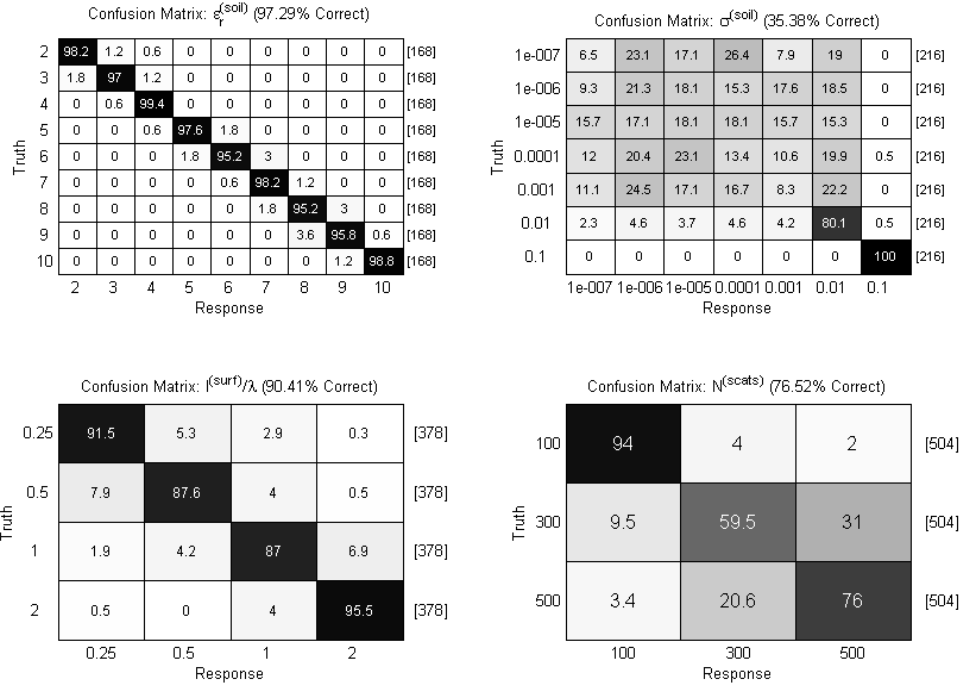


FIGURE 2.7: Confusion matrices illustrating results of RVM classification of $\epsilon_r^{(soil)}$ (top-left), $\sigma^{(soil)}$ (top-right), $l^{(soil)}/\lambda_c$ (bottom-left), and $N^{(scats)}$ (bottom-right) for the simulated data experiment. Vertical axes indicate the the true labels, and horizontal axes indicate the classifier response.

However, others have illustrated that the overall effect of soil conductivity on GPR is minimal unless the conductivity is very high [20]. The performance of the RVM in classifying B-scans by soil conductivity confirms these observations.

2.3.5 Regression Results

Unlike classification, which makes “hard” decisions, regression allows for the quantitative estimation of the underlying soil parameters. RVM-based regression results are shown in Figure 2.8. Each plot shows the regression output for each observation (dashed line) and the true values of the soil parameters (solid line). The goodness-of-fit is summarized by the RMS error, which is shown above each plot.

As in classification, RVM regression was able to very accurately estimate $\epsilon_r^{(soil)}$

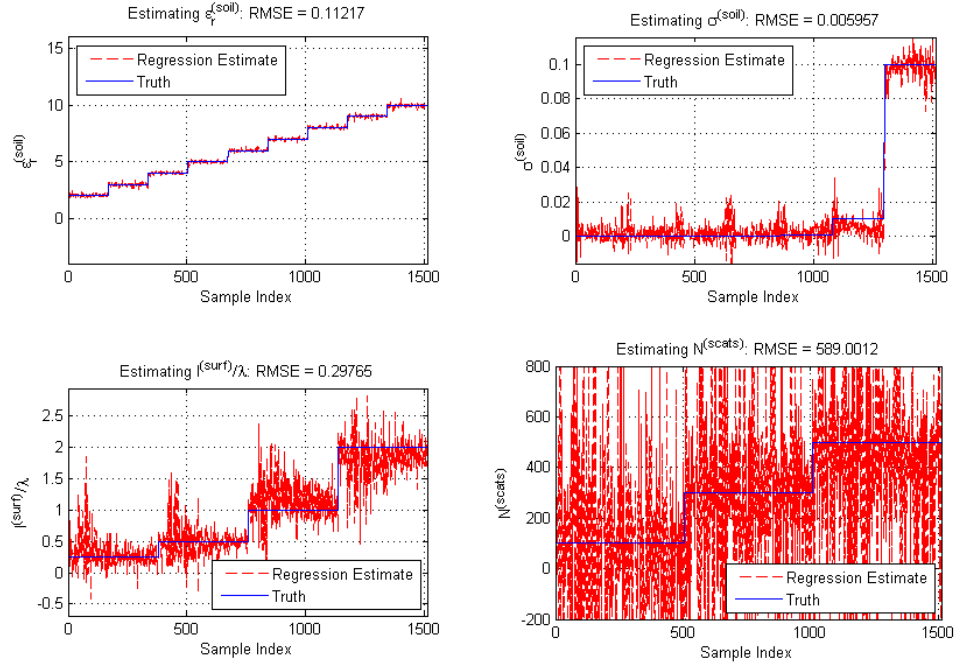


FIGURE 2.8: Results of RVM regression for predicting $\epsilon_r^{(soil)}$ (top-left), $\sigma^{(soil)}$ (top-right), $l^{(soil)}$ (bottom-left), and $N^{(scats)}$ (bottom-right) for the simulated data experiment. The blue line in each plot indicates the true values of each parameter, and the dashed red line indicates the regression estimate.

(RMSE = 0.11217) and $l^{(soil)}/\lambda_c$ (RMSE = 0.29765). Furthermore, estimation of $\sigma^{(soil)}$ was accurate only for the highest values, yielding an RMSE of 0.006. The only major difference between the regression and classification results was the estimation of $N^{(scats)}$, which yielded a RMSE of 589. This may be due to the fact the $N^{(scats)}$ is the *expected* number of subsurface scatterers, rather than the exact number.

Overall, however, regression results illustrate that it is possible to not only predict the different environmental conditions that were imposed on the simulated data, but also *how much* different those conditions are. In practical applications, it may be useful for a context-dependent processing strategy to tell when the underlying soil context changes dramatically (such as after a heavy rainfall) rather than subtly (such as a light misting of rain).

2.4 Evaluating GPR Context Features: Field Data Experiment

A second experiment was performed to evaluate the GPR features on field-collected data. The features were used to predict measurements of soil moisture and temperature by a meteorological station at an Eastern U.S. government test site. The following subsections describe the data set used in this experiment, and present the results of RVM regression.

2.4.1 *Field-collected data set*

The GPR data used in this experiment was collected at an temperate Eastern U.S. government test site for a total of 12 days, over 4 campaigns of 2-5 days each between March and August, 2008. The data collection site was comprised of two dirt and three gravel test lanes in which anti-tank landmines were emplaced. As data was collected, the GPR operator maintained an array height of approximately 7-8" above the ground. The GPR made several overlapping passes down each lane, in opposite directions, to ensure than the entire width of the lane was covered. For this experiment, only the first and last passes on each lane from each day are considered to ensure maximum possible change in soil conditions between passes.

A meteorological station was installed at the test site to collect various data regarding air and soil conditions. Figure 2.9 shows a photograph of the meteorological station located between a dirt lane and a gravel lane. The station recorded air temperature, humidity, atmospheric pressure, wind speed, wind direction, precipitation, dirt temperature (at depths of 1/2, 2, 4, and 8 in.), gravel temperature (at depths of 1/2, 2, 4, and 8 in.), soil moisture (at depths of 2, 4, and 8 in.), short-wave radiation (both up and down-welling), and long-wave radiation (both up and down-welling) at 5-minute intervals.

Only measurements of dirt temperature, gravel temperature, and soil moisture



FIGURE 2.9: Photograph of the meteorological station located at the Eastern US test site. The station is located between a dirt and gravel lane, with soil probes embedded in each lane.

were used in this experiment, since the other measurements were determined to be irrelevant or did not show significant variation. The soil measurements were averaged over depth since the only variation with respect to depth appeared to be scaling. The soil measurements were also averaged over each day since accurate timestamp information was not available to cross-register the GPR data with the meteorological data.

2.4.2 Feature Extraction

After the data was collected, a prescreener was run on the raw GPR data to flag locations of detected anomalies, and background B-scans of length 100 were extracted prior to each prescreener alarm. Examples of the background data prior to prescreener alarms are shown in Figure 2.10. The 23-dimensional contextual features were then extracted from the background B-scans. Table 2.3 lists the parameters that were set for performing feature extraction on the field data.

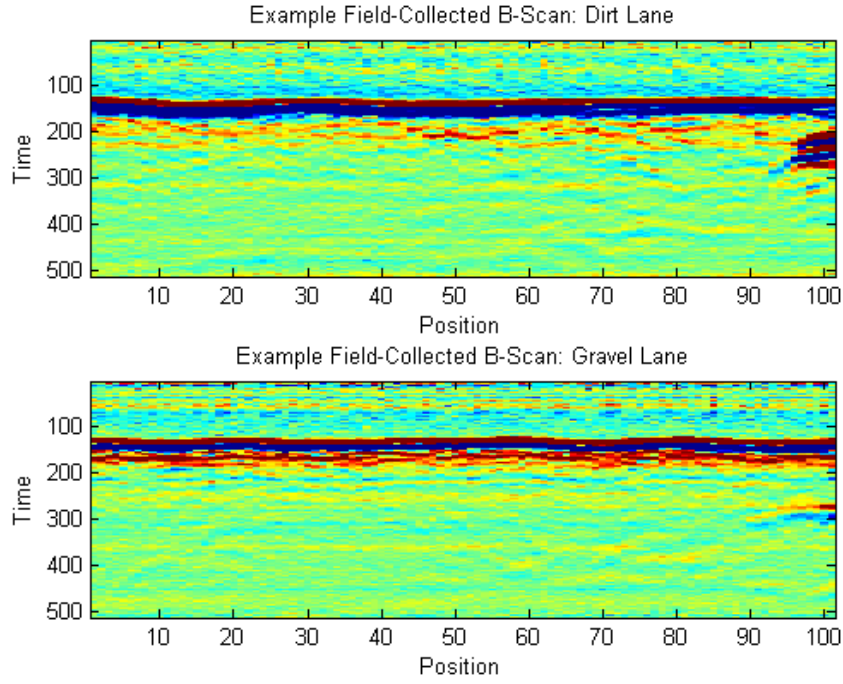


FIGURE 2.10: Example B-scans of field-collected GPR data collected on dirt (top) and gravel (bottom) lanes at an Eastern US test site. The images show background data collected prior to a prescreener alarm. The anomaly that was flagged can be seen at the far right of each image. © 2012 IEEE.

Table 2.3: Feature extraction parameters for field data experiment

Parameter	Description	Value
N	Number of A-scans per B-scan	100
T	1m temporal sampling rate	512
σ_w	Window width parameter	5
δ_0	MP convergence threshold	0.01
ω	Width of MP dictionary elements	25
t'	B-scan row indices	0,30,...,300
K	Linear predictor filter order	4

2.4.3 Correlation Analysis

To assess the efficacy of each individual feature in characterizing the meteorological data, each feature was correlated with the soil measurements and the correlations are plotted in Figure 2.11. The results are quite intuitive; soil moisture is most correlated with the energy, reflection coefficient, and early-time LP power features, since moisture has a great impact on overall soil permittivity. Because the measurements of dirt and gravel temperature were very similar, both measurements are correlated with the late-time MP histogram. If we recall the results of the simulated data experiment, the late-time MP histogram was most correlated with conductivity. A relationship has been shown to exist between soil conductivity and temperature [85], and is probably related to the drying of soils as temperature increases. Therefore, the correlation analysis suggests that the matching pursuits histogram may be indicative of soil temperature as well as conductivity.

2.4.4 Regression Results

As in the simulated data experiment, regression was performed on the kernel-mapped features using the RVM and evaluated via 10-fold cross-validation. Results of using RVM regression to predict the soil measurements from the contextual features are shown in Figure 2.12. Because the measurements of dirt and gravel temperature were similar, the regression performance was also similar, achieving estimation accuracy within 5-6 degrees (the RMSE for dirt temperature was 5.05, and for gravel was 6.05). More importantly, these results illustrate that the RVM is able to distinguish between major differences in temperature. Soil moisture was estimated with relative accuracy for higher values (≥ 0.14), but there appears to be an offset in regression estimates for lower values. Lower values of moisture correspond to lower conductivity, and as was seen in the simulated data experiment, it is difficult to estimate low values of conductivity using these features.

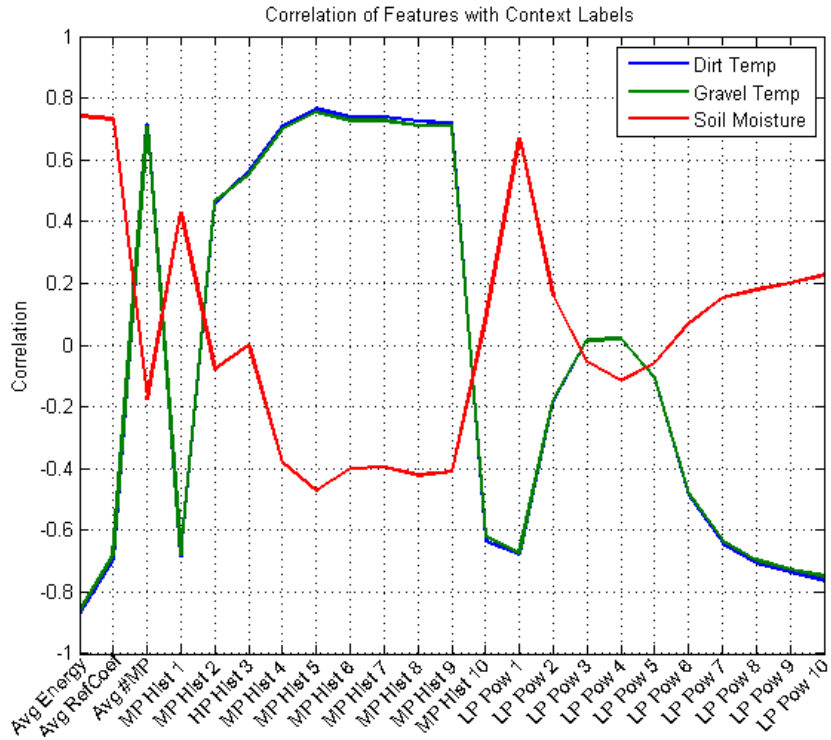


FIGURE 2.11: Plot of correlations between features (horizontal axis) and measured soil properties (line color) for the simulated GPR experiment. © 2012 IEEE.

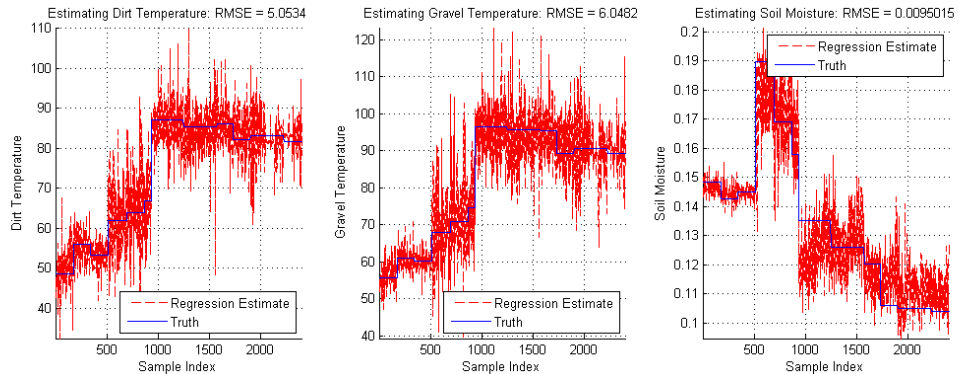


FIGURE 2.12: Results of RVM regression to predict dirt temperature (left), gravel temperature (center), and soil moisture (right) from contextual features extracted from field data. The RMS error is shown at the top of each plot.

2.5 Discussion

Physics-based features were developed to provide a low-dimensional representation of GPR data that may be useful for context learning. To verify the efficacy of the proposed contextual features, experiments were performed using simulated and field-collected GPR data. In these experiments, the proposed features were extracted from a variety of B-scans collected over varying environmental conditions, and a RVM was then applied to the features for predicting the underlying soil properties. In the simulated data experiment, it was shown that several underlying soil parameters were predictable from the features. In the field data experiment, the quantitative estimates of subsurface temperature and moisture were obtained via RVM regression.

Although in both experiments, some soil properties were more accurately predicted than others, in all cases the proposed features were characteristic of major differences in the soil context. In context-dependent learning for detecting buried threats in GPR data, it may be important for the algorithm to tell when such major contextual shifts take place. Given features that are indicative of environmental context, a statistical model could be used to group contextually-similar observations into distinct clusters. Then, a mixture of context-specific classifiers could potentially be learned as an alternative to global classification. The next several chapters will discuss several context modeling techniques that apply statistical mixture models for clustering the features proposed in this chapter into distinct contexts.

Basic Context Learning Techniques

Although many algorithms have been developed to automate buried threat detection in GPR data, past comparisons have shown that certain algorithms perform best under specific environmental conditions [41]. In the previous chapter, it was shown that multiple environmental factors can be characterized from GPR data by using the proposed *contextual features*. This chapter presents two basic techniques for clustering these features into distinct *contexts*, a process referred to as *context learning*, using both supervised and unsupervised techniques. Each of the learned contexts should be representative of a unique set of environmental conditions. If supervised learning is employed, the contexts should correspond to known contextual labels (e.g. soil type). Unsupervised learning, however, may cluster the features in a more informative way. For example, a broad category of observations with the context label “dirt” could potentially be clustered into many sub-contexts using unsupervised learning.

After context learning is performed, unique classifiers may be trained on the data from each context. In this work, an ensemble of relevance vector machines (RVMs) are used to perform *context-dependent algorithm fusion*. Context-dependent learning

allows for the fusion weights of several algorithms to be learned according to their relative performance in different environments. Therefore, it would be expected that context-dependent fusion would yield better overall target discrimination performance than a similar *global fusion* approach, which does not incorporate any contextual information.

The following sections introduce basic techniques which have been proposed in past work for supervised [86–88] and unsupervised [89] approaches to context learning. The RVM is also introduced as a classification model that can be implemented in a context-dependent learning framework. Experimental results using field-collected GPR data are presented to highlight the benefits and disadvantages of supervised and unsupervised context learning, and motivates the use of nonparametric Bayesian methods for achieving additional performance improvements.

3.1 Supervised Context Learning

If contextual ground truth is available for the training data, a supervised approach to context learning may be used. For example, if training data is collected over M several distinct soil types with labels $c = 1, 2, \dots, M$, the individual soil labels could potentially be useful in learning the model parameters. A simple technique for M -ary supervised clustering of the contextual features $\mathbf{X}^{(C)}$ is a Gaussian hypothesis test [86, 87]. Using Bayes’ theorem, posterior inference can be performed by

$$\begin{aligned} p(c_n = m | \mathbf{x}_n^{(C)}) &= \frac{p(\mathbf{x}_n^{(C)} | c_n = m)p(c_n = m)}{\sum_{j=1}^M p(\mathbf{x}_n^{(C)} | c_n = j)p(c_n = j)} \\ &= \frac{\mathcal{N}(\mathbf{x}_n^{(C)} | \hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m)p(c_n = m)}{\sum_{j=1}^M \mathcal{N}(\mathbf{x}_n^{(C)} | \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)p(c_n = j)}, \end{aligned} \tag{3.1}$$

where $\hat{\boldsymbol{\mu}}_m$ and $\hat{\boldsymbol{\Sigma}}_m$ are the maximum-likelihood estimates of the mean and variance of $\mathbf{X}^{(C)}$ conditioned on context m . If a uniform prior is assumed, i.e. $p(c_n = m) = 1/M$

for $m = 1, 2, \dots, M$, (3.1) can be simplified as

$$p(c_n = m | \mathbf{x}_n^{(C)}) = \frac{\mathcal{N}(\mathbf{x}_n^{(C)} | \hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m)}{\sum_{j=1}^M \mathcal{N}(\mathbf{x}_n^{(C)} | \hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)}. \quad (3.2)$$

Often for visualization or interpretation purposes, one may wish to make “hard” classifications of individual observations $\mathbf{x}_n^{(C)}$. In that case, individual points may be assigned to the maximum *a priori* (MAP) class:

$$c_n = \underset{m}{\operatorname{argmax}} p(c_n = m | \mathbf{x}_n^{(C)}) \quad (3.3)$$

If the prior on c is uniform, (3.3) simplifies to

$$c_n = \underset{m}{\operatorname{argmax}} \mathcal{N}(\mathbf{x}_n^{(C)} | \hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m). \quad (3.4)$$

Although supervised context modeling allows for characterizing known context labels from the contextual features, obtaining such labels for training data can be very difficult. There may be little apparent variation between soils over which data was collected, eliminating the possibility of using qualitative labels, or equipment for measuring soil properties could be too expensive or unavailable. If equipment is available, how to properly threshold soil measurements to yield distinct contexts is still an open question. In contrast, unsupervised learning allows for clustering without the need for discrete labels, eliminating many of these potential issues. A basic technique for unsupervised context learning is presented in the following section.

3.2 Unsupervised Context Learning

Several techniques for unsupervised clustering exist for grouping together proximate observations in a multidimensional feature space [70–72]. To draw a parallel to the supervised context learning technique that was described in Section 3.1, consider a Gaussian mixture model (GMM) as an unsupervised context model. Observations

drawn from a GMM come from a weighted sum of M Gaussian densities, each with its own mean $\boldsymbol{\mu}_m$ and covariance $\boldsymbol{\Sigma}_m$, according to the mixture proportions π_m . The likelihood function of the GMM is given by

$$p(\mathbf{x}_n^{(C)}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{m=1}^M \pi_m \mathcal{N}(\mathbf{x}_n^{(C)}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m). \quad (3.5)$$

The parameters of the GMM may be learned several ways. Conventionally, the expectation-maximization (EM) algorithm is used to iteratively maximize the likelihood of the data given the parameters [90]. Alternatively, Variational Bayesian (VB) inference provides an alternative technique for learning the full posterior densities of the model parameters [91]. After estimating the model parameters ($\hat{\pi}_m$, $\hat{\boldsymbol{\mu}}_m$ and $\hat{\boldsymbol{\Sigma}}_m$, for $m = 1, 2, \dots, M$), posterior probabilities of the resulting contexts may be obtained by

$$p(c_n = m|\mathbf{x}_n^{(C)}) = \frac{\hat{\pi}_m \mathcal{N}(\mathbf{x}_n^{(C)}|\hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m)}{\sum_{j=1}^M \hat{\pi}_j \mathcal{N}(\mathbf{x}_n^{(C)}|\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)}. \quad (3.6)$$

3.3 Within-Context Target Classification

After contexts are learned in the contextual space defined by features $\mathbf{X}^{(C)}$, unique classifiers must be learned for each context using the target features $\mathbf{X}^{(T)}$. In this work, the relevance vector machine (RVM) [83, 84] is used as a classification model due to its sparseness properties and probabilistic output. The RVM is a Bayesian solution to inference for the logistic discriminant classifier, given by

$$y_n = \mathbf{w}^T \phi(\mathbf{x}_n^{(T)})^T, \quad (3.7)$$

$$p(t_n|\mathbf{x}_n^{(T)}) = \sigma(y_n)^{t_n} [1 - \sigma(y_n)]^{1-t_n}, \quad (3.8)$$

where $\mathbf{w}$ are the D -dimensional classifier weights, t_n is a binary class label (0, 1) for observation n , $\sigma(\cdot)$ denotes the logistic sigmoid function, and $\phi(\cdot)$ denotes a kernel

transformation. The RVM incorporates sparseness-promoting priors on $\mathbf{w}$, given by

$$w_d \sim \mathcal{N}(0, \alpha_d^{-1}), \quad d = 1, 2, \dots, D, \quad (3.9)$$

$$\alpha_d \sim \text{Gamma}(a_0 = 10^{-6}, b_0 = 10^{-6}). \quad (3.10)$$

Sparseness is promoted in the RVM weights by assuming the weights are statistically independent of one another, and a non-informative Gamma prior is placed on the precisions α governing the weights. By performing Bayesian inference to obtain *a posteriori* estimates of the weights, the values of α tend to infinity for weights corresponding to irrelevant inputs. This yields a posterior infinitely peaked at zero, and the irrelevant dimensions in effectively receive a weight of zero. Those dimensions receiving nonzero weight are referred to as *relevance vectors*. Details regarding RVM inference are provided in Appendix B.

In context-dependent learning, classification is set up as a *mixture* of RVMs. Extending (3.7) and (3.8) to a mixture of classifiers yields

$$y_{nm} = \mathbf{w}_m^T \phi(\mathbf{x}_n^{(T)})^T, \quad m = 1, 2, \dots, M, \quad (3.11)$$

$$p(t_n | c_{nm} = 1, \mathbf{x}_n^{(T)}) = \sigma(y_{nm})^{t_n} [1 - \sigma(y_{nm})]^{1-t_n}, \quad (3.12)$$

where c_{nm} is a binary-coded latent variable that is equal to 1 if the true context of $\mathbf{x}_n^{(C)}$ is context m . The latent variables are inferred from the results of context identification. If a supervised context model is known, c is not random and the mixture of RVMs essentially can be learned as M individual RVMs trained for each of the known contexts. Otherwise, c must be treated probabilistically and be incorporated into the learning of each $\mathbf{w}_m$. Details regarding learning mixtures of RVMs are also included in Appendix B.

An advantage of the RVM over other sparse kernel machines, such as the support vector machine (SVM) [92], is that it yields probabilistic outputs - i.e. posterior

probabilities of t_n . The probabilistic RVM output can be easily used in context-dependent learning by integrating them over uncertainty in the underlying context. The SVM, by contrast, yields distances from the decision boundary that are not easily interpretable in a Bayesian framework. Another advantage of the RVM is that ϕ need not be a kernel function that satisfies Mercer's conditions [70, 83]. Therefore, the *direct kernel*, i.e. $\phi(\mathbf{x}_n^{(T)}) = [\mathbf{1}, \mathbf{x}_n^{(T)}]$ may be used in training the RVM. The effect of using a direct kernel is that irrelevant features will receive zero weight, so training a direct-kernel RVM is therefore a *de facto* method for feature selection. In context-dependent learning, using direct-kernel RVMs provides an intuitive way for performing *context-dependent feature selection*; features that are relevant for classification in a particular context will receive nonzero weight from the classifier trained for that context [86–89].

After training the (supervised or unsupervised) context model on the contextual features $\mathbf{X}^{(C)}$ and the mixture of RVMs on the target features $\mathbf{X}^{(T)}$, the outputs of both must be combined to yield a posterior probability of an observation being a target. This is accomplished by integrating the *within-context target posteriors* obtained from the RVM, $p(t_n | c_{nm} = 1, \mathbf{x}_n^{(T)})$, over the *context posteriors* obtained from the target model, $p(c_{nm} = 1 | \mathbf{x}_n^{(C)})$:

$$p(H_1 | \mathbf{x}_n^{(C)}) = \sum_{m=1}^M p(t_n | c_{nm} = 1, \mathbf{x}_n^{(T)}) p(c_{nm} = 1 | \mathbf{x}_n^{(C)}) \quad (3.13)$$

The resulting posterior probability, $p(H_1 | \mathbf{x}_n^{(C)})$, may then be thresholded for the purpose of making hard decisions. Overall performance may be measured by evaluating the probability of detection (PD) and false alarm rate (FAR) as a function of the decision threshold, and plotting the receiver operating characteristic (ROC) curve.

Table 3.1: Alarm Distribution by Soil Type and Ground Truth

Soil	Clutter (%)	Targets (%)	Total (%)
Dirt	9,356 (72.7%)	933 (54.3%)	10,289 (70.5%)
Gravel	2,658 (20.6%)	393 (22.9%)	3,051 (20.9%)
Asphalt	245 (1.9%)	212 (12.4%)	457 (3.1%)
Concrete	620 (4.8%)	178 (10.4%)	798 (5.5%)
ALL	12,879 (100%)	1,716 (100%)	14,595 (100%)

3.4 GPR Data for Evaluating Landmine/IED Detection Performance

Both techniques for basic context-dependent learning were evaluated on a large set of GPR data collected between 2009-2010 at two different government test sites in the continental U.S. One site was located in an arid region of the Southwestern U.S., and the other site was located in a temperate region of the Eastern U.S.. Data was collected with the NIITEK GPR over prepared dirt, gravel, asphalt, and concrete lanes with emplaced targets and clutter objects. The targets included 10 different types of AT landmines with varied metal content, 155mm artillery shells, and several IED targets consisting of a pressure plate, main charge, and command wire. Several metal and nonmetal clutter objects, including empty holes, were also considered as potential false alarm sources. The GPR made several passes down each test lane to ensure the entire area was covered, yielding a total of 171 target encounters and 524 clutter encounters over a total collection area of 92,340 m².

A derivative of the LMS prescreener [38] was run offline on the GPR data, and detected a total of 14,595 anomalies. These locations are referred to as *alarms*, and are passed to feature-based algorithms for classification as targets or clutter. Table 6.1 illustrates the distribution of prescreener alarms across the four types of lane construction (referred to henceforth as “soils”):

Contextual features were extracted from a 512×100 B-scan from the same channel

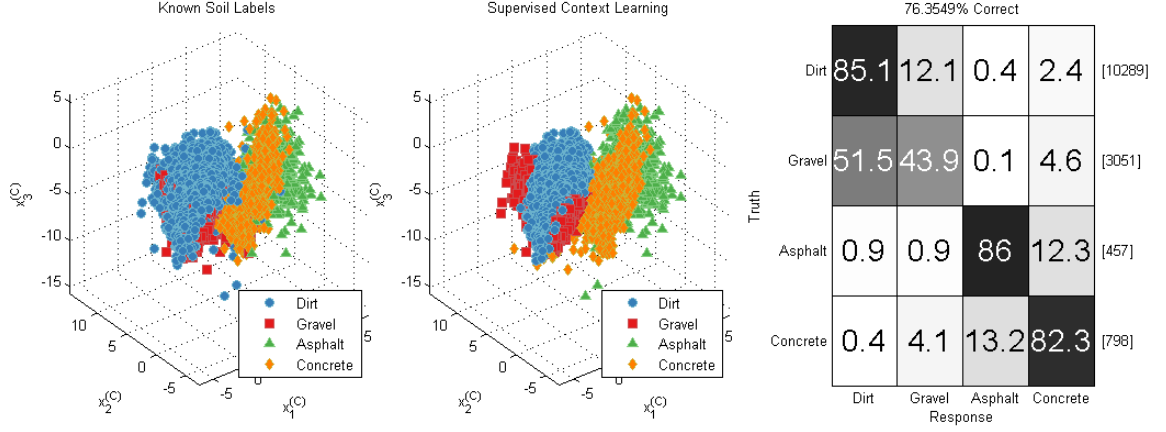


FIGURE 3.1: Left: Scatter plot of 3-D PCA projection of contextual features, with points colored by qualitative soil label. Center: Same scatter plot, but with points colored by MAP supervised context. Right: Confusion matrix illustrating overall performance of supervised context learning, evaluated by 10-fold cross-validation.

of each alarm consisting of the previous background data. Furthermore, the edge histogram descriptor (EHD) [44], the spectral correlation features (SCF) [49], and the hidden Markov model (HMM) [42] algorithms were run on the anomalous responses to yield confidence values for each alarm. In performing algorithm fusion, the target feature vector $\mathbf{x}^{(T)}$ consisted of the prescreener, EHD, SCF, and HMM confidence values. Unless otherwise noted, all evaluations on GPR data that are presented in this dissertation were performed on this data set.

3.5 Experimental Results

3.5.1 Supervised Context Learning

Figure 3.1 illustrates supervised modeling being performed on the 3-D principal components analysis (PCA) projection of the 23-dimensional contextual GPR features discussed in the previous chapter. Because this data was collected over four soil types, one may want to leverage that prior contextual information and train a supervised context model to infer the soil type from the background features.

In Figure 3.1, the leftmost plot illustrates the scatter of the projected features

colored according to the known soil labels. The points tend to cluster according to soil type, which was expected since the previous chapter illustrated their efficacy in characterizing multiple soil properties. The center plot illustrates the supervised classification result of each point, with each point colored according to the MAP class determined by the Gaussian hypothesis test. The results are summarized in the confusion matrix at right. Overall, 76.4% of observations' contexts were identified correctly. However, the misclassifications show some interesting results. Data collected over dirt and gravel are often confused with one another, as are asphalt and concrete. This result suggests a degree of commonality between these pairs of contexts. Additionally, gravel is confused with dirt much more often than dirt is mistaken for gravel. This result, coupled with the fact that there are over three times as many dirt observations than gravel, suggests that perhaps the dirt context could potentially be sub-divided into several smaller sub-contexts with distinct properties.

The advantages and disadvantages of supervised context modeling are clearly illustrated by these results. Depending on the labels being used, supervised learning can be an easy way to verify that the contextual features are indicative of underlying environmental factors. However, this is only true if the labels are relevant. If the labels are irrelevant or redundant, supervised context learning may be forced to differentiate between labeled contexts that have similar or no impact on sensor performance. Conversely, if the labels are too broad, the resulting clusters may not be indicative of underlying contextual factors.

3.5.2 Unsupervised Context Learning

Figure 3.2 illustrates examples of unsupervised context learning performed on the same PCA-projected features as in Figure 3.1. The top two plots illustrate the result of training a 3-component GMM. As shown by the scatterplot at top-left, the GMM converged to one large Gaussian cluster and two smaller ones. Two contexts are

primarily composed of dirt and gravel points, and the third context is spread across all four soil types. In contrast, consider the bottom two plots of Figure 3.2, which consider training an 8-component GMM on the PCA-projected features. In this case, the asphalt and concrete data are assigned to different contexts; concrete data are mostly assigned to Context 2, and asphalt data are mostly assigned to Context 7. The remaining contexts are split between dirt and gravel data.

The differences between the 3-component GMM and the 8-component GMM illustrate that the performance of unsupervised context learning can be substantially affected by the order of the model. Although the results obtained from the various clusterings can be interpreted in a variety of ways, they may not necessarily be indicative of the underlying phenomenology. For example, asphalt and concrete were grouped together by the 3-component GMM and discriminated by the 8-component GMM. Although an argument could be made that both are paved roads, one could also argue that each may constitute a unique propagation environment. Therefore, it is important for the model order to be selected carefully; If M is too small, the model will be too simple and could be *under-trained*, and if M is too large, the model will be too complex and may run the risk of *over-training*. This dilemma is addressed by the nonparametric Bayesian learning techniques that are proposed in the following chapters.

3.5.3 Context-Dependent Fusion Results

The following examples illustrate the results obtained from training RVMs for algorithm fusion using the contextual information obtained through supervised and unsupervised context modeling. In these examples, RVMs were trained on the different confidence values obtained for each alarm that was flagged in the data set. The prescreener, EHD, SPSCF, and HMM algorithms utilize complementary information and it has been shown that algorithm fusion aids in performance [41]. Figure 3.4

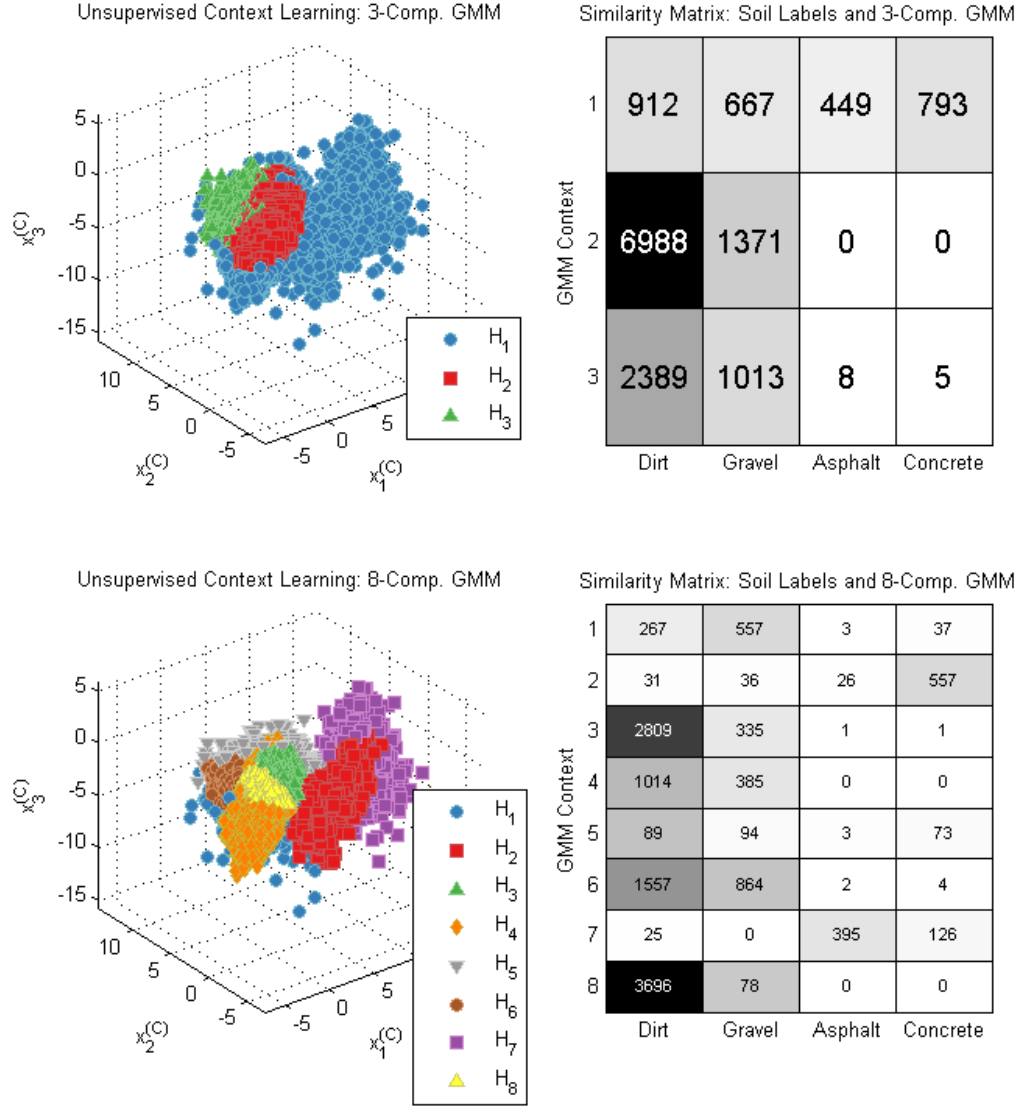


FIGURE 3.2: Top-Left: Scatter plot of 3-D PCA projection of contextual features, with points colored by MAP context determined by a 3-component GMM. Top-Right: Similarity matrix comparing the makeup of the 3 unsupervised contexts to the known soil labels. Bottom-Left: Scatter plot of 3-D PCA projection of contextual features, with points colored by MAP context determined by an 8-component GMM. Bottom-Right: Similarity matrix comparing the makeup of the 8 unsupervised contexts to the known soil labels.

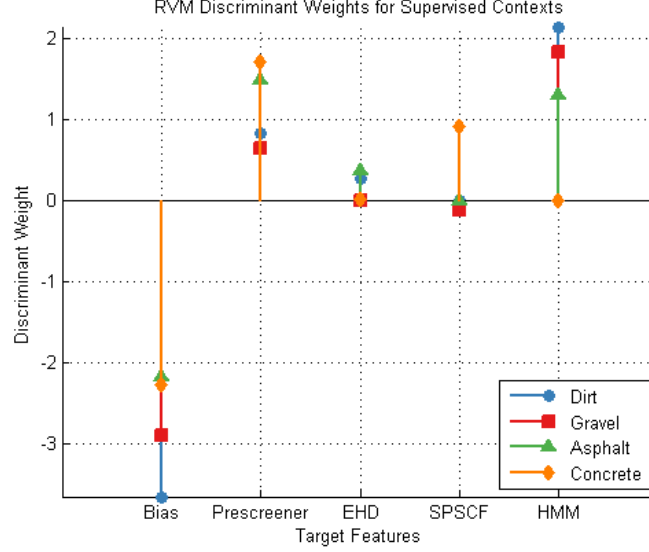


FIGURE 3.3: RVM discriminant weights learned for algorithm fusion in each supervised context. Each stem represents a particular dimension of the target feature space, the vertical axis represents the weight value, and soil contexts are indicated by line color.

illustrates the discriminant weights obtained by the RVMs for algorithm fusion in the labeled dirt, gravel, asphalt, and concrete soil contexts. Each context, illustrated by the different colors of lines, requires a unique weighting of the four algorithms' confidences. This result suggests that the contextual labels are relevant, otherwise the weighting would be the same across all four contexts. Because RVMs are being used to learn the discriminant weights, a unique subset of the algorithms are selected as relevant for each context while irrelevant algorithms are completely ignored. It also appears that other than the prescreener, no one algorithm is universally relevant since each of the three feature-based algorithms receives zero weight in at least one context.

Figure 3.4 shows similar results, but with unsupervised context modeling. The top plot illustrates the RVM weights obtained for each of the 3 unsupervised contexts, and the bottom plot illustrates the weights obtained for 8 contexts. Interpretation of these results can be very difficult and requires experimenting with different orders of

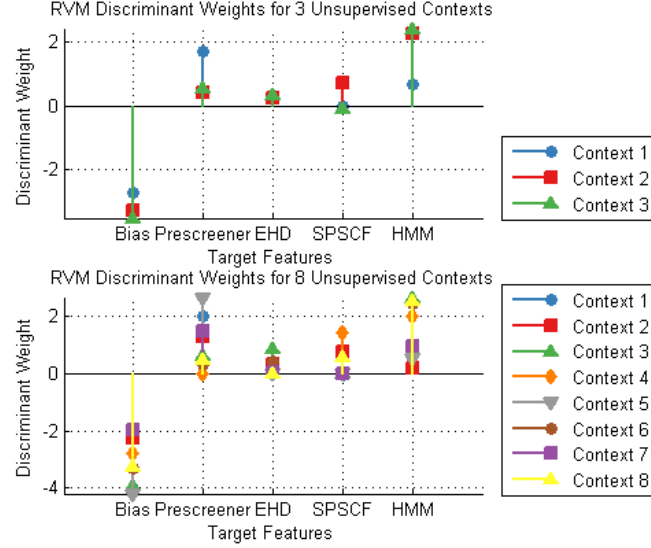


FIGURE 3.4: RVM discriminant weights learned for algorithm fusion in both 3 (top) and 8 (bottom) unsupervised contexts. Each stem represents a particular dimension of the target feature space, the vertical axis represents the weight value, and soil contexts are indicated by line color.

context models and evaluating performance for each case. As shown in the 3-context case, SPSCF is the only algorithm to ever receive zero weight, and does so in 2 of the 3 contexts, while all other algorithms receive nonzero weight in all contexts. In the 8-context case, the prescreener, EHD, and SPSCF are all irrelevant in at least one context each, and the HMM appears to be relevant in all 8 contexts. For both of these cases, the weights are difficult to interpret and performance may be better-evaluated from the ROC curves.

3.5.4 Detection Performance

Classification performance was evaluated using 10-fold cross-validation over emplaced *objects*, rather than alarms, to ensure that training and testing did not occur on different observations of the same object. Multiple alarms on the same object were consolidated in scoring by taking the maximum of all alarm confidences over a single pass registered within a radius of 0.25 m from an object’s center. Scoring was per-

formed using the Mine Detection Algorithm Scoring (MIDAS) tool provided by the Institute for Defense Analyses [93].

Figure 3.5 illustrates the ROC curves for basic context-dependent fusion, using both supervised and unsupervised context models, and compares performance to a globally-implemented RVM that incorporates no contextual information. The FAR for benchmark PDs of 0.85, 0.90, and 0.95 for each algorithm are shown in the legend. The context-dependent techniques, plotted as solid lines, illustrate varying degrees of improvement over the RVM, the performance of which the 90% confidence region is shaded. Somewhat surprisingly, supervised context learning yielded little improvement to performance. The ROC for context-dependent fusion with the supervised context model shows lower FAR than the RVM at low PD levels (< 0.65), but at high PD (> 0.85) the performance is essentially the same. This result suggests that perhaps the soil labels that were used are not reflective of the true contextual factors in this problem.

Meanwhile, unsupervised context learning appears to yield more useful contextual information. However, the degree of improvement is dependent on the order of the context model. If the model order is chosen correctly, significant improvements over the single RVM are possible at high PD. These results suggest that although unsupervised context modeling has the potential to leverage contextual information that is beyond qualitative context labels, performance is highly dependent on the context model order which must be determined experimentally.

3.6 Discussion

In this chapter, basic techniques for context-modeling and context-dependent fusion were introduced. Supervised and unsupervised techniques were proposed for modeling context distributions in the features $\mathbf{X}^{(C)}$. Evaluation of the supervised context model yielded intuitive results, and the interpretation of the unsupervised context

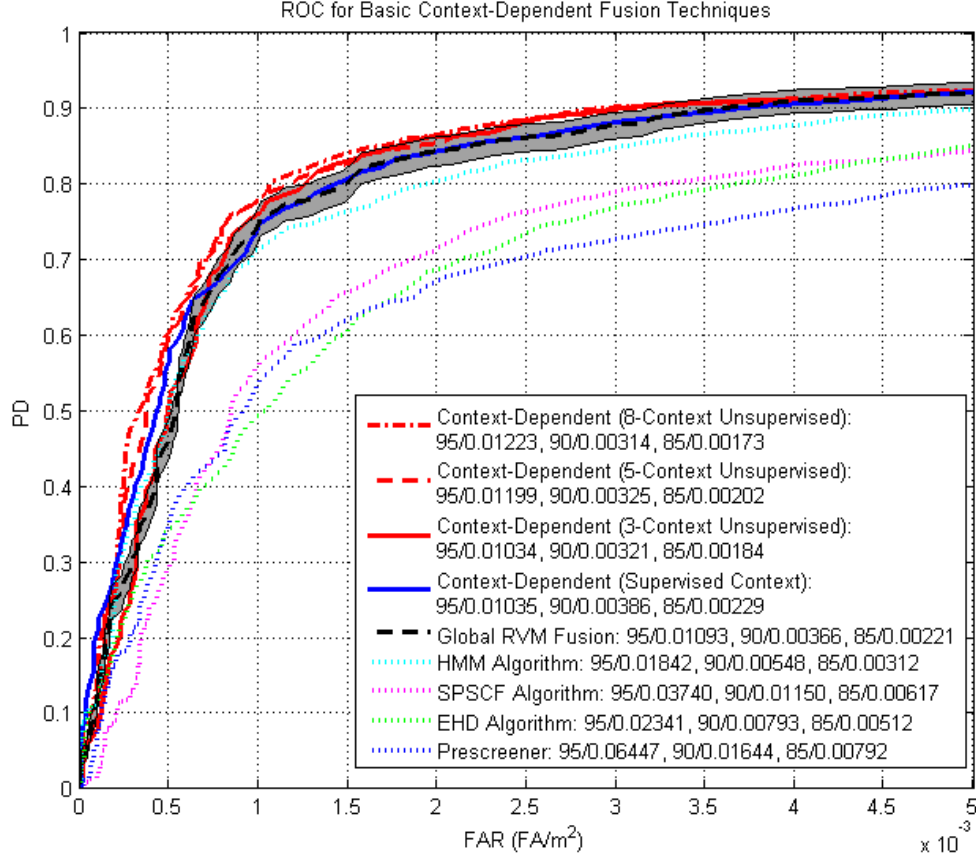


FIGURE 3.5: ROC curves for basic context-dependent fusion techniques, compared to non-context-dependent RVM fusion (black dashed) and the individual fused algorithms (dotted). The ROC consists of PD versus FAR, measured in false alarms per square meter, as a function of decision threshold.

model's behavior was dependent on the model order. Relevance vector machines were also introduced for the purpose of training context-specific classifiers on the target features $\mathbf{X}^{(T)}$. The choice of using a supervised or unsupervised context model appeared to have substantial impact on RVM training and overall model behavior. Finally, performance of context-dependent algorithm fusion was evaluated on a large, geographically-diverse GPR data set consisting of landmine and IED signatures and many false alarms. The potential for context-dependent fusion to improve upon the performance of non-context-dependent RVM fusion was illustrated, although the de-

degree of performance improvement depends on the specific model being used. While unsupervised context modeling led to the greatest performance improvements, the degree of improvement was highly dependent on the model order, i.e. the number of contexts being considered.

These results clearly illustrate that although unsupervised context learning may be advantageous, conventional techniques for clustering that depend on prior knowledge of the model order, M , may be prone to over- or under-training. If M is too small, too few unique contexts will be learned, which results in an under-trained model. If M is too large, too many contexts will be learned, resulting in an over-trained model. Furthermore, parametric models such as GMMs are difficult to implement in high-dimensional spaces due to the oft-cited *curse of dimensionality* [70–72], hence the use of PCA in projecting the 23-D context features to 3-D. Rather than set the order of the context model experimentally by evaluating performance with different numbers of contexts, it may be preferable for an algorithm to learn the optimal number of contexts automatically. Likewise, it may also be preferable to use all of the available contextual features rather than potentially sacrifice information through dimensionality reduction. These items are addressed in remainder of this dissertation, which proposes Bayesian inference for nonparametric context models that facilitate learning of the optimal model order and the discovery of latent features in high-dimensional spaces.

Generative Nonparametric Context Learning

The previous chapter illustrated that unsupervised context learning has potential benefits over supervised learning. Contexts learned from an unsupervised model may be more informative than subjective context labels, which may yield improvements to overall detection performance. However, unsupervised context learning is also a problem of model order selection, which translates to specifying the number of contexts to learn. Learning too many or too few contexts may run the risk of over- or under-training.

An alternative to specifying the model order is to use a *nonparametric* mixture model that facilitates learning an *effective* number of mixture components. In this chapter, two nonparametric context models are proposed. The first model was based on the Dirichlet Process Gaussian Mixture Model (DPGMM), originally published by Blei and Jordan [67]. The DPGMM consists of an infinite-order GMM with a sparseness-promoting Dirichlet process (DP) prior, and is useful for clustering when the number of clusters is unknown but can be learned from the data.

It is possible that some contexts may be characterized by different contextual factors, and some contexts may require more or less information to distinguish them

from others. The second context model proposed in this chapter, the DP Mixture of Factor Analyzers (DPMFA), is motivated by this hypothesis. Like the DPGMM, the DPMFA can be used to learn the number of clusters present in a data set as well as the latent features describing each cluster. The DPMFA model used in this work was originally proposed by Wang et al. [94].

Both nonparametric context models proposed in this chapter were trained using a *generative* learning. In other words, the context models were trained on the contextual features only, without regard to the target features and target/clutter labels for each observation. Both models were learned using variational Bayesian (VB) inference.

The following sections introduce the concept of VB inference, nonparametric models and the DP. Both nonparametric context models are then introduced through synthetic data examples. Finally, experimental results are presented to compare the merits of using these models in context-dependent algorithm fusion for buried threat detection with GPR.

4.1 Bayesian Inference and Variational Learning

4.1.1 Point Estimation of Model Parameters

Robust parameter estimation is particularly important in unsupervised learning, since labels cannot be used to verify the accuracy of the model. As was alluded to with the GMM presented in Section 3.2, conventional parameter estimation techniques yield point estimates of model parameters. The most common method for parameter estimation is maximum-likelihood (ML). The ML estimates, Θ^{ML} , of the model parameters Θ are found by maximizing the likelihood of the training data $\mathbf{X}$, given by:

$$\Theta^{ML} = \underset{\Theta}{\operatorname{argmax}} p(\mathbf{X}|\Theta) \quad (4.1)$$

In some cases, ML estimates can be found analytically (e.g., estimating the mean and variance of a Gaussian distribution) and in other cases estimates must be found iteratively (e.g., the EM algorithm applied to GMMs) [90]. A common criticism of ML parameter estimation is that it is prone to over-fitting [71, 95]. Therefore, an alternative to ML is maximum *a posteriori* (MAP) parameter estimation. MAP estimation is less prone to over-fitting because prior information is used to regularize inference. This can be seen in Bayes' theorem,

$$p(\Theta|\mathbf{X}) = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\Theta)p(\Theta)}{\int_{\Theta} p(\mathbf{X}|\Theta)p(\Theta)d\Theta}, \quad (4.2)$$

where $p(\mathbf{X}|\Theta)$ is referred to as the *likelihood*, $p(\Theta)$ is the *prior*, and $p(\mathbf{X})$ is the *evidence*. The MAP estimate is obtained by maximizing the posterior density, $p(\Theta|\mathbf{X})$:

$$\Theta^{MAP} = \underset{\Theta}{\operatorname{argmax}} p(\Theta|\mathbf{X}) \quad (4.3)$$

However, Θ^{MAP} is still a point estimate, effectively approximating the posterior uncertainty as a Dirac delta function, when a full posterior density may be desired for some applications. Calculating the posterior density involves solving to obtain the functional form of $p(\Theta|\mathbf{X})$. This procedure is known as *Bayesian inference*.

4.1.2 Bayesian Inference

According to (4.2), the only information required to solve for the posterior density are the likelihood and prior densities. The likelihood is obtained from the statistical model chosen for the problem, the prior density expresses uncertainty in the parameters' values, and the evidence is effectively a normalizing constant. In many cases, the evidence integral may be difficult to compute. A common technique for circumventing these potential issues is by assuming a *conjugate prior* distribution [95, 96].

A conjugate prior is defined as a distribution that, when paired with a particular model, yields a posterior distribution with the same functional form. The benefit

of using conjugate priors is that the parameters of the posterior density can be calculated as a function of the prior density's parameters (known as *hyperparameters*) and the data. Since the posterior is known to be of the same form as the observation model, the full posterior density can be determined *exactly* by simply updating its parameters.

For a simple example, consider a Bernoulli process as the observation model. Under this model, $x = (0, 1)$ with $p(x = 1) = \theta$ and $p(x = 0) = 1 - \theta$. The likelihood function of n successes ($x = 1$) and m failures ($x = 0$) is therefore

$$p(n, m|\theta) = \frac{n!}{m!(n-m)!} \theta^n (1-\theta)^m \quad (4.4)$$

For a Bernoulli model, the corresponding conjugate prior is the Beta density with hyperparameters α and β given by

$$p(\theta) = \begin{cases} \frac{(\alpha-1)!}{(\beta-1)!(\alpha+\beta-1)!} \theta^{\alpha-1} (1-\theta)^{\alpha+\beta-1}, & 0 \leq \theta \leq 1 \\ 0, & \text{otherwise.} \end{cases} \quad (4.5)$$

Solving for the posterior density yields

$$\begin{aligned} p(\theta|n, m) &= \frac{p(n, m|\theta)p(\theta)}{\int_{\theta} p(n, m|\theta)p(\theta)d\theta} \\ &= \frac{\frac{n!}{m!(n-m)!} \theta^n (1-\theta)^m \frac{(\alpha-1)!}{(\beta-1)!(\alpha+\beta-1)!} \theta^{\alpha-1} (1-\theta)^{\alpha+\beta-1}}{\int_0^1 \frac{n!}{m!(n-m)!} \theta^n (1-\theta)^m \frac{(\alpha-1)!}{(\beta-1)!(\alpha+\beta-1)!} \theta^{\alpha-1} (1-\theta)^{\alpha+\beta-1} d\theta} \\ &= \frac{\frac{n!(\alpha-1)!}{m!(n-m)!(\beta-1)!(\alpha+\beta-1)!} \theta^{n+\alpha-1} (1-\theta)^{n+\alpha-m-\beta-1}}{\frac{(\alpha-1)!n!(m+\beta-1)!(n+\alpha-m-\beta-1)!}{(\beta-1)!(\alpha+\beta-1)!m!(n-m)!(\alpha+n-1)!} \times \int_0^1 \frac{(n+\alpha-1)!}{(m+\beta-1)!(n+\alpha-m-\beta-1)!} \theta^{n+\alpha-1} (1-\theta)^{n+\alpha-m-\beta-1} d\theta}. \end{aligned} \quad (4.6)$$

Note that the integrand in the denominator of (4.6) is $Beta(n + \alpha, m + \beta)$, and

therefore integrates to 1. Simplifying (4.6) therefore yields

$$\begin{aligned} p(\theta|n, m) &= \frac{(n + \alpha - 1)!}{(m + \beta - 1)!(n + \alpha - m - \beta - 1)!} \theta^{n+\alpha-1} (1 - \theta)^{n+\alpha-m-\beta-1} \\ &= \text{Beta}(n + \alpha, m + \beta). \end{aligned} \tag{4.7}$$

In this example, the posterior is simply another Beta distribution, as was the prior. The posterior parameters are also a function of the hyperparameters. Therefore, rather than solving Bayes' theorem explicitly, the posterior can be expressed in terms of the updated parameters. If Bayes' theorem is being applied sequentially, updating the posterior as more data is observed, the posterior becomes the prior for the next iteration. By exploiting conjugate priors, Bayesian inference can provide a computationally-efficient technique for obtaining a full expression of posterior uncertainty.

Although conjugate priors are chosen to facilitate mathematical tractability, they should still reflect *a priori* information about the data. Fortunately, many conjugate priors offer a wide range of uncertainty expression through various parameter settings, including settings that represent little or no prior information, such as Beta(1,1) or a Gaussian with large variance. It is important in any Bayesian inference problem to use a prior that makes sense for the problem, and choosing parameters that allow for controlled regularization, since in many cases certain parameter settings may yield unexpected over- or under-regularization.

Conjugate priors are useful for determining the full posterior uncertainty in parameters of canonical distributions, but complex models often do not lend to a fully-conjugate solution. Often, these types of models involve latent variables; examples include the GMM and HMM, from which draws are conditioned on a finite mixture of component densities. Although point estimates of model parameters could still be obtained via iterative techniques such as the EM algorithm [70–72, 90], the

same concerns regarding over-fitting discussed previously still apply. Alternatively, one could seek to *approximate* the posterior by making certain assumptions about the *a priori* dependence of the model parameters. One technique for estimating the posterior, which has roots in statistical physics, is variational inference.

4.1.3 Variational Bayesian Inference

Variational (VB) Bayesian inference is a technique for approximate posterior inference in many problems where the evidence integral is intractable [67,68,71,84,91,97–101]. Since the evidence integral cannot be computed directly, variational inference is used to maximize a lower bound on it. Using this approximation for the evidence, a subsequent approximation to the posterior can be calculated and is referred to as the *variational posterior*:

$$q(\mathbf{Z}) = \hat{p}(\mathbf{\Theta}|\mathbf{X}) \quad (4.8)$$

To determine the lower bound on the evidence integral, first rewrite the evidence as

$$\begin{aligned} p(\mathbf{X}) &= \frac{p(\mathbf{X}, \mathbf{\Theta})}{p(\mathbf{\Theta}|\mathbf{X})} \\ &= \frac{p(\mathbf{X}, \mathbf{\Theta})q(\mathbf{\Theta})}{p(\mathbf{\Theta}|\mathbf{X})q(\mathbf{\Theta})}. \end{aligned} \quad (4.9)$$

Taking the logarithm of both sides of (4.9) yields

$$\begin{aligned} \log[p(\mathbf{X})] &= \int_{\mathbf{\Theta}} \log[p(\mathbf{X})] q(\mathbf{\Theta}) d\mathbf{\Theta} \\ &= \int_{\mathbf{Z}} \log \left[\frac{p(\mathbf{X}, \mathbf{\Theta})q(\mathbf{\Theta})}{p(\mathbf{Z}|\mathbf{X})q(\mathbf{\Theta})} \right] q(\mathbf{\Theta}) d\mathbf{\Theta} \\ &= \int_{\mathbf{Z}} \log \left[\frac{p(\mathbf{X}, \mathbf{\Theta})}{q(\mathbf{\Theta})} \right] q(\mathbf{\Theta}) d\mathbf{\Theta} + \int_{\mathbf{Z}} \log \left[\frac{q(\mathbf{\Theta})}{p(\mathbf{\Theta}|\mathbf{X})} \right] q(\mathbf{\Theta}) d\mathbf{\Theta} \\ &= \mathcal{F}[q(\mathbf{\Theta})] + \text{KLD}[q(\mathbf{\Theta})||p(\mathbf{\Theta}|\mathbf{X})] \end{aligned} \quad (4.10)$$

The first term of (4.10) is the negative of what is known in statistical physics as *free energy* [102], and is therefore referred to here as *negative free energy* (NFE) $\mathcal{F}[q(\boldsymbol{\Theta})]$:

$$\mathcal{F}[q(\boldsymbol{\Theta})] = \int_{\boldsymbol{\Theta}} \log \left[\frac{p(\mathbf{X}, \boldsymbol{\Theta})}{q(\boldsymbol{\Theta})} \right] q(\boldsymbol{\Theta}) d\boldsymbol{\Theta}. \quad (4.11)$$

The second term of (4.10) is the Kullback-Leibler divergence (KLD) between the variational posterior, $q(\boldsymbol{\Theta})$, and the true posterior, $p(\boldsymbol{\Theta}|\mathbf{X})$. The KLD is a distance metric between two probability densities, and by definition is always positive. Therefore, rearranging (4.10) illustrates that the NFE is a true lower bound on the log-evidence [99]:

$$\mathcal{F}[q(\boldsymbol{\Theta})] = \log[p(\mathbf{X})] - \text{KLD}[q(\boldsymbol{\Theta})||p(\boldsymbol{\Theta}|\mathbf{X})] \quad (4.12)$$

The NFE thereby serves as the objective function for the VB optimization problem since maximizing $\mathcal{F}[q(\boldsymbol{\Theta})]$ also maximizes the lower bound on the evidence integral. However, the true posterior cannot be calculated explicitly (hence the purpose of VB). A calculable definition of NFE can be obtained by rewriting the numerator of the fraction term in (4.11) as $p(\mathbf{X}|\boldsymbol{\Theta})p(\boldsymbol{\Theta})$. This yields an expression of the NFE as the difference between the expected log-likelihood (with respect to the variational posterior) and the KLD between the variational posterior and the prior:

$$\mathcal{F}[q(\boldsymbol{\Theta})] = \mathbb{E} [\log p(\mathbf{X}|\boldsymbol{\Theta})] - \text{KLD}[q(\boldsymbol{\Theta})||p(\boldsymbol{\Theta})] \quad (4.13)$$

Maximizing the NFE is generally a high-dimensional optimization problem, due to the dimensionality of $\mathbf{X}$ and the number of free parameters. To facilitate the optimization procedure, $q(\boldsymbol{\Theta})$ is generally restricted to a *factorized density* given by

$$q(\boldsymbol{\Theta}) = \prod_i q(\boldsymbol{\theta}_i). \quad (4.14)$$

This approach is referred to as the *mean-field* approximation [98]. The factorized density given by (4.14) restricts inference by implicitly assuming that $\boldsymbol{\Theta}$ can be partitioned into disjoint groups of statistically independent parameters (indexed by i).

Note that there is no restriction on the functional forms of the individual $q(\boldsymbol{\theta}_i)$. The factorized density allows for the optimization of the NFE with respect to one parameter at a time. This can be illustrated by first substituting (4.14) into (4.11). To further consolidate notation, let $q_i = q(\boldsymbol{\theta}_i)$, $q_{-i} = \prod_{j \neq i} q_j$. The following derivation is based on that found in [71]:

$$\begin{aligned}
\mathcal{F}(q) &= \int q(\boldsymbol{\Theta}) \log \left[\frac{p(\mathbf{X}, \boldsymbol{\Theta})}{q(\boldsymbol{\Theta})} \right] d\boldsymbol{\Theta} \\
&= \int \prod_i q_i \left[\log p(\mathbf{X}, \boldsymbol{\Theta}) - \sum_i \log q_i \right] d\boldsymbol{\Theta} \\
&= \int \left[\prod_i q_i \right] \log p(\mathbf{X}, \boldsymbol{\Theta}) d\boldsymbol{\Theta} - \int \left[\prod_i q_i \right] \sum_i \log q_i d\boldsymbol{\Theta} \\
&= \int q_i \left[\int q_{-i} \log p(\mathbf{X}, \boldsymbol{\Theta}) d\boldsymbol{\Theta}_{-i} \right] d\boldsymbol{\Theta}_i - \int q_{-i} \left[\int q_i \log q_i d\boldsymbol{\theta}_i \right] d\boldsymbol{\Theta}_{-i} \\
&\quad - \int q_i \left[\int q_{-i} \log q_{-i} d\boldsymbol{\Theta}_{-i} \right] d\boldsymbol{\Theta}_i \tag{4.15}
\end{aligned}$$

The second and third terms of (4.15) can be consolidated by noting that $\int q_i d\boldsymbol{\theta}_i = \int q_{-i} d\boldsymbol{\Theta}_{-i} = 1$, since q must be a valid PDF. This yields

$$\begin{aligned}
\mathcal{F}(q) &= \int q_i \left[\int q_{-i} \log p(\mathbf{X}, \boldsymbol{\Theta}) d\boldsymbol{\Theta}_{-i} \right] d\boldsymbol{\Theta}_i - \int q_i \log q_i d\boldsymbol{\Theta}_i - \int q_{-i} \log q_{-i} d\boldsymbol{\Theta}_{-i} \\
&= \int q_i \left[\int q_{-i} \log p(\mathbf{X}, \boldsymbol{\Theta}) d\boldsymbol{\Theta}_{-i} \right] d\boldsymbol{\Theta}_i - \int q_i \log q_i d\boldsymbol{\Theta}_i - \mathbb{H}[q_{-i}] \\
&= \int q_i \mathbb{E}_{q_{-i}}[\log p(\mathbf{X}, \boldsymbol{\Theta})] d\boldsymbol{\Theta}_i - \int q_i \log q_i d\boldsymbol{\Theta}_i - \mathbb{H}[q_{-i}] \\
&= \int q_i \log \tilde{p}(\mathbf{X}, \boldsymbol{\Theta}) d\boldsymbol{\Theta}_i - \int q_i \log q_i d\boldsymbol{\Theta}_i, \tag{4.16}
\end{aligned}$$

where $\mathbb{H}[q_{-i}]$ indicates the entropy operator applied to q_{-i} . Here, $\mathbb{E}_{q_{-i}}[\log p(\mathbf{X}, \boldsymbol{\Theta})]$ is defined as the *expected* joint log-density of $\mathbf{X}$ and $\boldsymbol{\Theta}$, with respect to the variational

density q_{-i} . The new density, $\tilde{p}(\mathbf{X}, \boldsymbol{\Theta})$, is given by

$$\log \tilde{p}(\mathbf{X}, \boldsymbol{\Theta}) = \mathbb{E}_{q_{-i}} [\log p(\mathbf{X}, \boldsymbol{\Theta})] - \mathbb{H} [q_{-i}] \quad (4.17)$$

$$\tilde{p}(\mathbf{X}, \boldsymbol{\Theta}) = \frac{\exp\{\mathbb{E}_{q_{-i}} [\log p(\mathbf{X}, \boldsymbol{\Theta})]\}}{\exp\{\mathbb{H} [q_{-i}]\}}. \quad (4.18)$$

One may recognize (4.16) as the negative KLD between q_i and $\tilde{p}(\mathbf{X}, \boldsymbol{\Theta})$. Therefore, $\mathcal{F}(q)$ will be maximized when $KLD[q_i || \tilde{p}(\mathbf{X}, \boldsymbol{\Theta})]$ is minimized, which occurs when the two densities are equal. The variational density of parameter $\boldsymbol{\theta}_i$ that maximizes the NFE is then given by

$$\log q_i = E_{q_{-i}} [\log p(\mathbf{X}, \boldsymbol{\Theta})] - \mathbb{H} [q_{-i}] \quad (4.19)$$

$$q_i = \frac{\exp\{\mathbb{E}_{q_{-i}} [\log p(\mathbf{X}, \boldsymbol{\Theta})]\}}{\exp\{\mathbb{H} [q_{-i}]\}}. \quad (4.20)$$

An interesting observation is that (4.20) is very similar to Bayes' theorem, except that it involves expectations of $p(\mathbf{X}, \boldsymbol{\Theta})$. By using (4.19) and (4.20), an iterative process for optimizing the NFE follows. In a process very similar to the EM algorithm, the variational posterior of each parameter is updated by using expectations computed with respect to the other parameters. Since the bound to log-evidence is convex, each iteration is guaranteed to increase the NFE [98]. When all of the parameters are updated, the NFE may be re-evaluated and updates continue until it converges, defined as increasing less than a predetermined amount.

4.2 Dirichlet Process

The Dirichlet process (DP) is a common choice of prior density for nonparametric mixture models that facilitates learning of latent variables [103]. The DP has been described as a distribution of distributions that is governed by the scaling parameter α and a base distribution G_0 :

$$G \sim \mathcal{DP} (G_0, \alpha) \quad (4.21)$$

Random draws $(\theta_n, n = 1, 2, \dots)$ from G exhibit clustering properties described by a Pólya urn scheme [104], which implies that some of the θ_i will have identical values represented by θ_m^* , $m = 1, 2, \dots$. This process is typically referred to as a *Chinese restaurant process*, which is described as follows: Customer x_n walks into a restaurant in which there are an infinite number of tables, and the customer must choose a table denoted as θ_n . The probability that a customer chooses to sit at a particular table is as follows:

$$p(\theta_n | \theta_1, \theta_2, \dots, \theta_{n-1}) = \begin{cases} \theta_m^* & \text{with probability } \frac{\text{num}_{n-1}(\theta_m^*)}{n-1+\alpha} \\ \text{New draw from } G & \text{with probability } \frac{\alpha}{n-1+\alpha} \end{cases} \quad (4.22)$$

In (4.22), $\text{num}_{n-1}(\theta_m^*)$ denotes the number of people who are already sitting at table θ_m^* , and α is the DP scaling parameter. The processes described by (4.22) suggests that as the restaurant fills up with people, new customers will be more likely to select tables at which a large number of people are sitting. However, there is a probability (which is a function of α) that a new customer will select an unoccupied table. From the Chinese restaurant process, G can be described as a discrete probability density that assigns mass to an infinite number of *atoms*,

$$G = \sum_{m=1}^{\infty} \pi_m \delta_{\theta_m^*}, \quad (4.23)$$

where the atoms are delta functions located at each θ_m^* . The mixing proportions given by π_m can be estimated by sampling from G and calculating the proportions of customers seated at each table.

Another construction of the DP that constrains the proportions to sum to unity is the *stick-breaking process* [105]. In a stick-breaking construction, the values of π_m can be expressed as the relative proportions of an infinite number of random pieces

sequentially broken off a unit-length stick:

$$\pi_m(\mathbf{v}) = v_m \prod_{j=1}^{m-1} (1 - v_j), \quad m = 1, 2, \dots, \infty \quad (4.24)$$

$$v_m \sim \text{Beta}(1, \alpha) \quad (4.25)$$

The sizes of the individual pieces, v_m , that are broken off the remainder of the stick are drawn from a Beta distribution controlled by the α parameter. Similar to the Chinese restaurant process, the stick-breaking process yields a G consisting of a countably infinite set of atoms, for which the vast majority have negligible proportion:

$$G = \sum_{m=1}^{\infty} \pi_m(\mathbf{v}) \delta_{\theta_m^*} \quad (4.26)$$

However, in this case, the values of π_m are a function of $\mathbf{v}$ according to (4.24).

The DP has been shown to be useful as a prior density in *nonparametric mixture models*, and the stick-breaking process is particularly amenable to variational learning since it can be incorporated into a fully-conjugate graphical model [67, 69, 100, 101]. Nonparametric models differ from parametric models not in that they have no parameters, but in that the number of unique parameters (i.e., the effective model order) controls model complexity rather than just the shape of the PDF. A DP prior is a useful mechanism for regulating the number of parameters, thereby effectively determining the model order and avoiding overfitting. These approaches are generally referred to as *Dirichlet process mixtures*, in which G is a conjugate prior density for an infinite number of mixture components. Therefore, the unique draws θ_m^* are the parameters that govern the m th component, for $m = 1, 2, \dots, \infty$.

In the following sections, two types of DP mixtures are presented to automate learning of an unsupervised context model of unknown order. First is the DP Gaussian mixture model (DPGMM), which facilitates learning the number of GMM com-

ponents. The second model is the DP mixture of factor analyzers that, like the DPGMM, will automate learning of the number of clusters as well as a locally-reduced dimensionality for each cluster.

4.3 Dirichlet Process Gaussian Mixture Model

The GMM was shown in Chapter 3 as an example of a basic unsupervised context model that clusters the contextual features $\mathbf{X}^{(C)}$ into M contexts, with each context represented by a single Gaussian mixture component. However, the behavior and overall benefit of using a fixed-order GMM depends on whether the model order (i.e., the number of contexts) was set correctly [89]. The DPGMM improves upon the fixed-order GMM by allowing the effective number of mixture components to be learned from the data by performing Bayesian inference [67]. The likelihood function of the DPGMM context model is given by

$$p(\mathbf{x}^{(C)}|\mathbf{v}, \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) = \sum_{m=1}^{\infty} \pi_m(\mathbf{v}) \mathcal{N}(\mathbf{x}^{(C)}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m^{-1}), \quad (4.27)$$

where $\boldsymbol{\mu}_m$ are the component means, $\boldsymbol{\Lambda}_m$ are the component precision (inverse covariance) matrices, and $\pi_m(\mathbf{v})$ are the mixing proportions drawn from the stick-breaking process given by (4.24). VB inference can be performed on this model by assuming conjugate priors on all of the model parameters, as well as the hyperparameter, α , controlling the stick-breaking process. The data-generating process for a fully-conjugate DPGMM is as follows:

1. Draw $\alpha \sim \text{Gamma}(\tau_{10}, \tau_{20})$
2. Draw $v_m|\alpha \sim \text{Beta}(1, \alpha)$
3. Draw $\boldsymbol{\theta}_m^*|G_0 \sim \mathcal{N}(\boldsymbol{\mu}_m^*|\boldsymbol{\rho}_0, u_0^{-1}\boldsymbol{\Lambda}_m^{*-1}) \mathcal{W}(\boldsymbol{\Lambda}_m^*|\mathbf{B}_0, \nu_0), m = 1, 2, \dots$
4. Calculate mixture proportions $\pi_m(\mathbf{v}) = v_m \prod_{j=1}^{m-1} (1 - v_j), m = 1, 2, \dots$

5. For $n = 1, 2, \dots, N$

(a) Draw indicator variable $\mathbf{c}_n | \mathbf{v} \sim \text{Multinomial}(\boldsymbol{\pi})$

(b) Draw data $(\mathbf{x}_n^{(C)} | c_{nm} = 1) \sim \mathcal{N}(\mathbf{x}_n^{(C)} | \boldsymbol{\theta}_m^*)$

In practice, the DPGMM is initialized with T clusters, where T is an arbitrarily large number, using the k -means algorithm. In the experiments presented in this chapter, the following hyperparameter settings were used: $u_0 = 1$, $\tau_{10} = \tau_{20} = 1$, $\nu_0 = D^{(C)}$, $\mathbf{B}_0 = D^{(C)} \mathbf{I}_{D^{(C)}}$, and $\boldsymbol{\rho}_0$ was set equal to the sample mean of $\mathbf{X}^{(C)}$. These hyperparameter settings were not optimized for any particular problem, as the DPGMM did not appear to be very sensitive to their settings for the problems that were considered. Details on variational inference for the DPGMM, including derivations of all posterior update equations and the negative free energy, are included in Appendix C.

The stick-breaking prior imposes a clustering effect on the parameters of each cluster that consolidates them to a few unique values. For the purpose of context-dependent learning, a pruning criterion was imposed to ensure that all contexts contained enough points for performing classification. Therefore, all clusters accounting for less than 1% of points were pruned from the model to yield M clusters such that $M \ll T$. The following variational posteriors were obtained for the model parameters:

$$q(\boldsymbol{\mu}_m, \boldsymbol{\Gamma}_m) = \mathcal{N}(\boldsymbol{\mu}_m | \boldsymbol{\rho}_m, u_m^{-1} \boldsymbol{\Lambda}_m^{-1}) \mathcal{W}(\boldsymbol{\Lambda} | \nu_m, \mathbf{B}_m), m = 1, 2, \dots, T \quad (4.28)$$

$$q(\mathbf{c}_n) = \text{Multinomial}(\boldsymbol{\phi}_n) \quad (4.29)$$

For new (test) values of $\mathbf{x}^{(C)}$, context posteriors are obtained by integrating out the model parameters to yield an *a posteriori* mixture of Student's t -distributions

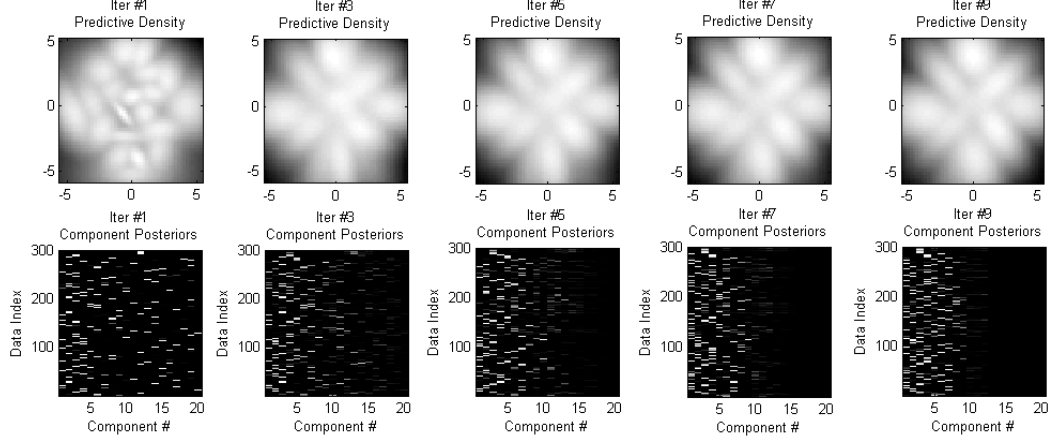


FIGURE 4.1: Example of the DPGMM learned on a mixture of 9 Gaussian distributions. The top row illustrates the predictive density, and the bottom row illustrates the component membership matrix, at learning iterations 1-9.

given by

$$p(c_{nm} = 1 | \mathbf{x}_n^{(C)}) = \frac{t_{\omega_m}(\mathbf{x}_n^{(C)} | \boldsymbol{\rho}_m, \mathbf{W}_m)}{\sum_{j=1}^M t_{\omega_j}(\mathbf{x}_n^{(C)} | \boldsymbol{\rho}_j, \mathbf{W}_j)}, \quad (4.30)$$

where context m is represented by a Student's t -distribution with $\omega_m = \nu_m + 1 - D^{(C)}$ degrees of freedom, mean $\boldsymbol{\rho}_m$, and covariance $\mathbf{W}_m = [(u_m + 1) / u_m \omega_m] \mathbf{B}_m^{-1}$ [91].

Figure 4.1 illustrates an example of the DPGMM being trained on a mixture of 9 bivariate Gaussian distributions arranged in a diamond shape. The top row illustrates the predictive density obtained by integrating over the model parameters at VB iterations 1, 3, 5, 7, and 9. The bottom row illustrates the membership matrix for each of 300 training points. Variational inference was initialized with $T = 20$ mixture components, so the membership matrix is initially distributed evenly between the 20 columns. As the number of iterations increases, the memberships consolidate to 9 columns. Furthermore, the predictive density converges to a mixture of the 9 true densities from which the training data was drawn.

In many model parameter estimation problems, it is difficult to perform inference reliable with high-dimensional data. This is referred to as the *curse of dimensionality* in most statistics texts [70–72], and the problem manifests itself in the DPGMM. Each covariance matrix requires the estimation of $D^{(C)}(D^{(C)} + 1)/2$ unique parameters, where $D^{(C)}$ is the dimensionality of the context feature space, and this could be very expensive computationally if $D^{(C)}$ is large. Furthermore, the number of samples N must be much greater than $D^{(C)}$ in order to avoid over-fitting, which becomes more difficult to achieve if $D^{(C)}$ is large. Therefore, the DPGMM was trained on the 3-D PCA projection of the contextual features for this work.

It is possible that the various contexts over which data was collected may be characterized by different contextual factors, suggesting that a unique number of features might characterize each context. Therefore, another nonparametric context model is proposed in the following section for learning a low-dimensional projection of each cluster. This model, the Dirichlet process mixture of factor analyzers (DPMFA) can potentially avoid the curse of dimensionality without having to specify the number of latent feature dimensions.

4.4 Dirichlet Process Mixture of Factor Analyzers

Recall Chapter 2 in which a variety of contextual features were proposed for characterizing multiple environmental factors from time-domain GPR data. It is possible that different environmental factors, and therefore features, may characterize the various contexts over which data was collected. For example, distinguishing between a dirt road and a concrete road may only need to be based on one factor, the soil dielectric constant. However, some concrete roads may be reinforced with rebar; therefore, subsurface heterogeneity may need to also be considered for distinguishing different types of concrete roads from a dirt road. The DPGMM assumes that all of the learned contexts have the same dimensionality, and therefore utilize the same

contextual information. In contrast, it could possibly be beneficial to use a context model that not only facilitates learning the *number* of contexts, but also the *local dimensionality* of each context.

One technique for dimensionality reduction that has been given a Bayesian treatment, and therefore is easily implemented in the proposed context-dependent learning framework, is factor analysis [71,98]. Closely related to PCA, a factor analysis model expresses the data $\mathbf{x}$ as a projection of K D -dimensional latent *factors*, $\mathbf{A}$, onto the $K \times 1$ *scores*, $\mathbf{s}$, biased by $D \times 1$ mean, $\boldsymbol{\mu}$. The projection error is assumed to be Gaussian with covariance matrix $\boldsymbol{\psi}^{-1}\mathbf{I}$, so that

$$p(\mathbf{x}|\mathbf{A}, \mathbf{s}, \boldsymbol{\mu}) = \mathcal{N}(\mathbf{x}|\mathbf{A}\mathbf{s} + \boldsymbol{\mu}, \boldsymbol{\psi}^{-1}\mathbf{I}). \quad (4.31)$$

Equation (4.31) is the same distribution assumed for the projection error of PCA, with the only difference is that in factor analysis the covariance $\boldsymbol{\psi}^{-1}\mathbf{I}$ is assumed to be diagonal rather than isotropic.

A *mixture* of factor analyzers (MFA) is similar to a GMM, except that each component is described by a local variant of (4.31):

$$p(\mathbf{x}|\mathbf{A}, \mathbf{s}, \boldsymbol{\mu}) = \sum_{m=1}^M \pi_m \mathcal{N}(\mathbf{x}|\mathbf{A}_m \mathbf{s} + \boldsymbol{\mu}_m, \boldsymbol{\psi}_m^{-1}\mathbf{I}) \quad (4.32)$$

A VB inference approach to the MFA was proposed in [68], and like the original VBGMM [91] assumed a Dirichlet prior on the mixture proportions. Furthermore, the MFA assigns an independent loading matrix $\mathbf{A}_m$ to each mixture component, for which learning may be difficult on a small data set or if outliers are present.

An more feasible approach is to impose a binary-coded $\mathbf{z}_m$ on each mixture component to select vectors from a common loading matrix [94]:

$$p(\mathbf{x}|\mathbf{A}, \mathbf{s}, \boldsymbol{\mu}) = \sum_{m=1}^M \pi_m \mathcal{N}(\mathbf{x}|\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s} + \boldsymbol{\mu}_m, \boldsymbol{\psi}_m^{-1}\mathbf{I}) \quad (4.33)$$

Although performing Bayesian inference on the MFA will yield a full posterior density for each of the M mixture components, the effective number of mixture components could only be found by using *a posteriori* point estimates of $\boldsymbol{\pi}$ and applying a threshold. Instead, a stick-breaking prior may be assumed for the mixing proportions $\boldsymbol{\pi}$, yielding a Dirichlet process mixture of factor analyzers (DPMFA):

$$p(\mathbf{x}) = \sum_{m=1}^M \pi_m(\mathbf{v}) \mathcal{N}(\mathbf{x} | \mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s} + \boldsymbol{\mu}_m, \boldsymbol{\psi}_m^{-1} \mathbf{I}) \quad (4.34)$$

Originally proposed as part of graphical model for classifying missing data [94], the DPMFA is used in this work for generative context modeling in the feature space $\mathbf{X}^{(C)}$. The stick-breaking prior on $\pi(\mathbf{v})$, given by (4.24), will impose a pruning effect on extraneous mixture components. This forces the corresponding elements of $\boldsymbol{\pi}$ to zero. A Bernoulli prior is also placed on the elements of $\mathbf{z}_m$ to automate factor selection for each local cluster. The data-generating process for a fully-conjugate DPMFA context model is as follows:

1. Draw $\alpha \sim \text{Gamma}(\tau_{10}, \tau_{20})$
2. Draw $v_m | \alpha \sim \text{Beta}(1, \alpha)$, $m = 1, 2, \dots$
3. Calculate mixture proportions $\pi_m(\mathbf{v}) = v_m \prod_{l=1}^{m-1} (1 - v_l)$, $m = 1, 2, \dots$
4. Draw $\gamma_{dk} \sim \text{gamma}(e_0, f_0)$, $d = 1, 2, \dots, D^{(C)}$, $k = 1, 2, \dots, K$
5. Draw $A_{dk} \sim \mathcal{N}(A_{dk} | 0, \gamma_{dk}^{-1})$, $d = 1, 2, \dots, D^{(C)}$, $k = 1, 2, \dots, K$
6. Draw $\zeta_{mk} \sim \text{Beta}(a_0/K, b_0(K-1)/K)$, $k = 1, 2, \dots, K$, $m = 1, 2, \dots$
7. Draw $z_{mk} \sim \text{Bernoulli}(z_{mk} | \zeta_{mk})$, $k = 1, 2, \dots, K$, $m = 1, 2, \dots$
8. Draw $\psi_{mk} \sim \text{Gamma}(\psi_{mj} | g_0, h_0)$, $k = 1, 2, \dots, K$, $m = 1, 2, \dots$

9. Draw $\boldsymbol{\mu}_m \sim \mathcal{N}(\boldsymbol{\mu}_m | \boldsymbol{\rho}_0, u_0^{-1} \text{diag}(\boldsymbol{\psi}_m^{-1}))$, $m = 1, 2, \dots$
10. Draw $\delta \sim \text{Gamma}(\delta_{10}, \delta_{20})$
11. For $n = 1, 2, \dots, N$
 - (a) Draw $\mathbf{s}_n \sim \mathcal{N}(\mathbf{s}_n | 0, \delta^{-1} \mathbf{I})$
 - (b) Draw indicator variable $\mathbf{c}_n | \mathbf{v} \sim \text{Multi}(\boldsymbol{\pi})$
 - (c) Draw data $(\mathbf{x}_n^{(C)} | c_{nm} = 1) \sim \mathcal{N}(\mathbf{x}_n^{(C)} | \mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m, \boldsymbol{\psi}_m^{-1} \mathbf{I})$

The DPMFA model was initialized with $T = 20$ clusters, using the k -means algorithm. Furthermore, the number of factors was capped at $K = 10$. In the experiments presented in this chapter, the following hyperparameter settings were used: $a_0 = 1$, $b_0 = 0.5$, $e_0 = g_0 = \delta_{10} = 0.1$, $f_0 = h_0 = \delta_{20} = 10$, and $\tau_{10} = \tau_{20} = 1$. These hyperparameter settings were chosen to limit sparseness in the factor loadings and scores, and allow selection to be governed by inference of $\mathbf{z}$. However, the values were not specifically optimized for any particular problem and were used for experiments with synthetic as well as real data. Variational inference yields the following variational posteriors on the model parameters:

$$q(A_{dk}) = \mathcal{N}(A_{dk} | \omega_{dk}, \sigma_{dk}) \quad (4.35)$$

$$q(\mathbf{s}_n) = \mathcal{N}_K(\mathbf{s}_n | \boldsymbol{\xi}_n, \boldsymbol{\Lambda}_n) \quad (4.36)$$

$$q(z_{mk}) = \text{Bernoulli}(\eta_{mk}) \quad (4.37)$$

$$q(\boldsymbol{\mu}_m) = \mathcal{N}_{D^{(C)}}(\boldsymbol{\mu}_m, \mathbf{U}_m) \quad (4.38)$$

$$q(\psi_{tj}) = \text{Gamma}(g_{tj}, h_{tj}) \quad (4.39)$$

$$q(\mathbf{c}_n) = \text{Multinomial}(\boldsymbol{\phi}_n) \quad (4.40)$$

In practice, the variational expectations of the factor loadings (ω_{dk}), scores ($\boldsymbol{\xi}_n$), selectors (η_{mk}), as well as the mixture component means ($\boldsymbol{\rho}_m$), variances (ψ_{tj}), and

memberships (ϕ_n) were used as the learned model parameters. Additionally, a pruning criterion was imposed on the mixture components. All components accounting for less than 1% of points were pruned from the model to yield M clusters such that $M \ll T$.

Figure 4.2 presents an example of a factor analysis model to highlight the behavior and performance of the DPMFA model. In this example, data was generated from a known factor loading matrix, mixture component means, and selection vectors while using random scores. The factor loading matrix was specified by having pairs of features share a single factor loading. These shared elements of $\mathbf{A}$ were set to one while the remaining elements were set to zero. The factor scores, $\mathbf{S}$, were randomly drawn from a zero-mean, unit-variance Gaussian distribution. The data was partitioned into three clusters, and unique factor selection vectors were specified for each. The first cluster (samples 1-500) utilized three factors, the second cluster (501-1000) utilized two factors that were distinct from those in the first cluster, and the third cluster (1001-1500) also utilized two factors, each shared with one of the two other clusters. Furthermore, the first cluster was biased by a mean value of 5, the second cluster was biased by a mean value of -5, and the third cluster remained zero-mean. White noise with a variance of 0.5 was then added to the data.

In this example, the model converged to a solution within 11 VB iterations, and yielded the expectations to the model parameters shown in Figure 4.3. Although the learned loading matrix does not match the structure of the true $\mathbf{A}$ from Figure 4.2, the factors are sparse and shared by two features at a time. This discrete selection of factors is summarized by the bottom-left image, which illustrates the learned factor selection vectors. Finally, the clustering results are summarized by the expected memberships at bottom-right, which are illustrated by the probabilities Φ . Clearly, three distinct clusters have been learned. The means show that the correct cluster locations were learned (having means of 5, -5, and 0), and the variances are close to

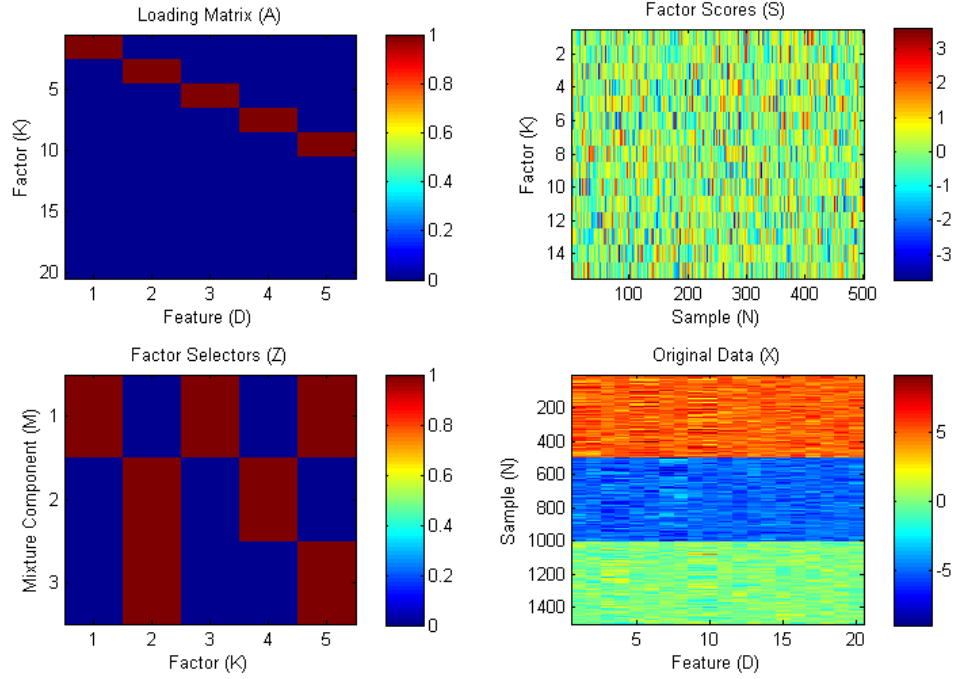


FIGURE 4.2: An example factor analysis problem to illustrate DPMFA model performance. Top-left: true factor loading matrix ($\mathbf{A}$); Top-right: factor scores ($\mathbf{S}$), Bottom-left: factor selectors ($\mathbf{Z}$), Bottom-right: Original data ($\mathbf{X}$).

the true variance of 0.5.

Latent feature models like the DPMFA can also be thought of as a technique for signal/image denoising. By learning the latent factors present in multi-dimensional data, the most informative parts of the data are retained. Considering this synthetic example, the data matrix $\mathbf{X}$ can be thought of as an $N \times D$ image, in which the features corresponding to the shared factors are the informative parts. Reproducing $\mathbf{X}$ by substituting the posterior expectations of the model parameters into 4.34 yields an image similar to the original data, but with the noise removed and the informative features retained.

As shown by the example, the DPMFA performs joint *clustering* and *feature selection* in an unsupervised manner. In context modeling, this is important because certain contexts may be explained by different environmental factors. For example,

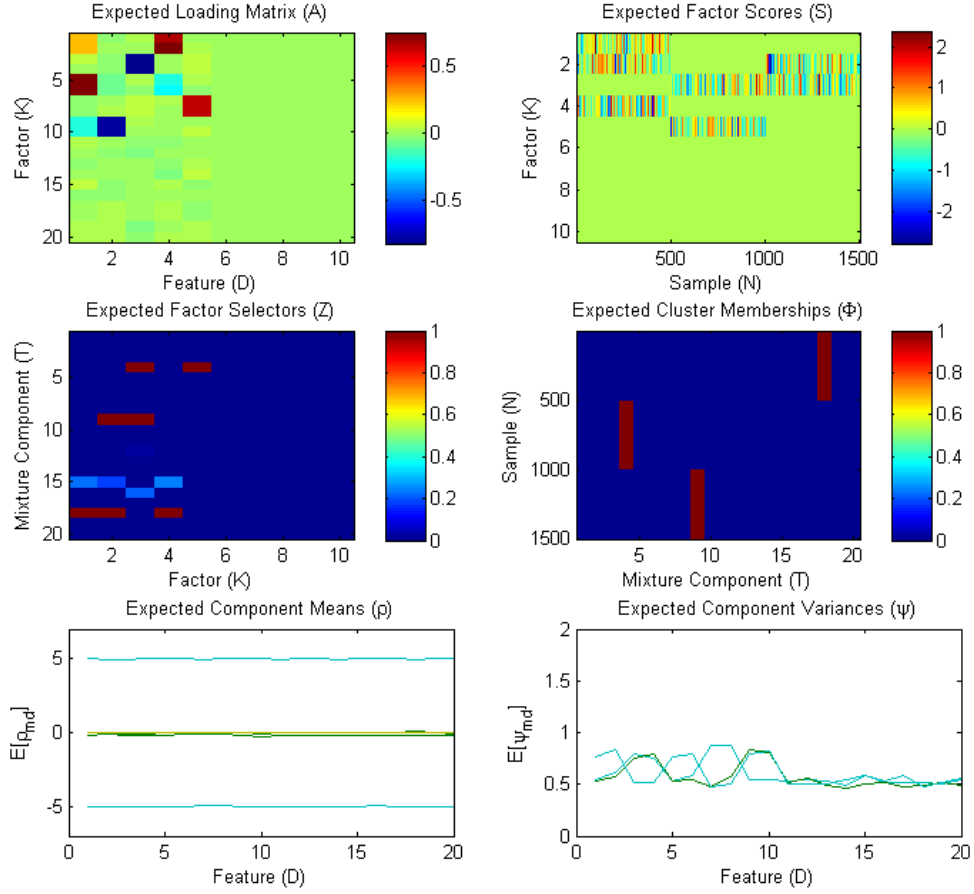


FIGURE 4.3: *A posteriori* expected values of the DPMFA model parameters learned from the example data shown in Figure 4.2. Top-left: learned loading matrix ($\mathbf{A}$); Top-right: learned factor scores ($\mathbf{S}$); Center-left: learned factor selectors ($\mathbf{Z}$); Center-right: learned cluster memberships (ϕ); Bottom-left: learned component means (ρ); Bottom-right: learned component variances (ψ)

distinguishing between GPR data collected on a homogeneous dirt lane, but in differing moisture conditions, may only be dependent on one feature (e.g., soil dielectric constant). However, distinguishing these contexts from paved or heavily-cluttered soils may require additional information (e.g. subsurface heterogeneity). Given a large number of contextual features for characterizing multiple environmental factors, the DPMFA is useful because it automates learning of the number of contexts

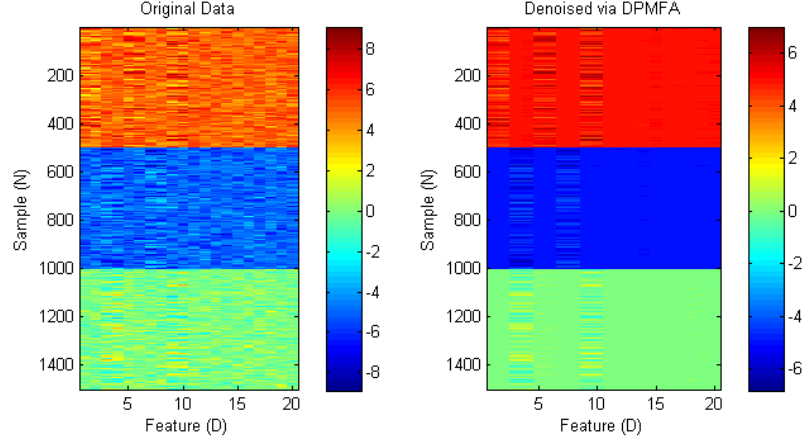


FIGURE 4.4: Results of denoising the example data shown in Figure 4.2 with DPMFA. Left: original data, shown in the bottom-right of Figure 4.2; Right: denoised data, calculated from learned DPMFA model parameters.

and also the local dimensionality of each.

4.5 Experimental Results

Preliminary results of using the DPGMM in context-dependent algorithm fusion were presented in [79] using a smaller data set considering only antitank landmine targets. In this section, experimental results are presented for using the DPGMM and DPMFA in context-dependent algorithm fusion on the data set that was summarized in Section 3.4. First, the results of context learning are analyzed by comparing the contexts learned by the DPGMM and DPMFA to the known labels. Then, the RVM weights learned for performing context-dependent algorithm fusion using either DPGMM or DPMFA contexts are compared. Finally, the detection performance of context-dependent algorithm fusion using both context models are compared to the basic approaches originally shown in Figure 3.5.

4.5.1 Context Learning with the DPGMM

The DPGMM was trained on the 3-D PCA projection of the normalized GPR context features. Initialization was set to $T = 30$ clusters using the k -means algorithm. The

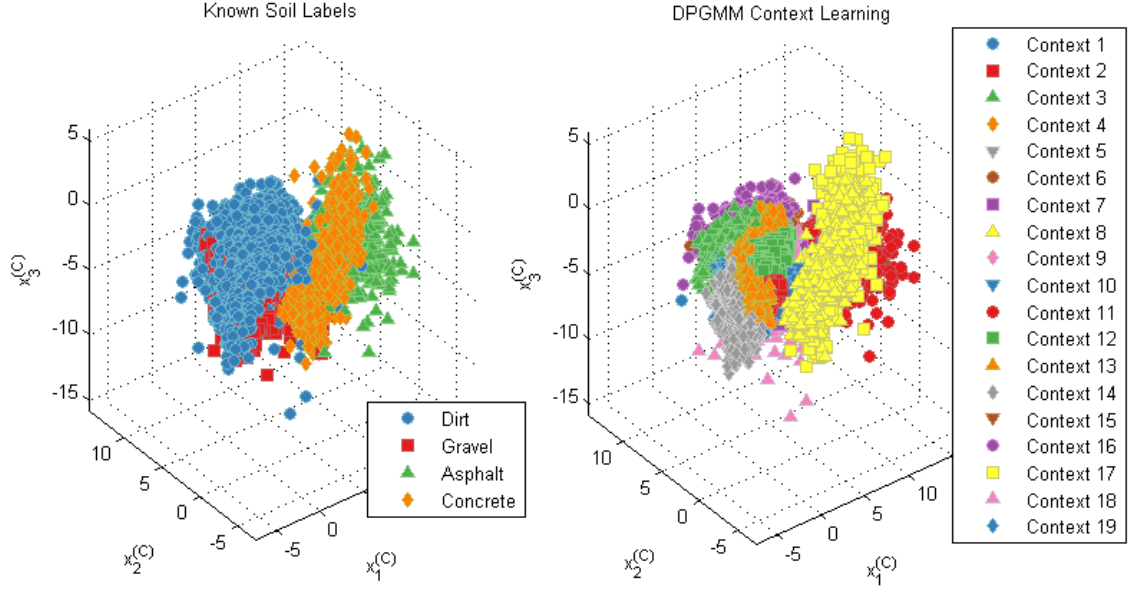


FIGURE 4.5: Scatterplot comparing results of context learning using the DPGMM on the GPR contextual features to the known soil labels. Left: Scatter plot of 3-D PCA projection of contextual features, with points colored by qualitative soil label. Right: Same scatter plot, but with points colored by MAP DPGMM component.

hyperparameters were set according to the same values used in the synthetic data example, with $D^{(C)} = 3$ since the PCA-projected context features were used. Of the 30 initial clusters, the DPGMM converged to 19 within the 1% pruning threshold. Figures 4.5 and 4.6 illustrates the performance of the DPGMM in clustering the context features. The left panel of Figure 4.5 illustrates the scatterplot of the PCA-projected context features, with the points colored by the known soil labels. The right panel shows the contexts obtained by assigning points to the MAP DPGMM component. The similarity matrix comparing the contexts learned by the DPGMM to the known labels is shown in Figure 4.6.

Contexts 2, 3, 4, 6, 9, 10, 12, 13, 14, and 16 were predominantly dirt. Context 18 was predominantly gravel. Contexts 1, 5, 7 and 15 were roughly split between dirt and gravel, suggesting a possible overlap of soil properties between these two labeled categories. Asphalt made up most of Context 11, and concrete made up most of

Similarity Matrix					
DPGMM Context	1	152	156	2	5
	2	1040	625	0	0
	3	370	59	0	1
	4	1160	126	0	0
	5	250	216	0	1
	6	431	11	7	19
	7	121	162	18	60
	8	15	2	8	422
	9	594	120	1	2
	10	693	264	0	6
	11	32	5	235	57
	12	2274	137	0	0
	13	1635	245	0	0
	14	514	158	0	0
	15	294	265	1	1
	16	333	30	0	1
	17	20	23	182	203
	18	71	304	0	19
	19	290	143	3	1
	Dirt	Gravel	Asphalt	Concrete	

FIGURE 4.6: Similarity matrix comparing DPGMM clustering results to the known soil labels.

context 8. Context 17 was roughly split between asphalt and concrete. No context overlapped significantly between one of the unpaved soil types and one of the paved categories. These results suggest that in a large GPR collection such as this, there may be a wealth of contextual information beyond the scope of the available soil labels that can be learned using a nonparametric model.

The learned model parameters are shown by Figure 4.7, which illustrates the cluster means, $\mathbf{p}_m$, and Figure 4.8, which illustrates the covariances $\mathbf{\Lambda}_m$. These plots illustrate that each context corresponds to a Student's- t distribution with unique mean and covariance.

4.5.2 Context Learning with the DPMFA

The DPMFA was trained on the full 23-dimensional GPR context features using the same hyperparameter settings from the synthetic example. Figure 4.9 illustrates the similarity matrix obtained by comparing the known soil labels to the MAP contexts assigned by DPMFA clustering. In this case, 12 contexts were learned that met the

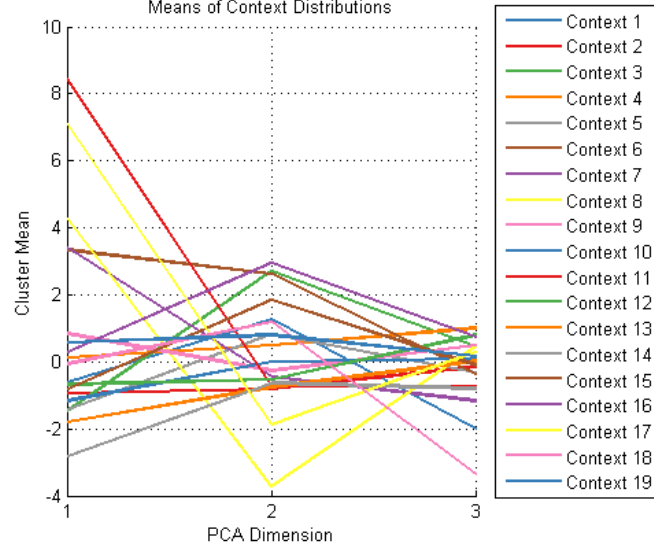


FIGURE 4.7: Means of clusters learned by the DPGMM context model. The horizontal axis represents the dimension of the PCA-projected features $\mathbf{X}^{(C)}$, the vertical axis represents the mean of each cluster that was learned, and colors represent the individual contexts.

1% pruning threshold. The data collected over dirt and gravel were split into many sub-contexts by the DPMFA. Most of these sub-contexts were split between dirt and gravel data, most of which were predominantly comprised of dirt data, but one (Context 7) was predominantly gravel. Asphalt was split into two distinct contexts (3 and 10), and was rarely confused with any of the other soil types. The majority of observations in context 2 were concrete.

Figure 4.10 illustrates the expected model parameters that were learned using VB inference on the DPMFA model. The learned factor loadings (each vector normalized to unit-magnitude for illustration purposes) are shown in the top-left panel, the learned scores (scaled by the corresponding factor magnitude) are shown at top-right, the learned selection vectors are shown at bottom-left, and the cluster membership probabilities are shown at bottom-right. The membership matrix illustrates that most observations fall into clusters 1, 3, 6, 7, 8, 9, 10, 11, 13, 15, 17, and 20. The factor selection matrix shows that most of these mixture components only utilize the

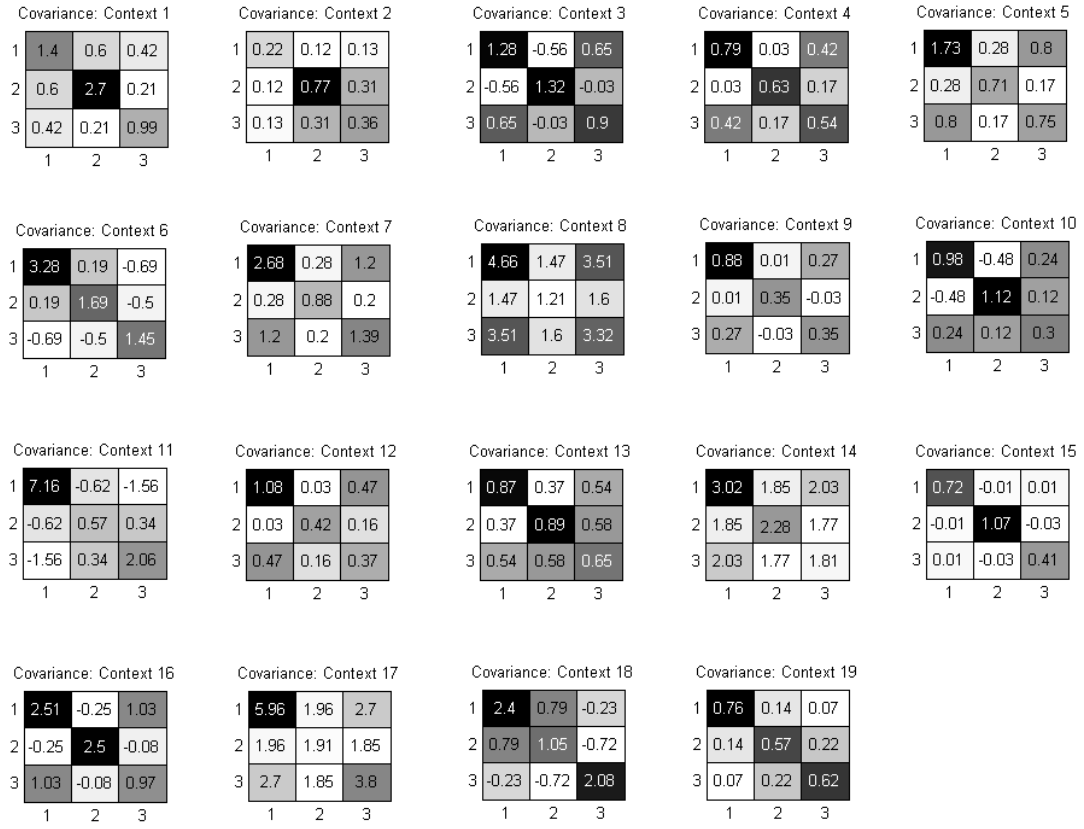


FIGURE 4.8: Covariance matrices of clusters learned by the DPGMM context model. Each panel represents the covariance matrix of the Student- t mixture components obtained by integrating over the DPGMM parameters.

first factor that was learned. However, clusters 1, 8, 13, 17, and 20 also utilize factor 3, but the scores assigned to factor 3 are small (as are the elements of the factor vector itself).

An interesting observation here is that the learned factors were constructed from projections of different contextual features. The features with the greatest magnitude in factor 1 are features 9-14, which correspond to the late-time portions of the MP histogram. Factor 1 may therefore characterize soil heterogeneity and attenuation properties. Meanwhile, the features with greatest magnitude in factor 3 are features 6 and 7, which correspond to early-time portions of the MP histogram. Therefore,

Similarity Matrix: DPMFA Context Learning

1	2652	803	0	0
2	21	4	7	519
3	5	0	284	77
4	171	95	1	14
5	554	298	3	4
6	329	24	2	91
7	86	397	1	9
8	138	65	1	72
9	1172	439	0	0
10	7	2	158	12
11	4595	845	0	0
12	559	79	0	0
	Dirt	Gravel	Asphalt	Concrete

FIGURE 4.9: Similarity matrix comparing DPMFA clustering results to the known soil labels.

factor 3 may characterize the near-surface properties of the soil. Surprisingly, features 1-2 (energy and reflection coefficient) have a very small magnitude in both factors.

Figure 4.11 illustrates the expected means, ρ_m , of each of the 23-dimensional clusters learned from the DPMFA. Figure 4.8 illustrates the variances (i.e. projection residual), ψ_m , of each dimension within each cluster. Each context is characterized by a unique mean and covariance. Several of the contexts have means located near zero, while others appear to be on the outskirts of the feature space.

The variances of the DPMFA-learned contexts should not be considered necessarily as variances in the Gaussian sense, but also the residual of the factor analysis projection of $\mathbf{X}^{(C)}$. Therefore, the features corresponding to *nulls* in variance are best characterized by the factors selected for that context. By this observation, each context appears to have a unique set of nulls (although contexts 1, 9, and 11 appear to be very similar), suggesting that each context uses different contextual feature information. Furthermore, each context yields high variance on features 14-23, which correspond to the LP power features. Recall from Chapter 2 that LP power decreases

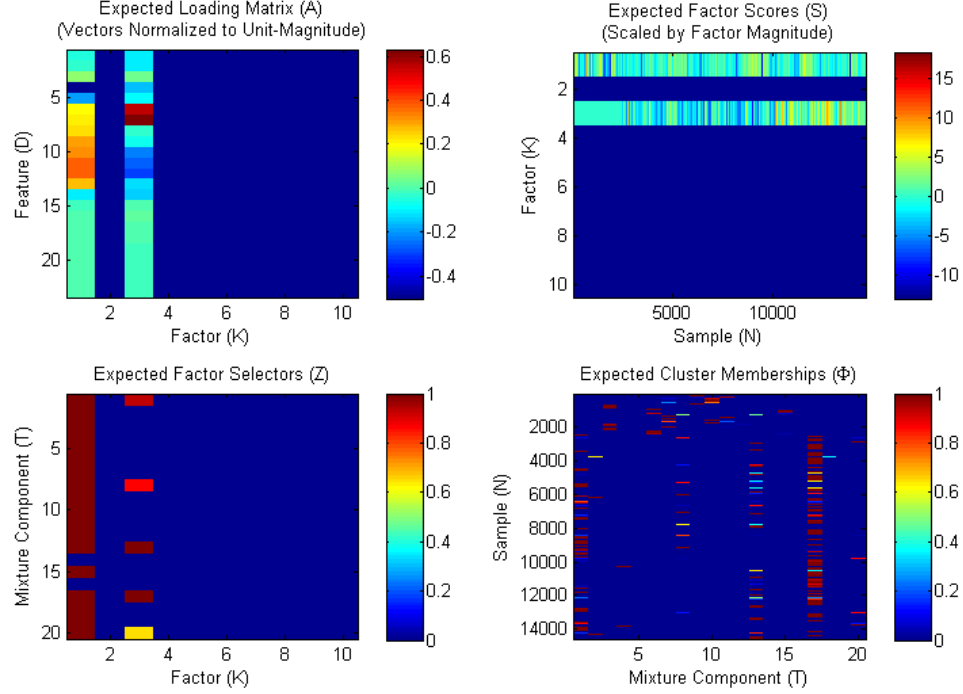


FIGURE 4.10: *A posteriori* expected values of the DPMFA model parameters learned from the GPR contextual features. Top-left: learned loading matrix ($\mathbf{A}$); Top-right: learned factor scores ($\mathbf{S}$); Bottom-left: learned factor selectors ($\mathbf{Z}$); Bottom-right: learned cluster memberships (ϕ).

exponentially with respect to temporal index, suggesting that the feature is characteristic of attenuation effects in soil. Therefore, the high variance that the DPMFA yielded for the LP power features may be an artifact of fitting a linear model to features that exhibit a nonlinear relationship.

4.5.3 Context-Dependent Fusion Results

Context-dependent algorithm fusion was evaluated using the DPGMM and DPMFA context models. Like the basic supervised and unsupervised context learning techniques presented in Chapter 3, posterior context probabilities obtained from the DPGMM and DPMFA were used in training a mixture of RVMs for weighting the confidences of the Prescreener [38], EHD [44], SPSCF [49], and HMM [42] algorithms.

The RVM weights obtained for the DPGMM contexts are plotted in Figure 4.13.

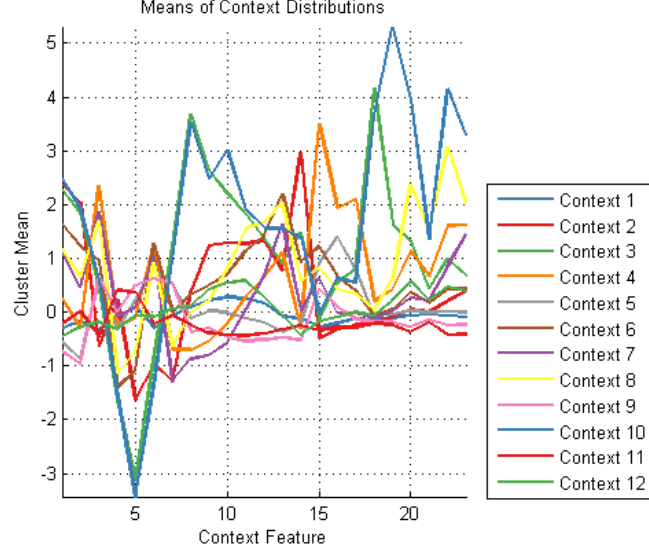


FIGURE 4.11: Means of clusters learned by the DPMFA context model. The horizontal axis represents the dimension of the features $\mathbf{X}^{(C)}$, the vertical axis represents the mean of each cluster that was learned, and colors represent the individual contexts.

The HMM, by far the best-performing single algorithm on this data set, received the most weight and was never irrelevant. Compared to the HMM, the other three algorithms were assigned very small weight and their relative weights varied with respect to context. Each algorithm, with the exception of HMM, was irrelevant in at least one context.

Figure 4.14 illustrates the RVM fusion weights obtained for each of the contexts learned from the DPMFA context model. In the DPMFA contexts, the HMM did not dominate fusion as much as it did with respect to the DPGMM contexts. In one context (context 8), it was actually irrelevant. Meanwhile, the prescreeener received large weight in several contexts, but it was irrelevant in one context (context 11). Each context therefore yielded a unique weighting of the four on-board algorithms, with each algorithm being considered irrelevant in at least one context.

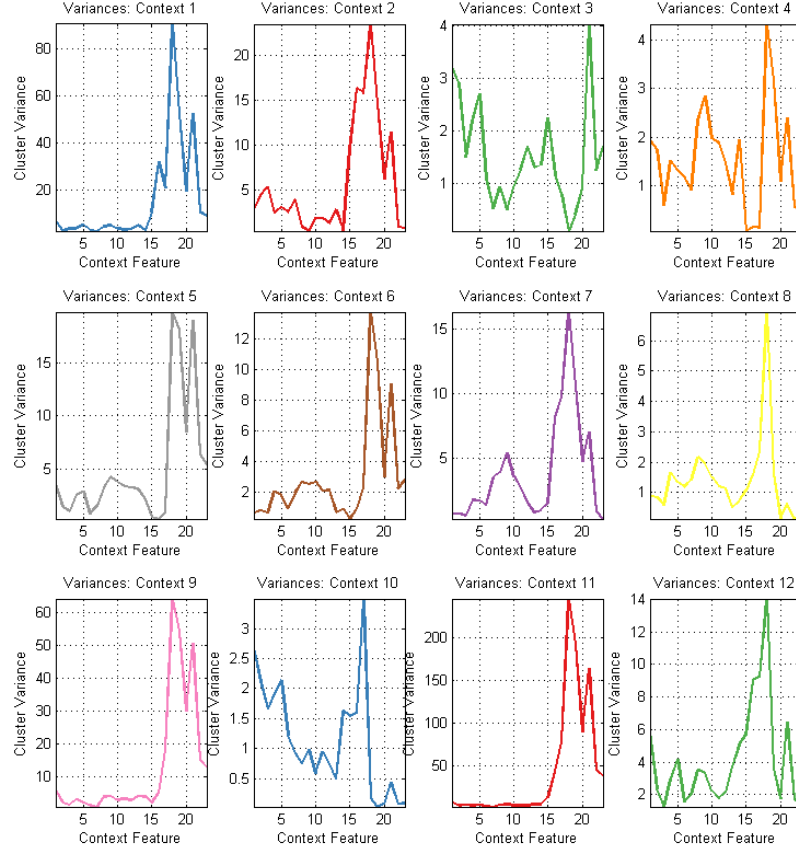


FIGURE 4.12: Covariance matrices of clusters learned by the DPGMM context model.

4.5.4 Detection Performance

Context-dependent fusion was evaluated using the same 10-fold, object-based cross-validation method used to generate the results shown in Chapter 3. The ROC curves plotted Figure 4.15 illustrates the results of context-dependent fusion using the DPGMM and DPMFA context models, which are respectively plotted in red and blue. Performance is compared to global RVM fusion (black dashed) which is not context-dependent, as well as the individual algorithms (dashed lines). The plot is on the same axes scale as Figure 3.3 for easy comparison to the basic context-dependent

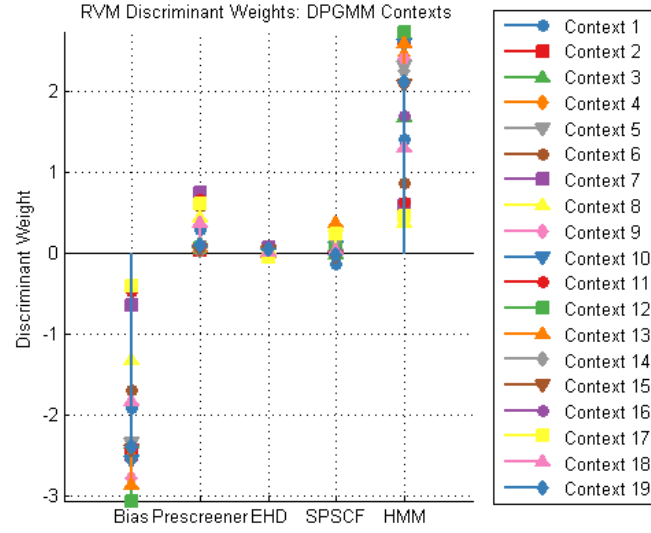


FIGURE 4.13: RVM discriminant weights learned for algorithm fusion in each DPGMM context. Each stem represents a particular dimension of the target feature space, the vertical axis represents the weight value, and the individual contexts are indicated by line color.

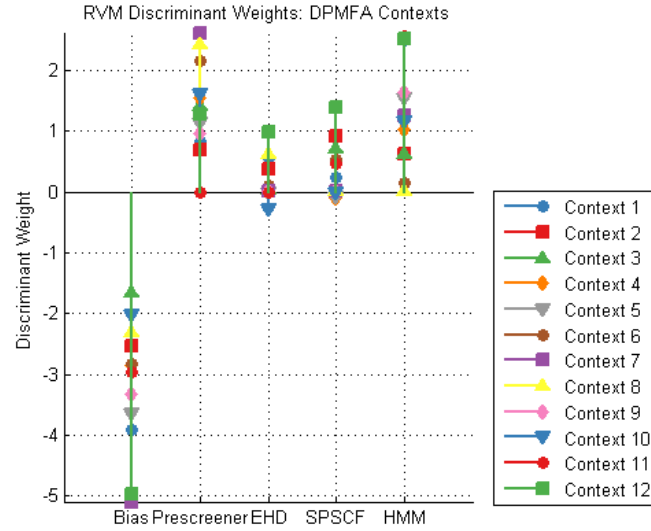


FIGURE 4.14: RVM discriminant weights learned for algorithm fusion in each DPMFA context. Each stem represents a particular dimension of the target feature space, the vertical axis represents the weight value, and the individual contexts are indicated by line color.

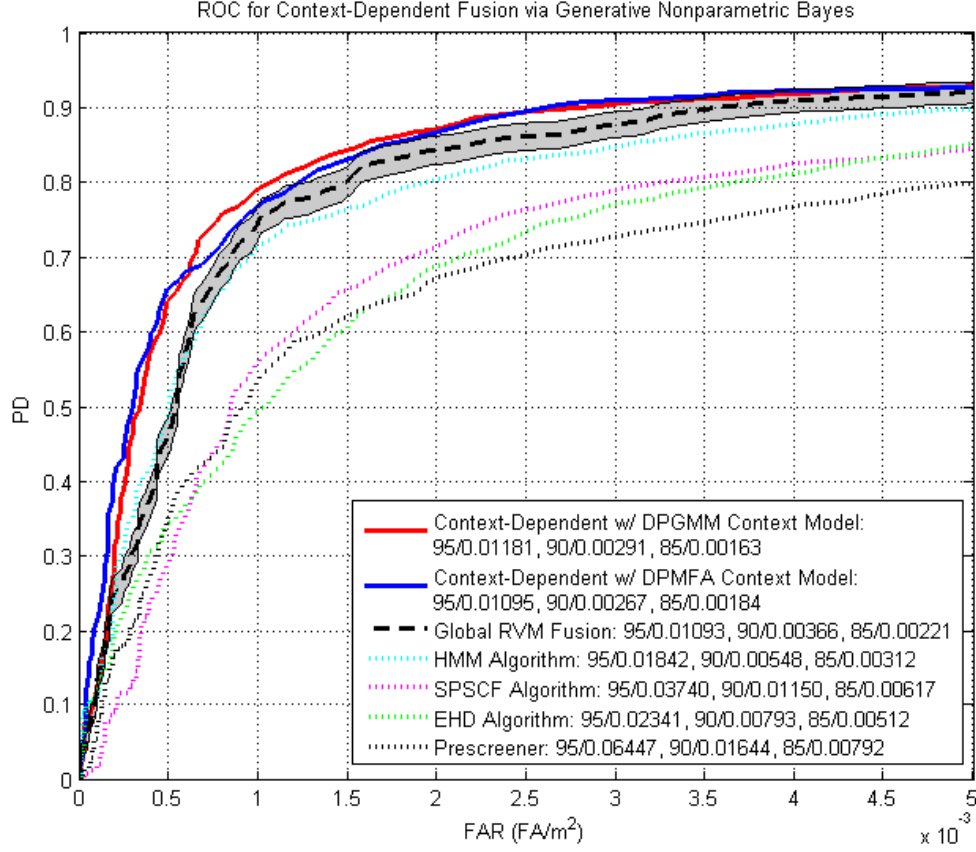


FIGURE 4.15: ROC curves for context-dependent fusion, using either the DPGMM or DPMFA context models, compared to non-context-dependent RVM fusion and the individual fused algorithms. The ROC consists of PD versus FAR, measured in false alarms per square meter, as a function of decision threshold.

techniques discussed in Chapter 3. Results illustrate that significant performance improvements (i.e., outside the 90% confidence bounds indicated by the shaded region) over the non-context-dependent RVM are possible by incorporating nonparametric Bayesian, generative context learning.

Context-dependent fusion with the DPGMM, which did not utilize soil labels and also did not require the specification of the number of contexts to learn, achieved significant reductions in FAR at $0.92 \geq PD \geq 0.25$. Context-dependent fusion using the DPMFA context model, which used even less *a priori* information than the

DPGMM, also yielded significant FAR reduction for the same PD range. It should be noted that both techniques performed better than context-dependent fusion using the supervised contexts trained according to the known soil labels, indicating that additional useful contextual information can be exploited using nonparametric models.

4.6 Conclusions

In this chapter, generative techniques for Bayesian learning nonparametric context models were presented and evaluated on the proposed GPR context features. The two context models were the DPGMM and the DPMFA. Both techniques utilize DP priors to facilitate learning of the number of clusters (contexts) present in the data. The DPGMM was trained on the 3-D PCA projection of the context features, while the DPMFA was able to learn a unique local dimensionality reduction for each cluster. Performance analysis showed that nonparametric models can potentially exploit information that is not described by available qualitative context labels. Experimental results on field-collected GPR data illustrated that using generative nonparametric context models to aid in context-dependent fusion yields significant reductions in FAR for a wide range of PD when compared to conventional fusion.

In contrast to the generative learning techniques that were proposed in this chapter, the following chapter presents *discriminative* techniques for GPR context modeling. In this chapter, context models were trained on the context features only without regard to the target features or the target/clutter labels of each observation. Alternatively, discriminative learning would find contexts that yield the best overall classification of targets from non-targets. Instead of considering both context identification and algorithm fusion as independent tasks, discriminative learning would consider them jointly to yield contexts that allow for the best classification of targets and clutter in each.

Discriminative Nonparametric Context Learning

In the previous chapter, two generative approaches to context-dependent learning were proposed in which the context model and classifiers were learned independently of one another. In generative context learning, contexts were learned based on the distribution of the context features and not with regard to the target/clutter class labels. In contrast, it may be desirable to learn a context-dependent classifier in a *discriminative* manner. Discriminative learning may be useful in finding contexts that allow for the best separation of the target and clutter classes.¹

In this chapter, two approaches are proposed for discriminative context learning. The first is a discriminative treatment of the DPGMM context model coupled with RVM classifiers. The second is a similar technique from the literature that utilizes non-sparse linear classifiers and operates on the joint context and target features. A comparison of both models' behavior is illustrated through several examples with synthetic data. Finally, both techniques were evaluated for GPR algorithm fusion and performance was compared to previous approaches.

¹ This chapter is derivative of previously published work [21]

5.1 Generative vs. Discriminative Learning

Statistical classification approaches often fall into one of two categories: generative or discriminative models. A generative model describes how likely the given data $\mathbf{X}$ was *generated*, and involves learning parameters Θ that define the likelihood function $p(\mathbf{X}|\Theta)$. Most density estimation techniques fall under the umbrella of generative models, including the GMM, HMM, and the k -nearest neighbor density estimate [70–72]. Discriminative models seek to describe how data is *classified*, and involve learning parameters of the conditional PDF of the labels $\mathbf{t}$, i.e. $p(\mathbf{t}|\mathbf{X}, \Theta)$. Most classifiers would therefore be considered discriminative models, including Fisher’s linear discriminant [72], support vector machines (SVMs) [92], and RVMS [83, 84].

In the previous chapters, generative techniques were proposed for training a context model (e.g., GMM, DPGMM, DPMFA) without regard to the target/clutter labels associated with each observation. Although the learned contexts may be reflective of underlying environmental factors, they may not necessarily allow for the best discrimination between targets and clutter. Because the ultimate goal of context-dependent learning is to improve target discrimination across varying environments, it is important to consider the potential benefits of discriminative context learning. Discriminative context models can be framed as a special case of the mixture-of-experts family of models, which are summarized in the following section.

5.2 Mixture-of-Experts Models

In many classification problems, a single linear model may not be sufficient for discriminating between classes. Therefore, many nonlinear classification models have been proposed. These including techniques such as polynomial discriminant analysis [72], decision trees [70] and random forests [106], neural networks [70, 72], and sparse kernel machines including SVMs [92] and RVMS [83, 84]. For each of these

techniques, several parameters must be “tuned” to avoid over- or under-training. Such tuning parameters include the order of a polynomial discriminant function, the pruning criteria used in tree-based methods, the number of hidden layers in a neural network, or the Gram matrix used in training an SVM or RVM. Context-dependent classification is a clear example of a problem requiring a nonlinear decision model. However, it is important to avoid the pitfall of insufficient training due to poor parameter selection while still maintaining the ability to discriminatively train the classifier.

Mixture-of-experts models are a family of classification and regression techniques that approximate a nonlinear model by an mixture of locally-linear “expert” models. The most representative of this family of classifiers is the *hierarchical mixture of experts* (HME) [107], in which the distribution of the binary class label, t , conditioned on each of $m = 1, 2, \dots, M$ experts is given by

$$p(t|\mathbf{x}, \mathbf{w}_m) = \sigma(\mathbf{w}_m^T \mathbf{x})^t [1 - \sigma(\mathbf{w}_m^T \mathbf{x})]^{1-t}, \quad (5.1)$$

where $\mathbf{w}_m$ are the weights associated with expert m , and $\sigma(\cdot)$ denotes the logistic sigmoid function.

The HME utilizes a linear gating network of $p = 1, 2, \dots, P$ nodes, each corresponding to an associated binary variable, $z_p = \{0, 1\}$. The value of z_p drawn from a Bernoulli distribution given by

$$p(z_p|\mathbf{x}, \mathbf{v}_p) = \sigma(\mathbf{v}_p^T \mathbf{x})^{z_p} [1 - \sigma(\mathbf{v}_p^T \mathbf{x})]^{1-z_p}, \quad (5.2)$$

where $\mathbf{v}_p$ are the parameters of the distribution governing node p .

Given the state of the gating network, the conditional distribution on the labels, $\mathbf{t}$, takes the form

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \boldsymbol{\tau}, \mathbf{z}) = \prod_{m=1}^M \left[\sigma(\mathbf{w}_m^T \mathbf{x})^t [1 - \sigma(\mathbf{w}_m^T \mathbf{x})]^{1-t} \right]^{\zeta_m}, \quad (5.3)$$

where

$$\zeta_m = \prod_{p=1}^P \bar{z}_p. \quad (5.4)$$

The parameter $\bar{z}_p$ allows for the nesting of sub-networks, so that

$$\bar{z}_i = \begin{cases} z_p & \text{if } m \text{ is in the left sub-tree of } p \\ 1 - z_p & \text{otherwise.} \end{cases} \quad (5.5)$$

The HME is learned discriminatively, and ML [107] and Bayesian [108] approaches have been proposed. However, the same caveats regarding model order that were discussed for probabilistic mixture models in Chapter 3 also apply to the HME. The order of the HME model is given by P , the number of unique nodes, and M , the number of experts. Both must be specified, and improper selection of P and M could lead to over- or under-training, which could result in poor performance.

This chapter considers two methods for discriminative context learning based on the HME paradigm, but the linear gating network is replaced with a network based on the DPGMM, which was originally presented in Chapter 4. The DPGMM gating network allows for a nonparametric model, facilitating learning of the number of expert component classifiers, using previously-developed learning methods.

The two methods being considered for discriminative context learning differ in the features used for classification and clustering, as well as their accommodation of sparse classification models. The first technique is based on those proposed in the Chapter 4, and involves replacing the linear gating network of the HME with a DPGMM, and the logistic experts with RVMs. Thus, this approach is referred to as the DPGMM-RVM. A novel property of the DPGMM-RVM is that it seeks to learn the DPGMM in the contextual features, while also training the RVMs on the target features [21].

The second discriminative context model is based on the infinite quadratically-

gated mixture of experts (IQGME) [94]. The IQGME also utilizes a DPGMM gating network, but performs classification and clustering in the same feature space. Therefore, it is not amenable to sparse classifiers. The derivations of both the DPGMM-RVM and IQGME are presented in greater detail in Section 5.3, and performance is compared in a series of synthetic data examples in Section 5.4.

5.3 Discriminative Context Models

Consider the DPGMM context model whose likelihood function is given by (4.27). The stick-breaking prior is initialized with a truncation level of T , and the DPGMM will cluster the contextual features $\mathbf{X}^{(C)}$ into M mixture components where $M < T$. Additionally, consider the RVM classifier whose likelihood function is given by (3.7) and (3.8). The RVM incorporates a sparseness-promoting prior on the weights ($\mathbf{w}$) that are used to classify the target features ($\mathbf{X}^{(T)}$) according to the labels ($\mathbf{t}$).

Inference could be performed on the DPGMM and RVM jointly using a discriminative model referred to here as the DPGMM-RVM. The likelihood function of the DPGMM-RVM is given by

$$p(\mathbf{t}, \mathbf{X}^{(C)} | \mathbf{X}^{(T)}, \mathbf{C}, \mathbf{W}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^N \prod_{m=1}^T \left[\sigma(\mathbf{w}_m^T \mathbf{x}_n^{(T)})^{t_n} [1 - \sigma(\mathbf{w}_m^T \mathbf{x}_n^{(T)})]^{1-t_n} \mathcal{N}_{D^{(C)}}(\mathbf{x}_n^{(C)} | \boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m^{-1}) \right]^{c_{nm}}, \quad (5.6)$$

where N denotes the number of observations, T denotes the truncation level, $D^{(C)}$ denotes the dimensionality of $\mathbf{X}^{(C)}$, and c_{nm} is the binary indicator that denotes the context of the n th observation.

The DPGMM-RVM model can be learned discriminatively by assuming conjugate priors and using VB inference. The data-generating process for the fully-conjugate DPGMM-RVM is as follows:

1. Draw $\alpha \sim \text{Gamma}(\tau_{10}, \tau_{20})$

2. For $m = 1, 2, \dots, T$

- (a) Draw $v_m | \alpha \sim \text{Beta}(1, \alpha)$
- (b) Draw $\boldsymbol{\theta}_m^* | G_0 \sim \mathcal{N}_{D^{(C)}}(\boldsymbol{\mu}_m^* | \boldsymbol{\rho}_0, u_0^{-1} \boldsymbol{\Lambda}_m^{*-1}) \mathcal{W}(\boldsymbol{\Lambda}_m^* | \mathbf{B}_0, \nu_0)$
- (c) Calculate mixture proportions $\pi_m(\mathbf{v}) = v_m \prod_{j=1}^{m-1} (1 - v_j)$
- (d) Draw $\beta_{md} \sim \text{Gamma}(a_0, b_0), d = 1, 2, \dots, D^{(T)}$
- (e) Draw $\mathbf{w}_m \sim \mathcal{N}_{D^{(T)}}(0, \text{diag}(\boldsymbol{\beta}_m)^{-1})$

3. For $n = 1, 2, \dots, N$

- (a) Draw indicator variable $\mathbf{c}_n \sim \text{Multi}(\boldsymbol{\pi})$
- (b) Draw data $\mathbf{x}_n^{(C)} | c_{nm} = 1 \sim \mathcal{N}_{D^{(C)}}(\mathbf{x}_n^{(C)} | \boldsymbol{\theta}_m^*)$
- (c) Draw label $t_n | c_{nm} = 1 \sim \sigma(\mathbf{w}_m^T \mathbf{x}_n^{(T)})^{t_n} [1 - \sigma(\mathbf{w}_m^T \mathbf{x}_n^{(T)})]^{1-t_n}$

Inference on the DPGMM-RVM will seek to perform clustering the $D^{(C)}$ -dimensional contextual features $\mathbf{X}^{(C)}$ while training sparse linear classifiers in the $D^{(T)}$ -dimensional target features $\mathbf{X}^{(T)}$. For all experiments, the following prior hyperparameter settings were used: $a_0 = b_0 = u_0 = 1$, $\tau_{10} = \tau_{20} = 0.01$, $\nu = D^{(C)}$, $\mathbf{B}_0 = D^{(C)} \mathbf{I}_{D^{(C)}}$, and $\boldsymbol{\rho}_0$ was set equal to the sample mean of $\mathbf{X}^{(C)}$. Variational inference was performed until the NFE converged within 0.01%. All details regarding VB for the DPGMM-RVM, including update equations and the NFE, are derived in Appendix E.

The structure of the DPGMM-RVM allows for mean-field updates of the DPGMM and RVM parameters to be performed independently of one another. Only in the update for the cluster responsibilities (the variational parameters governing the pos-

terior on $\mathbf{C}$), are both sets of parameters used:

$$\begin{aligned}
& \log \phi_{nm} \propto \log q(c_{nm} = 1) \\
& \propto \langle \log p(\mathbf{T}, \mathbf{X}^{(C)} | \mathbf{C}, \mathbf{X}^{(T)}, -) \rangle + \log p(\mathbf{C}) \\
& \propto \langle \log p(\mathbf{T} | \mathbf{C}, \mathbf{X}^{(T)}) \rangle + \langle \log p(\mathbf{X}^{(C)} | \mathbf{C}) \rangle + \log p(\mathbf{Z}) \\
& \propto \log \sigma(\xi_{nm}) + \frac{1}{2} ([2t_n - 1] \langle \mathbf{w}_m^T \mathbf{x}_n^{(T)} - \xi_{nm} \rangle - \lambda(\xi_{nm}) \left(\mathbf{x}_n^{(T)T} \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \mathbf{x}_n^{(T)} - \xi_{nm}^2 \right) \\
& \quad + \frac{1}{2} \langle \log |\mathbf{\Lambda}_m| \rangle - \frac{1}{2} \langle (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m)^T \mathbf{\Lambda}_m (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m) \rangle \\
& \quad + \langle \log v_m \rangle + \sum_{l < m} \langle \log (1 - v_l) \rangle,
\end{aligned} \tag{5.7}$$

where λ and ξ are defined in the RVM derivation found in Appendix B, and $\langle \cdot \rangle$ denotes variational expectation.

The first line of the final expression in (5.7) is the expectation of the RVM log-likelihood given by (B.38), the second line is the expectation of the GMM log-likelihood given by (C.8), and the third line is the stick-breaking prior. The prior will regularize the updates for both the DPGMM and RVM parameters, and the DPGMM and RVM will also regularize one another. Therefore, instead of learning a DPGMM that fits $\mathbf{X}^{(C)}$ well, or a set of RVMs that predict $\mathbf{t}$ well, the DPGMM-RVM will seek a model that satisfies *both* criteria.

An alternative approach would be to perform clustering and classification in the *combined* feature space $\tilde{\mathbf{X}} = [\mathbf{X}^{(C)}, \mathbf{X}^{(T)}]$ which has dimensionality $\tilde{D} = D^{(C)} + D^{(T)}$. The likelihood function of this model is similar to the DPGMM-RVM:

$$p(\mathbf{t}, \tilde{\mathbf{X}} | -, -) = \prod_{n=1}^N \prod_{m=1}^M \left[\sigma(\mathbf{w}_m^T \tilde{\mathbf{X}}_n)^{t_n} \left[1 - \sigma(\mathbf{w}_m^T \tilde{\mathbf{X}}_n) \right]^{1-t_n} \mathcal{N}_{\tilde{D}}(\tilde{\mathbf{X}}_n | \boldsymbol{\mu}_m, \mathbf{\Lambda}_h^{-1}) \right]^{c_{nm}}, \tag{5.8}$$

The model given by (5.8) was originally presented in [109] as the quadratically-

gated mixture of experts (QGME) for classification in problems with missing data. Incorporating a stick-breaking prior on the latent variables $\mathbf{C}$ yields the *infinite* quadratically-gated mixture of experts (IQGME) that was proposed in [94]. The QGME and IQGME were both originally proposed for classifying data with missing dimensions, so context-dependent learning is a novel application for this type of model.

It was suggested in [94] that it may not be desirable to enforce sparseness in the component classifiers if they are all jointly operating in the same feature space, since sparse component classifiers will yield a decision function that is discontinuous in the joint features $\tilde{\mathbf{X}}$. Unlike the DPGMM-RVM, the QGME and IQGME therefore utilize a common Normal-Gamma prior on the classifier weights given by

$$\mathbf{w}_m \sim \mathcal{N}_{\tilde{D}}(\boldsymbol{\xi}, \text{diag}(\boldsymbol{\beta})^{-1}), \quad (5.9)$$

$$(\boldsymbol{\xi}|\boldsymbol{\beta}) \sim \mathcal{N}_{\tilde{D}}(0, \gamma_0^{-1} \text{diag}(\boldsymbol{\beta})^{-1}), \quad (5.10)$$

$$\beta_p \sim \text{Gamma}(a_0, b_0), \quad p = 1, 2, \dots, \tilde{D}. \quad (5.11)$$

The data-generating process for the IQGME is very similar to the DPGMM-RVM, with the only differences being that clustering and classification are performed on the common features $\tilde{\mathbf{X}}$ and the prior given by (5.9)-(5.11) is imposed on the classifier weights. The hyperparameter settings for the IQGME in all experiments were very similar to the DPGMM-RVM; $a_0 = b_0 = u_0 = 1$, $\tau_{10} = \tau_{20} = 0.01$, $\gamma_0 = 1$, $\nu = \tilde{D}$, $\mathbf{B}_0 = \tilde{D}\mathbf{I}_{\tilde{D}}$, and $\boldsymbol{\rho}_0$ was set equal to the sample mean of $\tilde{\mathbf{X}}$. VB inference was also performed until the NFE converged within 0.01%.

Although the differences between the DPGMM-RVM and IQGME may appear to be subtle, the novel accommodation of sparse linear models through the DPGMM-RVM allows for markedly different performance. These differences will be analyzed in the following section through a series of synthetic data examples.

5.4 Synthetic Data Examples

In this section, the DPGMM-RVM and the IQGME are compared in three context-dependent learning problems using synthetic data. The first problem considers the case in which all features are informative; i.e. the classes are separable in the joint context and target features. The second problem is similar to the first, but the context features are made less informative by increasing the variance of each cluster. The third problem considers the case in which most of the target features are irrelevant in each context. This may occur in GPR algorithm fusion if one or more algorithms perform poorly in certain environments. In all examples, the DPGMM-RVM and IQGME were initialized with a clustering truncation of $T = 20$. The DPGMM-RVM and IQGME are compared based upon their context identification performance, learned discriminant weights, and overall classification accuracy.

Case 1: All Features Informative

Figure 5.1 provides scatterplots of the synthetic target and contextual features. The target features were drawn from Gaussian distributions conditioned on each class and context:

$$p(\mathbf{x}^{(T)}|H_0, c_1) = \mathcal{N}([-3, -2], 2\mathbf{I}), \quad p(\mathbf{x}^{(T)}|H_1, c_1) = \mathcal{N}([0, 0], 2\mathbf{I})$$

$$p(\mathbf{x}^{(T)}|H_0, c_2) = \mathcal{N}([-4, -1], 2\mathbf{I}), \quad p(\mathbf{x}^{(T)}|H_1, c_2) = \mathcal{N}([-2, 0], 2\mathbf{I})$$

$$p(\mathbf{x}^{(T)}|H_0, c_3) = \mathcal{N}([0, 0], 2\mathbf{I}), \quad p(\mathbf{x}^{(T)}|H_1, c_3) = \mathcal{N}([-3, -2], 2\mathbf{I})$$

$$p(\mathbf{x}^{(T)}|H_0, c_4) = \mathcal{N}([-2, 0], 2\mathbf{I}), \quad p(\mathbf{x}^{(T)}|H_1, c_4) = \mathcal{N}([-4, -1], 2\mathbf{I})$$

In the aggregate target feature space, the classes appear to overlap completely as shown in the left panel. The context features were drawn from four distinct Gaussian distributions:

$$p(\mathbf{x}^{(C)}|c_1) = \mathcal{N}([-2, 2], 2\mathbf{I})$$

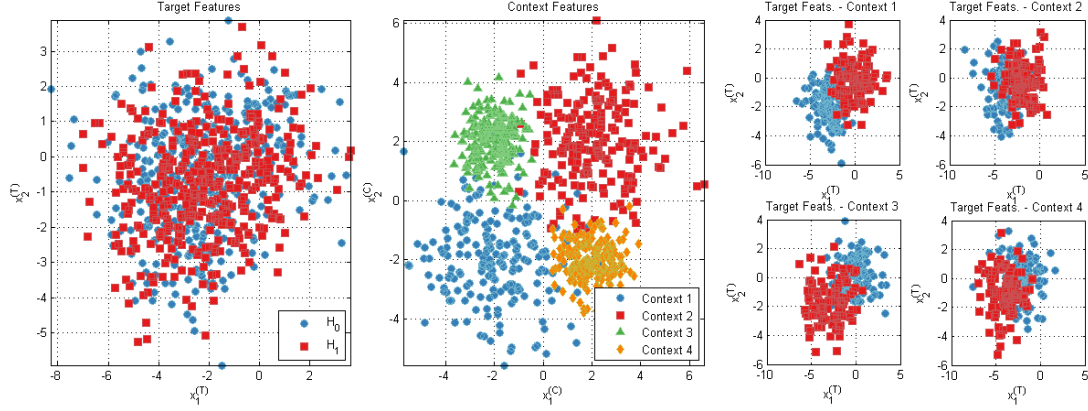


FIGURE 5.1: Scatterplot of target and context features for the first synthetic data example to illustrate discriminative context-dependent learning. Left: two-dimensional aggregate target feature space, with points colored by class; Center: two-dimensional context feature space, with points colored by context; Right: target features, split into individual contexts.

$$p(\mathbf{x}^{(C)}|c_2) = \mathcal{N}([2, 2], 2\mathbf{I})$$

$$p(\mathbf{x}^{(C)}|c_3) = \mathcal{N}([-2, 2], 0.5\mathbf{I})$$

$$p(\mathbf{x}^{(C)}|c_4) = \mathcal{N}([2, -2], 0.5\mathbf{I})$$

The center panel illustrates the two-dimensional context feature space and the distinct clusters are clearly visible. Conditioning the target features on the true underlying contexts reveals four classification problems that are almost linearly separable, as shown in the rightmost panels.

Figure 5.2 illustrates the clustering results obtained from the DPGMM-RVM in the contextual feature space. The left panel shows a scatterplot of the contextual feature space, with points colored by the MAP context assigned by the DPGMM-RVM. The similarity matrix between the learned contexts and the true context labels is shown in the right panel, illustrating that the four contexts that were learned correspond very closely to the true contexts.

The clustering results obtained from the IQGME are summarized in Figure 5.3. A total of 8 clusters were learned, and they appear to overlap in the contextual feature

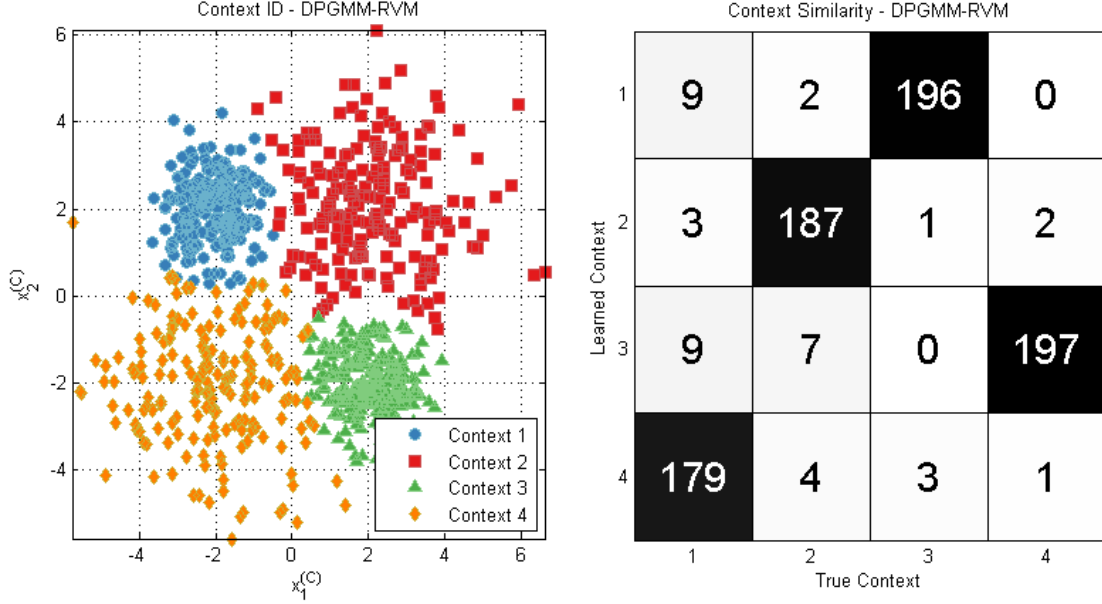


FIGURE 5.2: Results of context identification using the discriminative DPGMM-RVM model for the first synthetic data example.. Left: scatterplot of context features, with points colored by MAP context; Right: similarity matrix of true and learned context assignments.

space. However, recall that the IQGME performs clustering on the combined contextual and target features. Although the cluster assignments may appear to overlap heavily in the context feature space, they are distinct in the combined features.

The differences in clustering results for the DPGMM-RVM and IQGME are better-explained by comparing the classifiers learned by each. Figure 5.4 illustrates the classifiers corresponding to each of the contexts learned by the DPGMM-RVM. Each panel shows a local target feature space in which points are colored by class, and the linear decision models corresponding to each context are also shown. In the case of the DPGMM-RVM, each context is representative of a unique binary classification problem with approximately equal numbers of points from each class.

The classifiers learned by the IQGME are shown in Figure 5.5, and are markedly different from those learned by the DPGMM-RVM. Note that although IQGME performs classification in the joint context and target features, the illustrated classifica-

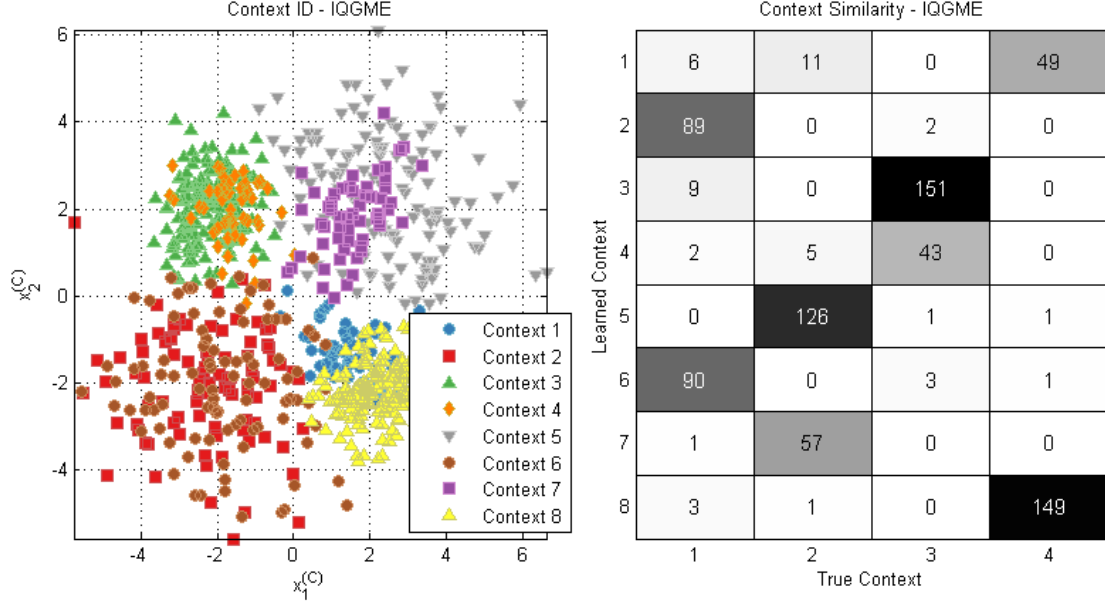


FIGURE 5.3: Results of context identification using the IQGME model for the first synthetic data example.. Left: scatterplot of context features, with points colored by MAP context; Right: similarity matrix of true and learned context assignments.

tion lines correspond only to the weights on the target features. Although Contexts 1, 3, 5, and 8 illustrate linearly-separable binary classification problems, Contexts 2, 4, 5, and 7 consist of mostly points from the H_0 class. The classifiers learned for these contexts could be highly over-trained because they do not incorporate much information about the H_1 class.

The differences between the behavior of the DPGMM-RVM and IQGME can be further highlighted through analysis of the discriminant weights, which are plotted in Figure 5.6. The top panel illustrates the weights learned by the DPGMM-RVM for each context, and the center panel illustrates the weights learned by the IQGME. The bottom plot shows the weights obtained from an “oracle” that trains a linear RVM on each of the context-specific classification problems.

The weights learned by the DPGMM-RVM agree nearly perfectly with the oracle weights. Because the IQGME operates on the joint target and context features, it

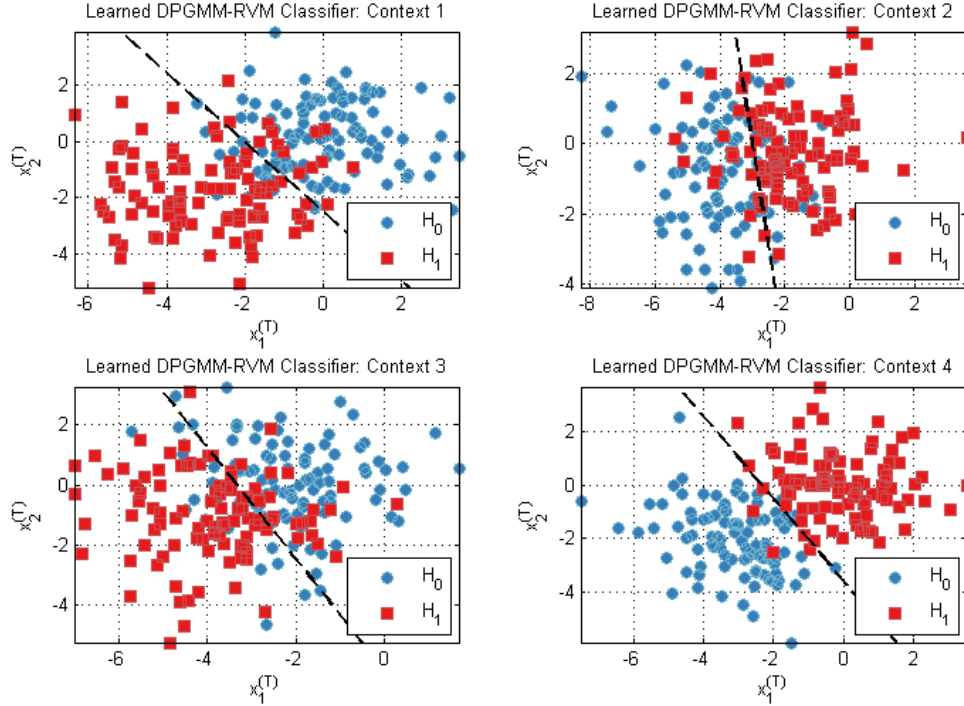


FIGURE 5.4: Component classifiers learned by the discriminative DPGMM-RVM model for the first synthetic data example. Each panel illustrates a two-dimensional scatterplot of the target features, corresponding to points from each learned context, with the decision boundary learned for each context overlaid.

assigns weights to the two context features as well. The weights assigned to the target features are of smaller magnitude than the weights assigned by the DPGMM-RVM and the oracle, and are of similar magnitude to the weights assigned to the context features. This result suggests that the IQGME also found the context features to be informative of class.

ROC curves for the DPGMM-RVM, IQGME, and the oracle are plotted in Figure 5.7. Results were evaluated by training and testing on different sets of data drawn from the same context and target feature distributions. The ROC for the DPGMM-RVM is shown in blue, and IQGME is shown in green. Performance is compared to generative context-dependent learning with the DPGMM-RVM (red), the oracle (black solid), and a linear RVM operating on the target features alone (black dashed).

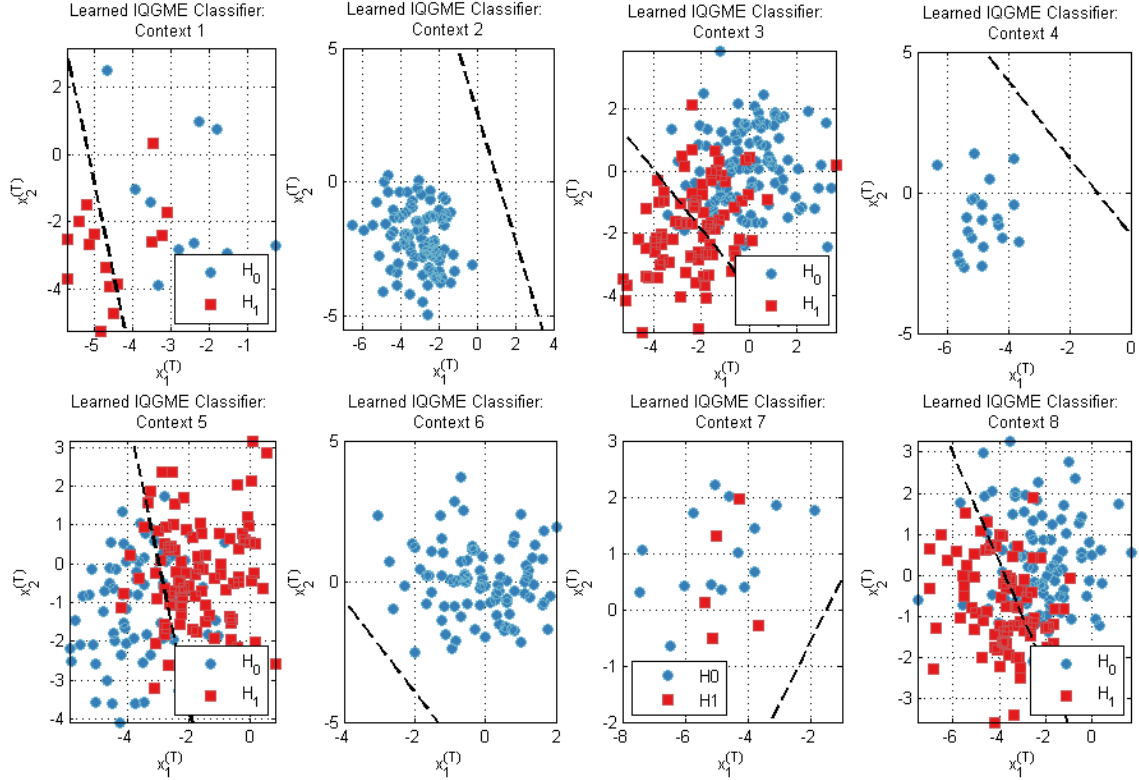


FIGURE 5.5: Component classifiers learned by the IQGME model for the first synthetic data example. Each panel illustrates a two-dimensional scatterplot of the target features, corresponding to points from each learned context, with the decision boundary learned for each context overlaid.

The performance of both the discriminative and generative DPGMM-RVM were similar, with the generative approach having slightly better performance. The IQGME did not perform as well as either DPGMM-RVM; it is likely that the IQGME was overtrained since it learned classifiers for contexts consisting of only data from one class.

Case 2: Less-Informative Context Features

In the second simulated data example, the context features were less informative since the clusters overlapped more in the feature space. This was achieved by increasing the variances of each dimension in each context. The distributions for Contexts 1

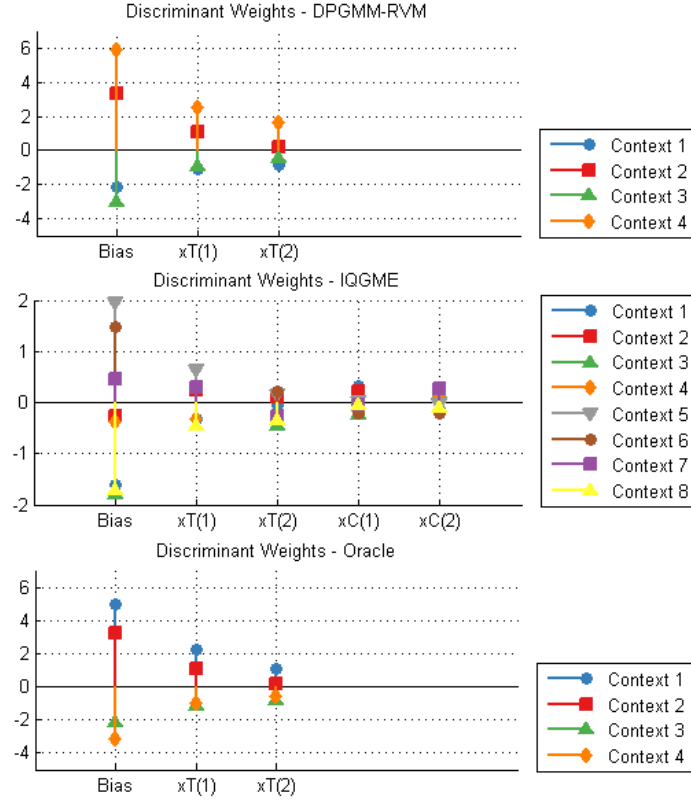


FIGURE 5.6: Discriminant weights learned by the DPGMM-RVM, IQGME, and the context oracle for the first synthetic data example, colored by context. Top: DPGMM-RVM weights; Center: IQGME weights; Bottom: RVM weights based on the context oracle.

and 2 had a covariance of $3\mathbf{I}$ and Contexts 3 and 4 had a covariance of $2\mathbf{I}$. Figure 5.8 illustrates scatterplots of the synthetic target and contextual features for the second synthetic data example.

Figure 5.9 illustrates the clustering results obtained from the DPGMM-RVM in the contextual feature space, as well as the similarity matrix between the learned contexts and the true context labels. In this case, the DPGMM-RVM learned more contexts than before, yielding a total of 7. Most of the data from each of the four true contexts are split between three or four learned contexts. This illustrates that when less obvious clustering exists in the contextual features, the number of contexts

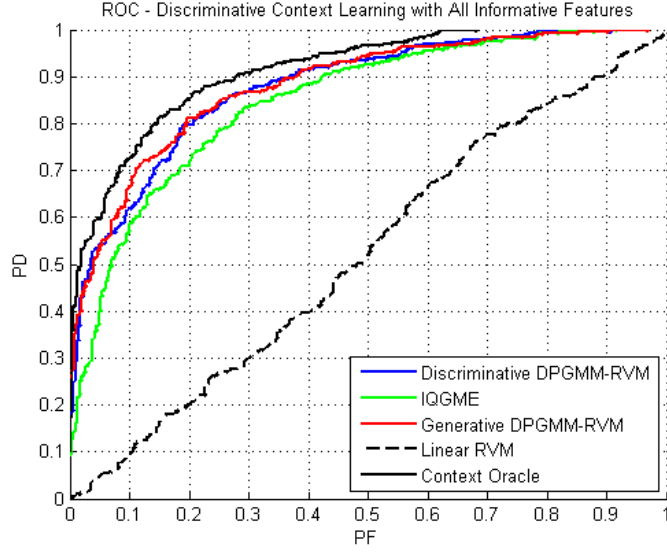


FIGURE 5.7: ROC curves comparing discriminative context-dependent learning on the first synthetic data example. Performance is compared between the DPGMM-RVM (blue), IQGME (green), generative context-dependent learning with the DPGMM-RVM (red), linear RVM learned on target features only (black dashed), and the context oracle (black solid).

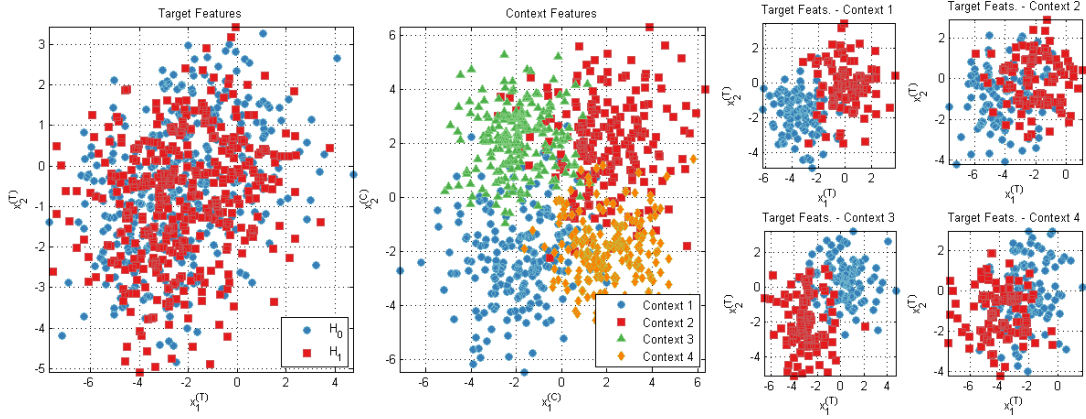


FIGURE 5.8: Scatterplot of target and context features for the second synthetic data example. Left: two-dimensional aggregate target feature space, with points colored by class; Center: two-dimensional context feature space, with points colored by context; Right: target features, split into individual contexts. [21]

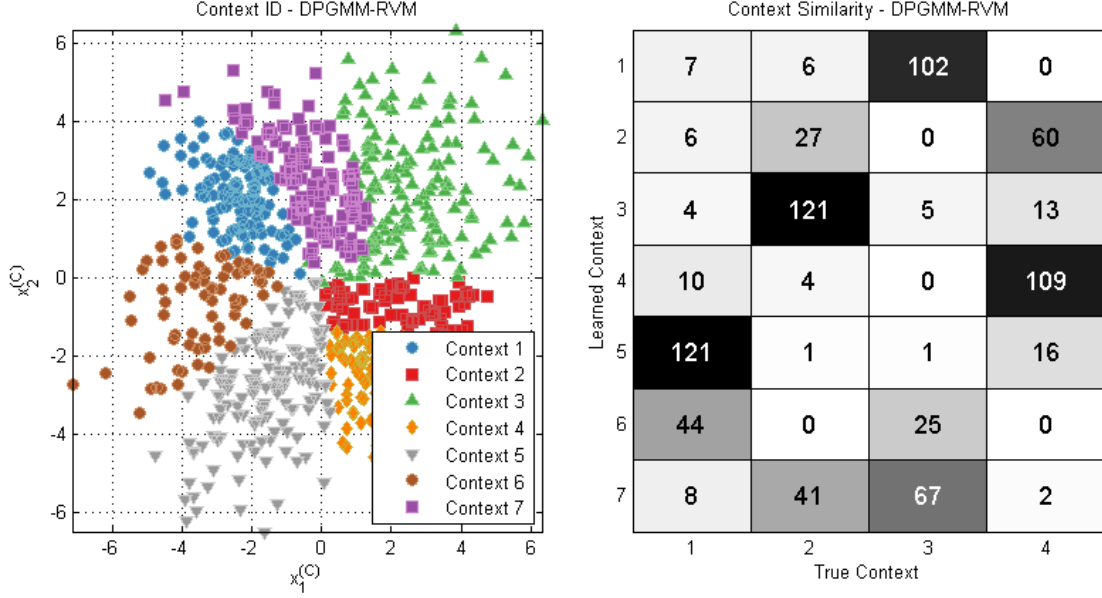


FIGURE 5.9: Results of context identification using the discriminative DPGMM-RVM model for the second synthetic data example. Left: scatterplot of context features, with points colored by MAP context; Right: similarity matrix of true and learned context assignments. [21]

learned by the DPGMM-RVM may increase.

The clustering results obtained from the IQGME are summarized by Figure 5.10. A total of 11 clusters were learned, and like before, they appear to overlap in the contextual feature space since clustering was performed on the combined contextual and target features. Furthermore, the higher number of clusters suggests that more locally-unique classification problems were learned from the IQGME than from the DPGMM-RVM.

The component classifiers learned by the discriminative DPGMM-RVM are shown in Figure 5.11. Although more contexts were learned in the case of less-informative context features, the DPGMM-RVM still finds linearly-separable sub-problems for each context. It is interesting to note that for most of the contexts, the learned decision boundary is either purely horizontal or vertical. This suggests that in these contexts, only one target feature is relevant.

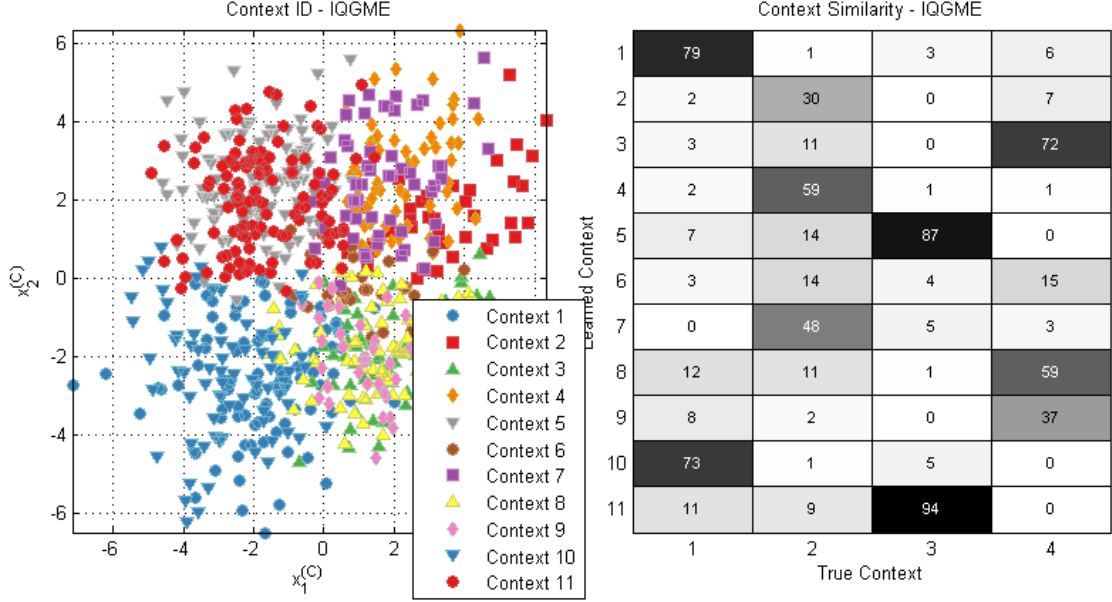


FIGURE 5.10: Results of context identification using the IQGME model for the second synthetic data example.. Left: scatterplot of context features, with points colored by MAP context; Right: similarity matrix of true and learned context assignments.

The classifiers learned by the IQGME are shown in Figure 5.12. Recall that IQGME performs classification in the joint context and target feature space; for visualization purposes, the classification lines shown in each panel are determined by the weights on the target features. Like the first example, most of the contexts learned by the IQGME consist of data from mostly one class. This is true for Contexts 1, 4, 5, 9, and 10. Based on these results, the IQGME would be expected to perform similarly as before.

The discriminant weights for the DPGMM-RVM, IQGME, and oracle are shown in Figure 5.13. Similar to the previous case, the DPGMM-RVM and oracle have weights of similar magnitude. However, since more than four contexts were learned, they do not match nearly as well as in the first example. However, the IQGME shows similar performance as before, assigning small weight to each of the target and context features.

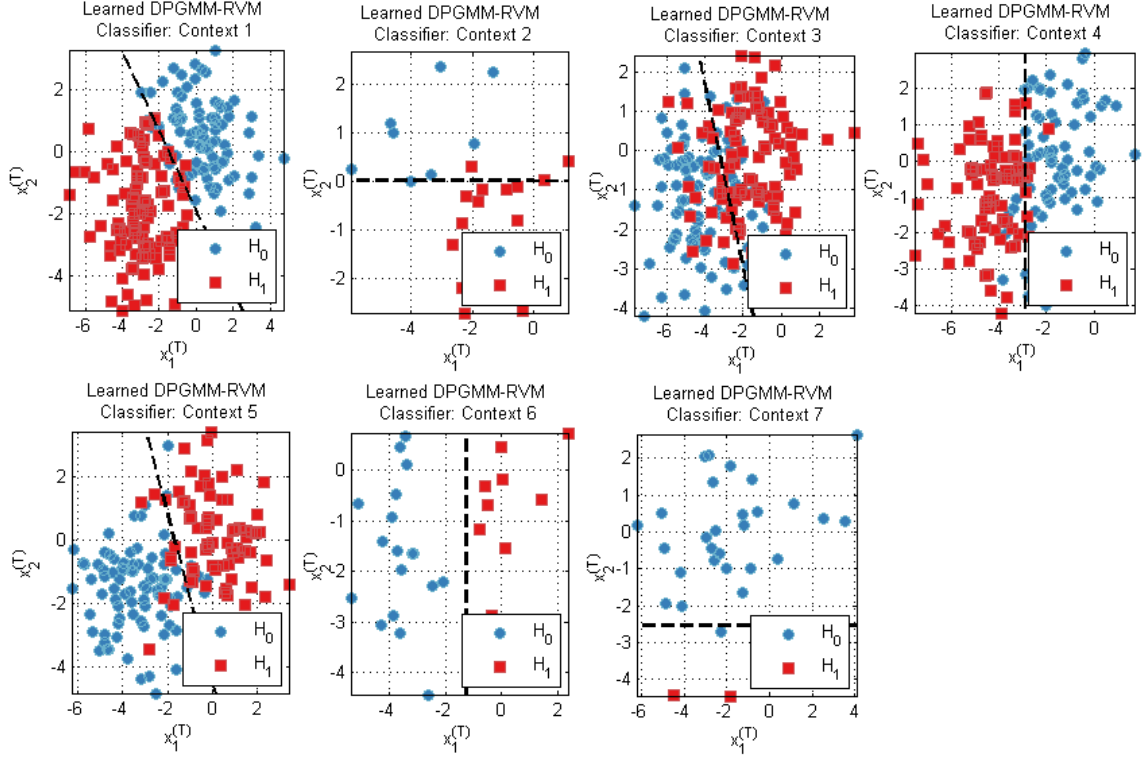


FIGURE 5.11: Component classifiers learned by the discriminative DPGMM-RVM model for the second synthetic data example. Each panel illustrates a two-dimensional scatterplot of the target features, corresponding to points from each learned context, with the decision boundary learned for each context overlaid. [21]

ROC curves for the second synthetic data example are shown in Figure 5.14. In this case, both discriminative approaches outperformed the generative approach. This is because the discriminative models learned contexts where classification could be performed effectively, while the generative model only sought to cluster the context features. Another interesting observation is that both discriminative models performed similarly to one another, suggesting that the IQGME was not as over-trained as the larger number of contexts may have suggested.

Case 3: Irrelevant Target Features

The third simulated data example addresses performance when some target features are irrelevant. This has implications for buried threat detection, in which the rele-

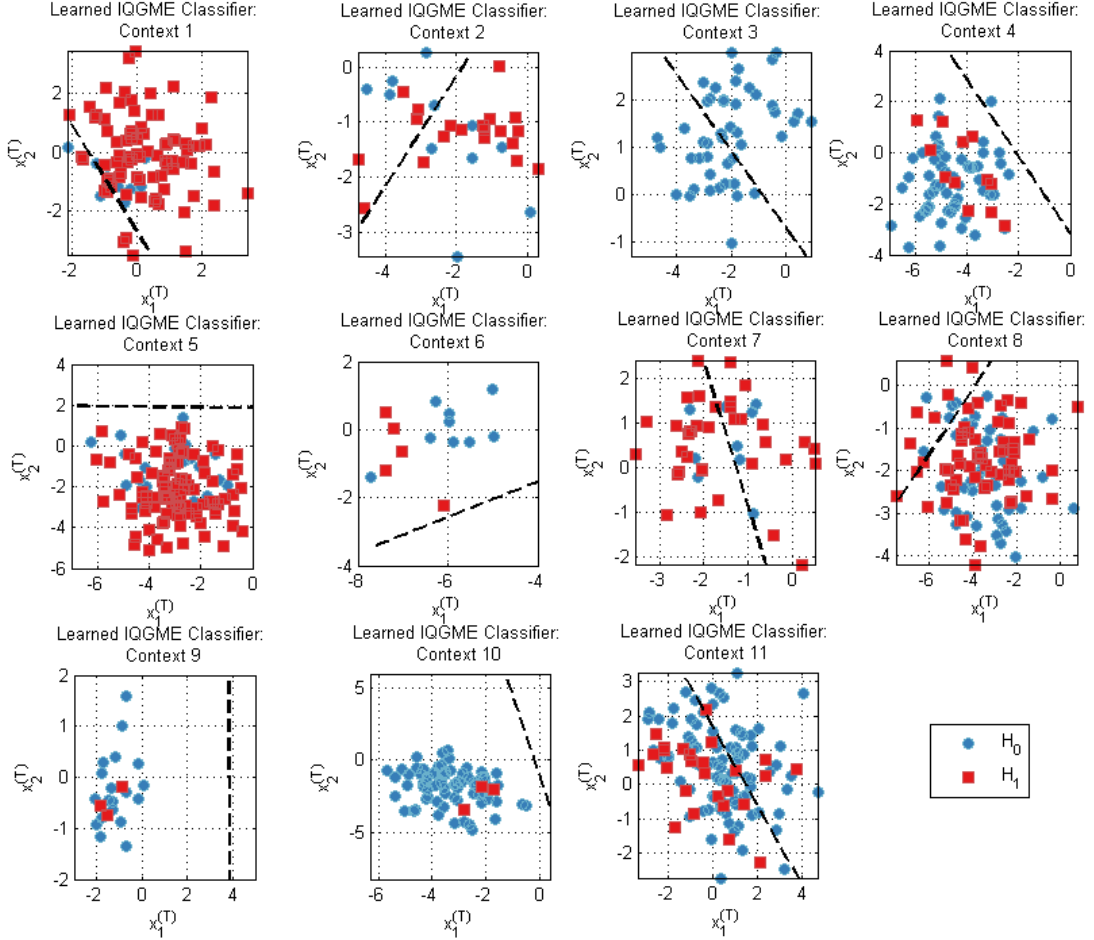


FIGURE 5.12: Component classifiers learned by the IQGME model for the second synthetic data example. Each panel illustrates a two-dimensional scatterplot of the target features, corresponding to points from each learned context, with the decision boundary learned for each context overlaid.

vance of detection algorithms may vary with respect to environment. For this case of simulated data, the target features were 10-dimensional, only two of which were relevant in each context. The two relevant features were drawn from the same distributions as in the previous example, and the irrelevant features were drawn from a Gaussian distribution with zero mean and variance of 2. The first two target features were relevant in Context 1, the last two were relevant in Context 2, features 1 and 10 were relevant in Context 3, and features 5 and 6 were relevant in Context 4. The contextual features were drawn from the same two-dimensional Gaussian

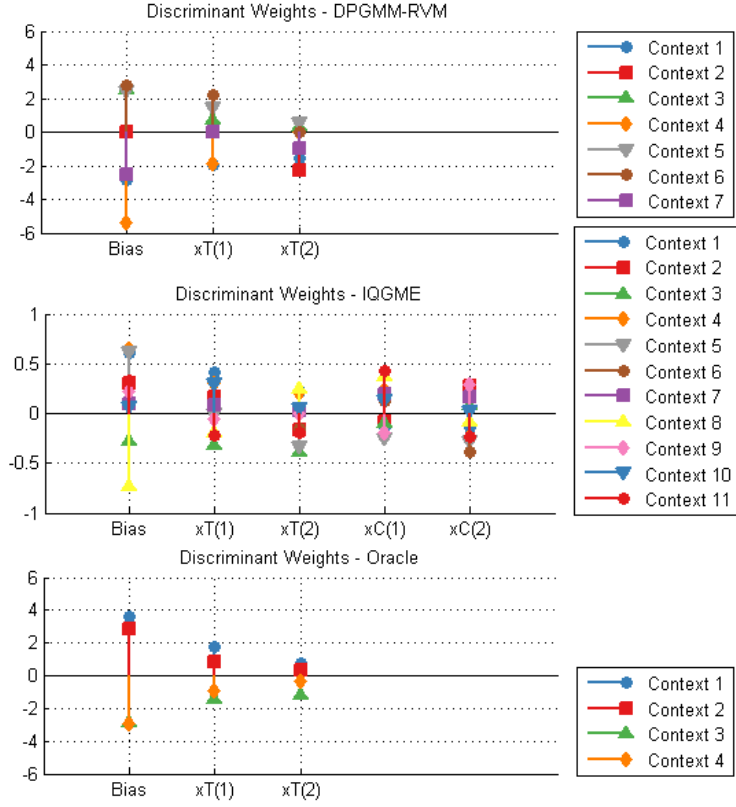


FIGURE 5.13: Discriminant weights learned by the DPGMM-RVM, IQGME, and the context oracle for the second synthetic data example, colored by context. Top: DPGMM-RVM weights; Center: IQGME weights; Bottom: RVM weights based on the context oracle. [21]

distributions used in the second example.

Figure 5.15 illustrates the clustering results obtained from DPGMM-RVM in the contextual feature space. The DPGMM-RVM performed similarly compared to the previous example, learning six contexts. Figure 5.16 illustrates the clustering results obtained from the IQGME. Compared to the previous example, the IQGME learned more contexts. A total of 17 contexts were learned, and like the previous examples, they overlapped heavily in the contextual feature space since clustering was performed on the joint context and target features.

More differences between the performance of the DPGMM-RVM and IQGME in

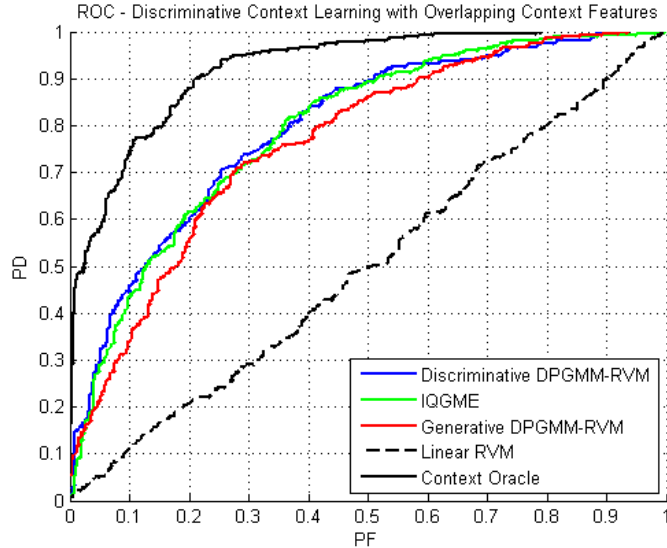


FIGURE 5.14: ROC curves comparing discriminative context-dependent learning on the second synthetic data example. Performance is compared between the DPGMM-RVM (blue), IQGME (green), generative context-dependent learning with the DPGMM-RVM (red), linear RVM learned on target features only (black dashed), and the context oracle (black solid). [21]

the presence of irrelevant features can be seen by analyzing the learned discriminant weights, which are plotted in Figure 5.17. The weights for the DPGMM-RVM are very similar to those learned by the oracle, illustrating that most of the weights for each context are zero, and the relevant features in each context receive nonzero weight. Meanwhile, the IQGME classifiers are not sparse, and most of the target and context features receive a relatively small weight.

The ROC curves comparing the performance of the DPGMM-RVM and IQGME in the presence of irrelevant features are provided in Figure 5.18. In this case, the DPGMM-RVM appears to be more robust than the IQGME since it was able to correctly model the relevance of the target features with respect to context. These results suggest that the DPGMM-RVM may be a superior model for context-dependent learning if different target features are expected to be irrelevant under certain environmental conditions.

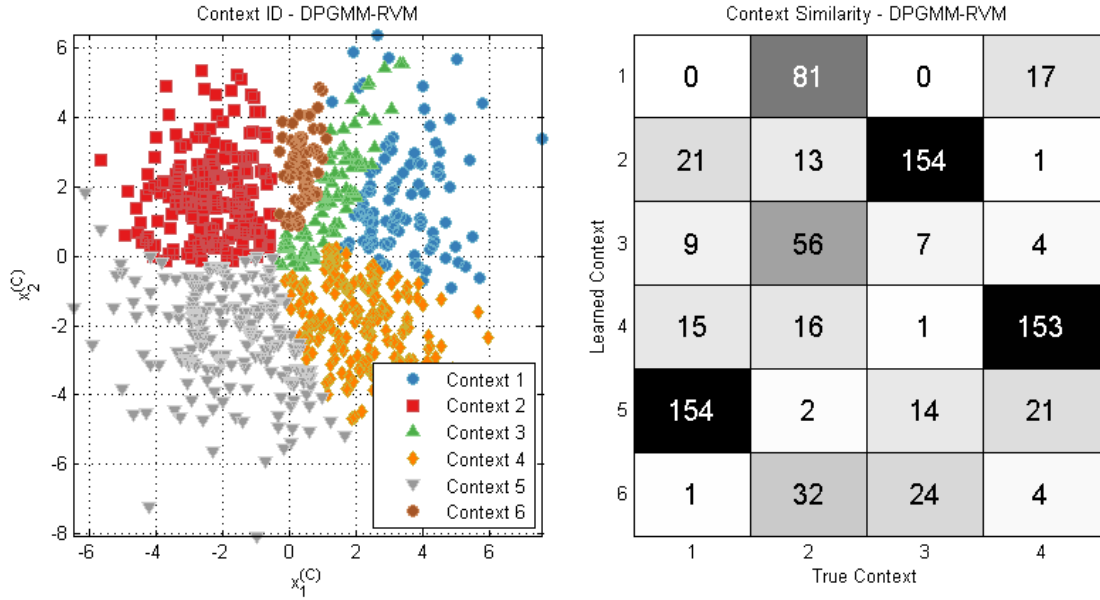


FIGURE 5.15: Results of context identification using the discriminative DPGMM-RVM model for the third synthetic data example. Left: scatterplot of context features, with points colored by MAP context; Right: similarity matrix of true and learned context assignments.

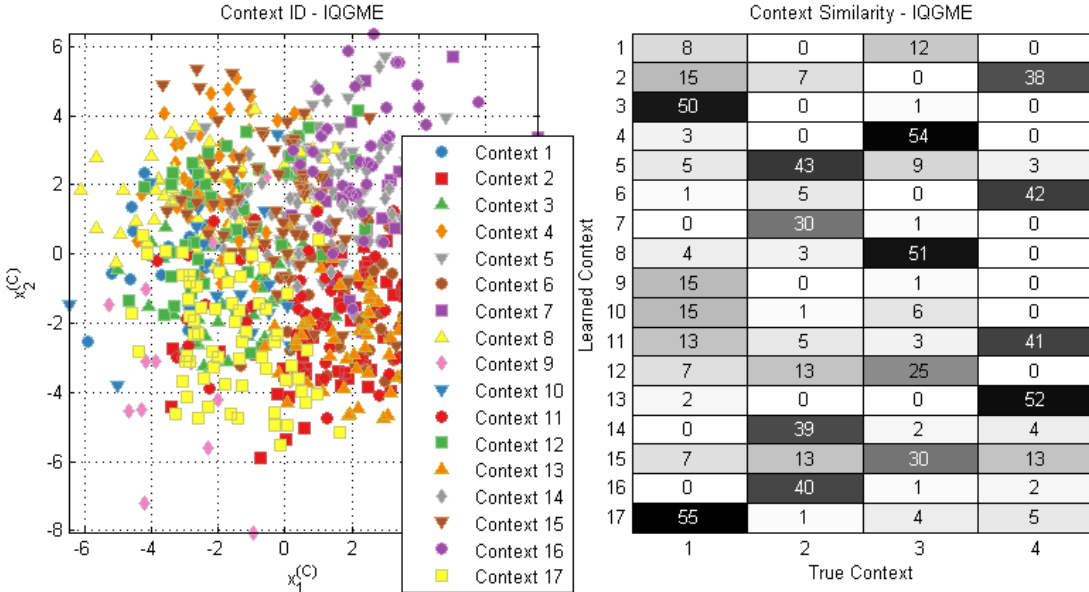


FIGURE 5.16: Results of context identification using the IQGME model for the third synthetic data example. Left: scatterplot of context features, with points colored by MAP context; Right: similarity matrix of true and learned context assignments.

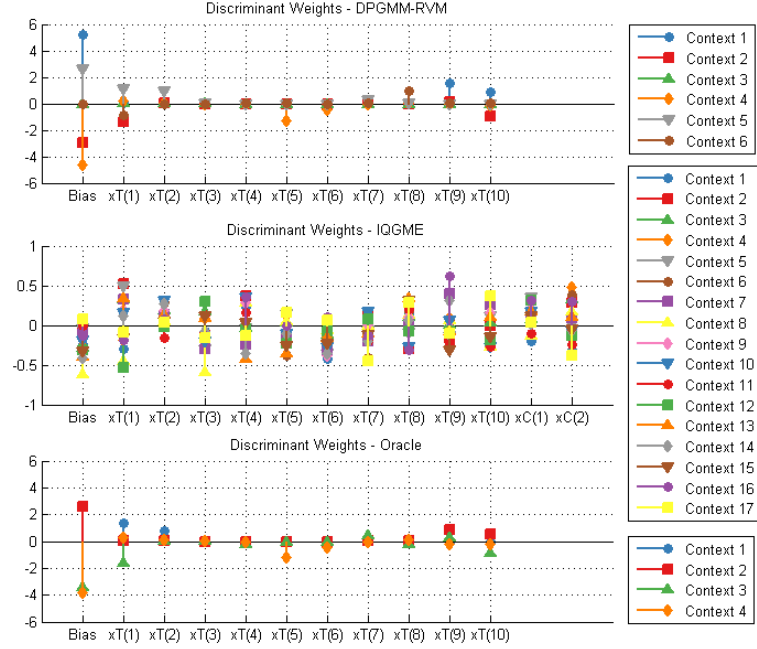


FIGURE 5.17: Discriminant weights learned by the DPGMM-RVM, IQGME, and the context oracle for the third synthetic data example, colored by context. Top: DPGMM-RVM weights; Center: IQGME weights; Bottom: RVM weights based on the context oracle.

In summary, the DPGMM-RVM and the IQGME are two similar approaches to discriminative context learning. However, their behavior on synthetic data highlights important differences as to when each is appropriate to use. The IQGME performs clustering and classification in a common feature space, and therefore is not amenable to sparse classifiers. In contrast, the DPGMM-RVM performs clustering on the contextual features, while also performing classification in the features designed for discriminating targets.

The synthetic data examples showed that in the case where all features are equally informative, and the context features form distinct clusters, generative context learning may be the best approach. However, if the contextual features do not cluster well, discriminative context learning can improve overall classification performance. The final example considered the case in which some target features were non-informative,

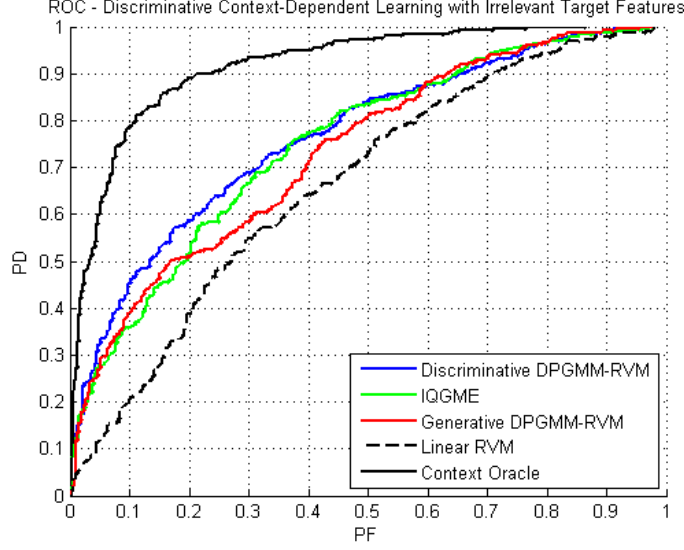


FIGURE 5.18: ROC curves comparing discriminative context-dependent learning on the third synthetic data example. Performance is compared between the DPGMM-RVM (blue), IQGME (green), linear RVM learned on target features only (red solid), linear RVM learned on both sets of features together (red dashed), and the context oracle (black).

illustrating that the DPGMM-RVM can effectively learn the context-dependent relevance of target features.

5.5 Experimental Results with GPR Data

The discriminative DPGMM-RVM and IQGME were used for context-dependent algorithm fusion and evaluated on the GPR data set used in Chapters 3 and 4. The target features consisted of the prescreeener, EHD, HMM, and SPSCF confidence values. As was done for the generative DPGMM in Chapter 4, the context features originally proposed in Chapter 2 were projected to 3-D via PCA. The DPGMM-RVM and IQGME discriminative models were trained using variational inference with the same hyperparameter settings from the synthetic examples. In addition, the truncation level for initializing the DPGMM-RVM was set to $T = 30$, and the truncation level for IQGME was set to $T = 20$. Due to the computational expense

of training these models, both were trained on a subset consisting of 6,864 alarms that included all target alarms and a 3:1 clutter-to-target ratio.

5.5.1 Context Identification Performance

The results of context identification using the DPGMM-RVM are summarized by Figure 5.19, which illustrates a scatterplot of the contextual features colored by soil label and by MAP contexts learned from the DPGMM-RVM. Additionally, Figure 5.20 shows the similarity matrix between the soil labels and DPGMM-RVM contexts. Results illustrate that the DPGMM-RVM learned a total of 21 contexts. This result appears very similar to what was obtained from the generative DPGMM in Chapter 4, which 19 contexts as shown in Figures 4.5 and 4.6. Similarities between generative and discriminative context learning include that the largest contexts contain mostly dirt observations, and that contexts composed of mostly asphalt and concrete data are distinct from those composed of mostly dirt and gravel. Another similarity is that gravel data held a majority in only a few contexts (Contexts 4 and 13), while holding a large minority of the population of many other contexts.

The IQGME behaved differently on the GPR features than it did in the synthetic data example. The IQGME identified *fewer* contexts than the DPGMM-RVM, yielding 13 contexts total. The scatterplots comparing the learned IQGME contexts to the known soil labels are shown in Figure 5.21, and the similarity matrix is shown in Figure 5.22. The scatterplot shows significant overlap of the context assignments, as it did in the synthetic examples. The vast majority of the data fall under Contexts 1 and 2, suggesting that the “typical” classification problem lies in these large contexts.

The similarity matrix shown in Figure 5.23 compares the context identification performance both the DPGMM-RVM and the IQGME. Because both techniques identified a large number of contexts, it was difficult to visually compare the context

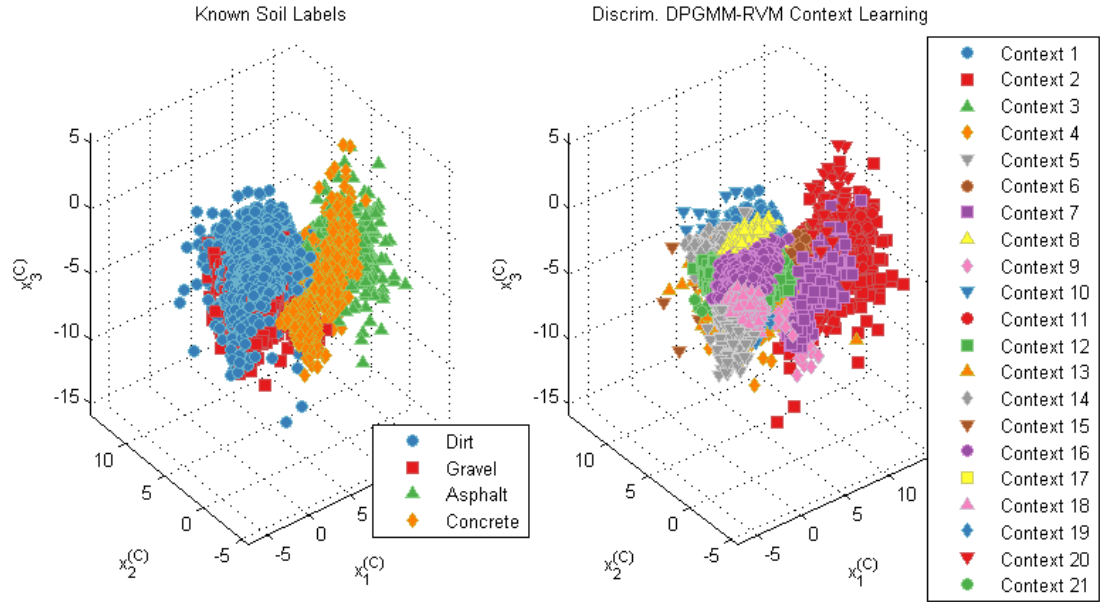


FIGURE 5.19: Scatterplot comparing results of context learning using the discriminative DPGMM-RVM on the GPR contextual features to the known soil labels. Left: Scatter plot of 3-D PCA projection of contextual features, with points colored by qualitative soil label. Right: Same scatter plot, but with points colored by MAP mixture component. [21]

Similarity Matrix					
Discrim. DPGMM-RVM Context	1	99	6	0	0
	2	10	4	86	8
	3	121	30	0	1
	4	47	177	0	5
	5	123	44	0	0
	6	54	26	12	29
	7	4	0	0	158
	8	583	70	0	0
	9	0	12	0	55
	10	137	66	1	4
	11	41	19	130	32
	12	659	339	0	0
	13	83	111	3	9
	14	258	119	0	0
	15	69	3	0	0
	16	1710	157	0	0
	17	72	89	0	0
	18	395	142	0	0
	19	188	7	1	2
	20	9	4	62	112
	21	39	28	0	0
		Dirt	Gravel	Asphalt	Concrete

FIGURE 5.20: Similarity matrix comparing DPGMM-RVM clustering results to the known soil labels.

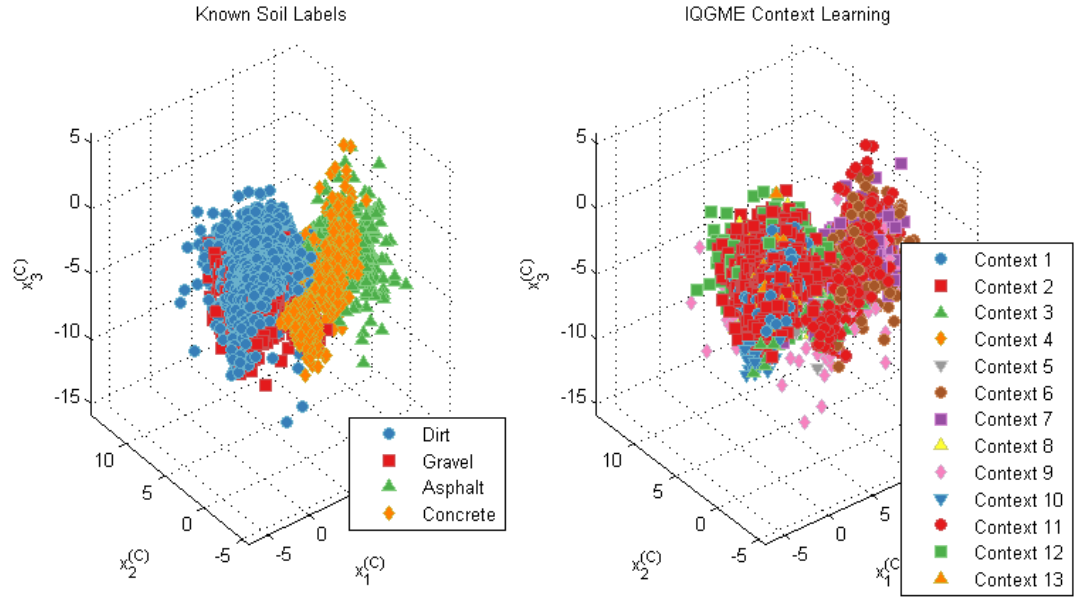


FIGURE 5.21: Scatterplot comparing results of context learning using IQGME on the GPR contextual features to the known soil labels. Left: Scatter plot of 3-D PCA projection of contextual features, with points colored by qualitative soil label. Right: Same scatter plot, but with points colored by MAP mixture component.

Similarity Matrix				
IQGME Context	1	2	3	4
	791	188	0	1
	2637	734	0	2
	166	135	4	28
	161	11	0	0
	116	33	0	5
	6	2	100	73
	41	8	96	70
	71	48	2	4
	39	55	20	21
	41	28	0	0
	3	2	67	190
	303	139	6	21
	326	70	0	0
	Dirt	Gravel	Asphalt	Concrete

FIGURE 5.22: Similarity matrix comparing IQGME clustering results to the known soil labels.

1	9	0	21	0	20	0	0	166	0	2	0	123	0	16	0	441	36	142	0	0	4
2	58	0	89	0	94	0	3	394	1	74	0	703	1	238	0	1246	103	329	4	0	36
3	2	2	3	47	7	35	8	0	16	15	34	2	71	8	28	0	1	0	43	4	7
4	3	0	15	1	3	2	0	17	0	13	0	31	1	21	0	43	4	9	3	0	6
5	1	1	4	38	1	0	1	1	1	21	0	8	30	20	3	7	0	0	14	1	2
6	0	46	0	0	0	6	22	0	8	0	42	0	2	0	0	0	0	0	0	55	0
7	0	25	1	1	0	24	16	0	5	1	72	0	10	0	0	0	0	0	12	48	0
8	11	0	2	23	0	9	0	3	2	13	5	1	14	12	2	1	0	1	26	0	0
9	0	24	0	52	0	4	2	0	11	4	6	0	6	0	15	0	0	0	3	8	0
10	0	0	0	0	31	0	0	1	0	3	0	22	0	1	0	3	3	4	0	0	1
11	0	7	0	0	0	8	110	0	23	1	42	0	1	0	0	0	0	0	0	70	0
12	15	3	6	67	0	33	0	6	0	59	21	19	70	43	24	5	0	0	93	1	4
13	6	0	11	0	11	0	0	65	0	2	0	89	0	18	0	121	14	52	0	0	7
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

FIGURE 5.23: Similarity matrix comparing IQGME context identification to DPGMM-RVM context identification. The horizontal axis represents the DPGMM-RVM contexts, and the vertical axis represents the IQGME contexts. The AMI of the two clusterings [110] is shown at top. [21]

assignments from the scatterplots in Figures 5.19 and 5.21. Instead, the adjusted mutual information (AMI) [110] was used to compare the results of context identification. The AMI can be used to compare two clusterings, each having different numbers of clusters, while correcting for the effect of chance agreement. The range of AMI is between zero and one; an AMI of one would be obtained for two identical clusterings, and an AMI of zero would be obtained for two clusterings with only chance similarity. The AMI between the contexts identified by the discriminative DPGMM-RVM and IQGME was 0.3002. Although there appears to be strong overlap between IQGME Contexts 1 and 2 and DPGMM-RVM Contexts 8, 12, 16, and 18, which contain the majority of observations and mostly correspond to the dirt soil type, the low AMI metric suggests that little information is shared between the clusterings. However, based on results from the synthetic data examples, the low degree of similarity between the DPGMM-RVM and IQGME contexts was expected.

5.5.2 Context-Dependent Fusion Results

The discriminant weights learned for the DPGMM-RVM and IQGME are shown in Figure 5.24. The DPGMM-RVM weights are shown in the top panel, and the IQGME weights are shown at the bottom. The first four dimensions are the target features, and in the case of IQGME, the last three are the context features. For the DPGMM-RVM, the weights on the feature values (not the bias) are mostly either positive or zero. The one exception to this is the prescreener weight in Context 8. Therefore, the DPGMM-RVM weights could be interpreted as each algorithm being either relied upon or ignored in each context, and only rarely discounted.

However, the IQGME weights for the target features appear somewhat evenly distributed around zero; some are positive, and others are negative. This suggests that the local classification problems discovered by IQGME are substantially different than those found by the DPGMM-RVM, and the negative weights will cause fusion to discount certain algorithms' confidences for some contexts. Therefore, fusion would tend to make a decision *opposite* of what the negatively-weighted algorithms may indicate in those contexts.

5.5.3 Detection Performance

The discriminative context-dependent fusion techniques were evaluated using the same cross-validation folds that were used to compute the ROC curves presented in Chapters 3 and 4. Both discriminative learning techniques, the DPGMM-RVM and IQGME, were evaluated and compared to the generative DPGMM-RVM presented in Chapter 4 as well as conventional fusion with a linear RVM.

Figure 5.25 illustrates the ROC curves obtained for each of the fusion approaches that were evaluated, as well as the prescreener, EHD, SPSCF and HMM algorithms. The global RVM curve, shown by the black solid line, is plotted along with a shaded region indicating the 90% confidence region. The ROC curve for the generative

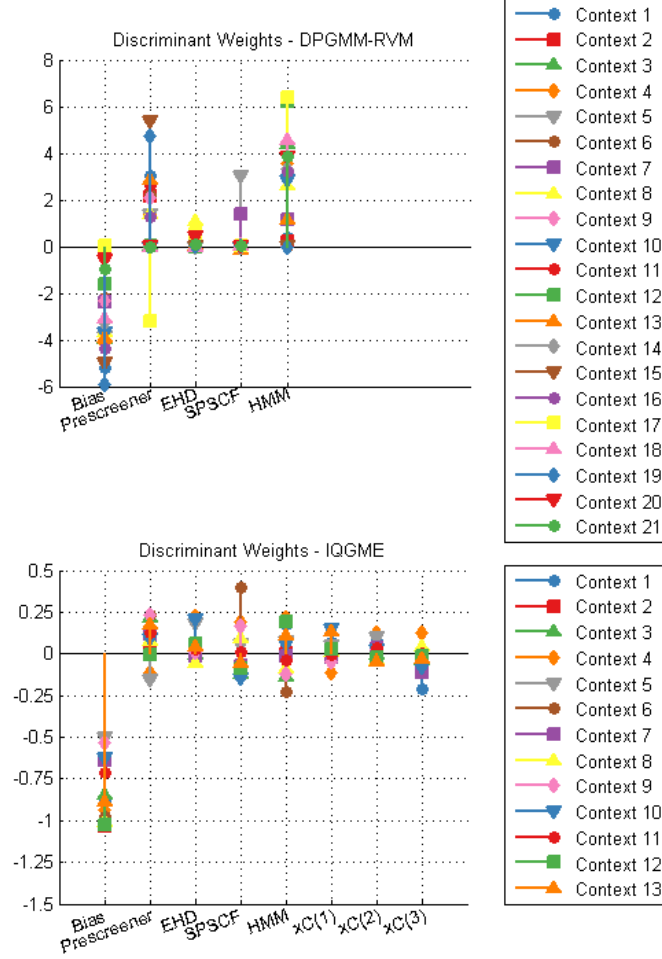


FIGURE 5.24: Discriminant weights learned by the DPGMM-RVM and IQGME for algorithm fusion on the GPR data set. Top: DPGMM-RVM weights; Bottom: IQGME weights.

DPGMM-RVM, originally shown as the red line in Figure 4.15, is plotted for reference. The discriminative DPGMM-RVM is shown by the green line, and the discriminative IQGME by the blue line. Results show that all three context-dependent fusion techniques yield significantly better performance than the single RVM. The discriminative context-dependent fusion techniques both show a lower FAR than the generative technique at low PD. The three ROC curves cross around $PD=0.65$. From $0.65 < PD < 0.85$, the generative context-dependent approach has the best performance. At higher PD, the performance of generative context-dependent fusion and

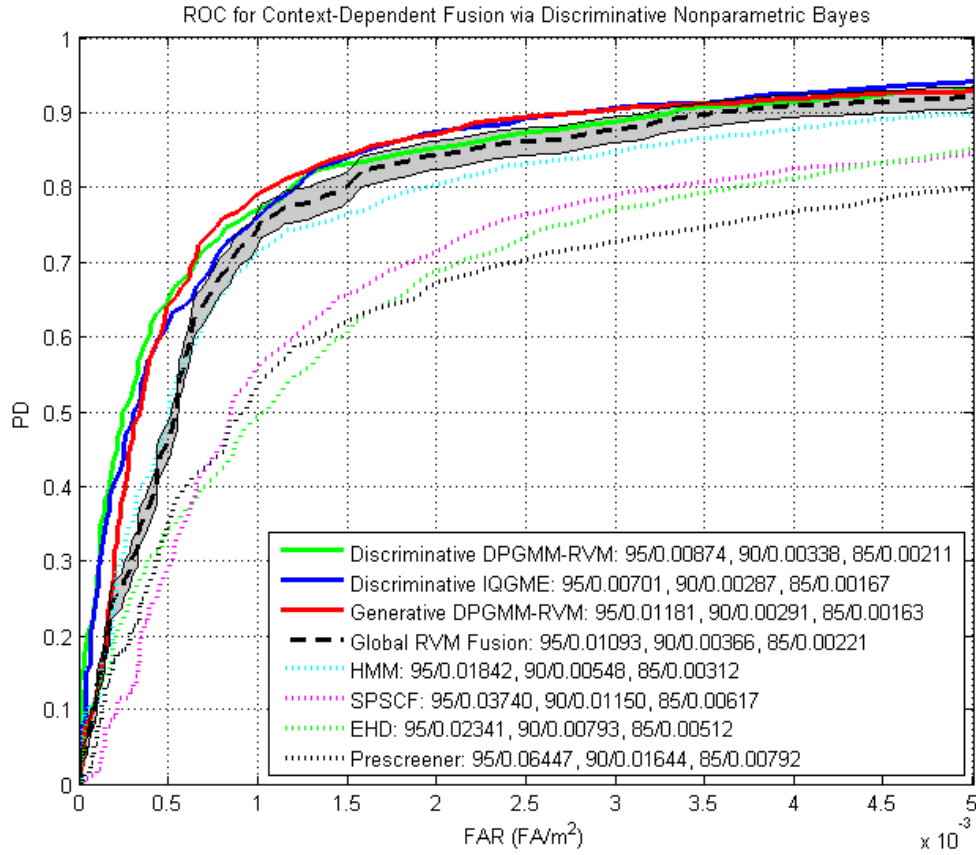


FIGURE 5.25: ROC curves for discriminative context-dependent fusion using the IQGME (blue) and DPGMM-RVM (green) compared to generative context-dependent fusion (red), non-context-dependent RVM fusion (black dashed), and the individual fused algorithms (dotted). The ROC consists of PD versus FAR, measured in false alarms per square meter, as a function of decision threshold. [21]

the discriminative IQGME are similar, while the discriminative DPGMM-RVM is not significantly better than the single RVM.

Of the three synthetic data examples, the results presented in Figure 5.25 appear to be most similar to the first case. Although it was expected that discriminative context-dependent fusion would yield the best performance, neither approach outperformed the generative context-dependent fusion technique presented in Chapter 4. However, it is interesting to see that both discriminative approaches performed sim-

ilarly despite incorporating contextual information in different ways. Furthermore, the similarity in performance between both discriminative approaches suggests that the target features are relevant across all contexts. Therefore, enforcing sparseness in the DPGMM-RVM discriminant weights did not improve performance. The superior performance of the generative approach suggests that the context features already cluster with respect to relevant contextual factors, and discriminative context learning may not be necessary.

5.6 Conclusion

In this chapter, two potential methods for discriminative context learning were presented. The first approach, referred to as the discriminative DPGMM-RVM, was based upon the generative techniques presented in the previous chapter but was learned based on the joint likelihood of the contextual features and class labels. The second approach, the IQGME, is similar in that the gating network is based on the DPGMM. However, the local experts are not sparse, and classification and clustering are performed on the joint target and context features.

Several examples using synthetic data were used to illustrate the differences in behavior between the two discriminative context learning approaches. The first example considered two-dimensional context and target features, in which all were informative. Comparison of the DPGMM-RVM and IQGME showed similarities in classification performance, although the contexts that were learned were substantially different. The discriminant weights learned for the IQGME were smaller in magnitude than those learned for the DPGMM-RVM, since the IQGME also incorporated context features. Furthermore, the IQGME appeared over-trained since it learned contexts consisting of only one class of data. Therefore, the DPGMM-RVM led to better performance, although generative context-dependent learning performed slightly better. The second example was similar to the first, with the only

difference being that the context features were less informative. In this case, both discriminative context-dependent classifiers performed similarly and yielded substantial performance improvements over generative context-dependent learning.

The third synthetic data example considered the case in which some target features were irrelevant, depending on the context. The DPGMM-RVM accurately identified the features that were relevant in each context. The IQGME yielded a model that was much less sparse, in terms of the number of learned contexts, than the DPGMM-RVM. In this example, the DPGMM-RVM achieved performance gains over the IQGME due to its ability to learn which features were relevant in which context.

For experiments with GPR data, it was expected that discriminative context-dependent learning would yield results similar to the second and third examples. However, experimental results appear to be similar to the first synthetic example. The similarity in performance of both discriminative techniques suggest that all of the target features may be relevant across contexts. Furthermore, the fact that generative context-dependent learning yielded better performance suggests that the proposed features are very informative of the underlying contextual factors, and that incorporating more information through discriminative context learning may not be necessary.

Additional sources of contextual information should still be considered for improving performance. One potential source is the *spatial* distribution of the context features. The context learning techniques presented up to this point considered individual prescreener alarms as statistically independent observations. However, a wealth of contextual information may be available in the large stretches of target-free data collected between prescreener alarms. By regularly sampling the background to extract contextual features, spatially-distributed contextual factors may be discovered. This information can be valuable for inferring the underlying context well

before a prescreener alarm is recorded. The following chapter investigates two techniques for achieving this goal through nonparametric spatial context modeling.

Nonparametric Spatial Context Models

In military route clearance applications, a vehicular GPR system such as the NI-ITEK HMDS may lead a convoy over many kilometers through varying terrain while searching for buried explosive threats. In Chapters 3-5, several context learning techniques were proposed for exploiting information regarding terrain differences to improve the detection performance achieved by algorithm fusion. These techniques utilized contextual information extracted near recorded prescreener alarms, and all alarms were treated as independent observations. In practice, it may be more advantageous to regularly extract contextual features from the background, and utilize the spatial dependency of observations for better inference of the underlying context.

This chapter proposes two methods for *nonparametric spatial context modeling*. While the previously-discussed context models operated on an alarm-by-alarm basis, the models proposed in this chapter are used to infer context as a function of space. This is achieved by extracting contextual features at regular downtrack intervals and performing inference on each sample. The first context model that will be considered is the DPGMM, which was originally presented for generative alarm-based context learning in Chapter 4. The second model to be considered is the stick-breaking

hidden Markov model (SBHMM), which is a nonparametric extension of the HMM originally proposed by Paisley and Carin [69]. Like the DPGMM, the SBHMM employs a stick-breaking prior to facilitate learning of the model’s order. However, unlike the DPGMM which assumes all samples are independent, the SBHMM context model allows for spatial dependency between samples.

6.1 Spatial Context Sampling

The context modeling approaches proposed in this chapter utilize contextual features extracted from the background at regular intervals over a given area. The feature extraction process is referred to as *context sampling*. There are several reasons for using context sampling as opposed to extracting features from prescreener alarms. The primary reason is that in route clearance patrols, the vast majority of GPR data collected in the field will be free of buried threats. In current processing strategies, the large stretches of background data are generally ignored after prescreening [41]. Although this background data may be target-free, it could potentially be rich in contextual information. Another reason to motivate context sampling is that certain contextual factors may be spatially-distributed. For example, consider a desert gulch, a local region of low elevation where moisture may accumulate in the event of a flash flood. It would be expected that the soil in a recently washed-out area may contain more moisture than surrounding areas at higher elevations.

Consider the example shown in Figure 6.1. The top panel illustrates raw GPR data collected on a concrete test lane, and the anomalies occurring around time sample 200 correspond to landmine signatures. In the late-time portion of the B-scan, a faint subsurface layer emerges around downtrack sample 1000 and becomes stronger around downtrack sample 2800. A second subsurface layer appears around down-track sample 4300. These distinct regions characterized by different subsurface layer responses could possibly correspond to unique, spatial context regions as illustrated

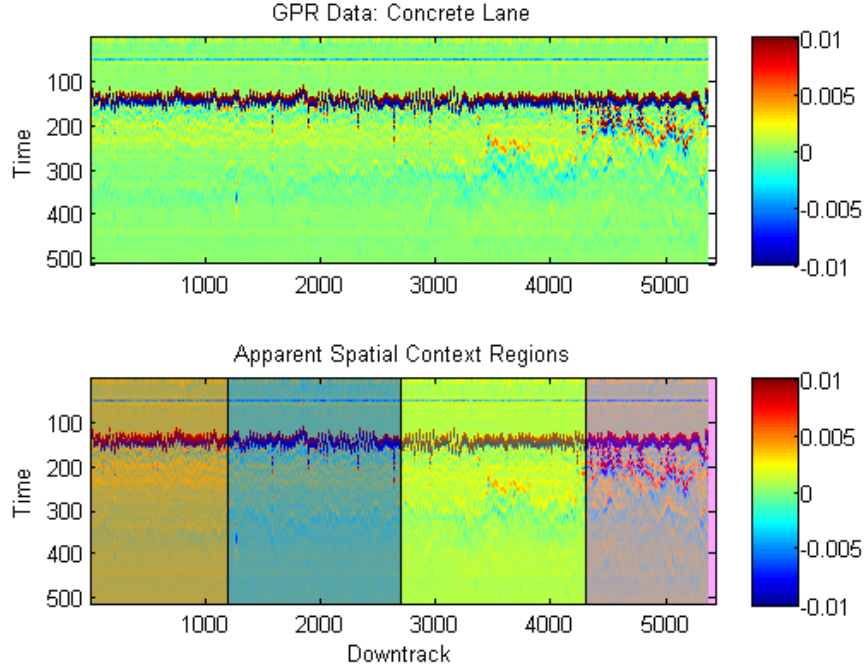


FIGURE 6.1: Example of GPR data collected on a concrete lane and apparent spatial context regions. Top: raw GPR data, Bottom: raw GPR data with apparent spatial contexts indicated by different shaded regions. Downtrack position is represented by the horizontal axis in both panels.

by the bottom panel.

In this work, context sampling was performed by extracting the contextual features proposed in Chapter 2 from the background at regular 10 cm intervals. The sequence of background features is denoted by $\mathbf{X}^{(C)} = [\mathbf{x}_1^{(C)}, \mathbf{x}_2^{(C)}, \dots, \mathbf{x}_N^{(C)}]$, where N is the length of the sequence. Although features were extracted from this data off-line, it is understood that real-time implementation will be necessary in fielded applications. Therefore, the sampling interval may need to be increased to facilitate real-time processing. Furthermore, context features were only extracted from the center channel (channel 24) of the GPR array. Although more contextual information could potentially be exploited by sampling the other channels, incorporating features from the other channels did not improve performance. In a similar vein

to previously-discussed context modeling techniques, the features were projected to 3-D via PCA. Since PCA implies an underlying Gaussian distribution, it facilitated training spatial context models based on Gaussian distributions. Furthermore, three principal components were used because the alarm-based DPGMM context model also performed best using the same number of components.

Figure 6.2 illustrates a zoomed-in portion of the GPR data shown in Figure 6.1 to illustrate the context sampling interval and how the background features are illustrative of contextual transitions. The top panel illustrates the portion of the lane where the early-time subsurface layer appears around downtrack sample 4275. The dashed lines represent the downtrack samples from where context features were extracted from the background. In the bottom plot, the contextual shift is reflected by a change in the values of the second principal component of the context features. It appears that there is a latency of about 30 samples from where the shift occurs and where the feature values change. This is likely due to the fact that features are extracted *causally*, using the 100 A-scans preceding each sample point.

After the feature sequence is extracted from the background, it is processed by a statistical context model. The context model yields posterior context probabilities, $p(c_{nm} = 1 | \mathbf{x}_n^{(C)})$, for each sample ($\mathbf{x}_n^{(C)}$) for $n = 1, 2, \dots, N$. If a prescreener alarm falls between two samples, it is associated with the context posterior of the earlier sample. Context posteriors for several distinct test lanes are illustrated in the experimental results presented in Section 6.4.1.

In this chapter, two spatial context models are proposed. Both models were learned using the generative approach. The first is an extension of the DPGMM originally presented in Chapter 4. The second is based upon the SBHMM, originally developed by Paisley and Carin [69]. While the DPGMM approach to context modeling treats all samples as independent observations, the SBHMM exploits dependencies between neighboring samples. The two context models are described in

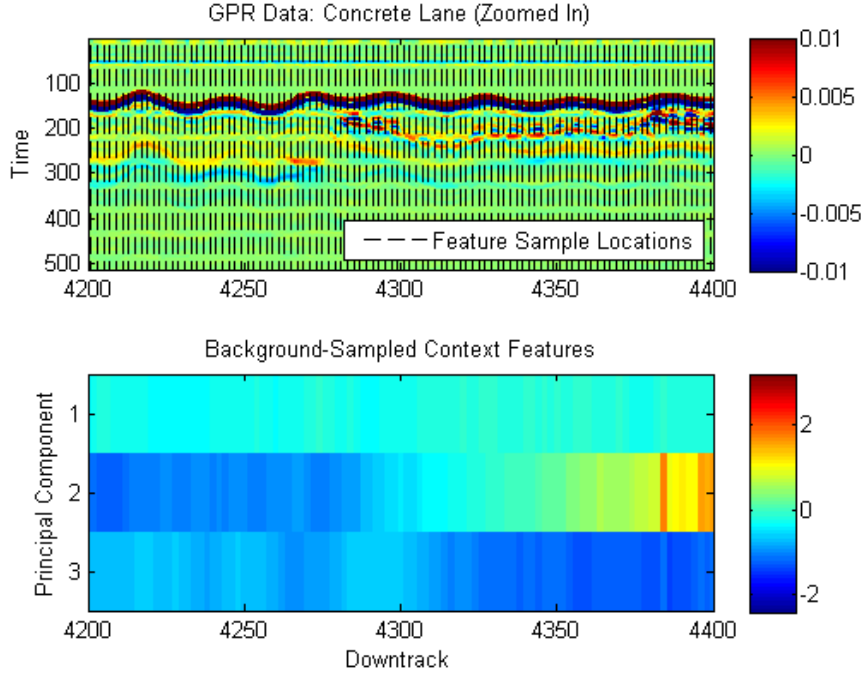


FIGURE 6.2: GPR data from Figure 6.1, zoomed in to illustrate background sampling near a contextual shift. Top: raw GPR data, with feature extraction locations noted by dashed lines, Bottom: 3-D PCA of background contextual features. Downtrack position is represented by the horizontal axis in both panels.

greater detail in the following sections.

6.2 DPGMM Spatial Context Model

The DPGMM was proposed in Chapter 4 for generative context learning, and was also utilized in Chapter 5 as the gating network for discriminative context learning. In both cases, the DPGMM was used to model the distribution of context features corresponding to prescreener alarms. In the case of spatial context modeling, the DPGMM was trained on the three-dimensional PCA projection of the contextual feature sequence ($\mathbf{X}^{(C)}$) extracted from regular background samples.

Refer to Section 4.3 for a description of the DPGMM generative model and likelihood functions. Details on VB inference of the model parameters that was developed

for this application can be found in Appendix C. The DPGMM was learned using the same hyperparameter settings as in Chapter 4: $u_0 = 1$, $\tau_{10} = \tau_{20} = 1$, $\nu_0 = D^{(C)}$, $\mathbf{B}_0 = D^{(C)}\mathbf{I}_{D^{(C)}}$, and $\boldsymbol{\rho}_0$ was set equal to the sample mean of $\mathbf{X}^{(C)}$. Note that since PCA is being used on the features, $D^{(C)} = 3$. Additionally, the truncation level T was set to 30. The only major difference in implementation was the cluster pruning criterion, which was set at 5% to prevent too many small contexts from being learned.

Recall the DPGMM likelihood function given by (4.27) and its data-generating process; by modeling context as the latent variable governing draws from a mixture of Gaussians, observations are treated as statistically independent. In terms of the Chinese restaurant process, which was described in Section 4.2, each customer selects a table based only on the *number* of people seated at each table and not necessarily what the previous customer’s choice was.

However, it was mentioned earlier that certain contextual factors may be spatially-distributed. The spatial dependency between feature samples may be a useful source of contextual information. The following section proposes using a nonparametric variant of the HMM to model context as a spatially-varying *state* underlying the background features $\mathbf{X}^{(C)}$.

6.3 SBHMM Spatial Context Model

The HMM is a popular choice for modeling time series that are dependent on an underlying state variable that is not directly observed, but can be inferred from data. While most notably used in speech recognition applications [111], HMMs have also been explored for modeling polyphonic music recordings [112, 113], speaker diarization [114], handwriting recognition [115, 116], acoustic sensing [117], and landmine detection [42, 48].

The HMM follows the structure of a Markov chain, in which a data sequence

$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ is assumed to be in one of M states at a given index n , i.e., $s_n \in \{S_1, S_2, \dots, S_M\}$, where M is the order of the model. The HMM incorporates a degree of statistical dependency between observations in the sequence through the *Markov property*, which states that the state of the current observation is only dependent on the state of the previous observation:

$$p(s_{n+1} = S_m | s_n = S_j, s_{n-1}, \dots, s_1) = p(s_{n+1} = S_m | s_n = S_j). \quad (6.1)$$

The “hidden” aspect of an HMM is that the underlying state is treated as an unknown latent variable. However, the state sequence can be inferred from $\mathbf{X}$ given the model parameters, $\{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\Theta}\}$. The $M \times 1$ vector, $\boldsymbol{\pi}$, consists of the *initial state probabilities*, which are given by

$$\pi_m = p(s_1 = S_m), \quad m = 1, 2, \dots, M, \quad (6.2)$$

and satisfy the following properties:

$$0 \leq \pi_m \leq 1 \quad (6.3)$$

$$\sum_{m=1}^M \pi_m = 1 \quad (6.4)$$

The $M \times M$ matrix, $\mathbf{A}$, consists of the *state transition probabilities* which are given by

$$a_{mj} = p(s_{n+1} = S_m | s_n = S_j), \quad m = 1, 2, \dots, M, \quad j = 1, 2, \dots, M, \quad (6.5)$$

are assumed to be constant with respect to time, and satisfy the following properties:

$$0 \leq a_{mj} \leq 1 \quad (6.6)$$

$$\sum_{j=1}^M a_{mj} = 1 \quad (6.7)$$

Finally, θ_i are the parameters for the *emission densities*, $p(\mathbf{x}_n | s_n = S_m)$. There is no restriction on the form of the emission densities, the only requirement being that they are valid PDFs.

The conventional method for learning the parameters of an HMM is via the Baum-Welch algorithm, which performs maximum-likelihood estimation for a model with fixed order (i.e., known number of states). Given a trained HMM, the Viterbi algorithm can then be used for calculating the most probable state sequence for a given observation sequence. Details regarding the Baum-Welch and Viterbi algorithms can be found in [111], while implementation details are discussed in [118].

Earlier work suggested modeling context in GPR data as a spatially-dependent state variable using an HMM of fixed order [119]. While a spatially-dependent HMM context model showed potential for improvement over alarm-based context-dependent fusion, performance varied significantly with respect to the number of states (contexts) being considered. Like GMMs, HMMs are susceptible to over- or under-training if the model order is specified incorrectly. A poorly-trained context model can then lead to poor performance in context-dependent fusion. Fortunately, the DP offers a potential solution to this problem as it did in the case of the GMM context model.

Since the elements of $\boldsymbol{\pi}$ and the rows of $\mathbf{A}$ are constrained to sum to one, they can be treated as parameters of a multinomial distribution from which the underlying state is drawn at any given point in the sequence, $\mathbf{X}$. If an HMM is assumed to be *infinite-order*, the DP can be used as a sparseness-promoting prior on the number of states since it is conjugate to the multinomial distributions parameterized by $\boldsymbol{\pi}$ and $\mathbf{A}$. Several methods for incorporating DP priors into HMM inference rely on Markov chain Monte Carlo (MCMC) sampling to approximate the posterior probabilities [114, 120]. To maintain consistency with the VB techniques used in the previous chapters, this work utilizes the VB approach based on the stick-breaking construction

as proposed in [69]. This model is referred to as the stick-breaking HMM (SBHMM).

The SBHMM imposes a stick-breaking prior on the rows of $\mathbf{A}$ as well as on $\boldsymbol{\pi}$ to facilitate learning an effective number of states given the training sequences. The priors are given by:

$$a_{mj} = v_{mj}^A \prod_{k=1}^{j-1} (1 - v_{mk}^A) \quad (6.8)$$

$$\pi_m = v_m^\pi \prod_{k=1}^{m-1} (1 - v_k^\pi) \quad (6.9)$$

where

$$v_{mj}^A \sim \text{Beta}(1, \alpha_{mj}^A) \quad (6.10)$$

$$v_m^\pi \sim \text{Beta}(1, \alpha_m^\pi) \quad (6.11)$$

Recall the discussion regarding stick-breaking priors from Section 4.2. If the distribution G is drawn from a stick-breaking process, model parameters drawn from G will take on distinct values θ_j^* , $j = 1, 2, \dots, \infty$, and G therefore translates to the discrete density given by (4.26). In the case of the HMM, the latent variable governing the mixture proportions corresponds to the underlying state. Therefore, imposing the stick-breaking prior on the state transition probabilities assumes that G is state-dependent and each state shares the same θ_j^* , such that,

$$G_n(m) = \begin{cases} \sum_{j=1}^{\infty} \pi_j \delta_{\theta_j^*}, & \text{if } n = 1 \\ \sum_{j=1}^{\infty} a_{mj} \delta_{\theta_j^*}, & \text{if } n > 1 \end{cases} \quad \forall m = 1, 2, \dots, \infty \quad (6.12)$$

VB inference can be performed on the SBHMM by assuming a truncation level T on the number of states and conjugate priors on all model parameters, including the parameters of the emission densities. In this work, the emission densities were treated as multivariate Gaussian with unknown mean and covariance, and therefore have Normal-Wishart priors. Therefore, the SBHMM used as a context model in

this work is very similar to the DPGMM proposed in the previous section, with the Markov property being the only major difference between the two. The data-generating process for the SBHMM used in this work is as follows:

1. For $m = 1, 2, \dots, T$
 - (a) Draw $\alpha_m^\pi \sim \text{Gamma}(c_0, d_0)$
 - (b) Draw $v_m^\pi | \alpha_m^\pi \sim \text{Beta}(1, \alpha_m^\pi)$
 - (c) Calculate initial state probabilities $\pi_m = v_m^\pi \prod_{k=1}^{m-1} (1 - v_k^\pi)$
 - (d) For $j = 1, 2, \dots, T$
 - i. Draw $\alpha_{mj}^A \sim \text{Gamma}(c_0, d_0)$
 - ii. Draw $v_{mj}^A | \alpha_{mj}^A \sim \text{Beta}(1, \alpha_{mj}^A)$
 - iii. Calculate state transition probabilities $a_{mj} = v_{mj}^A \prod_{k=1}^{j-1} (1 - v_{mk}^A)$
 - (e) Draw $\boldsymbol{\theta}_m^* | G_0 \sim \mathcal{N}(\boldsymbol{\mu}_m^* | \boldsymbol{\rho}_0, u_0^{-1} \boldsymbol{\Lambda}_m^{*-1}) \mathcal{W}(\boldsymbol{\Lambda}_m^* | \mathbf{B}_0, \nu_0)$
2. For $n = 1, 2, \dots, N$
 - (a) Draw indicator variable $s_n \sim \begin{cases} \text{Multinomial}(\boldsymbol{\pi}), & \text{if } n = 1 \\ \text{Multinomial}(\mathbf{a}_{s_{n-1}}) & \text{if } n > 1 \end{cases}$
 - (b) Draw data $\mathbf{x}_n^{(C)} | s_{nm} = 1 \sim \mathcal{N}(\mathbf{x}_n^{(C)} | \boldsymbol{\theta}_m^*), m = 1, 2, \dots, M$

The SBHMM emission densities were initialized into $T = 30$ clusters using k -means. The following hyperparameter settings were used for all experiments in this chapter, as recommended in [69]: $u_0 = 1$, $c_0 = 10^{-6}$, $d_0 = 0.1$, $\nu_0 = D^{(C)}$, $\mathbf{B}_0 = D^{(C)} \mathbf{I}_{D^{(C)}}$, and $\boldsymbol{\rho}_0$ was set equal to the sample mean of $\mathbf{X}^{(C)}$.

The following synthetic data example illustrates the performance of the SBHMM in modeling synthetic data. Figure 6.3 illustrates the parameters of an four-state HMM from which 100 sequences of length 25 were drawn. The emission densities are

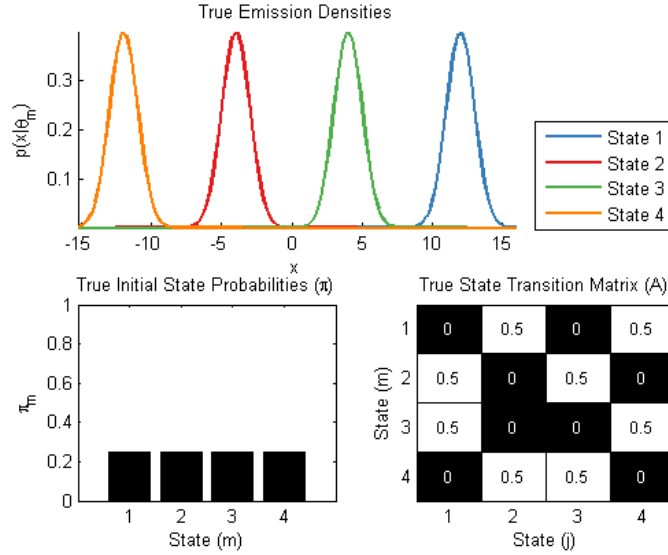


FIGURE 6.3: True parameters for the SBHMM synthetic data example. The top panel illustrates the emission densities, the bottom-left panel illustrates the initial state probabilities, and the bottom-right panel illustrates the state transition probability matrix.

four Gaussian distributions with means of -12, -4, 4, and 12 with unit variance. The initial state probabilities are uniform, i.e. $\pi_m = 0.25$, for $m = 1, 2, 3, 4$. Finally, the state transition matrix shows no probability of remaining in any given state - each state has equal probability of transitioning to one of two other states.

The SBHMM was learned using VB inference with a NFE convergence threshold of 10^{-4} . After convergence, states with too few samples were eliminated. The expected number of state transitions (denoted as $\tilde{\mathbf{A}} = \{\tilde{a}_{ij}\}$) was calculated from the variational posteriors on $\mathbf{A}$ to yield the top panel of Figure 6.4. To calculate the expected overall state occupancy, the columns of $\tilde{\mathbf{A}}$ were summed to yield the values shown in the bottom panel of Figure 6.4. All states with an occupancy of less than 1% were pruned from the model, and the remaining initial and transition probabilities were renormalized to sum to one.

The model parameters which remained after pruning are shown in Figure 6.5. In this example, all of the true parameters were approximated very closely. By

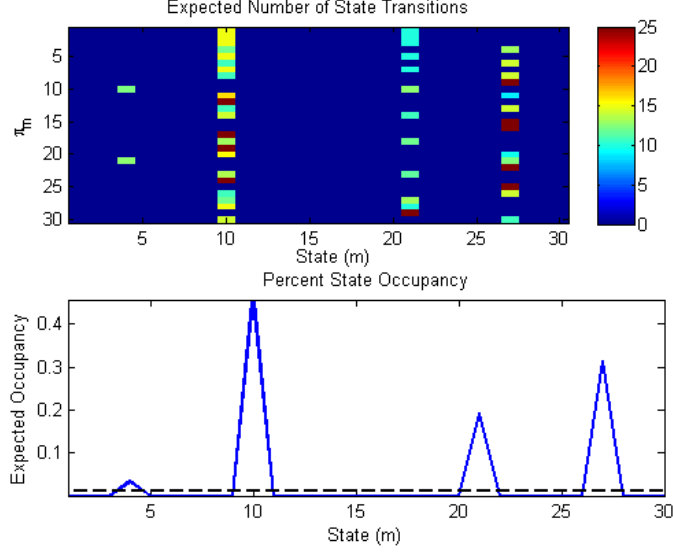


FIGURE 6.4: Illustration of state pruning from converged SBHMM. The top panel illustrates the expected number of state transitions as calculated from the variational posteriors. The bottom panel illustrates the expected state occupancy, with the 1% occupancy threshold shown.

comparing these results to the true HMM parameters shown in Figure 6.3, it is clear that the learned State 1 corresponds to the true State 2, the learned State 2 corresponds to the true State 1, the learned State 3 corresponds to the true State 4, and the learned State 4 corresponds to the true State 3.

For use as a GPR context model, an SBHMM was trained on sequences of PCA-projected background features ($D^{(C)} = 3$) using VB inference with the same hyperparameter settings as in the synthetic data example. After learning converged to a solution and extraneous states were pruned with a 5% occupancy criterion, the causal state posteriors at each downtrack position are given by the *forward variable*, α :

$$\alpha_n(m) = p\left(\mathbf{x}_1^{(C)}, \mathbf{x}_2^{(C)}, \dots, \mathbf{x}_n^{(C)}, s_{nm} = 1 | \boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\Theta}\right), \quad (6.13)$$

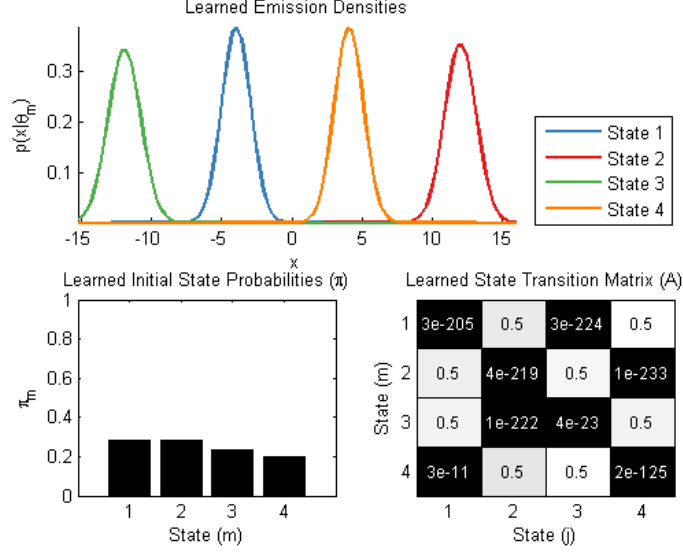


FIGURE 6.5: Learned parameters for the SBHMM synthetic data example. The top panel illustrates the learned emission densities, the bottom-left panel illustrates the learned initial state probabilities, and the bottom-right panel illustrates the learned state transition probability matrix.

where α is computed recursively via the following:

$$\alpha_1(m) = \pi_m p(\mathbf{x}_1^{(C)} | s_{nm} = 1) \quad (6.14)$$

$$\alpha_{n+1}(m) = \left[\sum_{m=1}^M \alpha_n(m) a_{nm} \right] p(\mathbf{x}_{n+1}^{(C)} | s_{nm} = 1). \quad (6.15)$$

The forward variable allows for the context of a given downtrack position to be computed using only the current and prior samples. Although the Markov property assumes that the state of a given sample is only dependent on the state of the previous sample, the recursive update allows for spatial dependency to be a factor in determining the context posterior of any location in the background sequence. When a prescreener alarm is encountered on the lane at location n , it is assigned the context posterior corresponding to the background sample $\mathbf{x}_n^{(C)}$. If the alarm falls between two background samples, the earlier sample's context posterior is used.

As in the previous approaches, the context posteriors were used in training an

ensemble of RVMs for context-dependent algorithm fusion on the prescreener, EHD, HMM, and SPSCF algorithm confidences. The RVMs were trained on each of the alarms' target features $\mathbf{x}_n^{(T)}$ using the mixture-of-RVMs approach described in Appendix B. For a test alarm at location n , each of the RVMs will yield a within-context target posterior, $p(H_1|\mathbf{x}_n^{(T)}, s_{nm} = 1)$. The forward variable, $\alpha_n(m)$, for that location is then calculated using the learned HMM parameters for $m = 1, 2, \dots, M$. Finally, a posterior confidence for the alarm can then be calculated by

$$\begin{aligned} p(H_1|\mathbf{x}_n^{(T)}, \mathbf{x}_n^{(C)}) &= \sum_{m=1}^M p(H_1|\mathbf{x}_n^{(T)}, s_{nm} = 1) p(\mathbf{x}_1^{(C)}, \mathbf{x}_2^{(C)}, \dots, \mathbf{x}_n^{(C)}, s_{nm} = 1 | \boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\Theta}) \\ &= \sum_{m=1}^M p(H_1|\mathbf{x}_n^{(T)}, s_{nm} = 1) \alpha_n(m) \end{aligned} \quad (6.16)$$

6.4 Experimental Results

An experiment was performed using a subset of the GPR data set that was used in previous chapters. A smaller dataset was used because the full data was too large for efficiently training the spatial context models with fine downtrack sampling. The data under consideration in this experiment was collected at an Eastern US test site under dry conditions in March 2009. Four test lanes (dirt, gravel, asphalt, and concrete) were present at the site. The target population consisted of 10 types of AT landmines plus 155mm artillery shells. Empty holes were present and scored as clutter. Overall, a total of 764 targets and 152 clutter objects were encountered over a total collection area was 12,383 m². The distribution of prescreener alarms with respect to the four lanes is summarized in Table 6.1.

Evaluation of alarm classification was performed using the same object-based cross-validation technique used in the previous experiments. However, the spatial DPGMM and SBHMM were trained outside of crossvalidation since they utilized

Table 6.1: Alarm Distribution by Soil Type and Ground Truth (Smaller Data Set)

Soil	Clutter (%)	Targets (%)	Total (%)
Dirt	387 (24.2%)	207 (25.8%)	594 (24.7%)
Gravel	350 (21.8%)	205 (25.6%)	555 (23.1%)
Asphalt	245 (15.3%)	212 (26.4%)	457 (19.0%)
Concrete	620 (38.7%)	178 (22.2%)	798 (33.2%)
ALL	1,602 (100%)	802 (100%)	2,404 (100%)

background feature sequences instead of prescreener alarms. The following subsections provide analysis of context-dependent fusion, including the performance of the context models, the context-specific RVMs, and overall discrimination performance.

6.4.1 Context Modeling Performance

Several unique realizations of the DPGMM and SBHMM context models were obtained through random k -means initializations. For purposes of comparison, we consider the case in which both models yielded seven contexts. Figure 6.6 compares the means of the context distributions that were learned from the spatial DPGMM and SBHMM models. Other than DPGMM Context 2, the means of the context distributions are very similar in both cases. In addition, the learned covariance matrices of the DPGMM context distributions are shown in Figure 6.7, the learned covariance matrices of the SBHMM context distributions are shown in Figure 6.8. The covariance matrices appear to be less similar than the means, but the overall scale and structure of each context’s covariance matrix appears similar between the two models. Comparing the Gaussian densities learned for both models therefore shows that the spatial dependency leveraged by the SBHMM has more of an impact on the learned emission covariances than the means.

The initial state probabilities learned for the SBHMM are plotted in Figure 6.9, and the state transition probability matrix is shown in Figure 6.10. The initial state

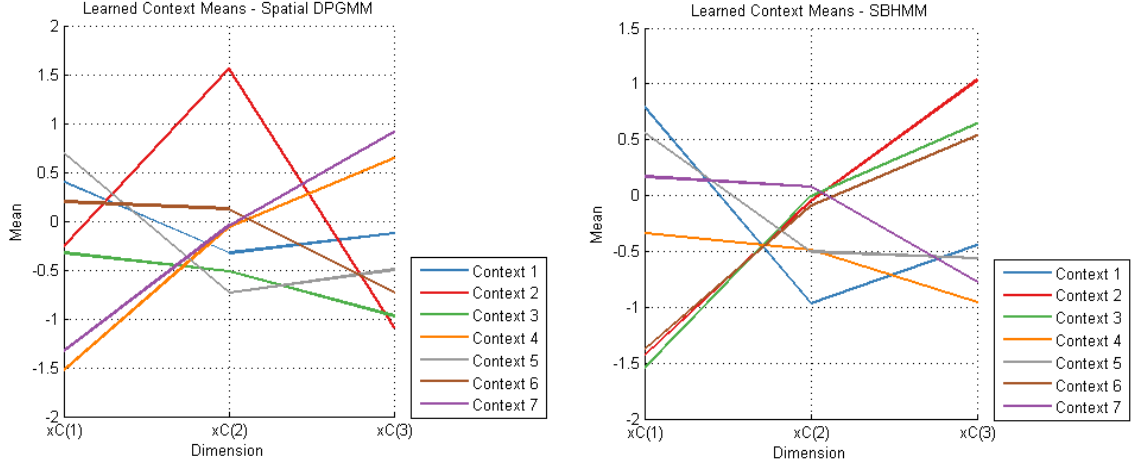


FIGURE 6.6: Learned context means for the spatial DPGMM and SBHMM context models on GPR data. Left: means learned from the spatial DPGMM, Right: means learned from the SBHMM. Feature dimension is represented by the horizontal axis.

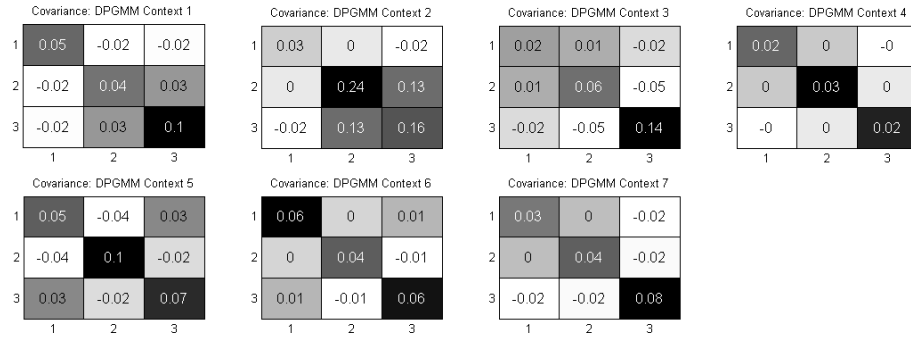


FIGURE 6.7: Covariance matrices of clusters learned by the spatial DPGMM context model. Each panel represents the covariance matrix of the Student- t mixture components obtained by integrating over the DPGMM parameters.

probabilities appear relatively uniform, with States 1, 3, and 5 having an initial probability close to 0.2 and States 2, 4, 6, and 7 having initial probabilities close to 0.1. The state transition matrix has a moderate diagonal, but the probabilities of remaining in one state are not as high as what would be expected. This result was somewhat surprising, since the test lanes over which data were artificially constructed and short in length, so they were expected to be relatively homogeneous.

Figures 6.11-6.14 illustrates examples of the raw data, background contextual

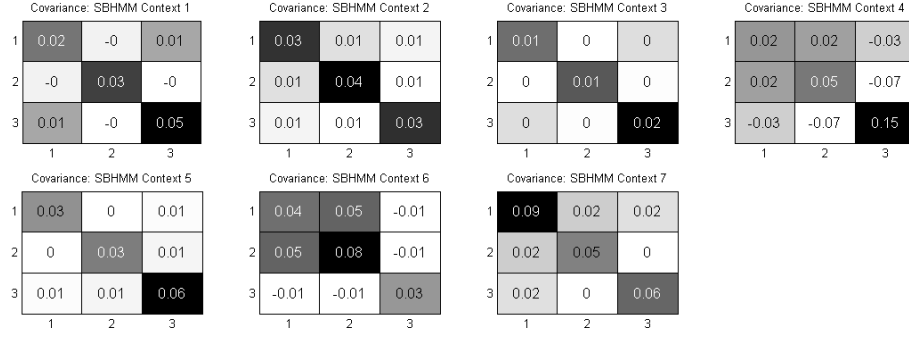


FIGURE 6.8: Covariance matrices of clusters learned by the SBHMM context model. Each panel represents the covariance matrix of the Gaussian emission density corresponding to each context.

features (projected to 3-D PCA), and the state posteriors for both the SBHMM and DPGMM context model for single passes down each of the four lanes. Figure 6.11 corresponds to the dirt lane. In this case, the SBHMM assigned high posterior probability of being in Context 7 for most of the lane, while the DPGMM assigned higher probability to either Context 1 or 2, and lower probability to Context 5 and 6. The gravel lane is shown in Figure 6.12, where the DPGMM and SBHMM context

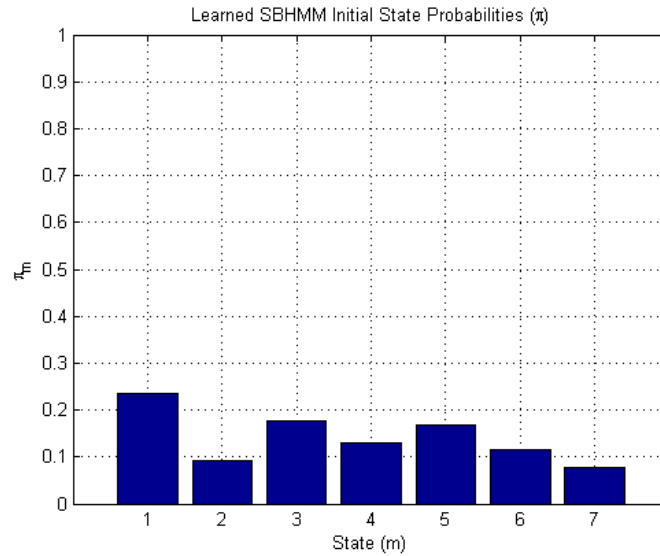


FIGURE 6.9: Initial state probabilities learned by the SBHMM context model. State (context) is represented by the horizontal axis.

Learned SBHMM State Transition Probabilities (A)

1	0.39	0.06	0.02	0.13	0.18	0.21	0
2	0.13	0.27	0.09	0.16	0.18	0.06	0.11
3	0.09	0	0.4	0.17	0.23	0	0.11
4	0.08	0.05	0.03	0.19	0.3	0.23	0.12
5	0.06	0.08	0	0.2	0.44	0.21	0.01
6	0.08	0	0.06	0.09	0.26	0.36	0.14
7	0	0	0.08	0.11	0.36	0.24	0.21
	1	2	3	4	5	6	7

State (j)

FIGURE 6.10: State transition probabilities learned by the SBHMM context model. State (context) is represented by the horizontal and vertical axes.

models appeared to behave somewhat similarly.

Figure 6.13 shows the results of spatial context modeling for the asphalt lane. The SBHMM assigned high posterior probability of being in Context 1, 4, or 7 at any given position. Meanwhile, the DPGMM context posteriors appear to be a more “smoothed-over” version of the SBHMM context posterior, assigning moderate probability to multiple contexts. Comparing the two models here shows a great similarity in where the contextual changes occurred in the lane. However, the SBHMM yielded sharp state transitions while the DPGMM favored gradual transitions.

Finally, the context posteriors for the concrete lane that was originally shown in Figures 6.1 and 6.2 are shown in Figure 6.14. A similar effect to what was seen on the asphalt lane is shown here, in that the SBHMM assigns posterior probabilities close to one or zero at each downtrack location, while the DPGMM yields moderate posteriors at transition points. Furthermore, it also appears that the SBHMM is utilizing more contexts on this lane. This can be seen in the first 2000 downtrack samples, where the SBHMM utilizes four contexts and the DPGMM utilizes three,

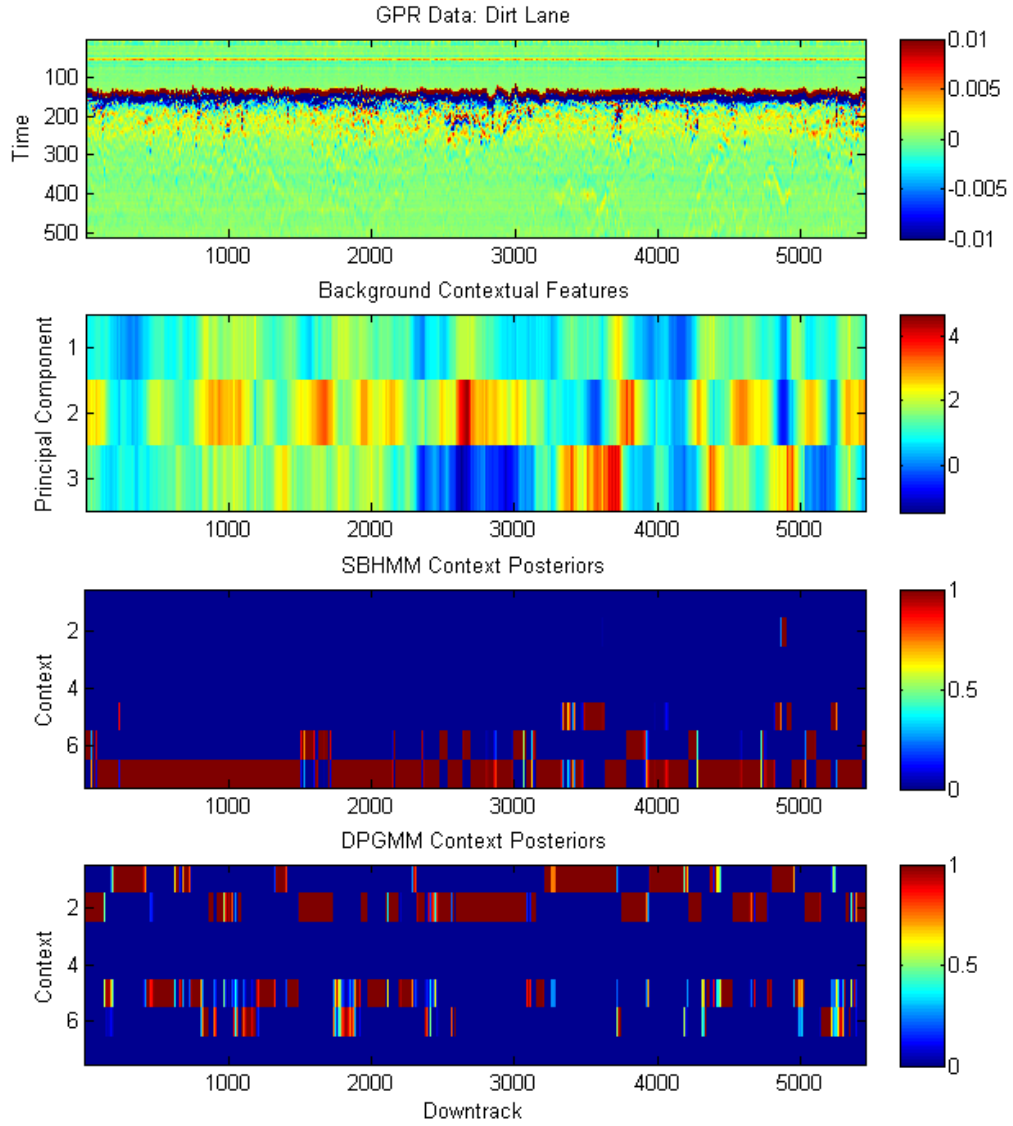


FIGURE 6.11: Example GPR data from the dirt lane and associated state posteriors from SBHMM and DPGMM context models. Top: GPR B-scan; Center: PCA of background context features; Bottom: SBHMM and DPGMM state posteriors. Downtrack position is represented by the horizontal axes.

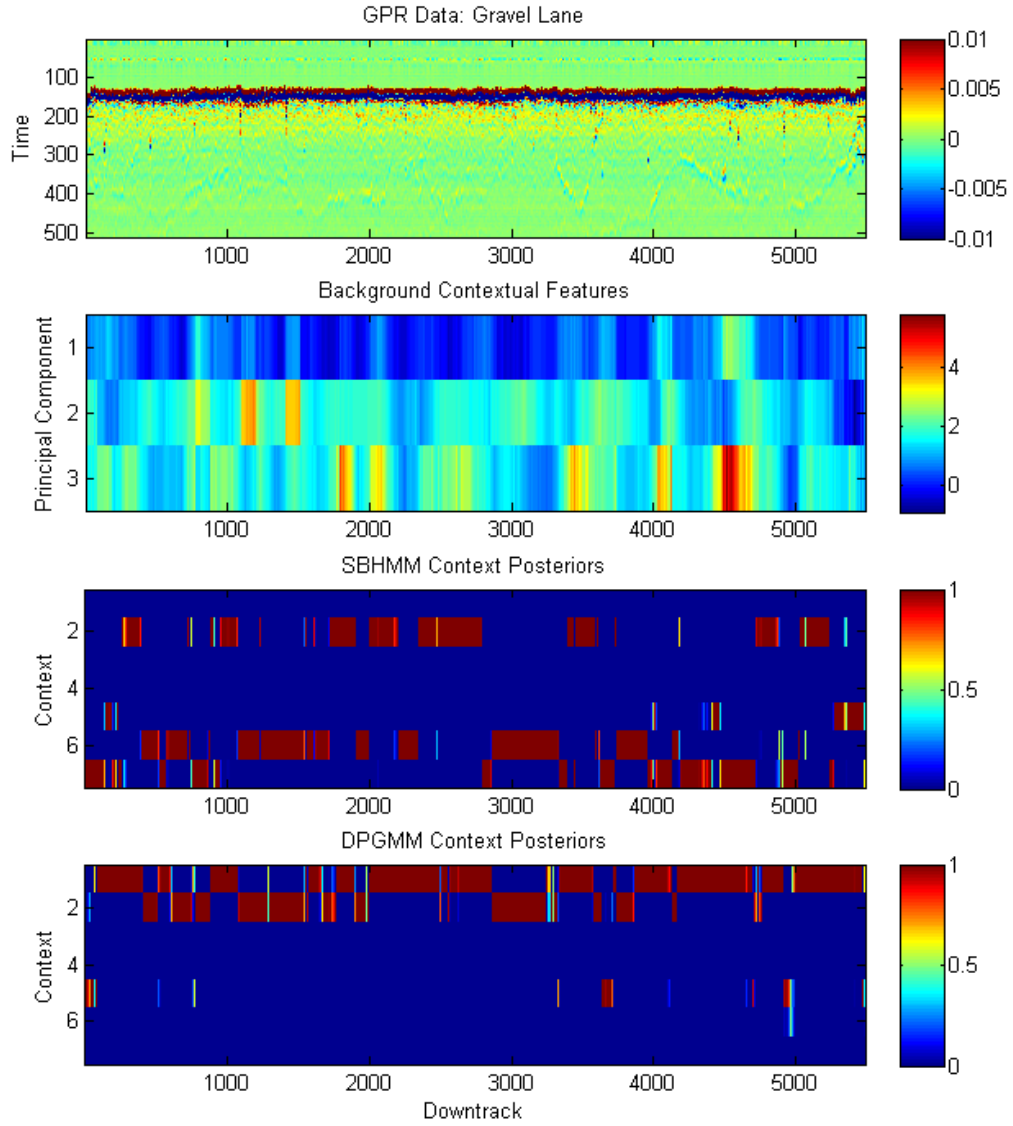


FIGURE 6.12: Example GPR data from the gravel lane and associated state posteriors from SBHMM and DPGMM context models. Top: GPR B-scan; Center: PCA of background context features; Bottom: SBHMM and DPGMM state posteriors. Downtrack position is represented by the horizontal axes.

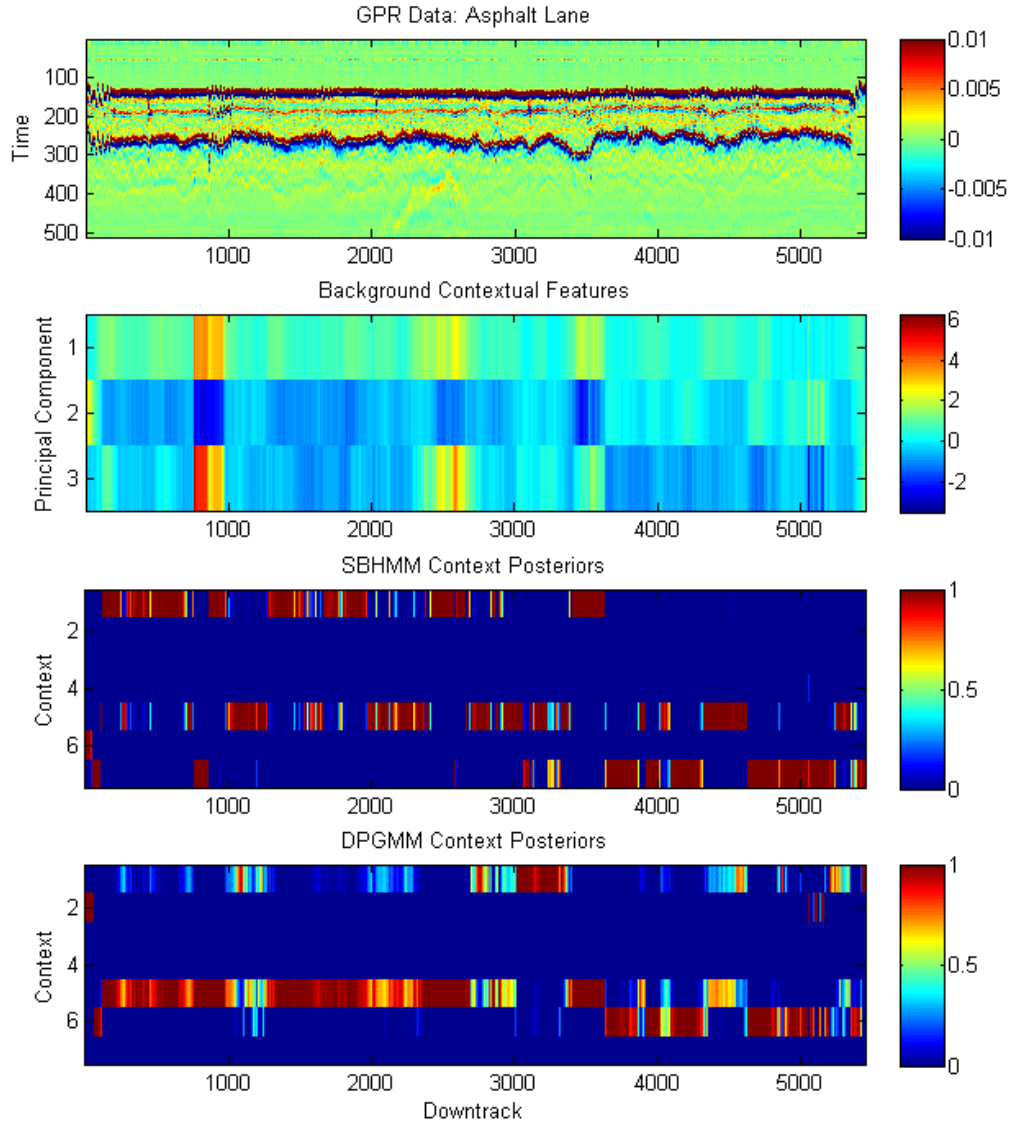


FIGURE 6.13: Example GPR data from the asphalt lane and associated state posteriors from SBHMM and DPGMM context models. Top: GPR B-scan; Center: PCA of background context features; Bottom: SBHMM and DPGMM state posteriors. Downtrack position is represented by the horizontal axes.

as well as in the remaining samples where the SBHMM utilizes three contexts and the DPGMM utilizes two.

It should be noted that the presence of landmine and clutter signatures in the GPR could have an effect on spatial context modeling, since their downtrack positions are likely to be sampled for extracting contextual features. The presence of anomalies corresponding to targets or clutter could possibly be a reason why the SBHMM tends to yield many state transitions in sections where the DPGMM suggests a single context. Because an anomaly does not appear similar to the previous observation in the feature sequence, the SBHMM considers the anomaly to be evidence of a state transition while the DPGMM considers it to be more of a statistical outlier. In previous work [119, 121], the background data was broken into segments between target positions, and the context model was trained on these target-free sequences. During this work, it was very difficult to extract target-free sections of the lanes that were long enough to effectively model the underlying contextual factors. It would also be impossible to train a context model in this manner using field data, since extracting target-free sections requires ground truth for the alarms that were encountered. Therefore, this approach was not used here although future work should investigate how to reliably train a spatially-dependent context model in the presence of known subsurface anomalies.

6.4.2 Context-Dependent Fusion Results

The spatial DPGMM and SBHMM context models assigned a posterior context probability to locations of prescreener alarms. As in previous chapters, these context posteriors were used in training context-specific RVMs for linearly-fusing the confidences of the prescreener, EHD, SPSCF, and HMM algorithms. Figures 6.16 and 6.15 illustrate the discriminant weights assigned by the RVMs to the algorithms in each of the 7 contexts identified by the DPGMM and SBHMM.

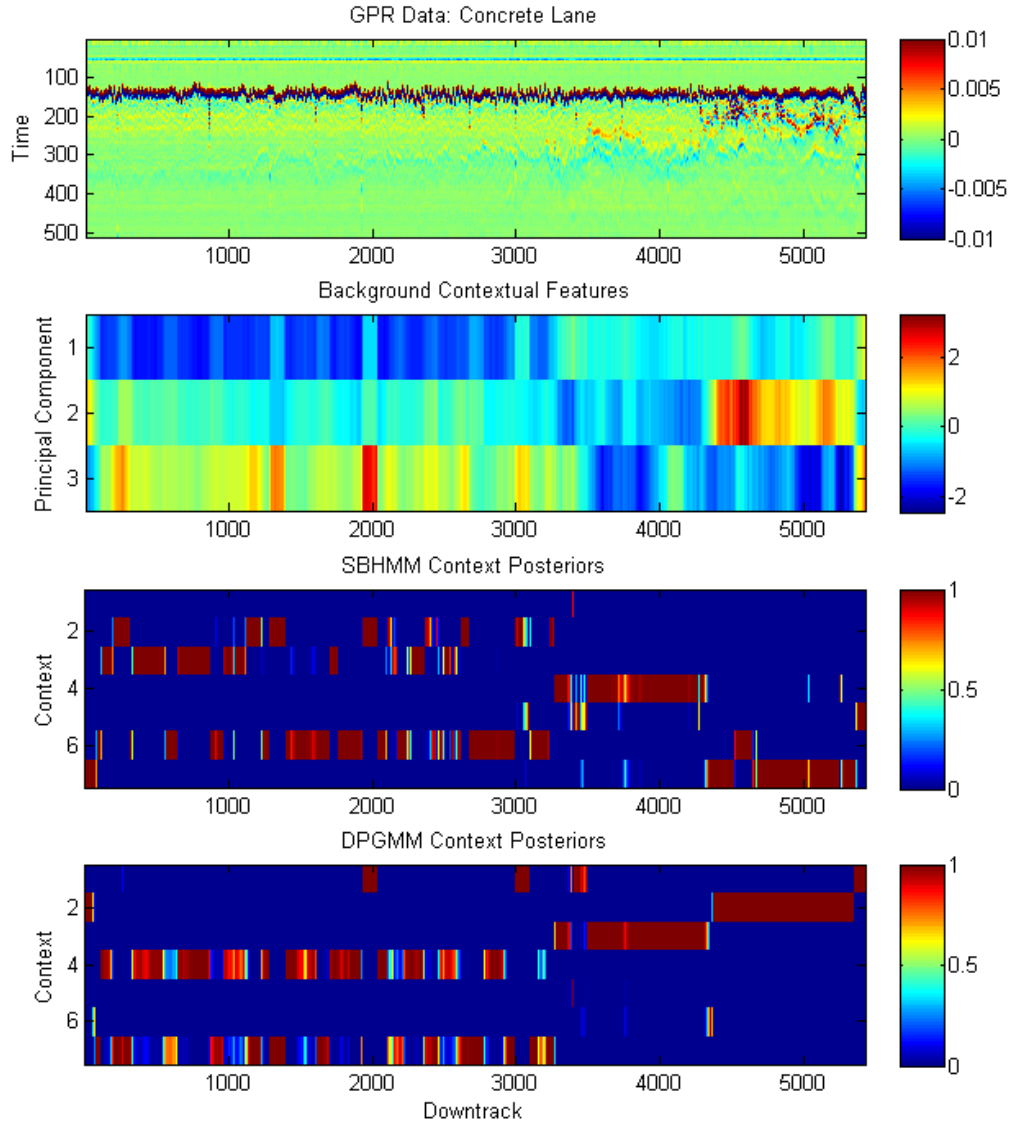


FIGURE 6.14: Example GPR data from the concrete lane and associated state posteriors from SBHMM and DPGMM context models. Top: GPR B-scan; Center: PCA of background context features; Bottom: SBHMM and DPGMM state posteriors. Downtrack position is represented by the horizontal axes.

A visual comparison of the fusion weights for each context modeling technique reveals a number of similarities. For both the DPGMM and SBHMM context model, the RVM assigns positive weight to the prescreener in three contexts, negative weight in two contexts, and zero weight in two contexts. The DPGMM contexts in which the prescreener receives negative weight are Contexts 5 and 7, and the SBHMM contexts are Contexts 1 and 2. The means and covariances for these contexts shown in Figures 6.6-6.8 suggest that DPGMM Context 5 and SBHMM Context 1 have similar densities, as do DPGMM Context 7 and SBHMM Context 2. The context posteriors shown in Figures 6.13 and 6.14 suggest that these contexts represent pavement - DPGMM Context 5 and SBHMM Context 1 correspond to portions of the asphalt lane, and DPGMM Context 7 and SBHMM Context 2 correspond to portions of the concrete lane. The negative fusion weight assigned to the prescreener for these contexts implies that its confidence should be *discounted*, perhaps because it flags too many false alarms due to anomalous responses from the pavement/soil subsurface layer.

Furthermore, the EHD, SPSCF, and HMM algorithms receive fusion weights that are quite similar between the two context modeling approaches. For DPGMM contexts, the EHD algorithm is relevant in six contexts while for SBHMM context it is relevant in five. The SPSCF algorithm is relevant in five contexts for both modeling approaches. Finally, the HMM receives nonzero weight in five DPGMM contexts and four SBHMM contexts. Although it appears that algorithms are generally more often relevant in DPGMM contexts than in SBHMM contexts, the values of the weights for each algorithm are similar between the two context models.

6.4.3 Detection Performance

Context-dependent algorithm fusion using the SBHMM and spatial DPGMM context models were evaluated via ten-fold object-based cross-validation, as the fusion

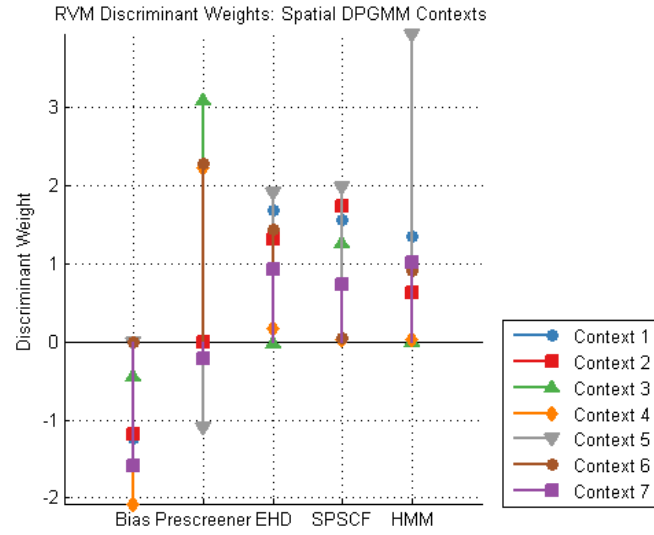


FIGURE 6.15: RVM discriminant weights learned for algorithm fusion in each spatial DPGMM context. Each stem represents a particular dimension of the target feature space, the vertical axis represents the weight value, and the individual contexts are indicated by line color.

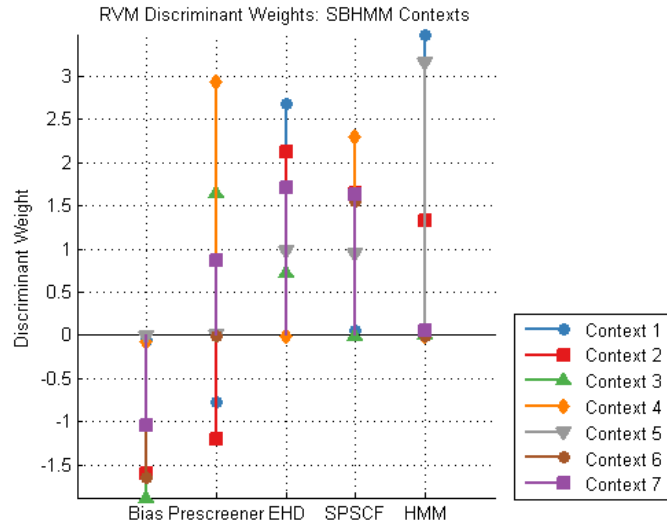


FIGURE 6.16: RVM discriminant weights learned for algorithm fusion in each SBHMM context. Each stem represents a particular dimension of the target feature space, the vertical axis represents the weight value, and the individual contexts are indicated by line color.

techniques presented in previous chapters were. In this experiment, the consistency of the SBHMM and DPGMM context models' performance were compared by using multiple random initializations of the VB learning algorithm. Five different realizations of the DPGMM and SBHMM context models were considered. For sake of comparison, five realizations of the alarm-based DPGMM context model (proposed earlier in Chapter 4) were also considered. Only one realization of the global RVM was necessary, since it did not require random initialization.

The ROC curves for context-dependent fusion, using the spatial DPGMM (green) and SBHMM (blue) context models as well as the alarm-based DPGMM (red), are plotted in Figure 6.17. ROC curves for five realizations of each model are shown, and their average FARs at benchmark PDs are shown in the legend. Performance is compared to the global RVM, which incorporates no contextual information, whose ROC is shown by the dashed black line and shaded by a 90% confidence region. Performance is also compared to the individual fused algorithms, whose ROC curves are shown by dotted lines.

As in previous GPR experiments, results illustrate that all three methods for context-dependent fusion achieved significantly better detection performance than global RVM fusion. Furthermore, both spatial context modeling techniques showed better fusion performance than the alarm-based DPGMM, with the most significant reductions of FAR occurring between PDs of 0 and 0.85. At $PD \geq 0.90$, all approaches operate at similar FARs although some realizations of the SBHMM result in better performance.

An interesting result is the differences in consistency between the three context-dependent fusion methods. Although the alarm-based DPGMM did not achieve the same level of performance as the spatial context models, the ROC curves for context-dependent fusion using the alarm-based DPGMM illustrate very consistent performance. On the other hand, the ROC curves obtained by using spatial context

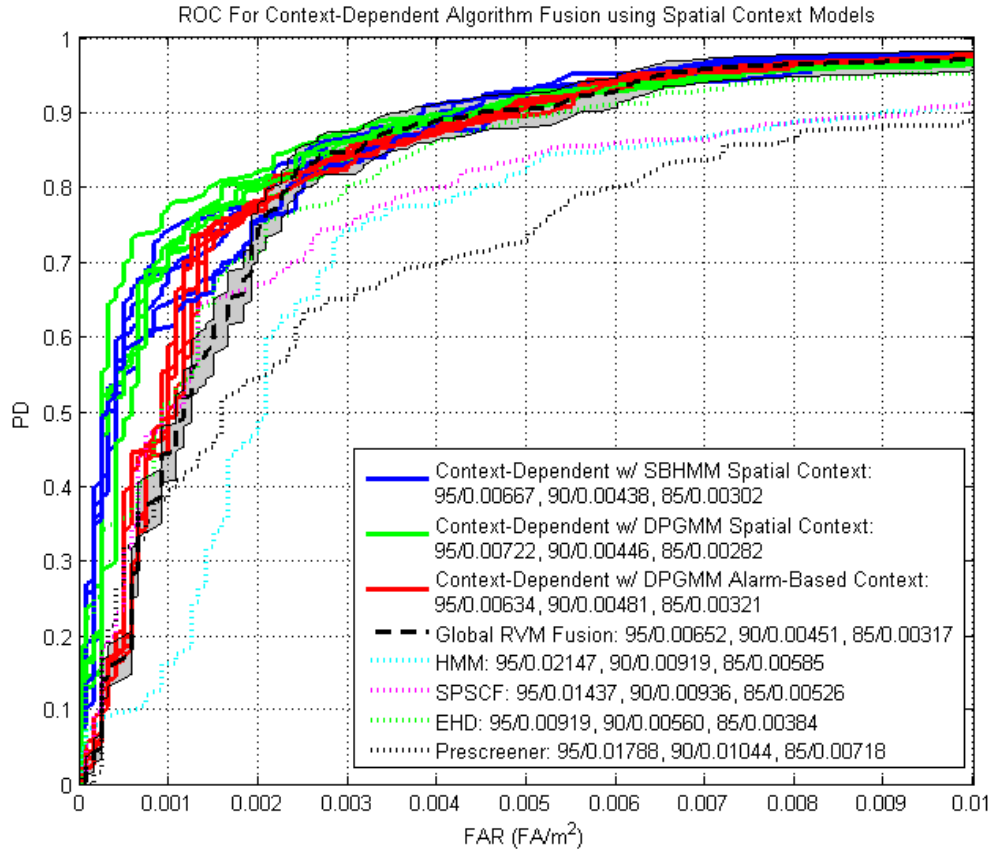


FIGURE 6.17: ROC curves for context-dependent fusion, using SBHMM (blue) and DPGMM (green) spatial context models, compared to alarm-based context-dependent fusion (red), global RVM fusion (black dashed), and the individual fused algorithms (dotted). The ROC consists of PD versus FAR, measured in false alarms per square meter, as a function of decision threshold.

models appear to be less consistent, with the SBHMM's performance fluctuating more than the DPGMM's. This could be due to several factors, such as poor choice of hyperparameters, not running VB for long enough, or insufficient context pruning to obtain a more consistent solution. Regardless of the reason, that the spatial DPGMM is a simpler model that appears to offer similar but more consistent performance, and would be a better choice for modeling context in this data.

6.5 Conclusion

In military route clearance applications for GPR, large stretches of target-free background data may be recorded on an excursion that could last for many kilometers. Although subsurface anomalies may not be present, background data collected for large periods could be a valuable source of contextual information. Therefore, the concept of context modeling was extended in this chapter, in which two methods were proposed for modeling context with respect to downtrack position.

The proposed approaches utilized features that were extracted through background sampling at regular intervals down the test lanes. The resulting feature sequences were then modeled using either a DPGMM or SBHMM to obtain posterior context probabilities at each downtrack location. While the DPGMM treated each sample as statistically independent, the SBHMM assumed a degree of dependency between neighboring samples. The incorporation of dependency via the SBHMM was motivated by the fact that many environmental factors, such as soil moisture, may be correlated spatially.

Experimental results illustrated that both spatial context modeling approaches were able to provide an intuitive description of how contextual factors vary over the course of a given area. Comparisons between the model parameters learned for the DPGMM and SBHMM illustrated that the learned contexts had similar probability density functions. Furthermore, comparisons of the context posteriors showed that both models generally agreed on where contextual transitions occurred in each lane. However, one major difference was that the DPGMM favored gradual transitions while the SBHMM implied that sharp transitions took place.

Evaluation of context-dependent fusion using the two spatial context models was performed on a subset of the data used in previous chapters. Performance was compared to context-dependent fusion using the alarm-based DPGMM that was

proposed in Chapter 4. The ROC curves showed that spatial context modeling provided additional performance benefits over alarm-based context modeling. However, the performance improvements obtained through the SBHMM were shown to be less consistent than those obtained from the spatial DPGMM. Therefore, it was concluded that the spatial DPGMM would be a better choice for spatial context modeling, since it is a simpler model that was more consistent and yielded similar detection performance.

Applications to Hyperspectral Sensing

In this chapter, the context-dependent learning framework originally developed for buried threat detection with GPR is applied to an alternative sensing modality, hyperspectral imagery (HSI). Airborne hyperspectral sensing is a particularly attractive option for detecting buried explosive threats, since it allows for greater standoff distance than GPR and can be used to survey wide areas quickly. Furthermore, disturbed earth yields a distinctive signature in HSI that can potentially be indicative of buried threats such as landmines and IEDs.

The following sections provide background information on HSI as well as the contextual factors affecting detection of buried threats. Two techniques for extracting contextual features are considered. The first technique utilizes a PCA projection of the background spectra, and is useful in characterizing different times of day. The second technique is based on spectral unmixing, which involves finding the spectra of the constituent materials present in the scene and how the abundances of those materials vary between observations. Finally, context-dependent band selection was used to classify prescreener alarms recorded over a wide area. Three context-dependent approaches are compared - supervised context learning, nonparametric

generative context learning with the DPGMM, and nonparametric discriminative context learning with the DPGMM-RVM. As was done previously for experiments with GPR data, performance is compared to a single RVM and several algorithms from the past literature.

7.1 Hyperspectral Imagery

7.1.1 Background

Hyperspectral sensors collect measurements of spectral radiance from many contiguous spectral bands. HSI is used in a variety of remote sensing applications, but system specifications vary widely with application area. For example, one of the most popular sensors in the research community is the NASA Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, which has been collecting images in 224 bands between 400 and 2500 nm for geological, agricultural, and urban mapping applications [122]. Meanwhile, the Airborne Hyperspectral Imager (AHI) developed by the University of Hawaii for subsurface and littoral sensing, is constrained to 70 bands in the long-wave infrared (LWIR) spectrum of wavelengths ranging from 8-12 μm [123].

Anomaly detection in HSI is typically performed by estimating background statistics and using a metric based on the likelihood ratio. An example of this approach is the popular RX detector [124], which uses adaptive whitening to estimate the local covariance of the background near pixels of interest. Examples of targets and false alarms detected by RX on a hyperspectral data set collected over a minefield at an arid site in the Western US are shown in Figures 7.1 and 7.2. Each figure displays a series of several 15x15 image chips, centered around detected anomalies. For visualization purposes, the chips were averaged over the 70 spectral bands.

The surface and volume scattering of recently-disturbed earth at the target location often yields a peak intensity called the *reststrahlen effect* [125]. In Figure 7.1,

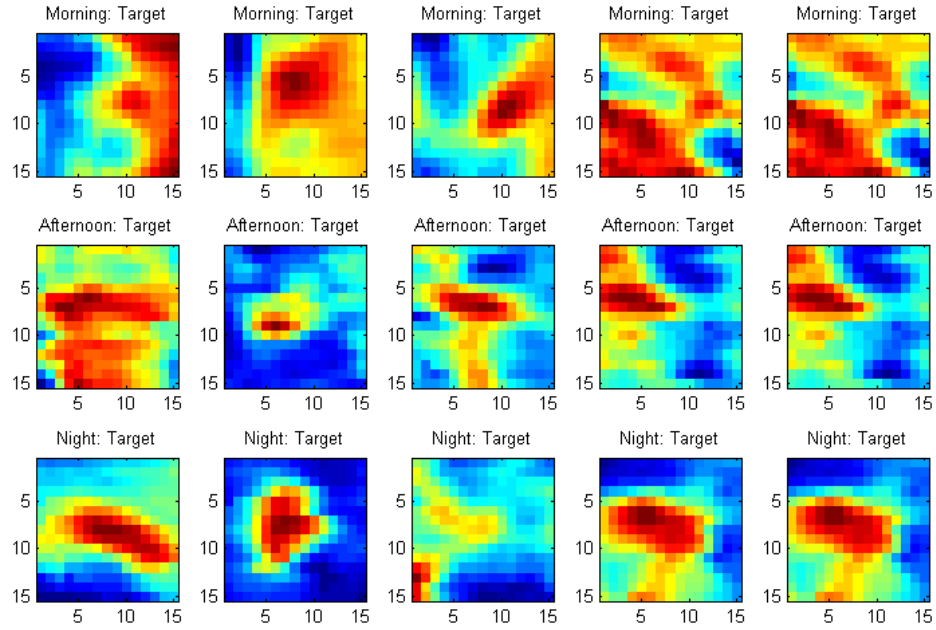


FIGURE 7.1: Example HSI image chips corresponding to antitank landmines recorded by the RX detector over a minefield located at an arid Western US test site.

the reststrahlen effect is evidenced by red peak at the center of many of the image chips. Meanwhile, most false alarms shown in Figure 7.2 do not exhibit the same type of signature. Because the reststrahlen signatures are confined to a small local area, the RX detector provides a good method for detecting these types of anomalies. However, the substantial false alarm rate has relegated its use to prescreening in past experiments with the AHI sensor [65, 126].

7.1.2 Environmental Effects on HSI Sensing

Because HSI measures spectral radiance over a wide spectral range that can include visible and/or infrared (IR) portions of the spectrum, several environmental factors can potentially affect the data. Occlusions such as clouds or heavy smog may impact the line-of-sight visibility of objects from the sensor's position, which may impact

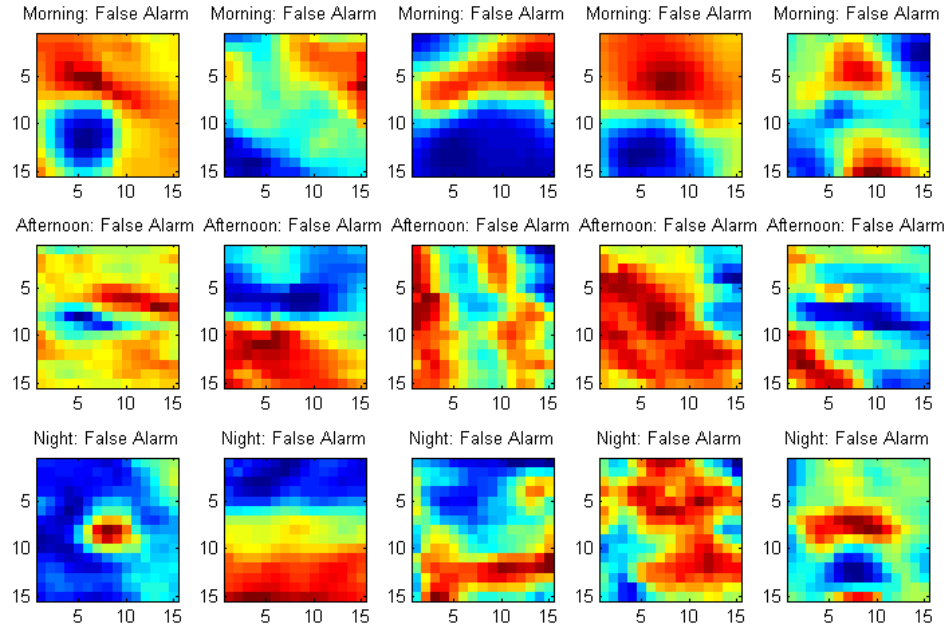


FIGURE 7.2: Example HSI image chips corresponding to false alarms recorded by the RX detector over a minefield located at an arid Western US test site.

measurements taken in the visible part of the spectrum [127]. For HSI collected in the IR spectrum, ambient solar radiance and temperature are important contextual factors since they affect the thermal emissions of the ground and objects that lie on the surface [123,126]. Figure 7.3 illustrates example spectra of three landmine targets (i.e. buried/surface AT landmines and/or disturbed earth) and false alarms (i.e. bare soil and/or vegetation) collected at three times of day: morning, afternoon, and night. Note the difference in magnitude between spectra collected at each time. The afternoon spectra, collected after the ground has absorbed much solar radiation, are of the highest magnitude. Meanwhile, the night spectra have the lowest magnitude.

Also note that the overall shape of target and false alarm spectra are quite similar and only subtle differences may distinguish targets of interest from non-threatening anomalies. Furthermore, the shape of target and false alarm spectra vary with respect

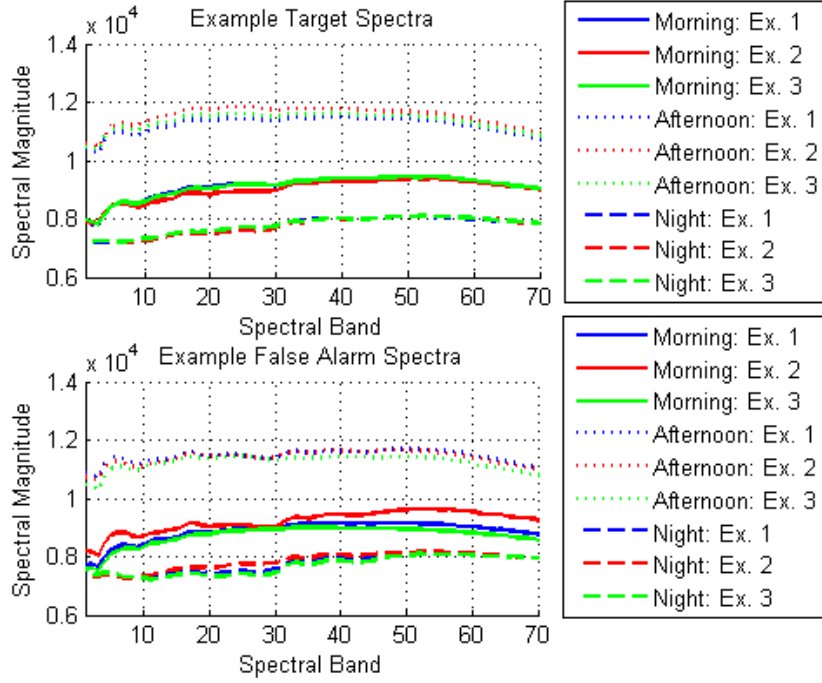


FIGURE 7.3: Example target and false alarm spectra from HSI collected in morning (solid lines), afternoon (dotted lines), and night (dashed lines). Target spectra are provided in the top panel, and false alarm spectra in the bottom panel.

to time of day. For example, the morning and afternoon spectra for both targets and false alarms exhibit a local peak around the fifth spectral band. Meanwhile, this peak does not appear in the night signatures. These differences between spectra suggest that time of day, and the lighting and temperature conditions associated with it, are contextual factors that should potentially be considered in target classification processing.

7.1.3 Buried Threat Detection with HSI in Changing Conditions

Anomaly detection in HSI requires proper modeling of the background in order for the spectra of interest to properly appear as anomalous. The most basic approach is to model the background as Gaussian-distributed with parameters estimated by maximum-likelihood statistics. This is the method employed by the RX detector,

which may use a sliding window to adaptively estimate the background mean and covariance, and declares outliers as anomalies [124]. A similar approach to mitigating local variations in background is by applying a multimodal statistical model, such as a mixture of Gaussians [128]. Another approach is to apply a transformation to the data that yields a feature space invariant to background changes, such by adaptive whitening and dewatering [129]. These past techniques were shown to be effective in cases where the background is spatially non-stationary. However, parametric models for high-dimensional data are difficult to learn robustly, and incorporate little to no prior knowledge regarding sensor phenomenology. Context-dependent learning is a potential method for exploiting knowledge of sensor phenomenology to improve detection performance across varying environments. In the HSI literature, local context-based processing was originally proposed for smoothing out segmentation maps [62, 63]. In this chapter, contextual information is utilized to improve anomaly classification in HSI using the same learning framework that was originally developed for a similar problem in GPR.

7.2 HSI Data Set

The HSI data used in this work was collected with the AHI sensor as part of a Wide Area Airborne Minefield Detection (WAAMD) platform, and has been used in several past evaluations of context-dependent landmine detection algorithms [65, 126, 130–132]. A total of 8 images (corresponding to individual flyovers) were collected over a minefield in the Southwestern US at times labeled “morning,” “afternoon,” and “night”. The minefield contained both surface-laid and buried metal anti-tank targets, as well as many empty holes which were also counted as targets. The RX detector was run on the data, and a total of 4,591 image chips, each consisting of a $15 \times 15 \times 70$ data block, were extracted around the detected anomalies. A total of 755 chips were labeled as targets (H_1), and 3,836 chips were labeled as clutter (H_0).

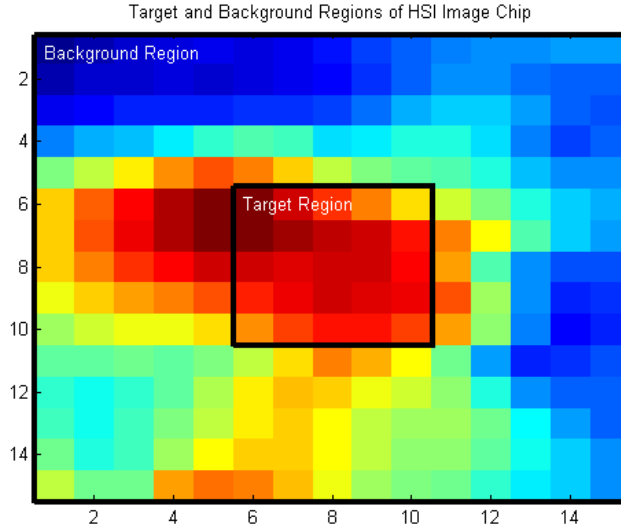


FIGURE 7.4: Illustration of the context and target feature extraction regions for a typical HSI image chip. Note that the image shown was averaged over all 70 spectral bands for visualization purposes.

The following section proposes methods for extracting contextual and target features from the HSI chips. Two contextual feature extraction approaches are considered, as was previously done in [132]. The first is based on the raw background spectra, which is useful for characterizing different times of day based on magnitude differences. The second is based on spectral unmixing, and is used to characterize the constituent spectra that make up the background.

7.3 Feature Extraction from HSI Data

Figure 7.4 illustrates the regions of a sample HSI chip where contextual and target features were extracted. As in alarm-based processing of GPR data, contextual features were extracted from the background data proximate to the detected anomaly. Meanwhile, target features were extracted from the 5×5 central region by averaging all of the pixels within that region to yield a 70-D target feature vector.

The first context learning technique is based on exploiting the magnitude differ-

ences that are associated with times of day. However, it is also important to consider the case if all samples were collected at the same time of day, which eliminates the possibility of temporal contextual effects. Therefore, the second technique is based on spectral unmixing to learn the constituent spectra of the pure materials present in different parts of the scene. The two contextual feature extraction techniques are described in the following sections.

7.3.1 Context Learning Based on Background Spectra

Recall Figure 7.3, which illustrated how the magnitude of target and false alarm spectra varies substantially with respect to time of day. These observations suggest that temporal context can be inferred directly from the raw HSI data. Therefore, the first context learning technique that was considered utilized the background region of the image chips (the outer square in Figure 7.4) to characterize whether the observation was collected in the morning, afternoon, or night.

Contextual feature extraction was performed by averaging the pixels in the background region, projecting the 70-D mean to 3-D with PCA, and then normalizing to zero-mean and unit-variance. Figure 7.5 illustrates a scatterplot of the principal components of the averaged background data, colored by time of day. The background data forms three distinct clusters for morning, afternoon, and night.

After extracting the 3-D background-based context features, they were provided as input to a statistical context model. In this chapter, three context models are compared. The basic supervised Gaussian hypothesis test, which as described in Section 3.1, serves as a baseline. In addition, two nonparametric context models were considered - the generative DPGMM, which was originally presented in Section 4.3, and the discriminative DPGMM-RVM, which was described in Section 5.3. In learning all three context models, any hyperparameter settings were set to the same values used in previous GPR experiments.

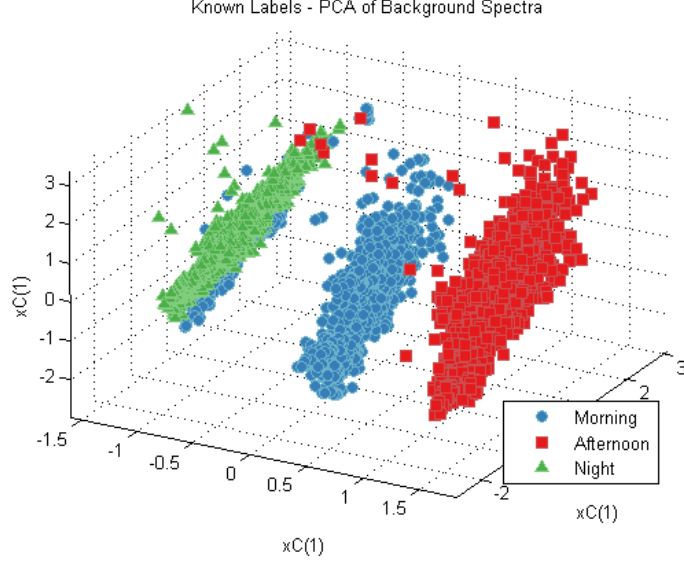


FIGURE 7.5: Scatterplot of the 3-D PCA projection of the averaged background pixels of each image chip, colored by time of day.

7.3.2 Context Learning Based on Spectral Unmixing

Another scenario to consider is if data was collected under similar lighting and temperature conditions. Although the data set used in this work was collected at different times of day, temporal effects were mitigated to simulate data collected at the same time of day. This was accomplished by subtracting the means from the background and target regions of all image chips recorded in a single flyover.

In the scenario where all observations are viewed under similar lighting and temperature, potential contextual factors could be obtained through the spectral composition of the background. This problem has been pursued extensively in the HSI literature as *spectral unmixing*, i.e. the expression of image pixels as a finite sum of known constituent spectra. These constituent spectra are known as *endmembers*, and are representative of the pure elements present in the scene. The following subsections discuss the *linear mixing model* as well as the technique used for extracting contextual features.

7.3.3 Linear Spectral Mixing Model

In Chapter 2, a simple phenomenological model was proposed for motivating contextual features from GPR data. This was the transmission line model, and although it is based on very broad physical assumptions it proved effective in characterizing quantitative properties of the soil environment. In HSI, a simple phenomenological model that is often used is the linear mixing model, which is based on the assumption that each of N pixels is a linear combination of M endmember spectra representing the pure elements present in the scene:

$$\mathbf{x}_n = \sum_{m=1}^M a_{nm} \mathbf{E}_m + \boldsymbol{\epsilon}_n \quad (7.1)$$

where

$$\sum_{m=1}^M a_{nm} = 1 \quad (7.2)$$

$$a_{nm} \geq 0 \quad (7.3)$$

In (7.1), $\mathbf{x}_n$ is the n th D -dimensional pixel in the image ($n = 1, 2, \dots, N$), $\mathbf{E}_m$ is the m th endmember spectrum ($m = 1, 2, \dots, M$) and the m th column of the endmember matrix ($\mathbf{E}$), a_{nm} is the *abundance* of endmember m in pixel n , and $\boldsymbol{\epsilon}_n$ is a random error term. The abundances are constrained to be greater than zero and sum to one. If there is no error, all of the pixels lie within an M -simplex in a D -dimensional space, where the endmembers correspond to the vertices of the simplex. The abundances form a simplex as well, but in M -dimensional space. However, since they must sum to one, the abundances contain redundant information. By projecting the abundances onto the simplex, they can serve as a feature space of dimensionality $M - 1$ from which a context model can be learned.

7.3.4 Endmember Extraction

A common problem in HSI is that the endmembers for a particular scene are often unknown, so they must be learned from the image data. Several *endmember extraction* algorithms have been proposed for unmixing HSI into its constituent spectra [133–136]. Endmember extraction algorithms often exploit the geometric interpretation of the linear mixing model. One of the earliest endmember extraction techniques is N-FINDR [133], which initializes the endmembers with random pixels and iteratively grows the simplex to include all pixels. However, a weakness of N-FINDR and similar techniques is that they inherently assume that at least one pure pixel is present in the image.

More recent approaches treat the endmember extraction task as an optimization problem [134–136]. For example, the iterative constrained endmembers (ICE) algorithm [134] optimizes a trade-off between minimizing the residual sum-of-squares (RSS) between the pixels and the linear mixing model, and minimizing the sum of squared distances (SSD) between the endmembers. RSS is calculated by

$$\text{RSS} = \sum_{n=1}^N \left(\mathbf{x}_n - \sum_{m=1}^M a_{nm} \mathbf{E}_m \right)^T \left(\mathbf{x}_n - \sum_{m=1}^M a_{nm} \mathbf{E}_m \right), \quad (7.4)$$

and SSD is calculated by

$$\text{SSD} = \sum_{m=1}^{M-1} \sum_{k=m+1}^M (\mathbf{E}_m - \mathbf{E}_k)^T (\mathbf{E}_m - \mathbf{E}_k) = M(M-1)V,$$

where V is the sum of the variances (over each band) of the endmembers. The objective function minimized by ICE in learning the endmembers is given by

$$\text{RSS}_{reg} = (1 - \mu) \frac{\text{RSS}}{N} + \mu V, \quad (7.5)$$

where μ is a parameter set to the trade-off between RSS and SSD (which is proportional to V). ICE uses an iterative process to minimize RSS_{reg} with respect to

the endmembers and abundances. Given a single row of the endmember matrix ($\mathbf{E}$), denoted by $\mathbf{e}_d$ for $d = 1, 2, \dots, D$, RSS is minimized (subject to the constraints on the abundances) using quadratic programming. This step yields an estimate of the abundances $\mathbf{A} = \{a_{nm}\}$. Then, given $\mathbf{A}$, the endmembers that minimize RSS_{reg} are given by

$$\mathbf{e}_d = \left[\mathbf{A}^T \mathbf{A} + \frac{N\mu}{(M-1)(1-\mu)} \left(\mathbf{I}_M - \frac{\mathbf{1}\mathbf{1}^T}{M} \right) \right]^{-1} \mathbf{A}^T \mathbf{x}_d, \quad (7.6)$$

where $\mathbf{x}_d$ is an $N \times 1$ vector consisting of the d th dimension of all pixels.

To illustrate the performance of ICE in extracting endmembers from hyperspectral data, a synthetic example is shown in Figure 7.6. Three-dimensional data was generated by 1000 draws from a Dirichlet(1,1,1) distribution. Since the data forms a simplex, ICE should find endmembers close to the simplex vertices. However, minimizing RSS alone ($\mu = 0$) would yield a simplex large enough to enclose all the pixels. Furthermore, minimizing SSD ($\mu = 1$) would yield another degenerate case in which all endmembers would converge to the mean of the data. Instead, the μ parameter must be set to balance the desired trade-off between the two. Figure 7.6 shows the result of ICE on the synthetic data using three different values of μ . Note that as μ increases, the learned endmembers move towards the mean of the pixel data.

Another illustration of the performance of ICE is shown in Figure 7.7. The top plot illustrates spectra of three materials from the US Geological Survey spectral library [137]. A total of 1000 random mixtures of these materials were simulated by drawing abundances from a Dirichlet(1,1,1) distribution. ICE was run on the mixed data with $M = 3$ and $\mu = .001$. The 3 endmembers extracted by ICE are shown in the bottom plot, and match the constituent spectra above very closely. Endmember 1 is approximately equal to the spectrum for rabbitbrush, Endmember 2 is approximately equal to the spectrum of juniper, and Endmember 3 is approximately equal to the

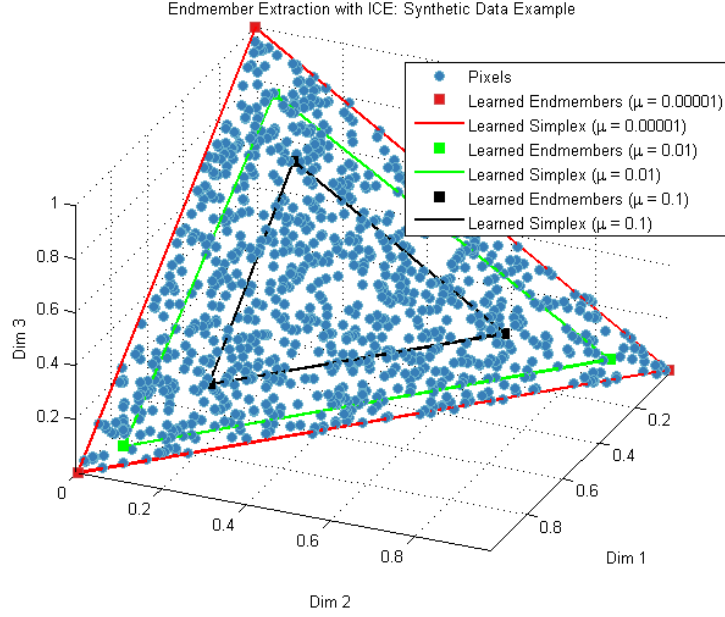


FIGURE 7.6: Results of endmember extraction using ICE on 3-dimensional toy data with $\mu = 0.1$ (black), $\mu = 0.01$ (green), and $\mu = 0.00001$ (red). The pixel data are represented by the blue points.

spectrum of grass.

In this work, ICE was used as a technique for contextual feature extraction for HSI. To eliminate temporal differences in spectral magnitude, the means of the background and target pixels for each time of day were subtracted from the images. Then, for each anomaly detected by RX, the pixels in the background region of each 15×15 chip were averaged. ICE was run with $M = 4$ and $\mu = .001$ on the aggregation of the averaged background spectra for all detected anomalies. A larger M could potentially be used, but experiments with larger values of M resulted in endmembers that were redundant or had negligible abundance. It should also be noted that a sparseness-promoting modification of ICE (SPICE) has been proposed for learning the number of endmembers [135]. However, the number of endmembers was fixed for the sake of comparison with the 3-D PCA-based features discussed in Section 7.3.1. The learned endmember spectra are shown in Figure 7.8, and they are clearly distinct

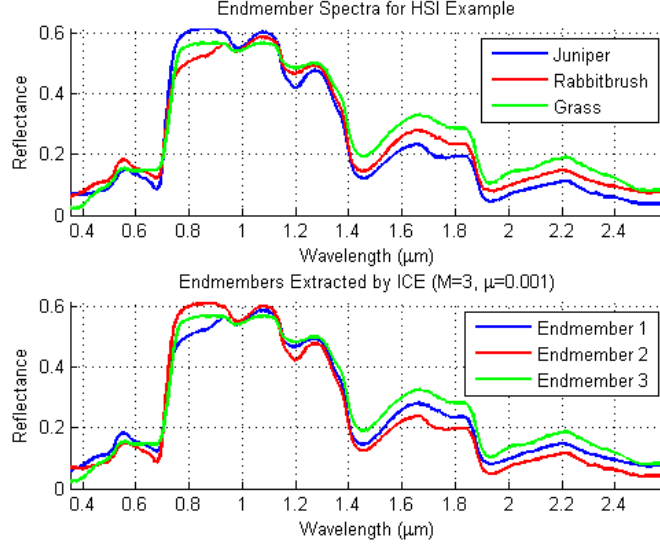


FIGURE 7.7: Results of endmember extraction using ICE on a synthetic mixture of endmember spectra from the USGS spectral library. The top plot illustrates the reflectance of three materials measured over many wavelengths by spectroradiometer. The bottom plot illustrates the endmembers extracted from $N = 1000$ random mixtures of the above spectra by ICE.

from one another.

After extracting the four endmembers and calculating their abundances for each chip, the abundances were projected onto the simplex to yield 3-D contextual features which were then normalized to zero-mean and unit variance. It is expected that different contexts should be characterized by differences in endmember abundances. Therefore, the contextual features should be amenable to clustering by a statistical mixture model. Like the background-based features proposed in the previous section, context learning was performed using the supervised, generative DPGMM, and discriminative DPGMM-RVM models.

7.4 Experimental Results

Experimental results are presented in this section, illustrating the results of context learning, context-dependent band selection, and overall detection performance on

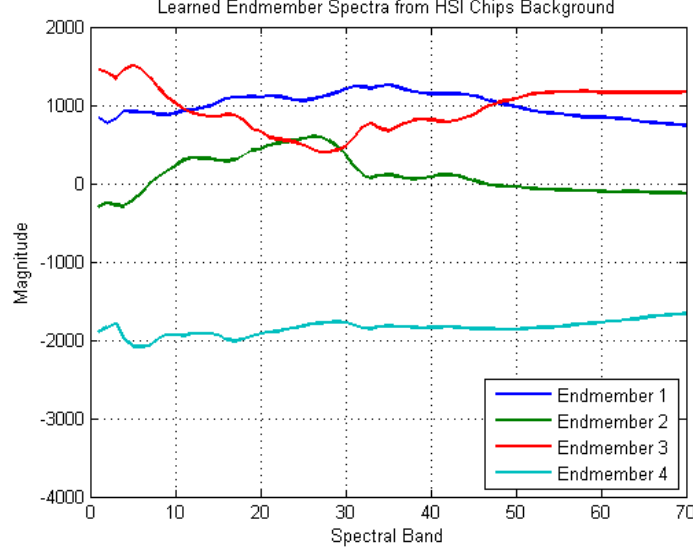


FIGURE 7.8: Endmember spectra extracted from background regions of AHI image chips using ICE with $\mu = 0.001$.

the HSI data described in Section 7.2. In each of the following subsections, results are compared for context learning using background-based and endmember-based features. By considering both sets of features separately, the relevant differences between exploiting temporal and environmental contexts can be seen. Detection performance is also compared to a single linear RVM, which incorporates no contextual information, as well as to methods that attempt to mitigate contextual effects via whitening/dewhitening [129] and a mixture of Gaussians [128], and the RX detector which was used as a prescreener [124].

7.4.1 Context Learning Results

As discussed in Section 7.3.1, the background spectra should be indicative of different times of day. Scatterplots of the PCA-projected background spectra are shown in Figure 7.9, with points colored according to their maximum *a posteriori* (MAP) temporal label (top-left), DPGMM-learned context (top-right), and discriminatively-learned context (bottom). The top-left plot shows that supervised context learning

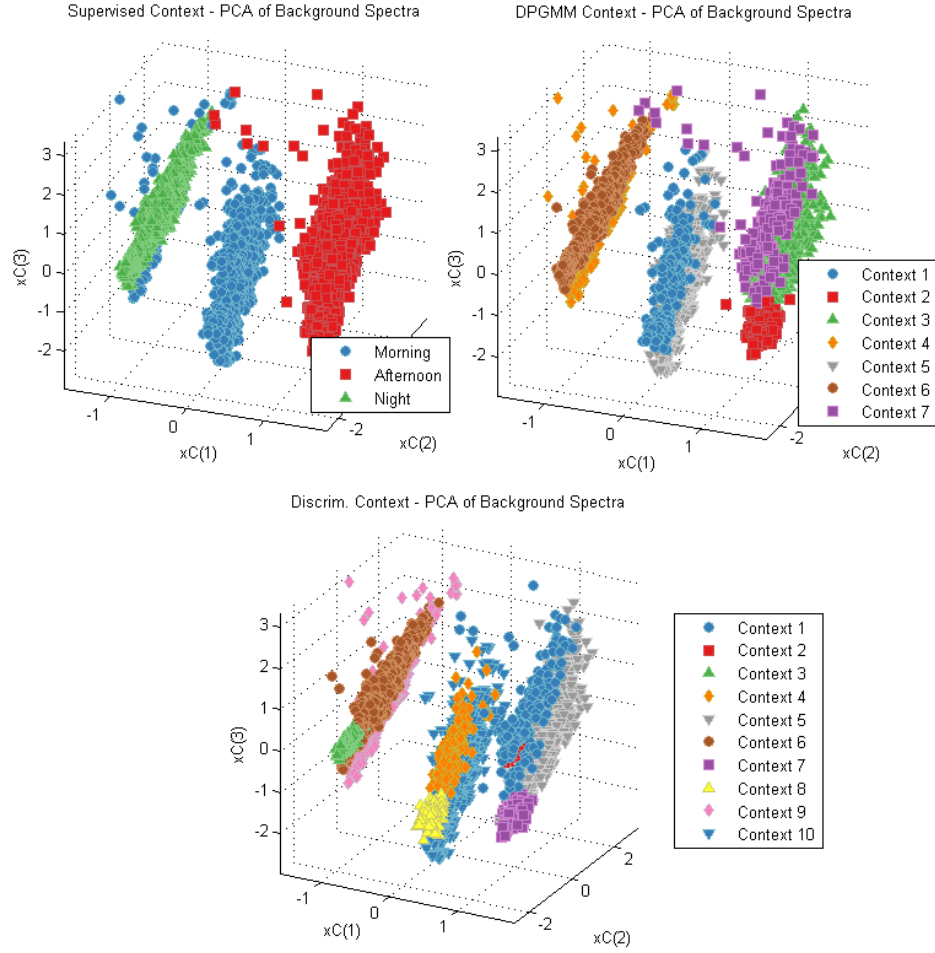


FIGURE 7.9: Scatterplots illustrating results of supervised (top-left), generative DPGMM (top-right), and discriminative (bottom) context learning from the PCA-projected background spectra. Points are colored according to their MAP context.

successfully classifies points according to the time-of-day labels originally shown in Figure 7.5. The top-right plot illustrates how the DPGMM splits the three temporal categories into sub-contexts that may be reflective of more subtle differences in spectrum magnitude. Finally, the bottom plot illustrates that discriminative context learning partitions the feature space in a different manner than generative context learning.

The similarity of the three context learning methods is compared in Table 7.1,

Table 7.1: AMI of HSI Context Models Trained on PCA of Background Spectra

	Supervised	DPGMM	Discriminative
Supervised	1	0.7594	0.6947
DPGMM	0.7594	1	0.8669
Discriminative	0.6947	0.8669	1

which compares the pairwise adjusted mutual information (AMI) [110] of the MAP context assignments. Recall that an AMI of one corresponds to identical cluster assignments, and an AMI of zero corresponds to a mutual information expected by chance. The DPGMM and discriminative context models are most similar, having an AMI of 0.8669. The supervised and discriminative models are most different, with an AMI of 0.6947. However, the tabulated AMI values are all relatively high, suggesting a great degree of similarity between the three models despite learning different numbers of contexts.

It was suggested in Section 7.3.2 that if time-of-day effects are corrected for (by subtracting the mean from the background and target regions), spectral unmixing may characterize the variations in endmember abundance throughout the scene. Endmember-based context learning was performed on the projection of the abundances onto the 3-simplex (a tetrahedron). Unlike the averaged background spectra, the endmember abundances were not expected to characterize time of day. Instead, they were expected to characterize local populations of observations where the endmember spectra mix differently in the background.

Figure 7.10 illustrates scatterplots of the endmember features, with points colored according to their MAP time of day (top-left), DPGMM-learned context (top-right), and discriminatively-learned context (bottom). The greatest difference between these scatterplots and those in Figure 7.9 is that the features do not cluster according to time of day. This comes as no surprise, since the HSI chips were mean-subtracted to eliminate the magnitude differences caused by temporal changes in lighting and

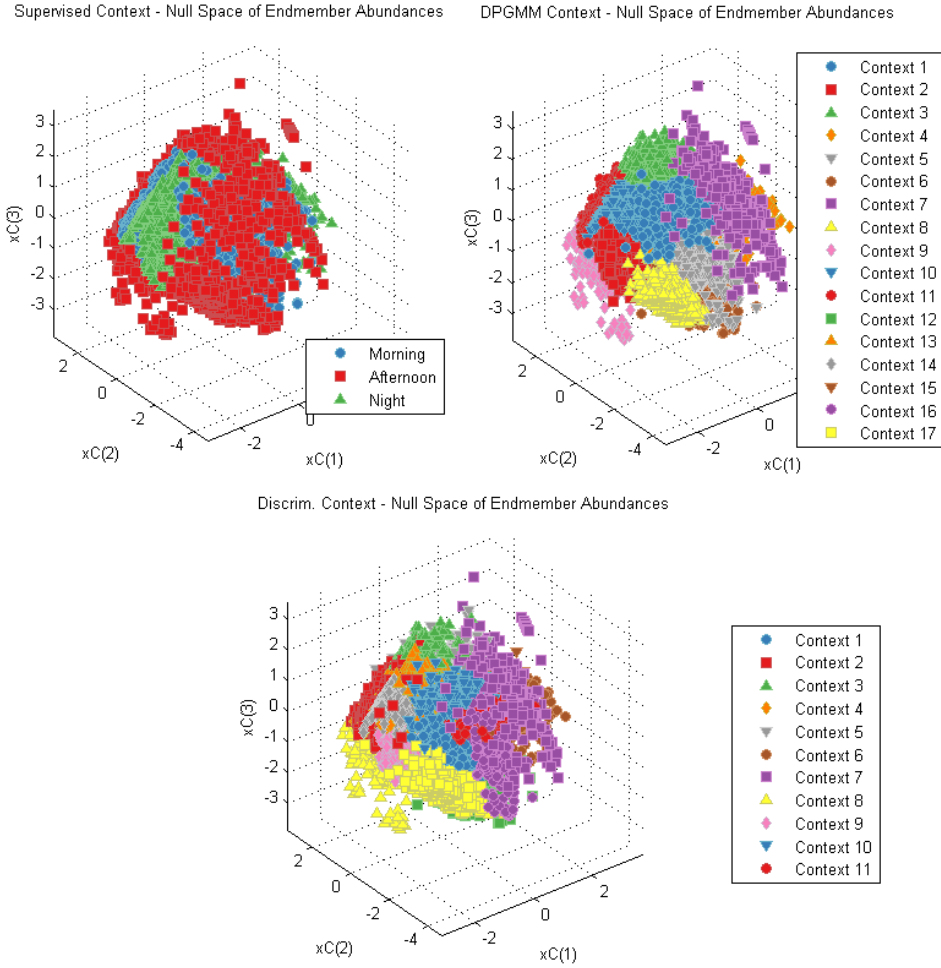


FIGURE 7.10: Scatterplots illustrating results of supervised (top-left), generative DPGMM (top-right), and discriminative (bottom) context learning from the endmember abundances learned by ICE. Points are colored according to their MAP context.

temperature. However, the top-right and bottom plots show that the nonparametric models allow for contexts to be learned in an unsupervised manner. Generative context learning with the DPGMM found 11 contexts, and discriminative context learning with the DPGMM-RVM found 17 contexts. In both cases, more contexts were learned from the endmember features than from the averaged background spectra, suggesting that the endmember features may be indicative of more localized contextual factors.

Table 7.2: AMI of HSI Context Models Trained on Endmember Abundances

	Supervised	DPGMM	Discriminative
Supervised	1	0.3402	0.3190
DPGMM	0.3402	1	0.6214
Discriminative	0.3190	0.6214	1

The AMI of the contexts learned from the endmember abundances are summarized in Table 7.2. The DPGMM and discriminative context models were the most mutually-informative, with an AMI of 0.6214. However, they were less mutually informative as they were when trained on the background spectra, suggesting that the models behave more differently when trained on endmember abundances. The two nonparametric context models had an AMI with the supervised model of about 0.3, which is much lower than when the models were trained on the background spectra. Low AMI between the supervised and nonparametric models was expected because temporal effects were eliminated.

7.4.2 Context-Dependent Band Selection Results

As was done previously for context-dependent fusion in GPR, the linear RVM was used as a context-specific classifier for discriminating targets from false alarms in HSI data. Target features were extracted from each image chip by averaging the pixels in the center region. Separate RVMs were learned for classifying the averaged target spectra in each context. Because the priors used in learning promote sparseness in the weights, the RVMs will apply nonzero weight to only a subset of the 70 spectral bands. Therefore, training an ensemble of RVMs based on the learned contexts will also perform *band selection* within each context.

The RVM weights corresponding to each context learned from the background spectra are shown in Figure 7.11. The top plot illustrates the weights for each of the three times of day. Most of the bands receiving nonzero weight are towards the left,

which roughly correspond to the “bump” in the spectra shown in Figure 7.3. In each context, a unique subset of the spectral bands receive nonzero discriminant weight, suggesting that the relevance of certain bands for classifying targets from clutter vary with time of day. The center plot illustrates the discriminant weights assigned for each of the generatively-learned DPGMM contexts, which are less sparse than those assigned for the temporally-labeled contexts. The bottom plot illustrates the weights obtained for each of the discriminatively-learned contexts, and they appear similar to those assigned for the temporally-labeled contexts. Note that the weights for the discriminatively-learned contexts appear to be more sparse than those learned for the generatively-learned contexts.

Figure 7.12 illustrates the RVM weights corresponding to each context learned from the endmember abundances. The top plot shows the weights for each of the three temporally-labeled contexts. Note that the weights in this panel are different than those in the top panel of Figure 7.11. This difference suggests that subtracting the mean from the target features, which eliminates temporal effects, also changes the relevance of certain spectral bands for classification purposes. The center plot illustrates the RVM weights assigned to each of the generatively-learned DPGMM contexts, and they appear to be more sparse than those shown in the center panel of Figure 7.11. These weights also appear similar to those in the bottom panel, which correspond to the discriminatively-learned contexts. Note that in the case of endmember-based context learning, a greater number of contexts were learned. The RVMs learned for each context tend to be very sparse, and some may only assign nonzero weight to two or three spectral bands. These results suggest that the reststrahlen characteristics of disturbed earth may manifest itself in only a few spectral bands that depend on the local soil context.

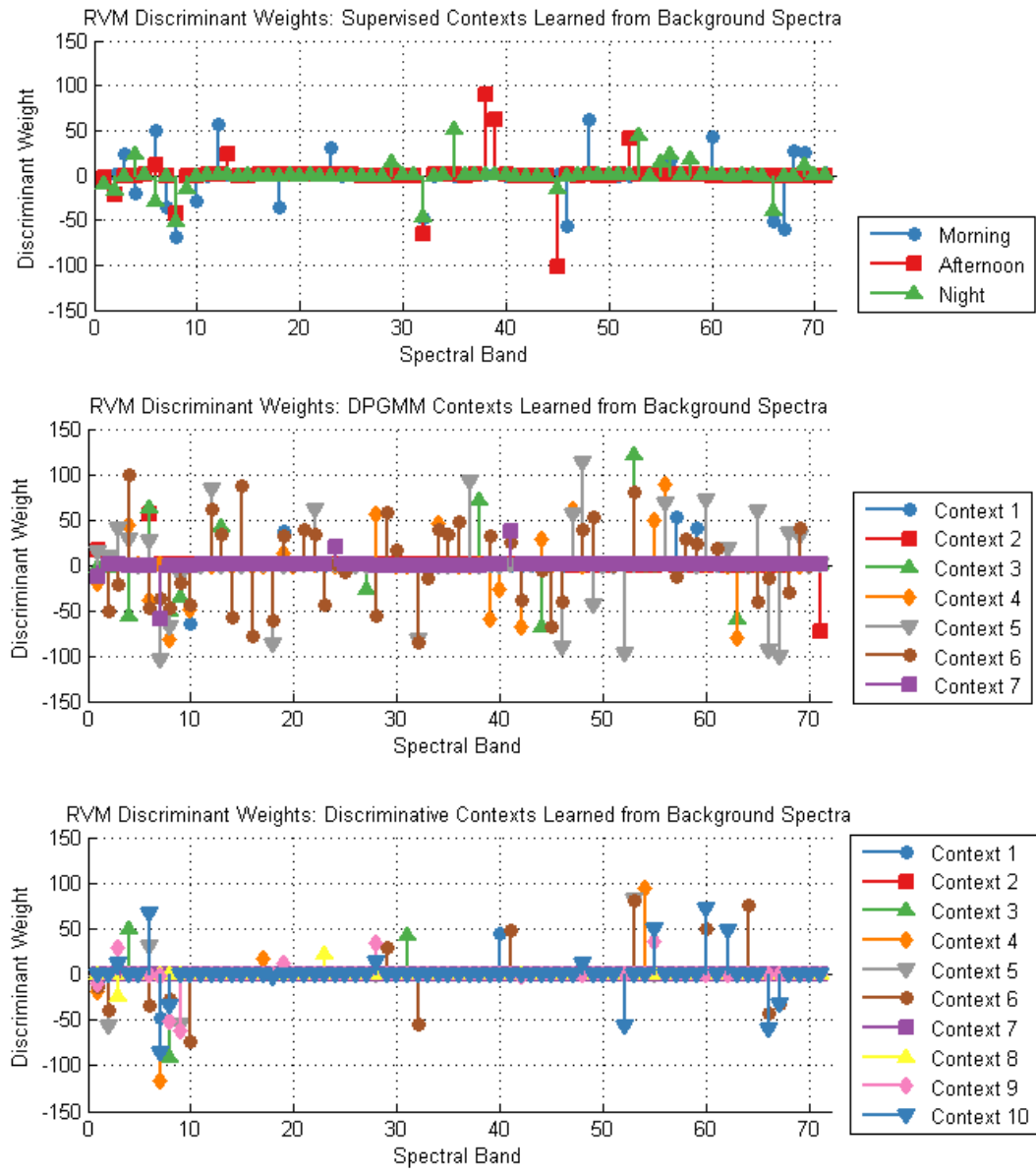


FIGURE 7.11: RVM discriminant weights corresponding to supervised (top), DPGMM (center), and discriminative (bottom) contexts learned from background spectra. The horizontal axes represent spectral band, and the vertical axes represent the value of the discriminant weights. Context is indicated by color.

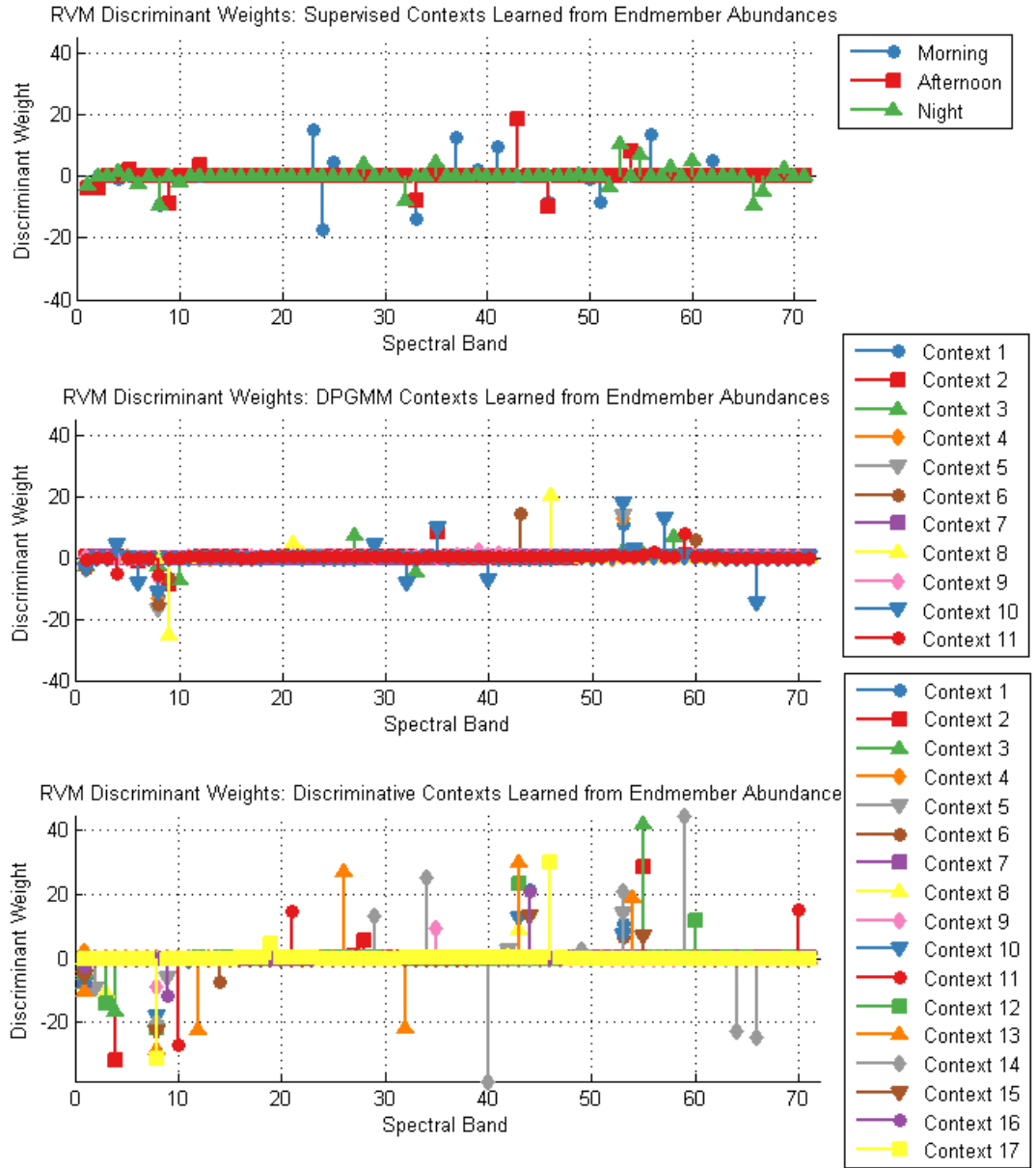


FIGURE 7.12: RVM discriminant weights corresponding to supervised (top), DPGMM (center), and discriminative (bottom) contexts learned from endmember abundances. The horizontal axes represent spectral band, and the vertical axes represent the value of the discriminant weights. Context is indicated by color.

7.4.3 Detection Performance

The discrimination performance of context-dependent learning was evaluated via 10-folds cross-validation over the image chips. The results of context-dependent classification using the background features are summarized by the ROC curves shown in Figure 7.13. The three black lines correspond to detectors from the literature that attempt to mitigate contextual effects. The black solid line illustrates the performance of the RX prescreener [124], the black dashed line illustrates performance of using whitening/dewhitening [129], and the black dotted lines illustrates performance of the mixture of Gaussians technique [128]. The colored lines indicate the performance of trained classifiers. The blue line corresponds to the performance of a single linear RVM that incorporates no contextual information. Context-dependent classification with the generative supervised context model is shown in magenta, and context-dependent classification with the generative DPGMM context model is shown in red. The performance of discriminative context-dependent learning with the DPGMM-RVM model is shown in green.

The order of performance is very similar to the GPR results that were shown in previous chapters. Generative context learning with the DPGMM yields the most performance improvement. Even at high PD, the ROC curve for context-dependent classification based on the DPGMM shows the most reduction in PF. Furthermore, all three context-dependent classification techniques yielded better performance than the global RVM, but the degree of improvement does not appear to be substantial.

The results of context-dependent learning based on the endmember features are summarized by the ROC curves in Figure 7.14. Note that in this case, subtracting the mean of each flyover’s target features caused the global RVM to perform worse than in Figure 7.13. However, the nonparametric methods yielded much greater improvement over the single RVM than they did when using background features.

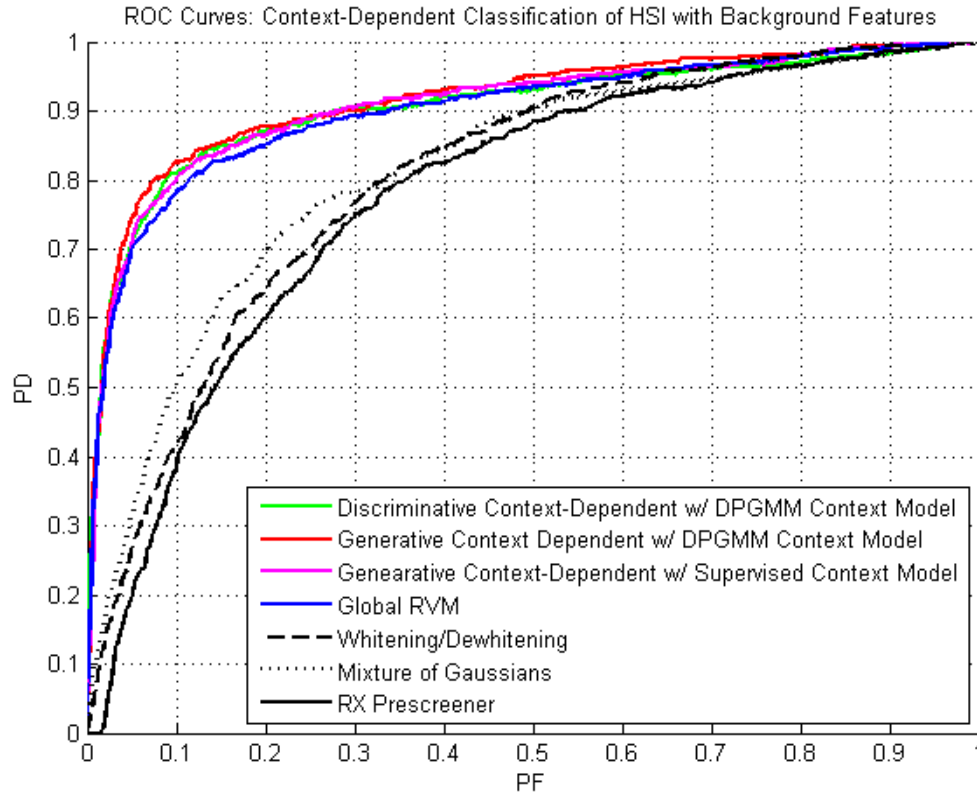


FIGURE 7.13: ROC curves for context-dependent classification of HSI data using background context features. Performance of the RX prescreener (black solid), whitening/dewhitening (black dashed), mixture of Gaussians (black dotted), global RVM (blue), generative context-dependent learning with supervised (magenta) and DPGMM (red) context models, and discriminative context-dependent learning (green) are compared. The horizontal axis represents probability of false alarm (PF) and the vertical axis represents probability of detection (PD).

Furthermore, the generative approach with the DPGMM context model did not yield the greatest performance improvement at high PD. For PDs greater than approximately 0.85, discriminative context learning illustrates the most substantial reduction in PF. These results suggest that if all HSI observations were collected under similar lighting and temperature conditions, endmember-based context learning has the potential to substantially improve classification performance.

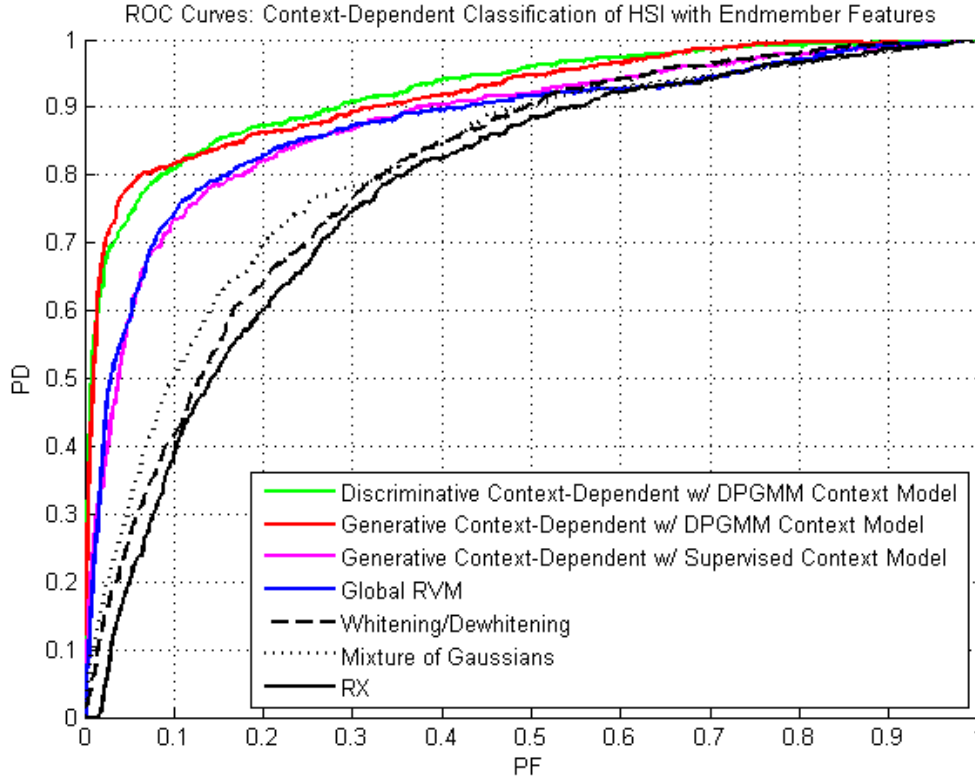


FIGURE 7.14: ROC curves for context-dependent classification of HSI data using endmember context features. Performance of the RX prescreener (black solid), whitening/dewhitening (black dashed), mixture of Gaussians (black dotted), global RVM (blue), generative context-dependent learning with supervised (magenta) and DPGMM (red) context models, and discriminative context-dependent learning (green) are compared. The horizontal axis represents probability of false alarm (PF) and the vertical axis represents probability of detection (PD).

7.5 Conclusions

This chapter presented another application of context-dependent learning for buried threat detection using a sensing modality complementary to GPR. Airborne HSI is a useful sensing technology for wide-area assessment whose phenomenology exploits the reflectance properties of disturbed earth known as the reststrahlen effect. Therefore, HSI sensors such as AHI that are tuned to special reststrahlen bands for disturbed earth may be useful in buried threat detection applications.

This chapter compared two approaches to context-dependent classification of HSI chips centered on anomalies detected by the RX algorithm, which served as a pre-screener. In both approaches, contextual features were extracted from the background data, which consisted of the pixels outside of the 5×5 center region of each 15×15 chip. The first set of context features were motivated by the differences in the magnitude of background spectra at different times of day. It was shown that spectra collected in the afternoon had higher magnitude than those collected at morning and night, and spectra collected at morning and night illustrated magnitude differences as well. Therefore, context features were extracted by averaging the pixels in the background region of each image chip, and context learning was performed in either a supervised manner using qualitative time-of-day labels, or through nonparametric models such as the DPGMM and the discriminative DPGMM-RVM. Results of context learning showed that the different times of day were easily characterized, and all three context learning techniques had high mutual information.

The second context learning approach considered the case where there were no temporal effects. For the purposes of simulating this case, the means of the background and target data were subtracted for each flyover. Features were extracted from the background data via spectral unmixing. The ICE algorithm was used to learn four endmember spectra from all chips' averaged background spectra. The endmember abundances were projected onto the corresponding simplex, and context learning was performed on the resulting 3-D features using supervised and nonparametric methods. Context learning results illustrated that although effects of temporal context were eliminated, nonparametric context learning found many distinct clusters in the endmember features.

Experimental results compared the performance of context-dependent classification using the background and endmember-based context features. Results from using background features, which was expected to exploit temporal differences in

spectral signatures, illustrated that context-dependent learning did not yield much improvement over the single RVM. However, in the case where temporal effects were removed and endmember features were used, context-dependent learning yielded substantial improvements over the RVM and discriminative context-dependent learning showed the best performance at high PD. These results illustrate that despite using similar context-dependent learning approaches, the degree of improvement over conventional classification can be highly dependent on the contextual information being exploited.

Conclusions and Future Work

In this dissertation, a variety of nonparametric Bayesian methods for context learning were proposed for improving the robustness of sensor systems used to detect buried explosive threats such as landmines or IEDs. However, the novel contributions of this work have broader application to a variety of current research areas. The following subsections summarize these contributions, propose avenues for future work, and discuss the broader implications of the novel context-dependent models that were developed.

8.1 Summary of Contributions

In Chapter 1, the threat of buried explosives was introduced as a problem of major concern to militaries and humanitarian organizations. GPR was then introduced as a valuable tool in detecting landmines and IEDs, since its phenomenology enables the detection of nonmetal objects. However, the phenomenology of GPR also makes it sensitive to effects from many aspects of the subsurface environment. Particular attention was paid to the effects of soil moisture [17–20], rough surface scattering [23–28], and subsurface heterogeneity [22, 28]. Although techniques based on

electromagnetic model inversion have been proposed for inferring an object’s true size and shape based on noisy GPR responses [29–31], these types of approaches are computationally slow and require *a priori* measurements of a target’s scattering properties. Because military route clearance requires real-time processing, and the IED threat is constantly redefining itself, iterative model inversion may not be the best approach to improving detection across varying environments.

In this work, a Bayesian learning framework referred to as *context-dependent classification* was proposed as a technique for maintaining robust performance across varying environments. Traditionally, statistical classification would be performed on a set of *target features* designed for characterizing target signatures from clutter. However, in varying environments there can be significant class overlap in the target feature space that cannot be modeled by a linear decision boundary. In context-dependent classification, a set of secondary *context features* were proposed for clustering observations collected under similar environmental conditions. By conditioning the classifiers operating in target-space on the clusters learned in context-space, a complex nonlinear classification problem could potentially be broken down into several simpler linear ones that are motivated by changes in the ambient sensing environment.

Although other researchers have proposed context-based learning techniques in the literature [64, 65], the definition of context used in this work differs from those used in the past. In this work, context is motivated by the physical state of the world from which an observation was drawn, and not from the properties of the observation itself. Regardless of whether a target is present at a particular location, the context of that location is still the same. It was therefore proposed that contextual information can be extracted from the background sensor data by exploiting *a priori* knowledge about the underlying phenomenology.

In Chapter 2, several physically-motivated *contextual features* were proposed for

training a statistical context model. The features were based upon a transmission line model for GPR A-scans. Although using the transmission line model implies major simplifying assumptions about the physics of wave propagation, deviations of the model from reality could be accounted for by analysis of the features' statistics. A variety of features were proposed to characterize different soil properties. For example, energy feature was proposed for characterizing such as soil permittivity, conductivity, and heterogeneity. A feature based on the reflection coefficient was proposed for characterizing the dielectric contrast at the air/ground interface. To compute the reflection coefficient, the ground bounce was isolated and basic radar ranging was applied. To characterize soil heterogeneity, features based on the matching pursuits algorithm were proposed for estimating the number of unique reflections that make up a single A-scan. Finally, features based on linear prediction were proposed for characterizing the stochastic properties of the background.

To evaluate the performance of these features in characterizing quantitative soil properties, experiments were performed using simulated and field-collected GPR data. For these experiments, the features were extracted from GPR data free of landmine signatures, and statistical regression and classification models were used to predict known soil properties from the features. In Section 2.3, it was shown that the features were informative in predicting soil dielectric constant, conductivity, surface correlation length (roughness), and the expected number of subsurface scatterers (heterogeneity) from simulated GPR data. The results of the experiment on field-collected data were presented in Section 2.4. In this experiment, the features were shown to be informative in estimating measurements of soil moisture and temperature collected from a nearby meteorological station. These results represent the first successful application of statistical inference for identifying soil properties from GPR features that are easy to extract in real-time operation.

In context-dependent classification, the contextual features were used to train a

statistical *context model* to partition the training data into M clusters known as *contexts*. Because the contextual features were shown to be characteristic of quantitative soil properties, performing clustering in that space can group together observations that were collected in similar environments. After learning the individual contexts, a unique classifier (the RVM [83] in this work) can be trained on the target features for discriminating targets from clutter in that context. In this work, the RVMs were trained on the confidence values of four currently-fielded detection algorithms - a process referred to as *context-dependent fusion*.

In Chapter 3, two basic context models were presented. The first was a *supervised* context model based on a Gaussian hypothesis test between known qualitative soil labels: dirt, gravel, asphalt, and concrete. By projecting the contextual features to 3-D via PCA, and learning a Gaussian distribution for each labeled soil type, test observations were classified according to the most likely soil type. Although this approach yielded excellent performance in distinguishing the four different soils, supervised context learning is highly dependent on quality of the labels. Therefore, another basic context model was proposed based on *unsupervised* learning, which is performed without labels. Basic unsupervised context learning was performed by estimating the parameters of an M -order GMM from the contextual features. Although this approach was able to sub-divide each of the four soils into multiple sub-clusters that could provide more physically-meaningful contextual information, choosing the order of the model is a separate and difficult task. As a result, it was concluded that context-dependent classification could potentially benefit from context models that facilitate learning the *number* of contexts, in addition to their statistical distribution in feature space.

Several Bayesian approaches for learning nonparametric context models were proposed in Chapters 4, 5, and 6. The common factor between the proposed nonparametric context models were that they were all infinite-order probabilistic mixtures

that incorporated *Dirichlet process* (DP) priors [103]. The DP prior was used to control model complexity and facilitate learning of an effective model order. This property was illustrated through discussion of the *Chinese restaurant process* and *stick-breaking process* in Section 4.2. Because posterior inference cannot be performed analytically for nonparametric mixture models, variational Bayesian (VB) inference was used to perform approximate inference. An overview of VB was given in Section E.10.

In Chapter 4, two models were proposed for *generative* nonparametric context learning. These models were learned on the context features alone, without regard to the class (target/clutter) labels or the target features. The first model was the DPGMM [67], which was able to learn an effective number of Gaussian contexts without having to specify the number of contexts *a priori*. The second model was the DPMFA (adapted from [68,94]) which lifted the restriction of having all contexts use the same underlying low-dimensional projection. The DPMFA was used to learn the number of contexts as well as the number of latent factors that characterize each. The DPGMM and DPMFA were both used to perform context-dependent fusion and the results were compared. Context-dependent fusion using either model performed significantly better than a single RVM incorporating no contextual information, but using the DPGMM led to better performance.

Chapter 5 explored *discriminative* nonparametric context learning. In contrast to generative context learning, which treated clustering in the context features and discrimination in the target features as independent tasks, discriminative context learning performed both tasks jointly. Two methods for discriminative context learning were compared. The first, referred to as the DPGMM-RVM, consisted of a mixture of RVMs with a DPGMM gating network. The DPGMM-RVM was shown to perform clustering in target space while also training the classifiers in target space. The other method was the IQGME, which was originally proposed in [94] for classification with

missing data. In contrast to the modified DPGMM-RVM, the IQGME performed classification and clustering in the joint feature space formed by concatenating the context and target features. The IQGME also did not utilize sparse component classifiers. A series of synthetic data examples compared the performance of the two discriminative context models under various scenarios. Furthermore, their performance in context-dependent fusion were compared. Although both showed significant improvement over the single RVM at many points on the ROC curve, performance only exceeded that of generative context learning at low PD levels. These results suggested that if the contextual features are effective in generatively clustering according to relevant contextual factors, discriminative context learning may be unnecessary.

The idea of context as a spatially-varying property was explored further in Chapter 6. In this chapter, contextual features were extracted from the background at regular intervals. This feature extraction technique was referred to as *context sampling*. By sampling context throughout all space, context was decoupled from the anomalies being classified. Instead, context was learned for large stretches of target-free data so that when an anomaly was encountered, its context would already have been inferred. Two spatial context models were proposed. The first was based on the DPGMM, but was trained on features extracted through context sampling. Although the DPGMM was trained on samples collected a large area, in the statistical sense each sample was treated as an independent observation. Therefore, a context model based on HMMs was also considered for incorporating the spatial dependency of samples into inference.

Spatially-dependent context models have a physical motivation, since many contextual factors (such as soil moisture) may be localized to a certain area. Therefore, it may be preferable to use more information from nearby locations when inferring the context of the present location. For context modeling, the SBHMM [69] was used as a nonparametric extension of the HMM that allowed for the inference of

the effective number of spatially-varying states. The performance of the DPGMM and SBHMM in spatial context modeling were compared, and it was shown that the SBHMM favored sharp transitions between different contexts while the DPGMM favored more gradual transitions. With regard to context-dependent fusion performance, both spatial context models provided additional performance improvements over the alarm-based techniques used in previous chapters. However, the DPGMM appeared to perform more consistently than the SBHMM when multiple realizations of the models were compared.

Finally, in Chapter 7, the context-dependent classification framework originally developed for GPR was applied to buried threat detection in airborne HSI data. Although the same statistical framework was applicable to this problem, different contextual factors needed to be exploited because the phenomenology of HSI differs from that of GPR. Two approaches were considered for extracting contextual information from HSI data. The first utilized the averaged background spectra near detected anomalies, which was indicative of the relative time of day (morning, afternoon, or night). However, it was also important to consider training data that did not exhibit such drastic temporal differences. Therefore, the second approach extracted contextual features from the background using *spectral unmixing* to yield the local abundances of several constituent *endmember* spectra. Context-dependent classification was performed on HSI data using the supervised, generative DPGMM, and discriminative DPGMM-RVM modeling techniques. For both sources of contextual information, performance was improved over a conventional linear classifier. However, context modeling based on spectral unmixing led to greater improvements in performance.

8.2 Considerations for Fielded Systems

Although the models and algorithms proposed in this dissertation were designed with fielded application (e.g., HMDS) in mind, a variety of factors have not yet been considered. In particular, greater attention should be paid to improving the efficiency of contextual feature extraction. The energy and reflection coefficient features require only simple calculations, but the process of extracting the matching pursuits and linear prediction features must be improved for real-time use. The efficiency of matching pursuits can be improved dramatically by careful design of the dictionary. This can include restricting the number of elements by limiting the number of pulse locations (in time), as well as adjusting the width of the pulses to better-reflect the pulses that make up a GPR A-scan. In this work, the pulse width was set to a single value that generally matched the width of the transmitted differentiated-Gaussian pulse. However, dispersion effects in soil propagation are inevitable, and the width of received pulses may change with time. Better understanding of this phenomenon may allow for the matching pursuits dictionary to be designed to better-reflect the structure of GPR A-scans.

The process of extracting contextual features based on linear prediction, as implemented in this work, involved training autoregressive models on individual segments of GPR data. In the case of alarm-based context learning, these segments consisted of the 100 A-scans collected before an alarm. In spatial context modeling, the segments were the 100 A-scans collected before the current background sample, and therefore much of the data used to compute these features at subsequent samples was redundant. Because linear prediction filtering is also a major component of the HMDS prescreening algorithm (see [38]), it may be more efficient to incorporate the prescreener’s internal calculations into contextual feature extraction. However, the prescreener was treated in this work as a “black box” and this idea was not explored.

Aside from feature extraction, as well as algorithm training (which is meant to be performed offline), all computations involve linear computations and/or canonical probability density functions (see Appendix A). Therefore, if the efficiency of feature extraction can be improved, context-dependent classification as proposed in this dissertation should be implementable for real-time processing on a fielded system. Performance, obviously, is dependent on sufficient training. The data used to train the algorithms used in this work was collected in 2009 on domestic military reservations, which present operating conditions that are more ideal than field conditions. Furthermore, although the target population consisted of real anti-tank landmines and a variety of simulated IEDs, it is a limited subset of the actual buried explosive threat. Recall from Chapter 1 that the IED threat is constantly changing and adapting to countermeasures. It is important that if context-dependent learning (and other buried threat detection algorithms) were to be deployed in a fielded system, the training data reflect field conditions as closely as possible.

8.3 Future Work

Beyond the questions of how to improve the efficiency of context-dependent fusion for fielded GPR systems, several unanswered theoretical questions should be the focus of future work. Future considerations must consider improving learning through sampling methods, as well as explore new challenges such as discriminative spatial context learning, comparing context-dependent learning to nonlinear classification models, and determining whether there is potential for online Bayesian context learning via nonparametric models.

Although context learning was performed using VB in this work, there is no reason why Markov chain Monte Carlo (MCMC) techniques such as Gibbs sampling cannot be used for approximate inference. It is well-known that MCMC is more accurate than VB, since it is based on sampling the posterior densities rather than iteratively

optimizing a lower bound from a randomized initial solution. Therefore, MCMC is not susceptible to converging to a local optimum solution, but this benefit comes at the expense of greater computational cost. However, because context learning is performed offline in this work, the greater computational cost of MCMC should not be a factor.

An question arising from the conclusions from Chapter 6 is whether a sequential context model can be learned discriminatively. Just as the DPGMM-RVM was used as a discriminative context model in Chapter 5, it may be possible to learn a discriminative SBHMM-RVM context model. A hybrid HMM-HME was originally proposed in [61] for speech recognition applications. Generalizing this model to accommodate a nonparametric HMM (i.e., SBHMM) and a mixture of sparse classifiers (i.e., RVMs) would be both academically interesting and practical for buried threat detection and speech recognition alike.

In all performance comparisons, context-dependent classification was compared to a single linear RVM that incorporated no contextual information. Comparisons to nonlinear classifiers, such as a kernel RVM, using the combined target and contextual features were never made. Although nonlinear classifiers may be competitive with alarm-based context-dependent learning, they would do so by including context as additional features of an observation rather than the state of the world at a given location. The comparisons made to IQGME in Chapter 5 address this issue, illustrating the advantages and disadvantages of performing context learning in separate feature spaces for alarm-based classification. However, it would be difficult to adapt IQGME or a nonlinear classifier that utilizes contextual information in the same fashion as the spatial context models presented in Chapter 6. By treating context as a property of a continuously-varying environment, and not a property of discrete observations within that environment, context-dependent learning satisfies our intuition in a way that nonlinear classification does not. Future work should consider a

series of synthetic data experiments, and other real-world examples outside of buried threat detection, to illustrate this important difference between context-dependent learning and standard nonlinear classification.

Future investigations must also consider what to do when new contexts are encountered in the field. This is a legitimate question, since the fielded algorithm would be trained on domestic data collected under somewhat idealized conditions. If a system using context-dependent classification as proposed in this work were to enter a previously-unseen context, the contextual features extracted from the data would appear to be statistical outliers. Although the likelihood of being in any one of the known contexts would be very small, the differences in likelihood between the contexts could be an order of magnitude (i.e. 10^{-4} versus 10^{-6}) and posterior inference would favor with surety the context with the greater likelihood. This could be modified by imposing a likelihood threshold, and if the likelihoods of all contexts fall below it each one would be treated as equally-unlikely. This would result in the system behaving in these conditions as if it were incorporating no contextual information at all.

However, online context learning may be a more attractive option for dealing with newly-encountered environments. This type of learning may be supported by the Dirichlet process. Recall the Chinese restaurant process; as more people enter the restaurant, tables that were empty at one time will fill up as time progress. Therefore, as more data is collected in the field, new context distributions can potentially be learned. To make online context learning viable, VB must be used to conserve processing resources. However, online VB for nonparametric models is still being explored for previously-developed nonparametric models [138, 139]. It should also be noted that the Chinese restaurant process does not support customers moving *between* tables. This suggests that although the DP may be a useful prior in learning new contexts as they are encountered, it may present difficulties in forgetting contexts

that are never seen in the field.

8.4 Broader Applications

Context-dependent classification as proposed in this dissertation may have broader application to areas outside of GPR and HSI sensing. The most general implication of this work is the notion that contextual factors can be *embraced*, rather than mitigated, to improve performance. This concept can be applied to a variety of statistical learning applications in the sensing field and beyond.

An example of another sensing technology that may benefit from context-dependent learning is laser induced breakdown spectroscopy (LIBS), which has shown potential for use in “fingerprinting” different chemical compounds [140,141]. LIBS operates by focusing a highly-powered laser onto a material to form a plasma. The plasma emits a distinctive spectrum that characterizes the material’s chemical composition. In applications of LIBS to classifying chemical, biological, radiological, nuclear, and explosive (CBRNE) residues, plasma will also be formed from the background substrate, resulting in the background spectrum being mixed with the spectrum of interest. Preliminary studies based on the work presented in this dissertation have suggested that residues could be better-classified by LIBS if the spectrum of the background is correctly identified first, suggesting that fieldability of LIBS sensors may be improved by embracing a context-dependent treatment of the classification problem [142].

Another area that may benefit from context-dependent classification may be in neurological prostheses, such as brain-computer interfaces (BCI). Systems such as the P300 speller exploit features of electroencephalogram (EEG) signals, recorded by an electrode cap worn by an amyotrophic lateral sclerosis (ALS) patient, to select characters as they become highlighted on a computer display [143]. A problem currently being investigated for BCI is channel selection, i.e. determining which electrodes are

most informative for identifying when the correct character was selected [144]. Just as *a priori* knowledge of GPR and HSI phenomenology was leveraged as a source of contextual information for improving detection of buried objects in that data, neuroscience may be a source of contextual information for performing channel selection in BCI. By anticipating which section of the brain would yield the most informative response, character identification performance and overall system throughput can potentially be improved.

Finally, context-dependent learning may someday find applications in the broader area of statistical data mining. Many experts have noted that society is entering the age of big data, and virtually all industries are demanding intelligent processing solutions to facilitate decision-making [145]. This problem has been brought to mainstream attention over recent years through a highly-publicized data mining competition spearheaded by Netflix, which sought to improve its movie recommendation algorithm [146]. As more customer data becomes available through Internet transactions and social networks, online service providers will have more consumer information available to them than ever before. For example, in recommending music to listeners, leading algorithms identify the genre of a recording solely based on frequency-domain features of the audio signal (e.g., [113, 147]). However, valuable contextual information is also available through associated metadata (artist biography, lyrics, subject matter) as well as in complementary media such as books or films that the customer also enjoys. The consumption of media by a user's friends may also be a source of contextual information in making a recommendation to a particular user.

In conclusion, significant contributions to the remote sensing and machine learning fields were made through this work. Several Bayesian techniques for context learning were proposed, and they were shown to provide useful information to improve the performance of GPR and HSI systems used for detecting landmines and

IEDs. Success was achieved in further bridging the fields of physics and statistics. It was illustrated that the statistics of data obtained through physical phenomena that are often overlooked can, in fact, be leveraged in making decisions. The results presented throughout this dissertation demonstrated that by thinking "out of the box", and approaching a problem from a different angle than previous researchers, fielded technologies can continue to be improved upon.

Appendix A

Probability Distributions

The probability distributions for all random variables considered in this dissertation are presented in this appendix. For each distribution, the functional form of the PDF, descriptions of the parameters, and moments necessary for all calculations are provided. The written formats of the PDFs are based on Bishop's text [71].

A.1 Bernoulli Distribution

The Bernoulli distribution is for a single binary variable, $x \in \{0, 1\}$, representing either a *positive* or *null* outcome of an experiment. The random variable, x , is denoted as Bernoulli-distributed by

$$x \sim \text{Bernoulli}(x|\theta). \quad (\text{A.1})$$

A.1.1 Parameters

The parameter of the Bernoulli distribution is θ , such that

$$\theta = p(x = 1). \quad (\text{A.2})$$

A.1.2 Probability Density Function

The density function for the Bernoulli distribution is

$$p(x|p) = x^p (1 - x)^{1-p}. \quad (\text{A.3})$$

A.1.3 Moments

The mean and variance of the Bernoulli distribution are

$$\mathbb{E}[x] = \theta, \quad (\text{A.4})$$

$$\text{Var}[x] = \theta(1 - \theta). \quad (\text{A.5})$$

A.1.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence (KLD) between two Bernoulli distributions is

$$\mathbb{KLD}[q(x|\theta_q) || p(x|\theta_p)] = \theta_q \log \frac{\theta_q}{\theta_p} + (1 - \theta_q) \log \frac{1 - \theta_q}{1 - \theta_p}. \quad (\text{A.6})$$

A.2 Binomial Distribution

The binomial distribution gives the probability of observing x positive Bernoulli trials in N experiments. The random variable, x , is denoted as binomial-distributed by

$$x \sim \text{Binomial}(x|N, \theta). \quad (\text{A.7})$$

A.2.1 Parameters

Like the Bernoulli distribution, the parameter of the binomial distribution is θ , such that

$$0 \leq \theta \leq 1 \quad (\text{A.8})$$

A.2.2 Probability Density Function

The density function for the binomial distribution is

$$p(x|N, \theta) = \binom{N}{x} \theta^x (1 - \theta)^{N-x}. \quad (\text{A.9})$$

A.2.3 Moments

The mean and variance of the binomial distribution are

$$\mathbb{E}[x] = N\theta, \quad (\text{A.10})$$

$$\text{Var}[x] = N\theta(1 - \theta). \quad (\text{A.11})$$

A.2.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence (KLD) between two Binomial distributions is:

$$\mathbb{KLD}[q(x|N_q, \theta_q) || p(x|N_p, \theta_p)] = \sum_{i=0}^N \binom{N_q}{i} \theta_q^i (1 - \theta_q)^{N_q-i} \log \left[\frac{\binom{N_q}{i} \theta_q^i (1 - \theta_q)^{N_q-i}}{\binom{N_p}{i} \theta_p^i (1 - \theta_p)^{N_p-i}} \right] \quad (\text{A.12})$$

A.3 Multinomial Distribution

The multinomial distribution is a multivariate generalization of the Bernoulli distribution to a D -dimensional binary variable $\mathbf{x}$, with elements $x_d \in \{0, 1\}$ constrained to sum to unity, i.e. $\sum_d x_d = 1$. The random vector, $\mathbf{x}$, is denoted as multinomial-distributed by

$$\mathbf{x} \sim \text{Multinomial}(\mathbf{x}|\boldsymbol{\theta}). \quad (\text{A.13})$$

A.3.1 Parameters

The parameter of the multinomial distribution is the probability vector $\boldsymbol{\theta}$, whose elements must satisfy the following:

$$0 \leq \theta_d \leq 1, \quad d = 1, 2, \dots, D, \quad (\text{A.14})$$

$$\sum_{d=1}^D \theta_d = 1. \quad (\text{A.15})$$

A.3.2 Probability Density Function

The density function for the multinomial distribution is

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{d=1}^D \theta_d^{x_d} \quad (\text{A.16})$$

A.3.3 Moments

The mean and variance of the multinomial distribution are:

$$\mathbb{E}[x_d] = \theta_d \quad (\text{A.17})$$

$$\text{Var}[x_d] = \theta_d(1 - \theta_d) \quad (\text{A.18})$$

A.3.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence (KLD) between two Multinomial distributions is:

$$\mathbb{KLD}[q(\mathbf{x}|\boldsymbol{\theta}_q) || p(\mathbf{x}|\boldsymbol{\theta}_p)] = \sum_{d=1}^D \theta_{qd} \log \frac{\theta_{qd}}{\theta_{pd}} \quad (\text{A.19})$$

A.4 Beta Distribution

The Beta distribution is for the continuous variable, $x \in [0, 1]$. Since the Beta distribution has finite support between zero and one, it is often used to represent uncertainty in the probability of an event. The random variable, x , is denoted as Beta-distributed by

$$x \sim \text{Beta}(x|a, b). \quad (\text{A.20})$$

A.4.1 Parameters

The parameters of the Beta distribution are a and b , such that

$$a > 0, \quad (\text{A.21})$$

$$b > 0. \quad (\text{A.22})$$

A.4.2 Probability Density Function

The density function for the Beta distribution is

$$p(x|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1} (1-x)^{b-1}, \quad (\text{A.23})$$

where $\Gamma(\cdot)$ denotes the Gamma function given by

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt \quad (\text{A.24})$$

A.4.3 Moments

The mean and variance of the Beta distribution are:

$$\mathbb{E}[x] = \frac{a}{a+b} \quad (\text{A.25})$$

$$\text{Var}[x] = \frac{ab}{(a+b)^2(a+b+1)} \quad (\text{A.26})$$

A.4.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence (KLD) between two Beta distributions is.

$$\begin{aligned} \mathbb{KLD}[q(x|a_q, b_q) || p(x|a_p, b_p)] &= \log \frac{\Gamma(a_q + b_q)}{\Gamma(a_p + b_p)} + \log \frac{\Gamma(a_p)}{\Gamma(a_q)} + \log \frac{\Gamma(b_p)}{\Gamma(b_q)} \\ &+ [a_q - a_p] [\psi(a_q) - \psi(a_q + b_q)] \\ &+ [b_q - b_p] [\psi(b_q) - \psi(a_q + b_q)]. \end{aligned} \quad (\text{A.27})$$

A.5 Dirichlet Distribution

The Dirichlet distribution is a multivariate extension of the Beta distribution for a D -dimensional vector, $\mathbf{x}$. The random vector, $\mathbf{x}$, is denoted as Beta-distributed by

$$\mathbf{x} \sim \text{Dir}(\mathbf{x}|\boldsymbol{\alpha}). \quad (\text{A.28})$$

A.5.1 Parameters

The parameters of the Dirichlet distribution are the elements of the vector, $\boldsymbol{\alpha}$, such that

$$0 \leq \alpha_d \leq 1, \quad d = 1, 2, \dots, D, \quad (\text{A.29})$$

$$\sum_{d=1}^D \alpha_d = 1. \quad (\text{A.30})$$

A.5.2 Probability Density Function

The density function for the Dirichlet distribution is

$$p(\mathbf{x}|\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{d=1}^D \alpha_d\right)}{\prod_{d=1}^D \Gamma(\alpha_d)} \prod_{d=1}^D x_d^{\alpha_d-1}. \quad (\text{A.31})$$

A.5.3 Moments

The mean and variance of the Dirichlet distribution are

$$\mathbb{E}[x_d] = \frac{\alpha_d}{\sum_{k=1}^D \alpha_k}, \quad (\text{A.32})$$

$$\text{Var}[x_d] = \frac{\alpha_d \left(-\alpha_d + \sum_{k=1}^D \alpha_k\right)}{\left(\sum_{k=1}^D \alpha_k\right)^2 \left(1 + \sum_{k=1}^D \alpha_k\right)}. \quad (\text{A.33})$$

A.5.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence between two Dirichlet distributions is

$$\begin{aligned} \text{KLD} [q(\mathbf{x}|\boldsymbol{\alpha}_q) || p(\mathbf{x}|\boldsymbol{\alpha}_p)] &= \log \frac{\Gamma\left(\sum_{d=1}^D \alpha_{q_d}\right)}{\Gamma\left(\sum_{d=1}^D \alpha_{p_d}\right)} + \sum_{d=1}^D \log \frac{\Gamma(\alpha_{p_d})}{\Gamma(\alpha_{q_d})} \\ &+ \sum_{d=1}^D [\alpha_{q_d} - \alpha_{p_d}] \left[\psi(\alpha_{q_d}) - \psi\left(\sum_{k=1}^D \alpha_{q_k}\right) \right]. \end{aligned} \quad (\text{A.34})$$

A.6 Gamma Distribution

The Gamma distribution is over a positive random variable, $x > 0$, governed by two positive parameters to ensure proper normalization. The random variable, x , is denoted as Gamma-distributed by

$$x \sim \text{Gamma}(x|a, b). \quad (\text{A.35})$$

A.6.1 Parameters

The parameters of the Gamma distribution are a and b , such that

$$a > 0, \quad (\text{A.36})$$

$$b > 0. \quad (\text{A.37})$$

A.6.2 Probability Density Function

The density function for the Gamma distribution is

$$p(x|a, b) = \frac{1}{\Gamma(a)} b^a x^{a-1} e^{-bx}, \quad (\text{A.38})$$

where $\Gamma(\cdot)$ denotes the Gamma function given by (A.24).

A.6.3 Moments

The mean and variance of the Gamma distribution are:

$$\mathbb{E}[x] = \frac{a}{b} \quad (\text{A.39})$$

$$\text{Var}[x] = \frac{a}{b^2} \quad (\text{A.40})$$

A.6.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence (KLD) between two Gamma distributions is

$$\begin{aligned} \text{KLD}[q(x|a_q, b_q) || p(x|a_p, b_p)] = & (a_q - 1)\psi(a_q) + \log b_q - a_q - \log \Gamma(a_q) + \log \Gamma(a_p) \\ & - a_p \log b_p - (a_p - 1)[\psi(a_q) - \log b_q] + \frac{a_q b_p}{b_q}, \end{aligned} \quad (\text{A.41})$$

where $\psi(\cdot)$ is the digamma function defined by

$$\psi(x) = \frac{d}{dx} \log \Gamma(x). \quad (\text{A.42})$$

A.7 Normal (Gaussian) Distribution

The Normal distribution of the continuous variable, x , has infinite support and is governed by the mean and variance parameters. The random variable, x , is denoted as Normally-distributed by

$$x \sim \mathcal{N}(x|\mu, \sigma). \quad (\text{A.43})$$

A.7.1 Parameters

The parameters of the Normal distribution are μ and σ , such that

$$-\infty < \mu < \infty, \quad (\text{A.44})$$

$$\sigma > 0. \quad (\text{A.45})$$

A.7.2 Probability Density Function

The density function for the Normal distribution is

$$p(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (\text{A.46})$$

A.7.3 Moments

$$\mathbb{E}[x] = \mu \quad (\text{A.47})$$

$$\text{Var}[x] = \sigma^2 \quad (\text{A.48})$$

A.7.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence (KLD) between two Normal distributions is

$$\mathbb{KLD}[q(x|\mu_q, \sigma_q)||p(x|\mu_p, \sigma_p)] = \frac{1}{2} \log \frac{\sigma_p^2}{\sigma_q^2} + \frac{\mu_q^2 + \mu_p^2 + \sigma_q^2 - 2\mu_q\mu_p}{2\sigma_p^2} - \frac{1}{2}. \quad (\text{A.49})$$

A.8 Multivariate Normal (Gaussian) Distribution

The multivariate extension of the Normal distribution is over the D -dimensional random vector, $\mathbf{x}$, whose elements $x_d \in (-\infty, \infty)$ for $d = 1, 2, \dots, D$. The distribution is governed by the mean vector and covariance matrix. The random vector, $\mathbf{x}$, is denoted as Normally-distributed by

$$\mathbf{x} \sim \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (\text{A.50})$$

A.8.1 Parameters

The parameters of the Normal distribution are $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, such that

$$-\infty < \mu_d < \infty, \quad d = 1, 2, \dots, D, \quad (\text{A.51})$$

$$\boldsymbol{\Sigma} \text{ is positive-definite.} \quad (\text{A.52})$$

A.8.2 Probability Density Function

The density function for the multivariate Normal distribution is

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-D/2} |\boldsymbol{\Sigma}|^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}. \quad (\text{A.53})$$

A.8.3 Moments

The mean and covariance elements are:

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \quad (\text{A.54})$$

$$\mathbb{E}[x_d x_k] = \mu_d \mu_k + \Sigma_{dk} \quad (\text{A.55})$$

$$\text{Cov}[\mathbf{x}] = \boldsymbol{\Sigma} \quad (\text{A.56})$$

A.8.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence (KLD) between two multivariate Normal distributions is

$$\begin{aligned} \text{KLD}[q(\mathbf{x}|\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)||p(\mathbf{x}|\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)] &= \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}_q|} + \frac{1}{2} \text{Tr}[\boldsymbol{\Sigma}_p^{-1} \boldsymbol{\Sigma}_q] \\ &\quad + \frac{1}{2} (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^T \boldsymbol{\Sigma}_p^{-1} (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p) - \frac{D}{2}. \end{aligned} \quad (\text{A.57})$$

A.9 Wishart Distribution

The Wishart distribution is over the $D \times D$ matrix $\boldsymbol{\Lambda}$, and is the conjugate prior for the precision (inverse covariance) matrix of a multivariate Normal distribution. The random matrix, $\boldsymbol{\Lambda}$, is denoted as Wishart-distributed by

$$\boldsymbol{\Lambda} \sim \mathcal{W}(\boldsymbol{\Lambda}|\mathbf{W}, \nu). \quad (\text{A.58})$$

A.9.1 Parameters

The parameters of the Wishart distribution are the degrees of freedom, ν , and the scale matrix, $\mathbf{W}$, which must satisfy the following:

$$\nu > D - 1 \quad (\text{A.59})$$

$$\mathbf{W} \text{ is positive definite.} \quad (\text{A.60})$$

A.9.2 Probability Density Function

The density function for the Wishart distribution is

$$p(\mathbf{\Lambda}|\mathbf{W}, \nu) = B(\mathbf{W}, \nu) |\mathbf{\Lambda}|^{\frac{\nu-D-1}{2}} \exp\left(-\frac{1}{2}\text{Tr}[\mathbf{W}^{-1}\mathbf{\Lambda}]\right), \quad (\text{A.61})$$

where

$$B(\mathbf{W}, \nu) = |\mathbf{W}|^{-\nu/2} \left[2^{\nu D/2} \pi^{D(D-1)/4} \prod_{d=1}^D \Gamma\left(\frac{\nu+1-d}{2}\right) \right]^{-1}. \quad (\text{A.62})$$

A.9.3 Moments

The expected values of $\mathbf{\Lambda}$ and $\log |\mathbf{\Lambda}|$ are

$$\mathbb{E}[\mathbf{\Lambda}] = \nu \mathbf{W} \quad (\text{A.63})$$

$$\mathbb{E}[\log |\mathbf{\Lambda}|] = \sum_{d=1}^D \psi\left(\frac{\nu+1-d}{2}\right) + D \log 2 + \log |\mathbf{W}|, \quad (\text{A.64})$$

where $\psi(\cdot)$ is the digamma function defined by (A.42).

A.9.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence (KLD) between two Wishart distributions is

$$\begin{aligned}
& \mathbb{KLD} [q(\mathbf{\Lambda}|\mathbf{W}_q, \nu_q) || p(\mathbf{\Lambda}|\mathbf{W}_p, \nu_p)] \\
&= \left(\frac{\nu_q - D - 1}{2} \right) \left(\sum_{d=1}^D \psi \left(\frac{\nu_q - d + 1}{2} \right) + D \log 2 + \log |\mathbf{W}_q| \right) \\
&\quad - \left(\frac{\nu_p - D - 1}{2} \right) \left(\sum_{d=1}^D \psi \left(\frac{\nu_p - d + 1}{2} \right) + D \log 2 + \log |\mathbf{W}_p| \right) \\
&\quad - \frac{\nu_q D}{2} + \frac{\nu_q}{2} \text{Tr} (\mathbf{B}_p^{-1} \mathbf{B}_q) + \log \frac{2^{\nu_p D/2} |\mathbf{B}_p|^{-\nu_p/2} \Gamma_D (\nu_p/2)}{2^{\nu_q D/2} |\mathbf{B}_q|^{-\nu_q/2} \Gamma_D (\nu_q/2)}.
\end{aligned} \tag{A.65}$$

A.10 Normal-Wishart Distribution

The Normal-Wishart distribution is a joint density over the $D \times 1$ vector, $\mathbf{x}$, and the $D \times D$ matrix, $\mathbf{\Lambda}$. It is the conjugate prior for a multivariate Normal distribution with unknown mean and precision (inverse covariance) matrix. The random variables, $(\mathbf{x}, \mathbf{\Lambda})$ are denoted as Normal-Wishart distributed by

$$(\mathbf{x}, \mathbf{\Lambda}) \sim \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, u^{-1} \mathbf{\Lambda}^{-1}) \mathcal{W}(\mathbf{\Lambda} | \mathbf{W}, \nu). \tag{A.66}$$

A.10.1 Parameters

The parameters of the Normal-Wishart distribution involve many of the same parameters of the Normal and Wishart distributions. They include the location (mean), $\boldsymbol{\mu}$; a precision scale, u ; the degrees of freedom, ν ; and the scale matrix, $\mathbf{W}$, which must satisfy the following:

$$-\infty \leq \mu_d \leq \infty, \quad d = 1, 2, \dots, D \tag{A.67}$$

$$u > 0 \tag{A.68}$$

$$\nu > D - 1 \tag{A.69}$$

$$\mathbf{W} \text{ is positive definite.} \tag{A.70}$$

A.10.2 Probability Density Function

The density function for the Normal-Wishart distributions is obtained by multiplication of the Normal and Wishart density functions, which yields

$$\begin{aligned}
p(\mathbf{x}, \mathbf{\Lambda} | \boldsymbol{\mu}, u, \mathbf{W}, \nu) &= \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, u^{-1} \mathbf{\Lambda}^{-1}) \mathcal{W}(\mathbf{\Lambda} | \mathbf{W}, \nu) \\
&= B(\mathbf{W}, \nu) (2\pi)^{-D/2} |u\mathbf{\Lambda}|^{1/2} |\mathbf{\Lambda}|^{\frac{\nu-D-1}{2}} \exp\left(-\frac{1}{2} \left[(\mathbf{x} - \boldsymbol{\mu})^T u \mathbf{\Lambda} (\mathbf{x} - \boldsymbol{\mu}) \right. \right. \\
&\quad \left. \left. + \text{Tr}(\mathbf{W}^{-1} \mathbf{\Lambda}) \right] \right).
\end{aligned} \tag{A.71}$$

where $B(\mathbf{W}, \nu)$ is defined in (A.62).

A.10.3 Moments

The expected values of $\mathbf{x}$ and $\mathbf{\Lambda}$ follow the Normal and Wishart distributions:

$$\mathbb{E}[\mathbf{x} | u^{-1} \mathbf{\Lambda}^{-1}] = \boldsymbol{\mu} \tag{A.72}$$

$$\mathbb{E}[\mathbf{\Lambda}] = \nu \mathbf{W} \tag{A.73}$$

$$\mathbb{E}[\log |\mathbf{\Lambda}|] = \sum_{d=1}^D \psi\left(\frac{\nu + 1 - d}{2}\right) + D \log 2 + \log |\mathbf{W}|, \tag{A.74}$$

where $\psi(\cdot)$ is the digamma function defined by (A.42).

A.10.4 Kullback-Leibler Divergence

The Kullback-Leibler Divergence between two Normal-Wishart distributions is

$$\begin{aligned}
&\mathbb{KLD}[q(\mathbf{x}, \mathbf{\Lambda} | \boldsymbol{\mu}_q, u_q, \mathbf{W}_q, \nu_q) || p(\mathbf{x}, \mathbf{\Lambda} | \boldsymbol{\mu}_p, u_p, \mathbf{W}_p, \nu_p)] \\
&= \frac{D}{2} \left(\frac{u_p}{u_q} + \log \frac{u_q}{u_p} - 1 \right) + \frac{1}{2} (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^T u_p \nu_q \mathbf{W}_q (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p) \\
&\quad + \mathbb{KLD}[\mathcal{W}(\mathbf{\Lambda} | \mathbf{W}_q, \nu_q) || \mathcal{W}(\mathbf{\Lambda} | \mathbf{W}_p, \nu_p)],
\end{aligned} \tag{A.75}$$

where $\mathbb{KLD}[\mathcal{W}(\mathbf{\Lambda}|\mathbf{W}_q, \nu_q) || \mathcal{W}(\mathbf{\Lambda}|\mathbf{W}_p, \nu_p)]$ is a Kullback-Leibler divergence between two Wishart distributions.

Appendix B

Relevance Vector Machines

The relevance vector machine (RVM), originally proposed by Tipping [83], was used in this work as a statistical model for classification and regression. In this appendix, the variational Bayesian update equations for a single RVM regressor/classifier [84] and a mixture of RVM classifiers are presented.

The RVM is a sparseness-promoting technique for Bayesian inference of regression and classification models. Like support vector machines (SVMs) [92], RVMs seek a sparse weighting of kernel-transformed features. While the SVM accomplishes this by maximizing the margin between classes, the RVM utilizes sparseness-promoting priors. The overall effect is a model that does not require tuning (due to the use of noninformative priors) and has different sparseness properties.

B.1 RVM Regression

B.1.1 Generative Model and Variable Definitions

$$y_n = \mathbf{w}^T \phi(\mathbf{x}_n)^T \quad (\text{B.1})$$

$$(t_n | \mathbf{w}, \mathbf{x}_n) \sim \mathcal{N}(t_n | y_n, \tau^{-1}) \quad (\text{B.2})$$

$n = 1, 2, \dots, N$ is observation index

$d = 1, 2, \dots, D$ is dimension index

$\mathbf{x}_n$ is feature vector of observation n

$\phi(\mathbf{x}_n)$ is a D -dimensional kernel transformation of $\mathbf{x}_n$

$\mathbf{w}$ is a D -dimensional weight vector

y_n is the model output for observation n

t_n is the target value for observation n

τ is the precision of t

B.1.2 Priors

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w} | 0, \mathbf{A}^{-1}), \text{ where } \mathbf{A} = \text{diag}(\boldsymbol{\alpha}) \quad (\text{B.3})$$

$$\alpha_d \sim \text{Gamma}(\alpha_d | a_0, b_0) \quad (\text{B.4})$$

$$\tau \sim \text{Gamma}(\tau | c_0, d_0) \quad (\text{B.5})$$

B.1.3 Variational Posterior on $\mathbf{w}$

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{w})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{w}) \propto \langle \log p(\mathbf{w} | -) \rangle \quad (\text{B.6})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(\mathbf{w}|-) = \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) + \log p(\mathbf{w}) - K, \quad (\text{B.7})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{w}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\log p(\mathbf{w}|-)$$

$$\begin{aligned} &= \sum_{n=1}^N \left[-\cancel{\frac{1}{2} \log 2\pi} + \cancel{\frac{1}{2} \log \tau} - \frac{\tau}{2} (t_n - y_n)^2 \right] - \cancel{\frac{P}{2} \log 2\pi} + \cancel{\frac{1}{2} \log |\mathbf{A}|} - \frac{1}{2} \mathbf{w}^T \mathbf{A} \mathbf{w} - K \\ &= -\frac{1}{2} \left(\tau \sum_{n=1}^N \left[t_n - \mathbf{w}^T \phi(\mathbf{x}_n) \right]^2 + \mathbf{w}^T \mathbf{A} \mathbf{w} \right) - K \\ &= -\frac{1}{2} \left(\cancel{\tau \sum_{n=1}^N t_n^2} - 2\mathbf{w}^T \tau \sum_{n=1}^N t_n \phi(\mathbf{x}_n) + \tau \sum_{n=1}^N \mathbf{w}^T \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \mathbf{w} + \mathbf{w}^T \mathbf{A} \mathbf{w} \right) - K \\ &= -\frac{1}{2} \left(-2\mathbf{w}^T \tau \sum_{n=1}^N t_n \phi(\mathbf{x}_n) + \mathbf{w}^T \left[\mathbf{A} + \tau \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right] \mathbf{w} \right) - K \end{aligned} \quad (\text{B.8})$$

Completing the square reveals that $\mathbf{w}$ is Gaussian:

$$\log p(\mathbf{w}|-) = \log \mathcal{N}(\mathbf{w}|\mathbf{m}, \Sigma) \quad (\text{B.9})$$

where

$$\mathbf{m} = \tau \Sigma \sum_{n=1}^N t_n \phi(\mathbf{x}_n) \quad (\text{B.10})$$

$$\Sigma = \left[\mathbf{A} + \tau \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right]^{-1}, \quad (\text{B.11})$$

Useful moments in VB updates for other model parameters:

$$\langle \mathbf{w} \rangle = \mathbf{m} \quad (\text{B.12})$$

$$\langle \mathbf{w} \mathbf{w}^T \rangle = \mathbf{m} \mathbf{m}^T + \Sigma \quad (\text{B.13})$$

B.1.4 Variational Posterior on α

It was derived in Section that the NFE is maximized by a variational posterior, $q(\boldsymbol{\alpha})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\boldsymbol{\alpha}) \propto \langle \log p(\boldsymbol{\alpha} | -) \rangle \quad (\text{B.14})$$

The true posterior may be calculated from Bayes' theorem:

$$p(\boldsymbol{\alpha} | -) \propto p(\mathbf{w} | \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) \quad (\text{B.15})$$

The variational posterior can be calculated by solving the true posterior as a function of α , then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} p(\boldsymbol{\alpha} | -) &\propto \prod_{d=1}^D (2\pi)^{-\frac{1}{2}} \alpha_d^{\frac{1}{2}} \exp\left(-\frac{\alpha_d w_d^2}{2}\right) \frac{b_0^{a_0}}{\Gamma(a_0)} \alpha_d^{a_0-1} \exp(-b_0 \alpha_d) \\ &\propto \prod_{d=1}^D \alpha_d^{\frac{1}{2}} \exp\left(-\frac{\alpha_d w_d^2}{2}\right) \alpha_d^{a_0-1} \exp(-b_0 \alpha_d) \\ &\propto \prod_{d=1}^D \alpha_d^{a_0 + \frac{1}{2} - 1} \exp\left(-\alpha_d \left[b_0 + \frac{1}{2} w_d^2\right]\right) \end{aligned} \quad (\text{B.16})$$

Therefore, the α 's are Gamma distributed:

$$p(\boldsymbol{\alpha} | -) = \prod_{d=1}^D \text{Gamma}(\alpha_d | a_d, b_d), \quad (\text{B.17})$$

where

$$a_d = a_0 + \frac{1}{2} \quad (\text{B.18})$$

$$b_d = b_0 + \frac{1}{2} w_d^2, \quad (\text{B.19})$$

Useful moments in VB updates for other model parameters:

$$\langle \alpha_d \rangle = \frac{a_d}{b_d} \quad (\text{B.20})$$

B.1.5 Variational Posterior on τ

It was derived in Section that the NFE is maximized by a variational posterior, $q(\tau)$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\tau) \propto \langle \log p(\tau|-) \rangle \quad (\text{B.21})$$

The true posterior may be calculated from Bayes' theorem:

$$p(\tau|-) \propto p(\mathbf{t}|\mathbf{X}, \tau) p(\tau) \quad (\text{B.22})$$

The variational posterior can be calculated by solving the true posterior as a function of τ , then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} p(\tau|-) &\propto \prod_{n=1}^N \left[\tau^{\frac{1}{2}} \exp \left(-\frac{\tau (t_n - y_n)^2}{2} \right) \right] \frac{d_0^{c_0}}{\Gamma(c_0)} \tau^{c_0-1} \exp(-d_0 \tau) \\ &\propto \tau^{\frac{N}{2}} \exp \left[-\frac{\tau \sum_{n=1}^N (t_n - y_n)^2}{2} \right] \tau^{c_0-1} \exp(-d_0 \tau) \\ &\propto \exp \left(-\tau \left[d_0 + \frac{\sum_{n=1}^N (t_n - \mathbf{w}^T \phi(\mathbf{x}_n))^2}{2} \right] \right) \tau^{\frac{N}{2} + c_0 - 1} \\ &\propto \exp \left[-\tau \left(d_0 + \frac{1}{2} \sum_{n=1}^N t_n^2 - \mathbf{w}^T \sum_{n=1}^N t_n \phi(\mathbf{x}_n) + \frac{1}{2} \sum_{n=1}^N \phi(\mathbf{x}_n)^T \mathbf{w} \mathbf{w}^T \phi(\mathbf{x}_n) \right) \right] \tau^{\frac{N}{2} + c_0 - 1} \end{aligned} \quad (\text{B.23})$$

Therefore, τ is Gamma-distributed:

$$p(\tau|-) = \text{Gamma}(\tau|c, d) \quad (\text{B.24})$$

where

$$c = c_0 + \frac{N}{2} \quad (\text{B.25})$$

$$d = d_0 + \frac{1}{2} \sum_{n=1}^N t_n^2 - \mathbf{w}^T \sum_{n=1}^N t_n \phi(\mathbf{x}_n) + \frac{1}{2} \sum_{n=1}^N \phi(\mathbf{x}_n)^T \mathbf{w} \mathbf{w}^T \phi(\mathbf{x}_n) \quad (\text{B.26})$$

,

Useful moments in VB updates for other model parameters:

$$\langle \tau \rangle = \frac{c}{d} \quad (\text{B.27})$$

$$\langle \log \tau \rangle = \psi(c) - \psi(d), \text{ where } \psi(\phi) = \frac{d}{d\phi} \log \Gamma(\phi) \quad (\text{B.28})$$

B.1.6 Negative Free Energy

The negative free energy (NFE) serves as the variational lower bound to the true log-evidence. Therefore, in it serves as an optimization criterion for variational learning. The NFE can be expressed as the difference between the expected log-likelihood and the Kullback-Leibler divergence (KLD) between the variational posteriors and the priors:

$$\begin{aligned} \mathcal{F} &= \langle \log p(\mathbf{t}|\mathbf{w}, \mathbf{X}) \rangle - \mathbb{KLD}[q(\mathbf{w})q(\mathbf{A})q(\tau) || p(\mathbf{w}|\mathbf{A})p(\mathbf{A})p(\tau)] \\ &= \langle \log p(\mathbf{t}|\mathbf{w}, \mathbf{X}) \rangle - \mathbb{KLD}[q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})] \\ &\quad - \sum_{d=1}^D \mathbb{KLD}[q(\alpha_d) || p(\alpha_d)] - \mathbb{KLD}[q(\tau) || p(\tau)] \\ &= -\frac{N}{2} \log 2\pi + \frac{N}{2} \langle \log \tau \rangle - \frac{\langle \tau \rangle}{2} \sum_{n=1}^N [t_n - \langle \mathbf{w}^T \rangle \phi(\mathbf{x}_n)]^2 \\ &\quad - \mathbb{KLD}[q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})] - \sum_{d=1}^D \mathbb{KLD}[q(\alpha_d) || p(\alpha_d)] - \mathbb{KLD}[q(\tau) || p(\tau)] \end{aligned} \quad (\text{B.29})$$

where $\mathbb{KLD}[q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})]$ is a KLD between two Gaussian distributions, $\mathbb{KLD}[q(\alpha_d) || p(\alpha_d)]$ is a KLD between two Gamma distributions, and $\mathbb{KLD}[q(\tau) || p(\tau)]$ is also a KLD between two Gamma distributions.

B.2 RVM Classification

B.2.1 Generative Model and Variable Definitions

$$y_n = \mathbf{w}^T \phi(\mathbf{x}_n)^T \quad (\text{B.30})$$

$$\sigma(y_n) = \frac{1}{1 + e^{-y_n}} \quad (\text{B.31})$$

$$(t_n | \mathbf{w}, \mathbf{x}_n) \sim \sigma(y_n)^{t_n} [1 - \sigma(y_n)]^{1-t_n} \quad (\text{B.32})$$

$n = 1, 2, \dots, N$ is observation index

$d = 1, 2, \dots, D$ is dimension index

$\mathbf{x}_n$ is feature vector of observation n

$\phi(\mathbf{x}_n)$ is a D -dimensional kernel transformation of $\mathbf{x}_n$

$\sigma(\cdot)$ is the logistic sigmoid function

$\mathbf{w}$ is a D -dimensional weight vector

y_n is the model output for observation n

t_n is the binary label for observation n

τ is the precision of t

B.2.2 Priors

$$\mathbf{w} \sim \mathcal{N}(\mathbf{w} | 0, \mathbf{A}^{-1}), \text{ where } \mathbf{A} = \text{diag}(\boldsymbol{\alpha}) \quad (\text{B.33})$$

$$\alpha_d \sim \text{Gamma}(\alpha_d | a_0, b_0), \text{ typically } a_0 = b_0 = 10^{-6} \quad (\text{B.34})$$

B.2.3 Approximate Likelihood

Because the binomial distribution on $\mathbf{t}$ does not offer conjugate updating for our choice of the prior on $\mathbf{w}$, we impose a lower-bound approximation to the likelihood,

$p(t_n|\mathbf{w}, \mathbf{x}_n)$, that was proved by Jakkola and Jordan [97]:

$$p(t_n|\mathbf{w}, \mathbf{x}_n) \geq \sigma(\xi_n) \exp \left[\frac{\gamma_n - \xi_n}{2} - \lambda(\xi_n) (\gamma_n^2 - \xi_n^2) \right] \quad (\text{B.35})$$

where ξ_n is a variational parameter and

$$\gamma_n = (2t_n - 1) y_n \quad (\text{B.36})$$

$$\lambda(\xi_n) = \frac{1}{4\xi_n} \tanh \left(\frac{\xi_n}{2} \right) \quad (\text{B.37})$$

Therefore, the log-likelihood will be approximated as

$$\log p(\mathbf{t}_n|\mathbf{w}, \mathbf{x}_n) \geq \log \sigma(\xi_n) + \frac{1}{2} (\gamma_n - \xi_n) - \lambda(\xi_n) (\gamma_n^2 - \xi_n^2) \quad (\text{B.38})$$

B.2.4 Variational Posterior on $\mathbf{w}$

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{w})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{w}) \propto \langle \log p(\mathbf{w}|-) \rangle \quad (\text{B.39})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(\mathbf{w}|-) = \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) + \log p(\mathbf{w}) - K, \quad (\text{B.40})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{w}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned}
& \log p(\mathbf{w}|-) \\
&= \sum_{n=1}^N \left[\log \sigma(\xi_n) + \frac{1}{2} (\gamma_n - \xi_n) - \lambda(\xi_n) (\gamma_n^2 - \xi_n^2) \right] - \frac{1}{2} [P \log 2\pi + \log |\mathbf{A}| + \mathbf{w}^T \mathbf{A} \mathbf{w}] - K \\
&= -\frac{1}{2} \left[\mathbf{w}^T \mathbf{A} \mathbf{w} + \sum_{n=1}^N (2\lambda(\xi_n) \gamma_n^2 - \gamma_n) \right] - K \\
&= -\frac{1}{2} \left[\mathbf{w}^T \mathbf{A} \mathbf{w} + \sum_{n=1}^N \left(2\lambda(\xi_n) \mathbf{w}^T \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \mathbf{w} - (2t_n - 1) \mathbf{w}^T \phi(\mathbf{x}_n) \right) \right] - K \\
&= -\frac{1}{2} \left[-2\mathbf{w}^T \left(\frac{1}{2} \sum_{n=1}^N (2t_n - 1) \phi(\mathbf{x}_n) \right) + \mathbf{w}^T \left(\mathbf{A} + 2 \sum_{n=1}^N \lambda(\xi_n) \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right) \mathbf{w} \right] - K
\end{aligned} \tag{B.41}$$

Completing the square reveals that $\mathbf{w}$ is Gaussian:

$$\log p(\mathbf{w}|-) = \log \mathcal{N}(\mathbf{w}|\mathbf{m}, \Sigma) \tag{B.42}$$

where

$$\mathbf{m} = \frac{1}{2} \Sigma \left(\sum_{n=1}^N (2t_n - 1) \phi(\mathbf{x}_n) \right) \tag{B.43}$$

$$\Sigma = \left(\mathbf{A} + 2 \sum_{n=1}^N \lambda(\xi_n) \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right)^{-1}, \tag{B.44}$$

Useful moments in VB updates for other model parameters:

$$\langle \mathbf{w} \rangle = \mathbf{m} \tag{B.45}$$

$$\langle \mathbf{w} \mathbf{w}^T \rangle = \mathbf{m} \mathbf{m}^T + \Sigma \tag{B.46}$$

B.2.5 Updating ξ

Since the variational parameter ξ is assumed *known* (i.e. no prior or posterior density), the updates cannot be found starting with Bayes' theorem. Update equations

can still be derived by directly optimizing the NFE:

$$\begin{aligned}\mathcal{F} &= \langle \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) \rangle - \text{KLD}[q(\mathbf{w}) || p(\mathbf{w})] - \text{KLD}[q(\mathbf{A}) || p(\mathbf{A})] \\ \frac{\partial \mathcal{F}}{\partial \boldsymbol{\xi}} &= \frac{\partial}{\partial \boldsymbol{\xi}} \langle \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) \rangle\end{aligned}\tag{B.47}$$

Substituting the approximation for $p(\mathbf{t}_n|\mathbf{w}, \mathbf{x}_n)$:

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial \boldsymbol{\xi}} &= \sum_{n=1}^N \left[\frac{1}{1 + e^{\xi_n}} - \frac{1}{2} + 2\xi_n \lambda(\xi_n) - \frac{\partial \lambda(\xi_n)}{\partial \xi_n} (\langle \gamma_n^2 \rangle - \xi_n^2) \right] \\ &= \sum_{n=1}^N \left[\frac{e^{-\xi_n/2}}{e^{\xi_n/2} + e^{-\xi_n/2}} + \frac{\frac{1}{2}e^{\xi_n/2} - \frac{1}{2}e^{-\xi_n/2}}{e^{\xi_n/2} + e^{-\xi_n/2}} - \frac{1}{2} - \frac{\partial \lambda(\xi_n)}{\partial \xi_n} (\langle \gamma_n^2 \rangle - \xi_n^2) \right] \\ &= \sum_{n=1}^N \left[\frac{\frac{1}{2}e^{\xi_n/2} + \frac{1}{2}e^{-\xi_n/2}}{e^{\xi_n/2} + e^{-\xi_n/2}} - \frac{1}{2} - \frac{\partial \lambda(\xi_n)}{\partial \xi_n} (\langle \gamma_n^2 \rangle - \xi_n^2) \right] \\ &= - \sum_{n=1}^N \frac{\partial \lambda(\xi_n)}{\partial \xi_n} (\langle \gamma_n^2 \rangle - \xi_n^2)\end{aligned}\tag{B.48}$$

Because the derivative of $\lambda(\xi_n)$ is purely negative, $\mathcal{F}$ is maximized at

$$\xi_n^2 = \langle \gamma_n^2 \rangle = \phi(\mathbf{x}_n)^T \langle \mathbf{w}\mathbf{w}^T \rangle \phi(\mathbf{x}_n)\tag{B.49}$$

B.2.6 Variational Posterior on α

It was derived in Section that the NFE is maximized by a variational posterior, $q(\boldsymbol{\alpha})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\boldsymbol{\alpha}) \propto \langle \log p(\boldsymbol{\alpha}|-) \rangle\tag{B.50}$$

The true posterior may be calculated from Bayes' theorem:

$$p(\boldsymbol{\alpha}|-) \propto p(\mathbf{W}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha})\tag{B.51}$$

The variational posterior can be calculated by solving the true posterior as a function of α , then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned}
p(\boldsymbol{\alpha}|-) &\propto \prod_{d=1}^D (2\pi)^{-\frac{1}{2}} \alpha_d^{\frac{1}{2}} \exp\left(-\frac{\alpha_d w_d^2}{2}\right) \frac{b_0^{a_0}}{\Gamma(a_0)} \alpha_d^{a_0-1} \exp(-b_0 \alpha_d) \\
&\propto \prod_{d=1}^D \alpha_d^{\frac{1}{2}} \exp\left(-\frac{\alpha_d w_d^2}{2}\right) \alpha_d^{a_0-1} \exp(-b_0 \alpha_d) \\
&\propto \prod_{d=1}^D \alpha_d^{a_0+\frac{1}{2}-1} \exp\left(-\alpha_d \left[b_0 + \frac{1}{2} w_d^2\right]\right)
\end{aligned} \tag{B.52}$$

Therefore, the α 's are Gamma distributed:

$$q(\boldsymbol{\alpha}) = \prod_{d=1}^D \text{Gamma}(\alpha_d | a_d, b_d), \tag{B.53}$$

where

$$a_d = a_0 + \frac{1}{2} \tag{B.54}$$

$$b_d = b_0 + \frac{1}{2} w_d^2, \tag{B.55}$$

Useful moments in VB updates for other model parameters:

$$\langle \alpha_d \rangle = \frac{a_d}{b_d} \tag{B.56}$$

B.2.7 Negative Free Energy

The NFE can be expressed as the difference between the expected log-likelihood and the Kullback-Leibler divergence (KLD) between the variational posteriors and the

priors:

$$\begin{aligned}
\mathcal{F} &= \langle \log p(\mathbf{t}|\mathbf{w}, \mathbf{X}) \rangle - \text{KLD} [q(\mathbf{w}) q(\mathbf{A}) || p(\mathbf{w}|\mathbf{A}) p(\mathbf{A})] \\
&= \langle \log p(\mathbf{t}|\mathbf{w}, \mathbf{X}) \rangle - \text{KLD} [q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})] - \sum_{d=1}^D \text{KLD} [q(\alpha_d) || p(\alpha_d)] \\
&= \sum_{n=1}^N \log \sigma(\xi_n) + \frac{1}{2} (\langle \gamma_n \rangle - \xi_n) - \lambda(\xi_n) (\langle \gamma_n \rangle^2 - \xi_n^2) \\
&\quad - \text{KLD} [q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})] - \sum_{d=1}^D \text{KLD} [q(\alpha_d) || p(\alpha_d)] \\
&= \sum_{n=1}^N \log \sigma(\xi_n) + \frac{1}{2} \left[(2t_n - 1) \phi(\mathbf{x}_n)^T \langle \mathbf{w} \rangle - \xi_n \right] - \lambda(\xi_n) \left(\phi(\mathbf{x}_n)^T \langle \mathbf{w} \mathbf{w}^T \rangle \phi(\mathbf{x}_n) - \xi_n^2 \right) \\
&\quad - \text{KLD} [q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})] - \sum_{d=1}^D \text{KLD} [q(\alpha_d) || p(\alpha_d)]
\end{aligned} \tag{B.57}$$

where $\text{KLD} [q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})]$ is a KLD between two Gaussian distributions, and $\text{KLD} [q(\alpha_d) || p(\alpha_d)]$ is a KLD between two Gamma distributions.

B.3 Mixture of RVM Classifiers

B.3.1 Generative Model and Variable Definitions

$$y_{nm} = \mathbf{w}_m^T \phi(\mathbf{x}_n)^T \tag{B.58}$$

$$\sigma(y_{nm}) = \frac{1}{1 + e^{-y_{nm}}} \tag{B.59}$$

$$(t_n | \mathbf{W}, \mathbf{x}_n, \mathbf{c}) \sim (\sigma(y_{nm})^{t_n} [1 - \sigma(y_{nm})]^{1-t_n}) c_{nm} \tag{B.60}$$

$n = 1, 2, \dots, N$ is observation index

$d = 1, 2, \dots, D$ is dimension index

$m = 1, 2, \dots, M$ is mixture component index

$\mathbf{x}_n$ is feature vector of observation n

$\phi(\mathbf{x}_n)$ is a D -dimensional kernel transformation of $\mathbf{x}_n$

$\sigma(\cdot)$ is the logistic sigmoid function

$\mathbf{w}_m$ is a D -dimensional weight vector

y_m is the model m output for observation n

t_n is the binary label for observation n

$\mathbf{c}_n = \{c_{nm}\}$ is a latent variable governing mixture component selection

τ is the precision of t

B.3.2 Priors

$$\mathbf{w}_m = \mathcal{N}(\mathbf{w}_m | 0, \mathbf{A}_m^{-1}), \text{ where } \mathbf{A}_m = \text{diag}(\boldsymbol{\alpha}_m) \quad (\text{B.61})$$

$$\alpha_{md} \sim \text{Gamma}(\alpha_{md} | a_0, b_0), \text{ typically } a_0 = b_0 = 10^{-6} \quad (\text{B.62})$$

B.3.3 Variational Posterior on $\mathbf{w}$

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{W})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{W}) \propto \langle \log p(\mathbf{W} | -) \rangle \quad (\text{B.63})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(\mathbf{W} | -) = \log p(\mathbf{t} | \mathbf{W}, \mathbf{X}, -) + \log p(\mathbf{W}) - K, \quad (\text{B.64})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{W}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned}
& \log p(\mathbf{W}|-) \\
&= \sum_{n=1}^N \sum_{m=1}^M c_{nm} \left[\log \sigma(\xi_{nm}) + \frac{1}{2} (\gamma_n - \xi_{nm}) - \lambda(\xi_{nm}) (\gamma_{nm}^2 - \xi_{nm}^2) \right] \\
&\quad - \frac{1}{2} \sum_{m=1}^M [P \log 2\pi + \log |\mathbf{A}_m| + \mathbf{w}_m^T \mathbf{A}_m \mathbf{w}_m] - K \\
&= -\frac{1}{2} \sum_{m=1}^M \left[\mathbf{w}_m^T \mathbf{A}_m \mathbf{w}_m + \sum_{n=1}^N c_{nm} (2\lambda(\xi_{nm}) \gamma_{nm}^2 - \gamma_{nm}) \right] - K \\
&= -\frac{1}{2} \sum_{m=1}^M \left[\mathbf{w}_m^T \mathbf{A}_m \mathbf{w}_m + \sum_{n=1}^N c_{nm} \left(2\lambda(\xi_{nm}) \mathbf{w}_m^T \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \mathbf{w}_m - (2t_n - 1) \mathbf{w}_m^T \phi(\mathbf{x}_n) \right) \right] - K \\
&= -\frac{1}{2} \sum_{m=1}^M \left[-2\mathbf{w}_m^T \left(\frac{1}{2} \sum_{n=1}^N c_{nm} (2t_n - 1) \phi(\mathbf{x}_n) \right) \right. \\
&\quad \left. + \mathbf{w}_m^T \left(\mathbf{A}_m + 2 \sum_{n=1}^N c_{nm} \lambda(\xi_{nm}) \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right) \mathbf{w}_m \right] - K
\end{aligned} \tag{B.65}$$

Completing the square reveals that $\mathbf{W}$ is Gaussian:

$$\log p(\mathbf{W}|-) = \sum_{m=1}^M \log \mathcal{N}(\mathbf{w}_m | \mathbf{m}_m, \Sigma_m) \tag{B.66}$$

where

$$\mathbf{m}_m = \frac{1}{2} \Sigma_m \left(\sum_{n=1}^N c_{nm} (2t_n - 1) \phi(\mathbf{x}_n) \right) \tag{B.67}$$

$$\Sigma_m = \left(\mathbf{A}_m + 2 \sum_{n=1}^N c_{nm} \lambda(\xi_{nm}) \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right)^{-1}, \tag{B.68}$$

Useful moments in VB updates for other model parameters:

$$\langle \mathbf{w}_m \rangle = \mathbf{m}_m \tag{B.69}$$

$$\langle \mathbf{w}_m \mathbf{w}_m^T \rangle = \mathbf{m}_m \mathbf{m}_m^T + \Sigma_m \quad (\text{B.70})$$

B.3.4 Updating ξ

Since the variational parameter ξ is assumed *known* (i.e. no prior or posterior density), the updates are found by directly optimizing the negative free energy:

$$\begin{aligned} \mathcal{F} &= \langle \log p(\mathbf{t}|\mathbf{X}, \mathbf{W}) \rangle - \text{KLD}[q(\mathbf{W}) || p(\mathbf{W})] - \text{KLD}[q(\mathbf{A}) || p(\mathbf{A})] \\ \frac{\partial \mathcal{F}}{\partial \xi} &= \frac{\partial}{\partial \xi} \langle \log p(\mathbf{t}|\mathbf{X}, \mathbf{W}) \rangle \end{aligned} \quad (\text{B.71})$$

$$\frac{\partial \mathcal{F}}{\partial \xi} = \sum_{n=1}^N \sum_{m=1}^M \frac{\partial}{\partial \xi_{nm}} \langle \log p(\mathbf{t}_n | \mathbf{x}_n, \mathbf{w}_m) \rangle$$

Substituting the approximation for $p(\mathbf{t}_n | \mathbf{W}, \mathbf{x}_n)$:

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \xi} &= \sum_{n=1}^N \sum_{m=1}^M c_{nm} \left[\frac{1}{1 + e^{\xi_{nm}}} - \frac{1}{2} + 2\xi_{nm}\lambda(\xi_{nm}) - \frac{\partial \lambda(\xi_{nm})}{\partial \xi_{nm}} (\langle \gamma_{nm}^2 \rangle - \xi_{nm}^2) \right] \\ &= \sum_{n=1}^N \sum_{m=1}^M c_{nm} \left[\frac{e^{-\xi_{nm}/2}}{e^{\xi_{nm}/2} + e^{-\xi_{nm}/2}} + \frac{\frac{1}{2}e^{\xi_{nm}/2} - \frac{1}{2}e^{-\xi_{nm}/2}}{e^{\xi_{nm}/2} + e^{-\xi_{nm}/2}} - \frac{1}{2} - \frac{\partial \lambda(\xi_{nm})}{\partial \xi_{nm}} (\langle \gamma_{nm}^2 \rangle - \xi_{nm}^2) \right] \\ &= \sum_{n=1}^N \sum_{m=1}^M c_{nm} \left[\frac{\frac{1}{2}e^{\xi_{nm}/2} + \frac{1}{2}e^{-\xi_{nm}/2}}{e^{\xi_{nm}/2} + e^{-\xi_{nm}/2}} - \frac{1}{2} - \frac{\partial \lambda(\xi_{nm})}{\partial \xi_{nm}} (\langle \gamma_{nm}^2 \rangle - \xi_{nm}^2) \right] \\ &= - \sum_{n=1}^N \sum_{m=1}^M c_{nm} \frac{\partial \lambda(\xi_{nm})}{\partial \xi_{nm}} (\langle \gamma_{nm}^2 \rangle - \xi_{nm}^2) \end{aligned} \quad (\text{B.72})$$

Because the derivative of $\lambda(\xi_n)$ is purely negative, $\mathcal{F}$ is maximized at

$$\xi_{nm}^2 = \langle \gamma_{nm}^2 \rangle = \phi(\mathbf{x}_n)^T \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \phi(\mathbf{x}_n) \quad (\text{B.73})$$

B.3.5 Variational Posterior on α

It was derived in Section that the NFE is maximized by a variational posterior, $q(\alpha)$, that is proportional to the variational expectation of the true log-posterior

with respect to all other model parameters:

$$q(\boldsymbol{\alpha}) \propto \langle \log p(\boldsymbol{\alpha} | -) \rangle \quad (\text{B.74})$$

The true posterior may be calculated from Bayes' theorem:

$$p(\boldsymbol{\alpha} | -) \propto p(\mathbf{W} | \boldsymbol{\alpha}) p(\boldsymbol{\alpha}) \quad (\text{B.75})$$

The variational posterior can be calculated by solving the true posterior as a function of $\boldsymbol{\alpha}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} p(\boldsymbol{\alpha} | -) &\propto \prod_{m=1}^M \prod_{d=1}^D (2\pi)^{-\frac{1}{2}} \alpha_{md}^{\frac{1}{2}} \exp\left(-\frac{\alpha_{md} w_{md}^2}{2}\right) \frac{b_0^{a_0}}{\Gamma(a_0)} \alpha_{md}^{a_0-1} \exp(-b_0 \alpha_{md}) \\ &\propto \prod_{m=1}^M \prod_{d=1}^D \alpha_{md}^{\frac{1}{2}} \exp\left(-\frac{\alpha_{md} w_{md}^2}{2}\right) \alpha_{md}^{a_0-1} \exp(-b_0 \alpha_{md}) \\ &\propto \prod_{m=1}^M \prod_{d=1}^D \alpha_{md}^{a_0 + \frac{1}{2} - 1} \exp\left(-\alpha_{md} \left[b_0 + \frac{1}{2} w_{md}^2\right]\right) \end{aligned} \quad (\text{B.76})$$

Therefore, the α 's are Gamma distributed:

$$p(\boldsymbol{\alpha} | -) = \prod_{m=1}^M \prod_{d=1}^D \text{Gamma}(\alpha_{md} | a_{md}, b_{md}), \quad (\text{B.77})$$

where

$$a_{md} = a_0 + \frac{1}{2} \quad (\text{B.78})$$

$$b_{md} = b_0 + \frac{1}{2} w_{md}^2, \quad (\text{B.79})$$

Useful moments in VB updates for other model parameters:

$$\langle \alpha_{md} \rangle = \frac{a_{md}}{b_{md}} \quad (\text{B.80})$$

$$\langle \mathbf{A}_m \rangle = \text{diag} \langle \boldsymbol{\alpha}_m \rangle \quad (\text{B.81})$$

B.3.6 Treatment of $\mathbf{c}$

The latent variable c defines which component of the RVM mixture is used. For a fully-conjugate model, $c_n \sim \text{Multinomial}(\boldsymbol{\rho}_n)$ and $\boldsymbol{\rho}_n \sim \text{Dir}(\lambda_0)$. However, this approach is not used in this work and therefore not discussed here.

In this work, the mixture of RVMs is used in a context-dependent learning framework. For cases in which supervised context modeling is used, the values c_{nm} are determined by the known context labels. Therefore, $c_{nm} = 1$ for observations collected in the m th labeled context. If unsupervised context modeling is used, c can be treated multinomial distributed and its density is determined *a posteriori* from context identification. Therefore $\langle c_{nm} \rangle = p(c_{nm} | \mathbf{x}_n^{(C)})$ regardless of the context model used to obtain these posterior probabilities. See Chapter 4 for more information. Discriminative context-dependent learning is a unique case that is described in Chapter 5 and the VB derivation can be found in Appendix E.

B.3.7 Negative Free Energy

The NFE can be expressed as the difference between the expected log-likelihood and the Kullback-Leibler divergence (KLD) between the variational posteriors and the

priors:

$$\begin{aligned}
\mathcal{F} &= \langle \log p(\mathbf{t}|\mathbf{w}, \mathbf{X}) \rangle - \mathbb{KLD} [q(\mathbf{W}) q(\mathbf{A}) || p(\mathbf{W}|\mathbf{A}) p(\mathbf{A})] \\
&= \langle \log p(\mathbf{t}|\mathbf{W}, \mathbf{X}) \rangle - \sum_{m=1}^M \mathbb{KLD} [q(\mathbf{w}_m) || p(\mathbf{w}_m|\mathbf{A}_m)] - \sum_{m=1}^M \sum_{d=1}^D \mathbb{KLD} [q(\alpha_{md}) || p(\alpha_{md})] \\
&= \sum_{n=1}^N \sum_{m=1}^M \langle c_{nm} \rangle \left[\log \sigma(\xi_{nm}) + \frac{1}{2} (\langle \gamma_n \rangle - \xi_{nm}) - \lambda(\xi_{nm}) (\langle \gamma_{nm}^2 \rangle - \xi_{nm}^2) \right] \\
&\quad \sum_{m=1}^M \mathbb{KLD} [q(\mathbf{w}_m) || p(\mathbf{w}_m|\mathbf{A}_m)] - \sum_{m=1}^M \sum_{d=1}^D \mathbb{KLD} [q(\alpha_{md}) || p(\alpha_{md})] \\
&= \sum_{n=1}^N \langle c_{nm} \rangle \left[\log \sigma(\xi_n) + \frac{1}{2} \left[(2t_n - 1) \phi(\mathbf{x}_n)^T \langle \mathbf{w}_m \rangle - \xi_{nm} \right] \right. \\
&\quad \left. - \lambda(\xi_{nm}) \left(\phi(\mathbf{x}_n)^T \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \phi(\mathbf{x}_n) - \xi_{nm}^2 \right) \right] - \mathbb{KLD} [q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})] \\
&\quad - \sum_{d=1}^D \mathbb{KLD} [q(\alpha_d) || p(\alpha_d)]
\end{aligned} \tag{B.82}$$

where $\mathbb{KLD} [q(\mathbf{w}) || p(\mathbf{w}|\mathbf{A})]$ is a KLD between two Gaussian distributions, and $\mathbb{KLD} [q(\alpha_d) || p(\alpha_d)]$ is a KLD between two Gamma distributions.

Appendix C

Dirichlet Process Gaussian Mixture Models

One of the generative context models presented in Chapter 4 is the Dirichlet process Gaussian mixture model (DPGMM). The DPGMM can be useful when performing unsupervised clustering in scenarios where the number of clusters is uncertain. This appendix presents the DPGMM of Blei and Jordan [67], as well as derivations for all variational Bayesian (VB) update equations and the negative free energy (NFE).

C.1 Generative Model and Variable Definitions

$$(\mathbf{x}|c_{nm} = 1) \sim \mathcal{N}_D(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m^{-1}) \quad (\text{C.1})$$

$\mathbf{x}$ is $D \times 1$ feature vector

$\mathbf{c}_n$ is $M \times 1$ binary-coded latent variable

$n = 1, 2, \dots, N$ is data index

$m = 1, 2, \dots, T$ is mixture component index (T is arbitrarily large)

$d = 1, 2, \dots, D$ is dimension index

C.2 Priors

$$(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) \sim \mathcal{N}_D(\boldsymbol{\mu}_m | \boldsymbol{\rho}_0, u_0^{-1} \boldsymbol{\Lambda}_m^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_m | \mathbf{B}_0, \nu_0) \quad (\text{C.2})$$

$$\mathbf{c}_n \sim \text{Multinomial}(\boldsymbol{\pi}) \quad (\text{C.3})$$

$$\pi_m = v_m \prod_{l < m} (1 - v_l) \quad (\text{C.4})$$

$$v_m \sim \text{Beta}(1, \alpha) \quad (\text{C.5})$$

$$\alpha \sim \text{Gamma}(\tau_{10}, \tau_{20}) \quad (\text{C.6})$$

C.3 Model likelihood

The joint likelihood of data given all model parameters is given by

$$p(\mathbf{X}|-) = \prod_{n=1}^N \prod_{m=1}^T \mathcal{N}_D(\mathbf{x}_n | \boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m^{-1})^{c_{nm}} \quad (\text{C.7})$$

$$\log p(\mathbf{X}|-) = -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[D \log 2\pi + \log |\boldsymbol{\Lambda}_m^{-1}| + (\mathbf{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Lambda}_m (\mathbf{x} - \boldsymbol{\mu}_m) \right] \quad (\text{C.8})$$

C.4 Variational Posterior on $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\boldsymbol{\mu}, \boldsymbol{\Lambda}) \propto \langle \log p(\boldsymbol{\mu}, \boldsymbol{\Lambda}|-) \rangle \quad (\text{C.9})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(\boldsymbol{\mu}, \boldsymbol{\Lambda}|-) = \log p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\Lambda}, -) + \log p(\boldsymbol{\mu} | \boldsymbol{\Lambda}) + \log p(\boldsymbol{\Lambda}) - K, \quad (\text{C.10})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned}
\log p(\boldsymbol{\mu}, \boldsymbol{\Lambda} | -) &= \\
&= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[\cancel{D \log 2\pi} + \log |\boldsymbol{\Lambda}_m^{-1}| + (\mathbf{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Lambda}_m (\mathbf{x} - \boldsymbol{\mu}_m) \right] \\
&\quad - \frac{1}{2} \sum_{m=1}^T \left[\cancel{D \log 2\pi} - \cancel{D \log u_0} + \log |\boldsymbol{\Lambda}_m^{-1}| + (\boldsymbol{\mu}_m - \boldsymbol{\rho}_0)^T u_0 \boldsymbol{\Lambda}_m (\boldsymbol{\mu}_m - \boldsymbol{\rho}_0) \right] \\
&\quad + \sum_{m=1}^T \left[\frac{\nu_0 - D - 1}{2} \log |\boldsymbol{\Lambda}_m| - \cancel{\frac{\nu_0 D}{2} \log 2} - \cancel{\frac{\nu_0}{2} \log |\mathbf{B}_0|} - \cancel{\Gamma_D\left(\frac{\nu_0}{2}\right)} \right. \\
&\quad \left. - \frac{1}{2} \text{Tr}(\mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) \right] - K \\
&= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} [\boldsymbol{\mu}_m^T \boldsymbol{\Lambda}_m \boldsymbol{\mu}_m - 2\boldsymbol{\mu}_m^T \boldsymbol{\Lambda}_m \mathbf{x} + \mathbf{x}^T \boldsymbol{\Lambda}_m \mathbf{x}] \\
&\quad - \frac{1}{2} \sum_{m=1}^T [\boldsymbol{\mu}_m^T u_0 \boldsymbol{\Lambda}_m \boldsymbol{\mu}_m - 2\boldsymbol{\mu}_m^T u_0 \boldsymbol{\Lambda}_m \boldsymbol{\rho}_0 + \boldsymbol{\rho}_0^T u_0 \boldsymbol{\Lambda}_m \boldsymbol{\rho}_0] \\
&\quad - \frac{1}{2} \sum_{m=1}^T \left[\text{Tr}(\mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D - 1 \right) \log |\boldsymbol{\Lambda}_m| \right] - K \\
&= -\frac{1}{2} \sum_{m=1}^T \left[-2\boldsymbol{\mu}_m^T \boldsymbol{\Lambda}_m \left(u_0 \boldsymbol{\rho}_0 + \sum_{n=1}^N c_{nm} \mathbf{x}_n \right) + \boldsymbol{\mu}_m^T \left(\sum_{n=1}^N c_{nm} + u_0 \right) \boldsymbol{\Lambda}_m \boldsymbol{\mu}_m \right. \\
&\quad + \boldsymbol{\rho}_m^T \left(\sum_{n=1}^N c_{nm} + u_0 \right) \boldsymbol{\Lambda}_m \boldsymbol{\rho}_m - \boldsymbol{\rho}_m^T \left(\sum_{n=1}^N c_{nm} + u_0 \right) \boldsymbol{\Lambda}_m \boldsymbol{\rho}_m + \sum_{n=1}^N c_{nm} \mathbf{x}^T \boldsymbol{\Lambda}_m \mathbf{x} \\
&\quad \left. + \boldsymbol{\rho}_0^T u_0 \boldsymbol{\Lambda}_m \boldsymbol{\rho}_0 + \text{Tr}(\mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D - 1 \right) \log |\boldsymbol{\Lambda}_m| \right] - K
\end{aligned} \tag{C.11}$$

Using the identity $\mathbf{a}^T \mathbf{B} \mathbf{a} = \text{Tr}(\mathbf{a} \mathbf{a}^T \mathbf{B})$ yields:

$$\begin{aligned}
& \log p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) \\
&= -\frac{1}{2} \sum_{m=1}^T \left[(\boldsymbol{\mu}_m - \boldsymbol{\rho}_m)^T u_m \boldsymbol{\Lambda}_m (\boldsymbol{\mu}_m - \boldsymbol{\rho}_m) - \text{Tr}(u_m \mathbf{S}_m \boldsymbol{\Lambda}_m) + \text{Tr}(\mathbf{C}_m \boldsymbol{\Lambda}_m) \right. \\
&\quad \left. + \text{Tr}(u_0 \mathbf{S}_0 \boldsymbol{\Lambda}_0) + \text{Tr}(\mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D - 1 \right) \log |\boldsymbol{\Lambda}_m| \right] - K
\end{aligned} \tag{C.12}$$

Where

$$u_m = u_0 + \sum_{n=1}^N c_{nm} \tag{C.13}$$

$$\boldsymbol{\rho}_m = \frac{u_0 \boldsymbol{\rho}_0 + \sum_{n=1}^N c_{nm} \mathbf{x}}{u_m} \tag{C.14}$$

$$\mathbf{S}_m = \boldsymbol{\rho}_m \boldsymbol{\rho}_m^T \tag{C.15}$$

$$\mathbf{C}_m = \sum_{n=1}^N c_{nm} \mathbf{x} \mathbf{x}^T \tag{C.16}$$

$$\mathbf{S}_0 = \boldsymbol{\rho}_0 \boldsymbol{\rho}_0^T \tag{C.17}$$

Consolidating the last four terms using the identity $\text{Tr}(\mathbf{A}) + \text{Tr}(\mathbf{B}) = \text{Tr}(\mathbf{AB})$ yields:

$$\begin{aligned}
& \log p(\boldsymbol{\mu}, \boldsymbol{\Lambda} | -) \\
&= -\frac{1}{2} \sum_{m=1}^T \left[(\boldsymbol{\mu}_m - \boldsymbol{\rho}_m)^T u_m \boldsymbol{\Lambda}_m (\boldsymbol{\mu}_m - \boldsymbol{\rho}_m) \right. \\
&\quad \left. + \text{Tr}(\mathbf{C}_m \boldsymbol{\Lambda}_m - u_m \mathbf{S}_m \boldsymbol{\Lambda}_m + u_0 \mathbf{S}_0 \boldsymbol{\Lambda}_0 + \mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D - 1 \right) \right. \\
&\quad \left. \log |\boldsymbol{\Lambda}_m| \right] - K \\
&= -\frac{1}{2} \sum_{m=1}^T \left[(\boldsymbol{\mu}_m - \boldsymbol{\rho}_m)^T u_m \boldsymbol{\Lambda}_m (\boldsymbol{\mu}_m - \boldsymbol{\rho}_m) \right. \\
&\quad \left. + \text{Tr}[(\mathbf{C}_m - u_m \mathbf{S}_m + u_0 \mathbf{S}_0 + \mathbf{B}_0^{-1}) \boldsymbol{\Lambda}_m] \right. \\
&\quad \left. - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D - 1 \right) \log |\boldsymbol{\Lambda}_m| \right] - K
\end{aligned} \tag{C.18}$$

Consolidating terms reveals that $\boldsymbol{\mu}, \boldsymbol{\Lambda}$ are Normal-Wishart:

$$\log p(\boldsymbol{\mu}, \boldsymbol{\Lambda} | -) = \sum_{m=1}^T \log [\mathcal{N}(\boldsymbol{\mu}_m | \boldsymbol{\rho}_m, u_m^{-1} \boldsymbol{\Lambda}_m^{-1}) \mathcal{W}(\boldsymbol{\Lambda} | \nu_m, \mathbf{B}_m)] \tag{C.19}$$

where $\boldsymbol{\rho}_m$ and u_m are defined above, and

$$\nu_m = \nu_0 + \sum_{n=1}^N c_{nm} \tag{C.20}$$

$$\mathbf{B}_m = (\mathbf{C}_m - u_m \mathbf{S}_m + u_0 \mathbf{S}_0 + \mathbf{B}_0^{-1})^{-1} \tag{C.21}$$

Useful moments in VB updates for other model parameters:

$$\langle \boldsymbol{\mu}_m \rangle = \boldsymbol{\rho}_m \tag{C.22}$$

$$\langle \mathbf{\Lambda}_m \rangle = \nu_m \mathbf{B}_m \quad (\text{C.23})$$

$$\langle \boldsymbol{\mu}_m \boldsymbol{\mu}_m^T \rangle = \mathbf{S}_m + u_m^{-1} \nu_m^{-1} \mathbf{B}_m^{-1} \quad (\text{C.24})$$

$$\langle \log |\mathbf{\Lambda}_m| \rangle = \sum_{d=1}^D \psi \left(\frac{\nu_m - d + 1}{2} \right) + D \log 2 + \log |\mathbf{B}_m| \quad (\text{C.25})$$

$$\langle (\mathbf{x}_n - \boldsymbol{\mu}_m)^T \mathbf{\Lambda}_m (\mathbf{x}_n - \boldsymbol{\mu}_m) \rangle = (\mathbf{x}_n - \boldsymbol{\rho}_m)^T \nu_m \mathbf{B}_m (\mathbf{x}_n - \boldsymbol{\rho}_m) + \frac{D}{u_m} \quad (\text{C.26})$$

C.5 Variational Posterior on $\mathbf{v}$

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{v})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{v}) \propto \langle \log p(v_m | -) \rangle \quad (\text{C.27})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(v_m | -) = \log p(\mathbf{C} | v_m) + \log p(v_m) - K, \quad (\text{C.28})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{v}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} \log p(v_m | -) &= \sum_{n=1}^N c_{nm} \log v_m + \sum_{n=1}^N \sum_{l>m} z_{nl} \log(1 - v_m) + (\alpha - 1) \log(1 - v_m) - K \\ &= \sum_{n=1}^N c_{nm} \log v_m + \left(\alpha + \sum_{n=1}^N \sum_{l>m} c_{nl} - 1 \right) \log(1 - v_m) - K \end{aligned} \quad (\text{C.29})$$

Therefore,

$$p(v_m | -) = \text{Beta}(\gamma_{m1}, \gamma_{m2}) \quad (\text{C.30})$$

where

$$\gamma_{m1} = 1 + \sum_{n=1}^N c_{nm} \quad (\text{C.31})$$

$$\gamma_{m2} = \alpha + \sum_{n=1}^N \sum_{l>m} c_{nl} \quad (\text{C.32})$$

Useful moments in VB updates for other model parameters:

$$\langle \ln v_m \rangle = \psi(\gamma_{m1}) - \psi(\gamma_{m1} + \gamma_{m2}) \quad (\text{C.33})$$

$$\langle \ln(1 - v_m) \rangle = \psi(\gamma_{m2}) - \psi(\gamma_{m1} + \gamma_{m2}) \quad (\text{C.34})$$

C.6 Variational Posterior on α

It was derived in Section that the NFE is maximized by a variational posterior, $q(\alpha)$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\alpha) \propto \langle \log p(\alpha | -) \rangle \quad (\text{C.35})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(\alpha | -) = \log p(\mathbf{v} | \alpha) + \log p(\alpha) - K, \quad (\text{C.36})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of α , then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} \log p(\alpha | -) &= \sum_{m=1}^{T-1} (\alpha - \mathcal{I}) \log(1 - v_m) - K \\ &\quad + \log \left[\frac{(M-1)}{\alpha} \exp(-\tau_{20}\alpha + \tau_{10} \log \alpha - \log \Gamma(\tau_{10}) + \tau_{10} \log \tau_{20}) \right] \\ &= \left[-\tau_{20} + \sum_{m=1}^{T-1} \log(1 - v_m) \right] \alpha + \sum_{\ell=1}^{N-1} (1 - v_m) + \log \alpha (\tau_{10} + T - 1) - K \end{aligned} \quad (\text{C.37})$$

Therefore,

$$p(\alpha|-) = \text{Gamma}(\tau_1, \tau_2) \quad (\text{C.38})$$

$$\tau_1 = \tau_{10} + T - 1 \quad (\text{C.39})$$

$$\tau_2 = \tau_{20} - \sum_{m=1}^{T-1} \log(1 - v_m) \quad (\text{C.40})$$

Useful moments in VB updates for other model parameters:

$$\langle \alpha \rangle = \frac{\tau_1}{\tau_2} \quad (\text{C.41})$$

C.7 Variational Posterior on $\mathbf{C}$

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{C})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{C}) \propto \langle \log p(\mathbf{C}|-) \rangle \quad (\text{C.42})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(\mathbf{C}|-) \propto \log p(\mathbf{X}|\mathbf{C}, -) + \log p(\mathbf{C}) - K, \quad (\text{C.43})$$

where K denotes a normalizing constant. The posterior will also be multinomial with parameters (responsibilities) Φ :

$$\begin{aligned} \log p(c_{nm} = 1|-) \\ &= -\cancel{\frac{D}{2} \log 2\pi} - \frac{1}{2} \log |\mathbf{\Lambda}_m^{-1}| - \frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_m)^T \mathbf{\Lambda}_m (\mathbf{x}_n - \boldsymbol{\mu}_m) + \log v_m + \sum_{l < m} \log(1 - v_l) - K \\ &= \frac{1}{2} \log |\mathbf{\Lambda}_m| - \frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_m)^T \mathbf{\Lambda}_m (\mathbf{x}_n - \boldsymbol{\mu}_m) + \log v_m + \sum_{l < m} \log(1 - v_l) - K \end{aligned} \quad (\text{C.44})$$

Therefore,

$$p(\mathbf{c}_n|-) \propto \text{Multinomial}(\boldsymbol{\phi}_n) \quad (\text{C.45})$$

Useful moments in VB updates for other model parameters:

$$\langle c_{nm} \rangle = \phi_{nm} \quad (\text{C.46})$$

C.8 Negative Free Energy

The NFE can be expressed as the difference between the expected log-likelihood and the Kullback-Leibler divergence (KLD) between the variational posteriors and the priors:

$$\begin{aligned} \mathcal{F} &= \langle \log p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{C}) \rangle - \mathbb{KLD} [q(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{C}, \mathbf{v}, \alpha) || p(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{C}, \mathbf{v}, \alpha)] \\ &= \langle \log p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{C}) \rangle - \sum_{n=1}^N \mathbb{KLD} [q(\mathbf{c}_n) || p(\mathbf{c}_n|\mathbf{v})] \\ &\quad - \sum_{m=1}^T \mathbb{KLD} [q(\boldsymbol{\mu}_m|\boldsymbol{\Lambda}_m)q(\boldsymbol{\Lambda}_m) || p(\boldsymbol{\mu}_m|\boldsymbol{\Lambda}_m)p(\boldsymbol{\Lambda}_m)] \\ &\quad - \sum_{m=1}^T \mathbb{KLD} [q(v_m) || p(v_m)] - \mathbb{KLD} [q(\alpha) || p(\alpha)] \\ &= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T \langle c_{nm} \rangle \left[D \log 2\pi - \langle \log |\boldsymbol{\Lambda}_m| \rangle + \langle (\mathbf{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Lambda}_m (\mathbf{x} - \boldsymbol{\mu}_m) \rangle \right] \\ &\quad - \sum_{n=1}^N \mathbb{KLD} [q(\mathbf{c}_n) || p(\mathbf{c}_n|\mathbf{v})] - \sum_{m=1}^T \mathbb{KLD} [q(\boldsymbol{\mu}_m|\boldsymbol{\Lambda}_m)q(\boldsymbol{\Lambda}_m) || p(\boldsymbol{\mu}_m|\boldsymbol{\Lambda}_m)p(\boldsymbol{\Lambda}_m)] \\ &\quad - \sum_{m=1}^T \mathbb{KLD} [q(v_m) || p(v_m)] - \mathbb{KLD} [q(\alpha) || p(\alpha)], \end{aligned} \quad (\text{C.47})$$

where $\mathbb{KLD} [q(\mathbf{c}_n) || p(\mathbf{c}_n|\mathbf{v})]$ is a KLD between two multinomial distributions,

$\mathbb{KLD} [q(\boldsymbol{\mu}_m|\boldsymbol{\Lambda}_m)q(\boldsymbol{\Lambda}_m) || p(\boldsymbol{\mu}_m|\boldsymbol{\Lambda}_m)p(\boldsymbol{\Lambda}_m)]$ is a KLD between two Normal-Wishart

distributions, $\mathbb{KLD}[q(v_m)||p(v_m)]$ is a KLD between two Beta distributions, and $\mathbb{KLD}[q(\alpha)||p(\alpha)]$ is a KLD between two Gamma distributions.

Appendix D

Dirichlet Process Mixture of Factor Analyzers

The Dirichlet process mixture of factor analyzers (DPMFA) model is used in this work for generative context learning. Like the Dirichlet process Gaussian mixture model (DPGMM), it is an unsupervised clustering technique that facilitates learning the number of clusters. Additionally, the use of the factor analysis model allows for a local latent, lower-dimensional structure to be learned for each mixture component. This is accomplished by selecting features from a shared loading matrix that is shared between all mixture components. This appendix presents the DPMFA, adapted from Ghahramani and Beal [68] and Wang et al. [94], including derivations for all variational Bayesian (VB) update equations and the negative free energy (NFE).

D.1 Model and Variable Definitions

$$(\mathbf{x}_n | c_{nm} = 1) \sim \mathcal{N}_D(\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m, \text{diag}(\boldsymbol{\psi}_m)^{-1}) \quad (\text{D.1})$$

$n = 1, 2, \dots, N$ is data index

$m = 1, 2, \dots, T$ is mixture component index (T is arbitrarily large)

$d = 1, 2, \dots, D$ is data dimension index

$k = 1, 2, \dots, K$ is factor index

$\mathbf{x}_n$ is $D \times 1$ data

$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_K]$ is $D \times K$ factor matrix

$\mathbf{z}_m = [z_{m1}, z_{m2}, \dots, z_{MK}]^T$ is $K \times 1$ binary-coded selection vector

$\mathbf{s}_n = [s_{1n}, s_{2n}, \dots, s_{KN}]^T$ is $K \times 1$ score vector

$\boldsymbol{\mu}_m = [\mu_{m1}, \mu_{m2}, \dots, \mu_{mD}]^T$ is $D \times 1$ component mean vector

$\boldsymbol{\psi}_m = [\psi_{m1}, \psi_{m2}, \dots, \psi_{MD}]^T$ is $D \times 1$ component precisions

c_{nm} is a binary-coded latent variable

D.2 Priors

$$p(A_{dk}|\gamma_{dk}) \sim \mathcal{N}(A_{dk}|0, \gamma_{dk}^{-1}) \quad (\text{D.2})$$

$$p(\mathbf{s}_n|\delta) \sim \mathcal{N}_K(\mathbf{s}_n|0, \delta^{-1}\mathbf{I}) \quad (\text{D.3})$$

$$p(z_{mk}|\eta_{mk}) \sim \text{Bernoulli}(z_{mk}|\eta_{mk}) \quad (\text{D.4})$$

$$p(\eta_{mk}) \sim \text{Beta}(\eta_{mk}|a_0/K, b_0(K-1)/K) \quad (\text{D.5})$$

$$p(\gamma_{dk}) \sim \text{Gamma}(\gamma_{dk}|e_0, f_0) \quad (\text{D.6})$$

$$p(\boldsymbol{\mu}_m|\boldsymbol{\psi}_m) \sim \mathcal{N}_D(\boldsymbol{\mu}_m|\boldsymbol{\rho}_0, u_0^{-1}\text{diag}(\boldsymbol{\psi}_m)^{-1}) \quad (\text{D.7})$$

$$p(\psi_{md}) \sim \text{Gamma}(\psi_{md}|g_0, h_0) \quad (\text{D.8})$$

$$p(\mathbf{c}_n) \sim \text{Multinomial}(\mathbf{c}_n|\boldsymbol{\pi}) \quad (\text{D.9})$$

$$\pi_m(\mathbf{v}) = v_m \prod_{l < m} (1 - v_l) \quad (\text{D.10})$$

$$p(v_m|\alpha) \sim \text{Beta}(v_m|1, \alpha) \quad (\text{D.11})$$

$$p(\alpha) \sim \text{Gamma}(\alpha|\tau_{10}, \tau_{20}) \quad (\text{D.12})$$

$$p(\delta) \sim \text{Gamma}(\delta|\delta_{10}, \delta_{20}) \quad (\text{D.13})$$

D.3 Model Likelihood

The joint likelihood of data given all model parameters is given by

$$p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \mathbf{Z}, \Psi) = \prod_{i=1}^N \prod_{m=1}^T \mathcal{N}_D(\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m, \text{diag}(\boldsymbol{\psi}_m)^{-1})^{c_{nm}} \quad (\text{D.14})$$

Use log-likelihood for analysis

$$\begin{aligned} & \log p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \mathbf{Z}, \Psi) \\ &= \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[-\frac{1}{2} \left(D \log 2\pi + \log |\text{diag}(\boldsymbol{\psi})^{-1}| \right. \right. \\ & \quad \left. \left. + [\mathbf{x}_n - (\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m)]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{x}_n - (\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m)] \right) \right] \\ &= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left(\mathbf{x}_n^T \text{diag}(\boldsymbol{\psi}_m) \mathbf{x}_n \right. \\ & \quad - [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m]^T \text{diag}(\boldsymbol{\psi}_m) \mathbf{x}_n - \mathbf{x}_n^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] \\ & \quad \left. + [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] \right. \\ & \quad \left. + \log |\text{diag}(\boldsymbol{\psi}_m)^{-1}| + D \log 2\pi \right) \\ &= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left(\mathbf{x}_n^T \text{diag}(\boldsymbol{\psi}_m) \mathbf{x}_n - 2\mathbf{x}_n^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] \right. \\ & \quad \left. + [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] \right. \\ & \quad \left. + \log |\text{diag}(\boldsymbol{\psi}_m)^{-1}| + D \log 2\pi \right) \end{aligned} \quad (\text{D.15})$$

Converting some of the terms to sums yields

$$\begin{aligned}
& \mathbf{x}_n^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] \\
&= \begin{bmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{nD} \end{bmatrix}^T \begin{bmatrix} \psi_{m1} & & & 0 \\ & \psi_{m2} & & \\ 0 & & \ddots & \\ & & & \psi_{mP} \end{bmatrix} \\
&\quad \left(\begin{bmatrix} A_{11} & A_{12} & \dots & A_{1K} \\ A_{21} & A_{22} & \dots & A_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ A_{P1} & A_{P2} & \dots & A_{DK} \end{bmatrix} \begin{bmatrix} z_{m1} & & & 0 \\ & z_{m2} & & \\ 0 & & \ddots & \\ & & & z_{mk} \end{bmatrix} \begin{bmatrix} s_{1n} \\ s_{2n} \\ \vdots \\ s_{Kn} \end{bmatrix} + \begin{bmatrix} \mu_{m1} \\ \mu_{m2} \\ \vdots \\ \mu_{mD} \end{bmatrix} \right) \\
&= \begin{bmatrix} x_{n1}\psi_{m1} \\ x_{n2}\psi_{m2} \\ \vdots \\ x_{nD}\psi_{mP} \end{bmatrix}^T \left(\begin{bmatrix} A_{11}z_{m1} & A_{12}z_{m2} & \dots & A_{1K}z_{mk} \\ A_{21}z_{m1} & A_{22}z_{m2} & \dots & A_{2K}z_{mk} \\ \vdots & \vdots & \ddots & \vdots \\ A_{P1}z_{m1} & A_{P2}z_{m2} & \dots & A_{DK}z_{mk} \end{bmatrix} \begin{bmatrix} s_{1i} \\ s_{2i} \\ \vdots \\ s_{Ki} \end{bmatrix} + \begin{bmatrix} \mu_{m1} \\ \mu_{m2} \\ \vdots \\ \mu_{mD} \end{bmatrix} \right) \\
&= \begin{bmatrix} x_{n1}\psi_{m1} \\ x_{n2}\psi_{m2} \\ \vdots \\ x_{nD}\psi_{mP} \end{bmatrix}^T \begin{bmatrix} \mu_{m1} + \sum_{k=1}^K A_{1k}z_{mk}s_{kn} \\ \mu_{m2} + \sum_{k=1}^K A_{2k}z_{mk}s_{kn} \\ \vdots \\ \mu_{mD} + \sum_{k=1}^K A_{DK}z_{mk}s_{kn} \end{bmatrix} \\
&= \sum_{d=1}^D x_{nd}\psi_{md}\mu_{md} + \sum_{d=1}^D \sum_{k=1}^K x_{nd}\psi_{md}A_{dk}z_{mk}s_{kn} \tag{D.16}
\end{aligned}$$

$$\begin{aligned}
& [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_h]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_h] \\
&= \begin{bmatrix} \mu_{m1} + \sum_{k=1}^K A_{1k} z_{mk} s_{kn} \\ \mu_{m2} + \sum_{k=1}^K A_{2k} z_{mk} s_{kn} \\ \vdots \\ \mu_{mD} + \sum_{k=1}^K A_{DK} z_{mk} s_{kn} \end{bmatrix}^T \begin{bmatrix} \psi_{m1} & & & 0 \\ & \psi_{m2} & & \\ & 0 & \ddots & \\ & & & \psi_{mP} \end{bmatrix} \begin{bmatrix} \mu_{m1} + \sum_{k=1}^K A_{1k} z_{mk} s_{kn} \\ \mu_{m2} + \sum_{k=1}^K A_{2k} z_{mk} s_{kn} \\ \vdots \\ \mu_{mD} + \sum_{k=1}^K A_{DK} z_{mk} s_{kn} \end{bmatrix} \\
&= \sum_{d=1}^D \psi_{md} \left(\mu_{md} + \sum_{k=1}^K A_{dk} z_{mk} s_{kn} \right)^2 \\
&= \sum_{d=1}^D \psi_{md} \mu_{md}^2 + 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} \mu_{md} A_{dk} z_{mk} s_{kn} + \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk}^2 z_{mk}^2 s_{kn}^2 \\
&+ 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk} z_{mk} s_{kn} \sum_{l < k} A_{dl} z_{ml} s_{ln} \tag{D.17}
\end{aligned}$$

where

$$\log |\text{diag}(\boldsymbol{\psi}_m)^{-1}| = - \sum_{j=1}^P \log \psi_{mj} \tag{D.18}$$

$$\mathbf{x}_n^T \text{diag}(\boldsymbol{\psi}_m) \mathbf{x}_n = \sum_{d=1}^D x_{nd}^2 \psi_{md} \tag{D.19}$$

Rephrasing the log-likelihood using these new quantities yields:

$$\log p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \mathbf{Z}, \Psi)$$

$$\begin{aligned}
&= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[\sum_{d=1}^D x_{nd}^2 \psi_{md} - 2 \sum_{d=1}^D x_{nd} \psi_{md} \mu_{md} - 2 \sum_{d=1}^D \sum_{k=1}^K x_{nd} \psi_{md} A_{dk} z_{mk} s_{kn} \right. \\
&\quad + \sum_{d=1}^D \psi_{md} \mu_{md}^2 + 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} \mu_{md} A_{dk} z_{mk} s_{kn} + \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk}^2 z_{mk}^2 s_{kn}^2 \\
&\quad \left. + 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk} z_{mk} s_{kn} \sum_{l < k} A_{dl} z_{ml} s_{ln} - \sum_{j=1}^P \log \psi_{md} + D \log 2\pi \right] \\
&= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[\sum_{d=1}^D \psi_{md} (x_{nd}^2 - 2x_{nd} \mu_{md} + \mu_{md}^2) + \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk}^2 z_{mk}^2 s_{kn}^2 \right. \\
&\quad - 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk} z_{mk} s_{kn} \left(x_{nd} - \sum_{l < k} A_{dl} z_{ml} s_{ln} - \mu_{md} \right) - \sum_{j=1}^P \log \psi_{md} + D \log 2\pi \left. \right] \\
&= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[\sum_{d=1}^D \psi_{md} (x_{nd}^2 - 2x_{nd} \mu_{md} + \mu_{md}^2) + \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk}^2 z_{mk}^2 s_{kn}^2 \right. \\
&\quad \left. - 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) - \sum_{j=1}^P \log \psi_{md} + D \log 2\pi \right]
\end{aligned} \tag{D.20}$$

where $x_{ndm}^{-k} = x_{nd} - \sum_{l < k}^K A_{dl} z_{ml} s_{ln}$.

D.4 Variational Posterior on A

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\mathbf{A})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{A}) \propto \langle \log p(\mathbf{A}|\mathbf{X}, -) \rangle \tag{D.21}$$

The true log-posterior may be calculated from Bayes' theorem

$$\log p(\mathbf{A}|\mathbf{X}, -) = \log p(\mathbf{X}|\mathbf{A}, -) + \log p(\mathbf{A}) - E, \quad (\text{D.22})$$

where E denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{A}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} \log p(\mathbf{A}|\mathbf{X}, -) &= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[\sum_{d=1}^D \psi_{md} (x_{nd}^2 - 2x_{nd}\mu_{md} + \mu_{md}^2) + \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk}^2 z_{mk}^2 s_{kn}^2 \right. \\ &\quad \left. - 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) - \sum_{j=1}^P \log \psi_{md} + D \log 2\pi \right] \\ &\quad - \frac{1}{2} \sum_{d=1}^D \sum_{k=1}^K A_{dk}^2 \gamma_{dk} - \frac{1}{2} \sum_{d=1}^D \sum_{k=1}^K \gamma_{dk} - \frac{DK}{2} \log 2\pi - E \\ &= \sum_{d=1}^D \sum_{k=1}^K -\frac{1}{2} \left[-2A_{dk} \sum_{m=1}^T \psi_{md} z_{mk} \sum_{n=1}^N c_{nm} s_{kn} (x_{ndm}^{-k} - \mu_{md}) \right. \\ &\quad \left. + A_{dk}^2 \left(\gamma_{dk} + \sum_{m=1}^T \psi_{md} z_{mk}^2 \sum_{n=1}^N c_{nm} s_{kn}^2 \right) \right] - E. \end{aligned} \quad (\text{D.23})$$

Completing the square reveals that A_{dk} is Gaussian:

$$\log p(\mathbf{A}|\mathbf{X}, -) = \sum_{d=1}^D \sum_{k=1}^K \log \mathcal{N}(\omega_{dk}, \sigma_{dk}), \quad (\text{D.24})$$

where

$$\sigma_{dk} = \left(\gamma_{dk} + \sum_{n=1}^N \sum_{m=1}^T c_{nm} \psi_{md} z_{mk}^2 s_{kn}^2 \right)^{-1} \quad (\text{D.25})$$

$$\omega_{dk} = \sigma_{dk} \left[\sum_{n=1}^N \sum_{m=1}^T c_{nm} \psi_{md} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) \right] \quad (\text{D.26})$$

Useful moments in VB updates for other model parameters:

$$\langle A_{dk} \rangle = \omega_{dk} \quad (\text{D.27})$$

$$\langle A_{dk}^2 \rangle = \omega_{dk}^2 + \sigma_{dk} \quad (\text{D.28})$$

$$\langle A_{dk} A_{dl} \rangle = \langle A_{dk} \rangle \langle A_{dl} \rangle \quad (\text{D.29})$$

D.5 Variational posterior on $\mathbf{S}$

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\mathbf{S})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{S}) \propto \langle \log p(\mathbf{S} | \mathbf{X}, -) \rangle \quad (\text{D.30})$$

The true log-posterior may be calculated from Bayes' theorem

$$\log p(\mathbf{s}_n | \mathbf{x}_n, -) = \log p(\mathbf{x}_n | \mathbf{s}_n, -) + \log p(\mathbf{s}_n) - E, \quad (\text{D.31})$$

where E denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{s}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned}
& \log p(\mathbf{s}_n | \mathbf{x}_n, -) \\
&= -\frac{1}{2} \sum_{m=1}^T c_{nm} \left(\cancel{\mathbf{x}_n^T \text{diag}(\boldsymbol{\psi}_m) \mathbf{x}_n} - 2\mathbf{x}_n^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] \right. \\
&\quad + [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] + \cancel{\log |\text{diag}(\boldsymbol{\psi}_m)^{-1}|} \\
&\quad \left. + \cancel{D \log 2\pi} \right) - \frac{1}{2} \mathbf{s}_n^T \delta \mathbf{I} \mathbf{s}_n - \cancel{\frac{P}{2} \log 2\pi} - E \\
&= -\frac{1}{2} \left(-2 \sum_{m=1}^T c_{nm} \mathbf{x}_n \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] \right. \\
&\quad \left. + \sum_{m=1}^T c_{nm} [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n + \boldsymbol{\mu}_m] + \mathbf{s}_n^T \delta \mathbf{I} \mathbf{s}_n \right) - E \\
&= -\frac{1}{2} \left(-2 \sum_{m=1}^T c_{nm} \mathbf{x}_n \text{diag}(\boldsymbol{\psi}_m) \mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n \right. \\
&\quad + \sum_{m=1}^T c_{nm} [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n] \\
&\quad \left. + 2 \sum_{m=1}^T c_{nm} [\mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n]^T \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\mu}_m + \cancel{\boldsymbol{\mu}_m^T \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\mu}_m} + \mathbf{s}_n^T \delta \mathbf{I} \mathbf{s}_n \right) - E \\
&= -\frac{1}{2} \left[-2\mathbf{s}_n \sum_{m=1}^T c_{nm} [\mathbf{A} \text{diag}(\mathbf{z}_m)]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{x}_n - \boldsymbol{\mu}_m] \right. \\
&\quad \left. + \mathbf{s}_n^T \left(\delta \mathbf{I} + \sum_{m=1}^T c_{nm} [\mathbf{A} \text{diag}(\mathbf{z}_m)]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m)] \right) \mathbf{s}_n \right] - E
\end{aligned} \tag{D.32}$$

Completing the square reveals that s_{kn} is Gaussian:

$$p(\mathbf{s}_n | \mathbf{x}_n, -) = \log \mathcal{N}_K(\boldsymbol{\xi}_n, \boldsymbol{\Lambda}_n) \tag{D.33}$$

where

$$\Lambda_n = \left(\delta \mathbf{I} + \sum_{m=1}^T c_{nm} [\mathbf{A} \text{diag}(\mathbf{z}_m)]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{A} \text{diag}(\mathbf{z}_m)] \right)^{-1} \quad (\text{D.34})$$

$$\boldsymbol{\xi}_n = \Lambda_n \left(\sum_{m=1}^T c_{nm} [\mathbf{A} \text{diag}(\mathbf{z}_m)]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{x}_n - \boldsymbol{\mu}_m] \right) \quad (\text{D.35})$$

Useful moments in VB updates for other model parameters:

$$\langle \mathbf{s}_n \rangle = \boldsymbol{\xi}_n \quad (\text{D.36})$$

$$\langle \mathbf{s}_n \mathbf{s}_n^T \rangle = \boldsymbol{\xi}_n \boldsymbol{\xi}_n^T + \Lambda_n \quad (\text{D.37})$$

D.6 Variational Posterior on $\mathbf{z}$

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\mathbf{Z})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{Z}) \propto \langle \log p(\mathbf{Z}|\mathbf{X}, -) \rangle \quad (\text{D.38})$$

The true log-posterior may be calculated from Bayes' theorem

$$\log p(\mathbf{Z}|\mathbf{X}, -) = \log p(\mathbf{X}|\mathbf{Z}, -) + \log p(\mathbf{Z}) - E, \quad (\text{D.39})$$

where E denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{z}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\log p(z_{mk} = 1| -) \propto \log p(\mathbf{X}|z_{mk} = 1, -) + \log(\pi_{mk}) - E \quad (\text{D.40})$$

$$\log p(z_{mk} = 0| -) \propto \log p(\mathbf{X}|z_{mk} = 0, -) + \log(1 - \pi_{mk}) - E \quad (\text{D.41})$$

Where $\log p(\mathbf{X}|z_{mk} = 1, -)$ and $\log p(\mathbf{X}|z_{mk} = 1, -)$ are given by (D.20) with z_{mk} set to equal 1 or 0:

$$\begin{aligned}
& \log p(\mathbf{X}|z_{mk} = 1, -) \\
&= -\frac{1}{2} \sum_{n=1}^N c_{nm} \left[\sum_{d=1}^D \psi_{md} (x_{nd}^2 - 2x_{nd}\mu_{md} + \mu_{md}^2) + \sum_{d=1}^D \psi_{md} A_{dk}^2 s_{kn}^2 \right. \\
&\quad \left. - 2 \sum_{d=1}^D \psi_{md} A_{dk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) - \sum_{j=1}^P \log \psi_{md} + D \log 2\pi \right] - E \\
&= -\frac{1}{2} \sum_{n=1}^N \sum_{d=1}^D c_{nm} \psi_{md} A_{dk}^2 s_{kn}^2 + \sum_{n=1}^N \sum_{d=1}^D c_{nm} \psi_{md} A_{dk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) - E \\
& \log p(\mathbf{X}|z_{mk} = 0, -) \\
&= -\frac{1}{2} \sum_{n=1}^N c_{nm} \left[\sum_{d=1}^D \psi_{md} (x_{nd}^2 - 2x_{nd}\mu_{md} + \mu_{md}^2) + \log \prod_{d=1}^P \psi_{md}^{-1} + D \log 2\pi \right] - E \\
&= E
\end{aligned} \tag{D.42}$$

Therefore, $p(z_{mk}|-) \sim \text{Bernoulli}(\rho_{mk})$, where

$$\rho_{mk} = \frac{\exp(\zeta_{mk}^{(1)})}{\exp(\zeta_{mk}^{(1)}) + \exp(\zeta_2)} \tag{D.43}$$

$$\begin{aligned}
\zeta_{mk}^{(1)} &= \log(\pi_{mk}) - \frac{1}{2} \sum_{n=1}^N \sum_{d=1}^D c_{nm} \psi_{md} A_{dk}^2 d_k^2 s_{kn}^2 \\
&\quad + \sum_{n=1}^N \sum_{d=1}^D c_{nm} \psi_{md} A_{dk} d_k s_{kn} (x_{ndm}^{-k} - \mu_{md})
\end{aligned} \tag{D.44}$$

$$\zeta_2 = \log(1 - \pi_{mk}) \tag{D.45}$$

Useful moments in VB updates for other model parameters:

$$\langle z_{mk} \rangle = \rho_{mk} \tag{D.46}$$

$$\langle z_{mk}^2 \rangle = \rho_{mk}, \text{ since } z_{mk} \text{ is binary.} \quad (\text{D.47})$$

$$\langle z_{mk} z_{ml} \rangle = \langle z_{mk} \rangle \langle z_{ml} \rangle \quad (\text{D.48})$$

D.7 Variational Posterior on μ

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\boldsymbol{\mu})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\boldsymbol{\mu}_m) \propto \langle \log p(\boldsymbol{\mu}_m | \mathbf{X}, -) \rangle \quad (\text{D.49})$$

The true log-posterior may be calculated from Bayes' theorem

$$\log p(\boldsymbol{\mu}_m | \mathbf{X}, -) = \log p(\mathbf{X} | \boldsymbol{\mu}_m, -) + \log p(\boldsymbol{\mu}_m) - E, \quad (\text{D.50})$$

where E denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\boldsymbol{\mu}_m$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned}
& \log p(\boldsymbol{\mu}_m | \mathbf{X}, -) \\
&= -\frac{1}{2} \sum_{n=1}^N c_{nm} \left([\mathbf{x}_n - \mathbf{A} \text{diag}(\mathbf{s}_n) - \boldsymbol{\mu}]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{x}_n - \mathbf{A} \text{diag}(\mathbf{s}_n) - \boldsymbol{\mu}_m] \right. \\
&\quad \left. + \log |\cancel{\text{diag}(\boldsymbol{\psi}_m)^{-1}}| + \cancel{D \log 2\pi} \right) - \frac{1}{2} \left([\boldsymbol{\mu}_m - \boldsymbol{\rho}_0]^T u_0 \text{diag}(\boldsymbol{\psi}_m) [\boldsymbol{\mu}_m - \boldsymbol{\rho}_0] \right. \\
&\quad \left. + \log |\cancel{u_0^{-1} \text{diag}(\boldsymbol{\psi}_m)^{-1}}| + \cancel{D \log 2\pi} \right) - E \\
&= -\frac{1}{2} \sum_{n=1}^N c_{nm} \left(\boldsymbol{\mu}_m^T \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\mu}_m - 2\boldsymbol{\mu}_m^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{x}_n - \mathbf{A} \text{diag}(\mathbf{s}_n)] \right. \\
&\quad \left. + [\mathbf{x}_n - \mathbf{A} \text{diag}(\mathbf{s}_n)]^T \text{diag}(\boldsymbol{\psi}_m) [\mathbf{x}_n - \mathbf{A} \text{diag}(\mathbf{s}_n)] \right) - \frac{1}{2} \left(\boldsymbol{\mu}_m^T u_0 \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\mu}_m \right. \\
&\quad \left. - 2\boldsymbol{\mu}_m^T u_0 \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\rho}_0 + \cancel{\boldsymbol{\rho}_0^T u_0 \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\rho}_0} \right) - E \\
&= -\frac{1}{2} \left[-2\boldsymbol{\mu}_m \left(u_0 \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\rho}_0 + \sum_{n=1}^N c_{nm} \text{diag}(\boldsymbol{\psi}_m) [\mathbf{x}_n - \mathbf{A} \text{diag}(\mathbf{s}_n)] \right) \right. \\
&\quad \left. + \sum_{n=1}^N c_{nm} \boldsymbol{\mu}_m^T \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\mu}_m + \boldsymbol{\mu}_m^T u_0 \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\mu}_m \right] - E
\end{aligned} \tag{D.51}$$

Therefore,

$$\log p(\boldsymbol{\mu} | \mathbf{X}, -) = \sum_{m=1}^T \log p(\boldsymbol{\mu}_m | \mathbf{X}, -) = \sum_{m=1}^T \log \mathcal{N}_D(\boldsymbol{\rho}_m, \mathbf{U}_m), \tag{D.52}$$

where

$$\mathbf{U}_m = \text{diag}(\boldsymbol{\psi}_m)^{-1} \left(u_0 + \sum_{n=1}^N c_{nm} \right)^{-1} \tag{D.53}$$

$$\boldsymbol{\rho}_m = \mathbf{U}_m \left(u_0 \text{diag}(\boldsymbol{\psi}_m) \boldsymbol{\rho}_0 + \sum_{n=1}^N c_{nm} \text{diag}(\boldsymbol{\psi}_m) [\mathbf{x}_n - \mathbf{A} \text{diag}(\mathbf{z}_m) \mathbf{s}_n] \right) \tag{D.54}$$

Useful moments in VB updates for other model parameters:

$$\langle \mu_m \rangle = \boldsymbol{\rho}_m \quad (\text{D.55})$$

$$\langle \mu_m \mu_m^T \rangle = \boldsymbol{\rho}_m \boldsymbol{\rho}_m^T + \mathbf{U}_m \quad (\text{D.56})$$

D.8 Variational Posterior on $\boldsymbol{\psi}$

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\boldsymbol{\psi})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\boldsymbol{\psi}) \propto \langle \log p(\boldsymbol{\psi} | \mathbf{X}, -) \rangle \quad (\text{D.57})$$

The true log-posterior may be calculated from Bayes' theorem

$$\log p(\boldsymbol{\psi} | \mathbf{X}, -) = \log p(\mathbf{X} | \boldsymbol{\psi}, -) + \log p(\boldsymbol{\psi}) - E, \quad (\text{D.58})$$

where E denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\boldsymbol{\psi}$, then taking the variational

expectation $\langle \cdot \rangle$:

$$\begin{aligned}
& \log p(\boldsymbol{\psi}|\mathbf{X}, -) \\
&= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[\sum_{d=1}^D \psi_{md} (x_{nd}^2 - 2x_{nd}\mu_{md} + \mu_{md}^2) + \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk}^2 z_{mk}^2 s_{kn}^2 \right. \\
&\quad \left. - 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) - \sum_{j=1}^P \log \psi_{md} + \cancel{D \log 2\pi} \right] \\
&\quad + \sum_{m=1}^T \sum_{d=1}^D [\cancel{g_0 \log h_0} - \log \Gamma(\cancel{g_0}) + (g_0 - 1) \log \psi_{md} - h_0 \psi_{md}] - E \\
&= \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[-\frac{1}{2} \sum_{d=1}^D \psi_{md} (x_{nd}^2 - 2x_{nd}\mu_{md} + \mu_{md}^2) + \sum_{k=1}^K A_{dk}^2 z_{mk}^2 s_{kn}^2 \right. \\
&\quad \left. - 2 \sum_{k=1}^K A_{dk} z_{mk} s_{kn} [x_{ndm}^{-k} - \mu_{md}] + \frac{1}{2} \sum_{j=1}^P \log \psi_{md} \right] \\
&\quad + \sum_{m=1}^T \sum_{d=1}^D (g_0 - 1) \log \psi_{md} - \sum_{m=1}^T \sum_{d=1}^D h_0 \psi_{md} - E
\end{aligned} \tag{D.59}$$

Exponentiating yields:

$$\begin{aligned}
& p(\boldsymbol{\psi}|\mathbf{X}, -) \\
& \propto \prod_{t=1}^T \prod_{j=1}^P \psi_{md}^{g_0-1} \exp(-h_0 \psi_{md}) \prod_{i=1}^N \left[\psi_{md}^{\frac{1}{2}} \exp\left(-\frac{1}{2} \psi_{md} \left[x_{nd}^2 - 2x_{nd} \mu_{md} + \mu_{md}^2 \right. \right. \right. \\
& \quad \left. \left. \left. + \sum_{k=1}^K A_{dk}^2 z_{mk}^2 s_{kn}^2 - 2 \sum_{k=1}^K A_{dk} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) \right] \right) \right]^{c_{nm}} \\
& \propto \prod_{t=1}^T \prod_{j=1}^P \psi_{md}^{g_0-1} \exp(-h_0 \psi_{md}) \prod_{i=1}^N \psi_{md}^{\frac{c_{nm}}{2}} \exp\left(-\frac{1}{2} \psi_{md} c_{nm} \left[x_{nd}^2 - 2x_{nd} \mu_{md} + \mu_{md}^2 \right. \right. \\
& \quad \left. \left. + \sum_{k=1}^K A_{dk}^2 z_{mk}^2 s_{kn}^2 - 2 \sum_{k=1}^K A_{dk} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) \right] \right) \\
& \propto \prod_{t=1}^T \prod_{j=1}^P \psi_{md}^{g_0-1} \exp(-h_0 \psi_{md}) \psi_{md}^{\frac{\sum_{n=1}^N c_{nm}}{2}} \exp\left(-\frac{1}{2} \psi_{md} \sum_{n=1}^N c_{nm} \left[x_{nd}^2 - 2x_{nd} \mu_{md} + \mu_{md}^2 \right. \right. \\
& \quad \left. \left. + \sum_{k=1}^K A_{dk}^2 z_{mk}^2 s_{kn}^2 - 2 \sum_{k=1}^K A_{dk} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) \right] \right) \\
& \propto \prod_{t=1}^T \prod_{j=1}^P \psi_{md}^{\frac{\sum_{n=1}^N c_{nm}}{2} + g_0 - 1} \exp\left(-\psi_{md} \left[h_0 + \frac{1}{2} \sum_{n=1}^N c_{nm} \left(x_{nd}^2 - 2x_{nd} \mu_{md} + \mu_{md}^2 \right. \right. \right. \\
& \quad \left. \left. \left. + \sum_{k=1}^K A_{dk}^2 z_{mk}^2 s_{kn}^2 - 2 \sum_{k=1}^K A_{dk} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) \right) \right] \right)
\end{aligned} \tag{D.60}$$

Therefore,

$$p(\boldsymbol{\psi}|\mathbf{X}, -) = \prod_{t=1}^T \prod_{j=1}^P \text{Gamma}(g_{md}, h_{md}), \tag{D.61}$$

where

$$g_{md} = \frac{\sum_{n=1}^N c_{nm}}{2} + g_0 \tag{D.62}$$

$$\begin{aligned}
h_{md} = & h_0 + \frac{1}{2} \sum_{n=1}^N c_{nm} \left(x_{nd}^2 - 2x_{nd}\mu_{md} + \mu_{md}^2 + \sum_{k=1}^K A_{dk}^2 d_k^2 z_{mk}^2 s_{kn}^2 \right. \\
& \left. - 2 \sum_{k=1}^K A_{dk} d_k z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) \right)
\end{aligned} \tag{D.63}$$

Useful moments in VB updates for other model parameters:

$$\langle \psi_{md} \rangle = \frac{g_{md}}{h_{md}} \tag{D.64}$$

$$\langle \log \psi_{md} \rangle = \text{Digamma}(g_{md}) - \log h_{md} \tag{D.65}$$

D.9 Variational Posterior on π

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\boldsymbol{\pi})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\pi_{mk}) \propto \langle \log p(\pi_{mk} | -) \rangle \tag{D.66}$$

The true posterior may be calculated from Bayes' theorem

$$\begin{aligned}
p(\pi_{mk} | -) & \propto p(z_{mk} | \pi_{mk}) p(\pi_{mk}) \\
& \propto \pi_{mk}^{z_{mk}} (1 - \pi_{mk})^{1-z_{mk}} \pi_{mk}^{\frac{a_0}{K}-1} (1 - \pi_{mk})^{\frac{b_0(K-1)}{K}} \\
& \propto \pi_{mk}^{z_{mk} + \frac{a_0}{K} - 1} (1 - \pi_{mk})^{z_{mk} + \frac{b_0(K-1)}{K} + 1 - 1}
\end{aligned} \tag{D.67}$$

Therefore,

$$p(\pi_{mk} | -) = \text{Beta}(a_{mk}, b_{mk}), \tag{D.68}$$

where

$$a_{mk} = z_{mk} + \frac{a_0}{K} \tag{D.69}$$

$$b_{mk} = z_{mk} + \frac{b_0(K-1)}{K} + 1 \tag{D.70}$$

Useful moments in VB updates for other model parameters:

$$\langle \pi_{mk} \rangle = \frac{a_{mk}}{a_{mk} + b_{mk}} \quad (\text{D.71})$$

$$\langle \log \pi_{mk} \rangle = \text{Digamma}(a_{mk}) - \text{Digamma}(a_{mk} + b_{mk}) \quad (\text{D.72})$$

$$\langle \log(1 - \pi_{mk}) \rangle = \text{Digamma}(b_{mk}) - \text{Digamma}(a_{mk} + b_{mk}) \quad (\text{D.73})$$

D.10 Variational Posterior on γ

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\gamma)$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\gamma_{dk}) \propto \langle \log p(\gamma_{dk} | -) \rangle \quad (\text{D.74})$$

The true posterior may be calculated from Bayes' theorem

$$\begin{aligned} p(\gamma_{dk} | -) &\propto p(A_{dk} | \gamma_{dk}) p(\gamma_{dk}) \\ &\propto (2\pi)^{-\frac{1}{2}} \gamma_{dk}^{\frac{1}{2}} \exp \left[-\frac{\gamma_{dk} A_{dk}^2}{2} \right] \gamma_{dk}^{e_0-1} \exp(-f_0 \gamma_{dk}) \\ &\propto \gamma_{dk}^{e_0+\frac{1}{2}} \exp \left(-\gamma_{dk} \left[f_0 + \frac{A_{dk}^2}{2} \right] \right) \end{aligned} \quad (\text{D.75})$$

Therefore,

$$p(\gamma_{dk} | -) \propto \text{Gamma}(e_{dk}, f_{dk}), \quad (\text{D.76})$$

where

$$e_{dk} = e_0 + \frac{1}{2} \quad (\text{D.77})$$

$$f_{dk} = f_0 + \frac{A_{dk}^2}{2} \quad (\text{D.78})$$

Useful moments in VB updates for other model parameters:

$$\langle \gamma_{dk} \rangle = \frac{e_{dk}}{f_{dk}} \quad (\text{D.79})$$

D.11 Variational Posterior on $\mathbf{C}$

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\mathbf{C})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{C}) \propto \langle \log p(\mathbf{C}|-) \rangle \quad (\text{D.80})$$

The true log-posterior may be calculated from Bayes' theorem

$$\log p(c_{nm} = 1|-) = \log p(\mathbf{x}_n | c_{nm} = 1, -) + \log p(c_{nm} = 1) - E, \quad (\text{D.81})$$

where E denotes a normalizing constant. The posterior will also be multinomial with parameters (responsibilities) Φ :

$$\begin{aligned} \log p(c_{nm} = 1|-) &= -\frac{1}{2} \left[\sum_{d=1}^D \psi_{md} (x_{nd}^2 - 2x_{nd}\mu_{md} + \mu_{md}^2) + \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk}^2 z_{mk}^2 s_{kn}^2 \right. \\ &\quad \left. - 2 \sum_{d=1}^D \sum_{k=1}^K \psi_{md} A_{dk} z_{mk} s_{kn} (x_{ndm}^{-k} - \mu_{md}) - \sum_{j=1}^P \log \psi_{md} + D \log 2\pi \right] \\ &\quad + \log v_m + \sum_{l < m} \log (1 - v_l) - E \end{aligned} \quad (\text{D.82})$$

Useful moments in VB updates for other model parameters:

$$\langle c_{nm} \rangle = \phi_{nm} \quad (\text{D.83})$$

D.12 Variational Posterior on $\mathbf{v}$

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{v})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{v}) \propto \langle \log p(v_m|-) \rangle \quad (\text{D.84})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(v_m|-) = \log p(\mathbf{C}|v_m) + \log p(v_m) - E, \quad (\text{D.85})$$

where E denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{v}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} \log p(v_m|-) &= \sum_{n=1}^N c_{nm} \log v_m + \sum_{n=1}^N \sum_{l>m} z_{nl} \log(1 - v_m) + (\alpha - 1) \log(1 - v_m) - K \\ &= \sum_{n=1}^N c_{nm} \log v_m + \left(\alpha + \sum_{n=1}^N \sum_{l>m} c_{nl} - 1 \right) \log(1 - v_m) - E \end{aligned} \quad (\text{D.86})$$

Therefore,

$$\log p(v_m|-) = \text{Beta}(\nu_{t1}, \nu_{t2}) \quad (\text{D.87})$$

where

$$\nu_{t1} = 1 + \sum_{n=1}^N c_{nm} \quad (\text{D.88})$$

$$\nu_{t2} = \alpha + \sum_{n=1}^N \sum_{s>m} c_{nm} \quad (\text{D.89})$$

Useful moments in VB updates for other model parameters:

$$\langle \ln v_m \rangle = \psi(\nu_{t1}) - \psi(\nu_{t1} + \nu_{t2}) \quad (\text{D.90})$$

$$\langle \ln(1 - v_m) \rangle = \psi(\nu_{t2}) - \psi(\nu_{t1} + \nu_{t2}) \quad (\text{D.91})$$

D.13 Variational Posterior on α

It was derived in Section that the NFE is maximized by a variational posterior, $q(\alpha)$, that is proportional to the variational expectation of the true log-posterior

with respect to all other model parameters:

$$q(\alpha) \propto \langle \log p(\alpha| -) \rangle \quad (\text{D.92})$$

The true posterior may be calculated from Bayes' theorem:

$$\log p(\alpha| -) = \log p(\mathbf{v}|\alpha) + \log p(\alpha) - E, \quad (\text{D.93})$$

where E denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of α , then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} \log p(\alpha| -) &= \sum_{t=1}^{T-1} (\alpha - \mathbb{I}) \log(1 - v_m) + \log \left[\frac{(T-1)}{\alpha} \exp(-\tau_{20}\alpha + \tau_{10} \log \alpha \right. \\ &\quad \left. - \log \Gamma(\tau_{10}) + \tau_{10} \log \tau_{20}) \right] - E \\ &= \left[-\tau_{20} + \sum_{t=1}^{T-1} \log(1 - v_m) \right] \alpha + \sum_{m=1}^{T-1} (1 - v_m) + \log \alpha (\tau_{10} + T - 1) - E \end{aligned} \quad (\text{D.94})$$

Therefore,

$$p(\alpha| -) \propto \text{Gamma}(\tau_1, \tau_2), \quad (\text{D.95})$$

where

$$\tau_1 = \tau_{10} + T - 1 \quad (\text{D.96})$$

$$\tau_2 = \tau_{20} - \sum_{m=1}^{T-1} \log(1 - v_m) \quad (\text{D.97})$$

Useful moments in VB updates for other model parameters:

$$\langle \alpha \rangle = \frac{\tau_1}{\tau_2} \quad (\text{D.98})$$

D.14 Variational Posterior on δ

It was derived in Section that the NFE is maximized by a variational posterior, $q(\delta)$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\delta) \propto \langle \log p(\delta|-) \rangle \quad (\text{D.99})$$

The true posterior may be calculated from Bayes' theorem:

$$\begin{aligned} p(\delta|-) &\propto p(\mathbf{S}|\delta)p(\delta) \\ &\propto \prod_{i=1}^N (2\pi)^{-P/2} \delta^{-K/2} \exp\left(-\frac{1}{2}\delta \sum_{k=1}^K s_{kn}^2\right) \delta^{\delta_{10}-1} \exp(-\delta_{20}\delta) \\ &\propto \delta^{\delta_{10}+KN/2-1} \exp\left[-\delta \left(\delta_{20} + \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K s_{kn}^2\right)\right] \end{aligned} \quad (\text{D.100})$$

Therefore,

$$q(\delta) \propto \text{Gamma}(\delta_1, \delta_2), \quad (\text{D.101})$$

where

$$\delta_1 = \delta_{10} + \frac{KN}{2} \quad (\text{D.102})$$

$$\delta_2 = \delta_{20} + \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K s_{kn}^2. \quad (\text{D.103})$$

Useful moments in VB updates for other model parameters:

$$\langle \delta \rangle = \frac{\delta_1}{\delta_2} \quad (\text{D.104})$$

D.15 Negative Free Energy

The NFE can be expressed as the difference between the expected log-likelihood and the Kullback-Leibler divergence (KLD) between the variational posteriors and the

priors:

$$\begin{aligned}
\mathcal{F} = & \langle \log p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \mathbf{Z}, \Psi) \rangle \\
& - \mathbb{KLD}[q(\mathbf{A}, \mathbf{S}, \mathbf{Z}, \boldsymbol{\mu}, \Psi, \boldsymbol{\pi}, \boldsymbol{\gamma}, \mathbf{C}, \mathbf{v}, \alpha, \delta) || p(\mathbf{A}, \mathbf{S}, \mathbf{Z}, \boldsymbol{\mu}, \Psi, \boldsymbol{\pi}, \boldsymbol{\gamma}, \mathbf{C}, \mathbf{v}, \alpha, \delta)] \\
= & \langle \log p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \mathbf{Z}, \Psi) \rangle - \sum_{d=1}^D \sum_{k=1}^K \mathbb{KLD}[q(A_{dk}) || p(A_{dk})] - \sum_{n=1}^N \mathbb{KLD}[q(\mathbf{s}_n) || p(\mathbf{s}_n)] \\
& - \sum_{m=1}^T \sum_{k=1}^K \mathbb{KLD}[q(z_{mk}) || p(z_{mk})] - \sum_{m=1}^T \mathbb{KLD}[q(\boldsymbol{\mu}_m) || p(\boldsymbol{\mu}_m)] \\
& - \sum_{m=1}^T \sum_{d=1}^D \mathbb{KLD}[q(\psi_{md}) || p(\psi_{md})] - \sum_{m=1}^T \sum_{k=1}^K \mathbb{KLD}[q(\pi_{tk}) || p(\pi_{tk})] \\
& - \sum_{d=1}^D \sum_{k=1}^K \mathbb{KLD}[q(\gamma_{dk}) || p(\gamma_{dk})] - \sum_{n=1}^N \mathbb{KLD}[q(\mathbf{c}_n) || p(\mathbf{c}_n)] \\
& - \sum_{m=1}^T \mathbb{KLD}[q(v_m) || p(v_m)] - \mathbb{KLD}[q(\alpha) || p(\alpha)] - \mathbb{KLD}[q(\delta) || p(\delta)]
\end{aligned} \tag{D.105}$$

where

$$\begin{aligned}
\langle \log p(\mathbf{X}|\mathbf{A}, \mathbf{S}, \mathbf{Z}, \Psi) \rangle = & -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T \langle c_{nm} \rangle \left[\sum_{d=1}^D \langle \psi_{md} \rangle (x_{nd}^2 - 2x_{nd} \langle \mu_{md} \rangle + \langle \mu_{md}^2 \rangle) \right. \\
& + \sum_{d=1}^D \sum_{k=1}^K \langle \psi_{md} \rangle \langle A_{dk}^2 \rangle \langle z_{mk}^2 \rangle \langle s_{kn}^2 \rangle \\
& - 2 \sum_{d=1}^D \sum_{k=1}^K \langle \psi_{md} \rangle \langle A_{dk} z_{mk} s_{kn} x_{ndm}^{-k} \rangle \\
& - 2 \sum_{d=1}^D \sum_{k=1}^K \langle \psi_{md} \rangle \langle A_{dk} \rangle \langle z_{mk} \rangle \langle s_{kn} \rangle \langle \mu_{md} \rangle \\
& \left. + \langle \log \prod_{j=1}^P \psi_{md}^{-1} \rangle + D \log 2\pi \right],
\end{aligned} \tag{D.106}$$

and $\mathbb{KLD}[q(A_{dk})||p(A_{dk})]$ is a KLD between two Gaussian distributions, $\mathbb{KLD}[q(\mathbf{s}_n)||p(\mathbf{s}_n)]$ is between two K -dimensional Gaussian distributions, $\mathbb{KLD}[q(z_{mk})||p(z_{mk})]$ is between two Bernoulli distributions, $\mathbb{KLD}[q(\boldsymbol{\mu}_m)||p(\boldsymbol{\mu}_m)]$ is between two D -dimensional Gaussian distributions, $\mathbb{KLD}[q(\psi_{md})||p(\psi_{md})]$ is between two Gamma distributions, $\mathbb{KLD}[q(\pi_{tk})||p(\pi_{tk})]$ is between two Beta distributions, $\mathbb{KLD}[q(\gamma_{dk})||p(\gamma_{dk})]$ is between two Gamma distributions, $\mathbb{KLD}[q(\mathbf{c}_n)||p(\mathbf{c}_n)]$ is between two multinomial distributions, $\mathbb{KLD}[q(v_m)||p(v_m)]$ is between two Beta distributions, $\mathbb{KLD}[q(\alpha)||p(\alpha)]$ is between two Gamma distributions, and $\mathbb{KLD}[q(\delta)||p(\delta)]$ is between two Gamma distributions.

Appendix E

Discriminative DPGMM-RVM

The DPGMM-RVM hybrid model is used for discriminative context learning in Chapter 5. The model is constructed based on the mixture-of-RVMs presented in Appendix B.3 where the latent mixing variables are governed by a DPGMM, which was described in Appendix C. Therefore, the derivations for the update equations and NFE for the DPGMM-RVM are very similar to the individual RVM and DPGMM models. Learning the DPGMM seeks to jointly cluster the contextual features ($\mathbf{X}^{(C)}$) and classify the target features ($\mathbf{X}^{(T)}$) according to the labels, $\mathbf{t}$.

E.1 Generative Model and Variable Definitions

$$y_{nm} = \mathbf{w}_m^T \mathbf{x}_n^{(T)} \quad (\text{E.1})$$

$$(t_n | c_{nm} = 1) \sim \sigma(y_{nm})^{t_n} [1 - \sigma(y_{nm})]^{1-t_n} \quad (\text{E.2})$$

$$(\mathbf{x}_n^{(C)} | c_{nm} = 1) \sim \mathcal{N}_{D^{(C)}}(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m^{-1}) \quad (\text{E.3})$$

$\mathbf{x}_n^{(T)}$ is $D^{(T)} \times 1$ target feature vector

$\mathbf{x}_n^{(C)}$ is $D^{(C)} \times 1$ contextual feature vector
 t_n is binary class label
 $\mathbf{c}_n$ is binary-coded latent variable
 $n = 1, 2, \dots, N$ is data index
 $m = 1, 2, \dots, T$ is mixture component index
 $d = 1, 2, \dots, D^{(C)}$ or $D^{(T)}$ is dimension index

E.2 Priors

$$(\boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m) \sim \mathcal{N}_{D^{(C)}}(\boldsymbol{\mu}_m | \boldsymbol{\rho}_0, u_0^{-1} \boldsymbol{\Lambda}_m^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_m | \mathbf{B}_0, \nu_0) \quad (\text{E.4})$$

$$\mathbf{w}_m \sim \mathcal{N}_{D^{(T)}}(0, \text{diag}(\boldsymbol{\beta}_m)^{-1}) \quad (\text{E.5})$$

$$\beta_{md} \sim \text{Gamma}(a_0, b_0) \quad (\text{E.6})$$

$$\mathbf{c}_n \sim \text{Multinomial}(\boldsymbol{\pi}_n) \quad (\text{E.7})$$

$$\pi_m = v_m \prod_{l < h} (1 - v_l) \quad (\text{E.8})$$

$$v_m \sim \text{Beta}(1, \alpha) \quad (\text{E.9})$$

$$\alpha \sim \text{Gamma}(\tau_{10}, \tau_{20}) \quad (\text{E.10})$$

E.3 Model Likelihood

The joint likelihood of labels and context features, given all model parameters is given by

$$p(\mathbf{t}, \mathbf{X}^{(C)} | -) = \prod_{n=1}^N \prod_{m=1}^T [\sigma(y_{nm})^{t_n} [1 - \sigma(y_{nm})]^{1-t_n} \mathcal{N}_{D^{(C)}}(\mathbf{x}_n^{(C)} | \boldsymbol{\mu}_m, \boldsymbol{\Lambda}_m^{-1})]^{c_{nm}} \quad (\text{E.11})$$

s:

$$\begin{aligned} \log p(\mathbf{t}, \mathbf{X}^{(C)} | -) = & \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left(t_n \log \sigma(y_{nm}) + (1 - t_n) \log [1 - \sigma(y_{nm})] \right. \\ & \left. - \frac{1}{2} \left[D^{(C)} \log 2\pi + \log |\mathbf{\Lambda}_m^{-1}| + (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m)^T \mathbf{\Lambda}_m (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m) \right] \right) \end{aligned} \quad (\text{E.12})$$

E.4 Variational Posterior on $\boldsymbol{\mu}$ and $\mathbf{\Lambda}$

It was derived in Section E.10 that the NFE is maximized by a variational posterior, $q(\boldsymbol{\mu}, \mathbf{\Lambda})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\boldsymbol{\mu}, \mathbf{\Lambda}) \propto \langle \log p(\boldsymbol{\mu}, \mathbf{\Lambda} | -) \rangle \quad (\text{E.13})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\begin{aligned} \log p(\boldsymbol{\mu}, \mathbf{\Lambda} | -) = & \log p(\mathbf{t} | \mathbf{X}^{(T)}, \mathbf{W}, -) + \log p(\mathbf{X}^{(C)} | \boldsymbol{\mu}, \mathbf{\Lambda}, -) \\ & + \log p(\boldsymbol{\mu} | \mathbf{\Lambda}) + \log p(\mathbf{\Lambda}) - K, \end{aligned} \quad (\text{E.14})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\boldsymbol{\mu}$ and $\mathbf{\Lambda}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned}
& \log p(\boldsymbol{\mu}, \boldsymbol{\Lambda} | -) \\
&= \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left(\cancel{t_n \log \sigma(y_{nm})} + (1 - t_n) \log [1 - \cancel{\sigma(y_{nm})}] - \frac{1}{2} \left[\cancel{D^{(C)} \log 2\pi} + \log |\boldsymbol{\Lambda}_m^{-1}| \right. \right. \\
&\quad \left. \left. + (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m)^T \boldsymbol{\Lambda}_m (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m) \right] \right) - \frac{1}{2} \sum_{m=1}^T \left[\cancel{D^{(C)} \log 2\pi} + \log u_0^{D^{(C)}} + \log |\boldsymbol{\Lambda}_m^{-1}| \right. \\
&\quad \left. + (\boldsymbol{\mu}_m - \mathbf{m}_0)^T u_0 \boldsymbol{\Lambda}_m (\boldsymbol{\mu}_m - \mathbf{m}_0) \right] + \sum_{m=1}^T \left[\frac{\nu_0 - D^{(C)} - 1}{2} \log |\boldsymbol{\Lambda}_m| - \cancel{\frac{\nu_0 D^{(C)}}{2} \log 2} \right. \\
&\quad \left. - \cancel{\frac{\nu_0}{2} \log |\mathbf{B}_0|} - \cancel{\Gamma_{D^{(C)}}\left(\frac{\nu_0}{2}\right)} - \frac{1}{2} \text{Tr}(\mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) \right] - K \\
&= -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left[\boldsymbol{\mu}_m^T \boldsymbol{\Lambda}_m \boldsymbol{\mu}_m - 2\boldsymbol{\mu}_m^T \boldsymbol{\Lambda}_m \mathbf{x}_n^{(C)} + \mathbf{x}_n^{(C)T} \boldsymbol{\Lambda}_m \mathbf{x}_n^{(C)} \right] \\
&\quad - \frac{1}{2} \sum_{m=1}^T \left[\boldsymbol{\mu}_m^T u_0 \boldsymbol{\Lambda}_m \boldsymbol{\mu}_m - 2\boldsymbol{\mu}_m^T u_0 \boldsymbol{\Lambda}_m \mathbf{m}_0 + \mathbf{m}_0^T u_0 \boldsymbol{\Lambda}_m \mathbf{m}_0 \right] \\
&\quad - \frac{1}{2} \sum_{m=1}^T \left[\text{Tr}(\mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D^{(C)} - 1 \right) \log |\boldsymbol{\Lambda}_m| \right] - K \\
&= -\frac{1}{2} \sum_{m=1}^T \left[-2\boldsymbol{\mu}_m^T \boldsymbol{\Lambda}_m \left(u_0 \boldsymbol{\rho}_0 + \sum_{n=1}^N c_{nm} \mathbf{x}_n \right) + \boldsymbol{\mu}_m^T \left(\sum_{n=1}^N c_{nm} + u_0 \right) \boldsymbol{\Lambda}_m \boldsymbol{\mu}_m \right. \\
&\quad \left. + \boldsymbol{\rho}_m^T u_m \boldsymbol{\Lambda}_m \boldsymbol{\rho}_m - \boldsymbol{\rho}_m^T u_m \boldsymbol{\Lambda}_m \boldsymbol{\rho}_m + \sum_{n=1}^N c_{nm} \mathbf{x}_n^{(C)T} \boldsymbol{\Lambda}_m \mathbf{x}_n^{(C)} + \mathbf{m}_0^T u_0 \boldsymbol{\Lambda}_m \mathbf{m}_0 + \text{Tr}(\mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) \right. \\
&\quad \left. - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D^{(C)} - 1 \right) \log |\boldsymbol{\Lambda}_m| \right] - K
\end{aligned} \tag{E.15}$$

Completing the square in the first two terms, and then using the identity $\mathbf{a}^T \mathbf{B} \mathbf{a} =$

$\text{Tr}(\mathbf{a}\mathbf{a}^T\mathbf{B})$ yields:

$$\begin{aligned}
& \log p(\boldsymbol{\mu}, \boldsymbol{\Lambda} | -) \\
&= -\frac{1}{2} \sum_{m=1}^T \left[(\boldsymbol{\mu}_m - \boldsymbol{\rho}_m)^T u_m \boldsymbol{\Lambda}_m (\boldsymbol{\mu}_m - \boldsymbol{\rho}_m) - \text{Tr}(u_m \mathbf{M}_m \boldsymbol{\Lambda}_m) + \text{Tr}(\mathbf{C}_m \boldsymbol{\Lambda}_m) \right. \\
&\quad \left. + \text{Tr}(u_0 \mathbf{M}_0 \boldsymbol{\Lambda}_0) + \text{Tr}(\mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m) - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D^{(C)} \right) \log |\boldsymbol{\Lambda}_m| \right] - K
\end{aligned} \tag{E.16}$$

Where

$$u_m = u_0 + \sum_{n=1}^N c_{nm} \tag{E.17}$$

$$\boldsymbol{\rho}_m = \frac{u_0 \boldsymbol{\rho}_0 + \sum_{n=1}^N c_{nm} \mathbf{x}_n^{(C)}}{u_m} \tag{E.18}$$

$$\mathbf{M}_m = \boldsymbol{\rho}_m \boldsymbol{\rho}_m^T \tag{E.19}$$

$$\mathbf{C}_m = \sum_{n=1}^N c_{nm} \mathbf{x}_n^{(C)} \mathbf{x}_n^{(C)T} \tag{E.20}$$

$$\mathbf{M}_0 = \boldsymbol{\rho}_0 \boldsymbol{\rho}_0^T \tag{E.21}$$

Consolidating the last four terms using the identity $\text{Tr}(\mathbf{A}) + \text{Tr}(\mathbf{B}) = \text{Tr}(\mathbf{AB})$ yields:

$$\begin{aligned}
& \log p(\boldsymbol{\mu}, \boldsymbol{\Lambda} | -) \\
&= -\frac{1}{2} \sum_{m=1}^T \left[(\boldsymbol{\mu}_m - \boldsymbol{\rho}_m)^T u_m \boldsymbol{\Lambda}_m (\boldsymbol{\mu}_m - \boldsymbol{\rho}_m) + \text{Tr} \left(\mathbf{C}_m \boldsymbol{\Lambda}_m - u_m \mathbf{M}_m \boldsymbol{\Lambda}_m \right. \right. \\
&\quad \left. \left. + u_0 \mathbf{M}_0 \boldsymbol{\Lambda}_0 + \mathbf{B}_0^{-1} \boldsymbol{\Lambda}_m \right) - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D^{(C)} - 1 \right) \log |\boldsymbol{\Lambda}_m| \right] - K \\
&= -\frac{1}{2} \sum_{m=1}^T \left[(\boldsymbol{\mu}_m - \boldsymbol{\rho}_m)^T u_m \boldsymbol{\Lambda}_m (\boldsymbol{\mu}_m - \boldsymbol{\rho}_m) \right. \\
&\quad \left. + \text{Tr} \left[(\mathbf{C}_m - u_m \mathbf{M}_m + u_0 \mathbf{M}_0 + \mathbf{B}_0^{-1}) \boldsymbol{\Lambda}_m \right] \right. \\
&\quad \left. - \left(\nu_0 + \sum_{n=1}^N c_{nm} - D^{(C)} - 1 \right) \log |\boldsymbol{\Lambda}_m| \right] - K
\end{aligned} \tag{E.22}$$

Consolidating terms reveals that $\boldsymbol{\mu}, \boldsymbol{\Lambda}$ are Normal-Wishart:

$$\log p(\boldsymbol{\mu}, \boldsymbol{\Lambda} | -) = \sum_{m=1}^T \log [\mathcal{N}(\boldsymbol{\mu}_m | \boldsymbol{\rho}_m, u_m^{-1} \boldsymbol{\Lambda}_m^{-1}) \mathcal{W}(\boldsymbol{\Lambda} | \nu_m, \mathbf{B}_m)] \tag{E.23}$$

where $\boldsymbol{\rho}_m$ and u_m are defined above, and

$$\nu_m = \nu_0 + \sum_{n=1}^N c_{nm} \tag{E.24}$$

$$\mathbf{B}_m = (\mathbf{C}_m - u_m \mathbf{M}_m + u_0 \mathbf{M}_0 + \mathbf{B}_0^{-1})^{-1} \tag{E.25}$$

Useful moments in VB updates for other model parameters:

$$\langle \boldsymbol{\mu}_m \rangle = \boldsymbol{\rho}_m \tag{E.26}$$

$$\langle \boldsymbol{\Lambda}_m \rangle = \nu_m \mathbf{B}_m \tag{E.27}$$

$$\langle \boldsymbol{\mu}_m \boldsymbol{\mu}_m^T \rangle = \boldsymbol{\rho}_m \boldsymbol{\rho}_m^T + u_m^{-1} \nu_m^{-1} \mathbf{B}_m^{-1} \tag{E.28}$$

$$\langle \log |\boldsymbol{\Lambda}_m| \rangle = \sum_{d=1}^{D^{(C)}} \psi \left(\frac{\nu_m - p + 1}{2} \right) + D^{(C)} \log 2 + \log |\mathbf{B}_m| \quad (\text{E.29})$$

$$\langle (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m)^T \boldsymbol{\Lambda}_m (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m) \rangle = (\mathbf{x}_n^{(C)} - \boldsymbol{\rho})^T \nu_m \mathbf{B}_m (\mathbf{x}_n^{(C)} - \boldsymbol{\rho}_m) + \frac{D^{(C)}}{u_m} \quad (\text{E.30})$$

E.5 Variational Posterior on w

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{W})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{W}) \propto \langle \log p(\mathbf{W}|-) \rangle \quad (\text{E.31})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\begin{aligned} \log p(\mathbf{W}|-) &= \log p(\mathbf{t}|\mathbf{W}, \mathbf{X}^{(T)}, -) + \log p(\mathbf{X}^{(T)}|-) \\ &\quad + \log p(\mathbf{W}) - K, \end{aligned} \quad (\text{E.32})$$

where K denotes a normalizing constant.

Because the binomial distribution on $\mathbf{t}$ does not offer conjugate updating for our choice of the prior on $\mathbf{w}$, we impose a lower-bound approximation to $p(\mathbf{t}_n|\mathbf{w}_m, \mathbf{x}_n^{(T)})$:

$$\begin{aligned} p(\mathbf{t}_n|\mathbf{w}_m) &= \sigma(y_{nm})^{t_n} [1 - \sigma(y_{nm})]^{1-t_n} \\ &\geq \sigma(\xi_{nm}) \exp \left[\frac{\gamma_{nm} - \xi_{nm}}{2} - \lambda(\xi_{nm}) (\gamma_{nm}^2 - \xi_{nm}^2) \right] \end{aligned} \quad (\text{E.33})$$

where ξ_{nm} is a variational parameter and

$$\gamma_{nm} = (2t_n - 1) y_{nm} \quad (\text{E.34})$$

$$\lambda(\xi_{nm}) = \frac{1}{4\xi_{nm}} \tanh \left(\frac{\xi_{nm}}{2} \right) \quad (\text{E.35})$$

Using the approximation for $p(\mathbf{t}_n|\mathbf{w}_m, \mathbf{x}_n^{(T)})$, the variational posterior can be calculated by solving for $p(\mathbf{W}|-)$ as a function of $\mathbf{W}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned}
& \log p(\mathbf{W}|-) \\
&= \sum_{n=1}^N \sum_{m=1}^T c_{nm} \left(\left[\log \sigma(\xi_{nm}) + \frac{1}{2} (\gamma_{nm} - \xi_{nm}) - \lambda(\xi_{nm}) (\gamma_{nm}^2 - \xi_{nm}^2) \right] \right. \\
&\quad \left. - \frac{1}{2} D^{(C)} \log 2\pi - \frac{1}{2} \log |\mathbf{\Lambda}_m^{-1}| - \frac{1}{2} (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m)^T \mathbf{\Lambda}_m (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m) \right) \\
&\quad - \frac{1}{2} \sum_{m=1}^T [P \log 2\pi + \log |\mathbf{A}| + \mathbf{w}_m^T \mathbf{A} \mathbf{w}_m] - K \\
&= -\frac{1}{2} \sum_{m=1}^T \left[\mathbf{w}_m^T \mathbf{A} \mathbf{w}_m + \sum_{n=1}^N c_{nm} (2\lambda(\xi_{nm}) \gamma_{nm}^2 - \gamma_{nm}) \right] \\
&= -\frac{1}{2} \sum_{m=1}^T \left[\mathbf{w}_m^T \mathbf{A} \mathbf{w}_m + \sum_{n=1}^N c_{nm} \left(2\lambda(\xi_{nm}) \mathbf{w}_m^T \mathbf{x}_n^{(T)} \mathbf{x}_n^{(T)T} \mathbf{w}_m - (2t_n - 1) \mathbf{w}_m^T \mathbf{x}_n^{(T)} \right) \right] \\
&= -\frac{1}{2} \sum_{m=1}^T \left[-2\mathbf{w}_m^T \left(\frac{1}{2} \sum_{n=1}^N c_{nm} (2t_n - 1) \mathbf{x}_n^{(T)} \right) \right. \\
&\quad \left. + \mathbf{w}_m^T \left(\mathbf{A} + 2 \sum_{n=1}^N c_{nm} \lambda(\xi_{nm}) \mathbf{x}_n \mathbf{x}_n^{(T)} \right) \mathbf{w}_m \right]
\end{aligned} \tag{E.36}$$

Completing the square reveals that $\mathbf{W}$ is Gaussian:

$$\log p(\mathbf{W}|-) = \sum_{m=1}^T \log \mathcal{N}(\mathbf{w}_m | \boldsymbol{\omega}_m, \boldsymbol{\Sigma}_m) \tag{E.37}$$

where

$$\boldsymbol{\omega}_m = \frac{1}{2} \boldsymbol{\Sigma}_m \left(\sum_{n=1}^N c_{nm} (2t_n - 1) \mathbf{x}_n^{(T)} \right) \tag{E.38}$$

$$\Sigma_m = \left(\mathbf{A} + 2 \sum_{n=1}^N c_{nm} \lambda(\xi_{nm}) \mathbf{x}_n^{(T)} \mathbf{x}_n^{(T)T} \right)^{-1} \quad (\text{E.39})$$

Useful moments in VB updates for other model parameters:

$$\langle \mathbf{w}_m \rangle = \boldsymbol{\omega}_m \quad (\text{E.40})$$

$$\langle \mathbf{w}_m \mathbf{w}_m^T \rangle = \boldsymbol{\omega}_m \boldsymbol{\omega}_m^T + \Sigma_m \quad (\text{E.41})$$

E.6 Variational Posterior on ξ

The updates for the variational parameter ξ are derived by directly optimizing the Negative Free Energy:

$$\begin{aligned} \mathcal{L} = & \langle \log p(\mathbf{t}, \mathbf{X}^{(C)} | -) \rangle - \text{KLD}[q(\boldsymbol{\mu}, \mathbf{A}) || p(\boldsymbol{\mu}, \mathbf{A})] - \text{KLD}[q(\mathbf{W}) || p(\mathbf{w})] \\ & - \text{KLD}[q(\mathbf{A}) || p(\mathbf{A})] - \text{KLD}[q(\mathbf{Z}) || p(\mathbf{Z})] - \text{KLD}[q(\mathbf{v}) || p(\mathbf{v})] - \text{KLD}[q(\alpha) || p(\alpha)] \end{aligned} \quad (\text{E.42})$$

$$\frac{\partial \mathcal{L}}{\partial \xi} = \frac{\partial}{\partial \xi} \langle \log p(\mathbf{t}, \mathbf{X}^{(C)} | -) \rangle = \frac{\partial}{\partial \xi} \langle p(\mathbf{t}_n | \mathbf{w}_m, \mathbf{x}_n^{(T)}) \rangle \quad (\text{E.43})$$

Substituting the approximation for $p(\mathbf{t}_n | \mathbf{w}_m, \mathbf{x}_n^{(T)})$:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \xi} &= \sum_{m=1}^T \sum_{d=1}^{D^{(T)}} \left[\frac{1}{1 + e^{\xi_{md}}} - \frac{1}{2} + 2\xi_{md}\lambda(\xi_{md}) - \frac{\partial \lambda(\xi_{md})}{\partial \xi_{md}} (\langle \gamma_{md}^2 \rangle - \xi_{md}^2) \right] \\ &= \sum_{m=1}^T \sum_{d=1}^{D^{(T)}} \left[\frac{e^{-\xi_{md}/2}}{e^{\xi_{md}/2} + e^{-\xi_{md}/2}} + \frac{\frac{1}{2}e^{\xi_{md}/2} - \frac{1}{2}e^{-\xi_{md}/2}}{e^{\xi_{md}/2} + e^{-\xi_{md}/2}} - \frac{1}{2} - \frac{\partial \lambda(\xi_{md})}{\partial \xi_{md}} (\langle \gamma_{md}^2 \rangle - \xi_{md}^2) \right] \\ &= \sum_{m=1}^T \sum_{d=1}^{D^{(T)}} \left[\frac{\frac{1}{2}e^{\xi_{md}/2} + \frac{1}{2}e^{-\xi_{md}/2}}{e^{\xi_{md}/2} + e^{-\xi_{md}/2}} - \frac{1}{2} - \frac{\partial \lambda(\xi_{md})}{\partial \xi_{md}} (\langle \gamma_{md}^2 \rangle - \xi_{md}^2) \right] \\ &= - \sum_{m=1}^T \sum_{d=1}^{D^{(T)}} \frac{\partial \lambda(\xi_{md})}{\partial \xi_{md}} (\langle \gamma_{md}^2 \rangle - \xi_{md}^2) \end{aligned} \quad (\text{E.44})$$

Because the derivative of $\lambda(\xi_{md})$ is purely negative, $\mathcal{L}$ is maximized at

$$\xi_{md}^2 = \langle \gamma_{md}^2 \rangle = \mathbf{x}^{(T)T} \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \mathbf{x}^{(T)} \quad (\text{E.45})$$

E.7 Variational Posterior on β

It was derived in Section that the NFE is maximized by a variational posterior, $q(\beta)$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\beta) \propto \langle \log p(\beta|-) \rangle \quad (\text{E.46})$$

The true posterior may be calculated from Bayes' theorem:

$$p(\beta|-) \propto p(\mathbf{W}|\beta) p(\beta) \quad (\text{E.47})$$

The variational posterior can be calculated by solving the true posterior as a function of β , then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} p(\beta|-) &\propto \prod_{m=1}^T \prod_{d=1}^{D^{(T)}} (2\pi)^{-\frac{1}{2}} \beta_{md}^{\frac{1}{2}} \exp\left(-\frac{\beta_{md} w_{md}^2}{2}\right) \frac{b_0^{a_0}}{\Gamma(a_0)} \beta_{md}^{a_0-1} \exp(-b_0 \beta_{md}) \\ &\propto \prod_{m=1}^T \prod_{d=1}^{D^{(T)}} \beta_{md}^{\frac{1}{2}} \exp\left(-\frac{\beta_{md} w_{md}^2}{2}\right) \beta_{md}^{a_0-1} \exp(-b_0 \beta_{md}) \\ &\propto \prod_{m=1}^T \prod_{d=1}^{D^{(T)}} \beta_{md}^{a_0 + \frac{1}{2} - 1} \exp\left(-\beta_{md} \left[b_0 + \frac{1}{2} w_{md}^2\right]\right) \end{aligned} \quad (\text{E.48})$$

Therefore, the β 's are Gamma distributed:

$$p(\beta|-) = \prod_{m=1}^T \prod_{d=1}^{D^{(T)}} \text{Gamma}(\beta_{md} | a_{md}, b_{md}), \quad (\text{E.49})$$

where

$$a_{md} = a_0 + \frac{1}{2} \quad (\text{E.50})$$

$$b_{md} = b_0 + \frac{1}{2} w_{md}^2 \quad (\text{E.51})$$

Useful moments in VB updates for other model parameters:

$$\langle \beta_{md} \rangle = \frac{a_{md}}{b_{md}} \quad (\text{E.52})$$

E.8 Variational Posterior on V

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{v})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{v}) \propto \langle \log p(v_m | -) \rangle \quad (\text{E.53})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(v_m | -) = \log p(\mathbf{C} | v_m) + \log p(v_m) - K, \quad (\text{E.54})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of $\mathbf{v}$, then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} \log p(v_m | -) &= \sum_{n=1}^N c_{nm} \log v_m + \sum_{n=1}^N \sum_{l>m} z_{nl} \log(1 - v_m) + (\alpha - 1) \log(1 - v_m) - K \\ &= \sum_{n=1}^N c_{nm} \log v_m + \left(\alpha + \sum_{n=1}^N \sum_{l>m} c_{nl} - 1 \right) \log(1 - v_m) - K \end{aligned} \quad (\text{E.55})$$

Therefore,

$$p(v_m | -) = \text{Beta}(\nu_{m1}, \nu_{m2}) \quad (\text{E.56})$$

where

$$\nu_{m1} = 1 + \sum_{n=1}^N c_{nm} \quad (\text{E.57})$$

$$\nu_{m2} = \alpha + \sum_{n=1}^N \sum_{l>m} c_{nl} \quad (\text{E.58})$$

Useful moments in VB updates for other model parameters:

$$\langle \ln v_m \rangle = \psi(\nu_{m1}) - \psi(\nu_{m1} + \nu_{m2}) \quad (\text{E.59})$$

$$\langle \ln(1 - v_m) \rangle = \psi(\nu_{m2}) - \psi(\nu_{m1} + \nu_{m2}) \quad (\text{E.60})$$

E.9 Variational Posterior on α

It was derived in Section that the NFE is maximized by a variational posterior, $q(\alpha)$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\alpha) \propto \langle \log p(\alpha| -) \rangle \quad (\text{E.61})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(\alpha| -) = \log p(\mathbf{v}|\alpha) + \log p(\alpha) - K, \quad (\text{E.62})$$

where K denotes a normalizing constant. The variational posterior can be calculated by solving the true log-posterior as a function of α , then taking the variational expectation $\langle \cdot \rangle$:

$$\begin{aligned} \log p(\alpha| -) &= \sum_{m=1}^{T-1} (\alpha - \mathcal{I}) \log(1 - v_m) - K \\ &\quad + \log \left[\frac{(M-1)}{\alpha} \exp(-\tau_{20}\alpha + \tau_{10} \log \alpha - \cancel{\log \Gamma(\tau_{10})} + \cancel{\tau_{10} \log \tau_{20}}) \right] \\ &= \left[-\tau_{20} + \sum_{m=1}^{T-1} \log(1 - v_m) \right] \alpha + \cancel{\sum_{t=1}^{N-1} (1 - v_m)} + \log \alpha (\tau_{10} + T - 1) - K \end{aligned} \quad (\text{E.63})$$

Therefore,

$$p(\alpha| -) = \text{Gamma}(\tau_1, \tau_2) \quad (\text{E.64})$$

$$\tau_1 = \tau_{10} + T - 1 \quad (\text{E.65})$$

$$\tau_2 = \tau_{20} - \sum_{m=1}^{T-1} \log(1 - v_m) \quad (\text{E.66})$$

Useful moments in VB updates for other model parameters:

$$\langle \alpha \rangle = \frac{\tau_1}{\tau_2} \quad (\text{E.67})$$

E.10 Variational Posterior on C

It was derived in Section that the NFE is maximized by a variational posterior, $q(\mathbf{C})$, that is proportional to the variational expectation of the true log-posterior with respect to all other model parameters:

$$q(\mathbf{C}) \propto \langle \log p(\mathbf{C}|-) \rangle \quad (\text{E.68})$$

The true log-posterior may be calculated from Bayes' theorem:

$$\log p(\mathbf{C}|-) \propto \log p(\mathbf{T}, \mathbf{X}^{(C)}|\mathbf{C}, -) + \log p(\mathbf{C}) - K, \quad (\text{E.69})$$

where K denotes a normalizing constant. The posterior will also be multinomial with parameters (responsibilities) Φ :

$$\begin{aligned} \log p(c_{nm} = 1|-) &= \log \rho_{nm} \\ &\propto \log \sigma(\xi_{nm}) + \frac{1}{2}(\gamma_{nm} - \xi_{nm}) - \lambda(\xi_{nm})(\gamma_{nm}^2 - \xi_{nm}^2) - \cancel{\frac{D^{(C)}}{2} \log 2\pi} - \frac{1}{2} \log |\Lambda_m^{-1}| \\ &\quad - \frac{1}{2}(\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m)^T \Lambda_m (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m) + \log v_m + \sum_{l < m} \log(1 - v_l) \\ &\propto \log \sigma(\xi_{nm}) + \frac{1}{2}([2t_n - 1] \mathbf{w}_m^T \mathbf{x}_n^{(T)} - \xi_{nm}) - \lambda(\xi_{nm}) \left(\mathbf{x}_n^{(T)T} \mathbf{w}_m \mathbf{w}_m^T \mathbf{x}_n^{(T)} - \xi_{nm}^2 \right) \\ &\quad + \frac{1}{2} \log |\Lambda_m| - \frac{1}{2}(\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m)^T \Lambda_m (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m) + \log v_m + \sum_{l < m} \log(1 - v_l) \end{aligned} \quad (\text{E.70})$$

Therefore,

$$p(\mathbf{c}_n|-) = \text{Multinomial}(\boldsymbol{\phi}_n) \quad (\text{E.71})$$

Useful moments in VB updates for other model parameters:

E.11 Negative Free Energy

The NFE can be expressed as the difference between the expected log-likelihood and the Kullback-Leibler divergence (KLD) between the variational posteriors and the priors:

$$\begin{aligned}
\mathcal{F} &= \langle \log p(\mathbf{t}, \mathbf{X}^{(C)} | -) \rangle - \mathbb{KLD} [q(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{W}, \boldsymbol{\beta}, \mathbf{C}, \mathbf{v}, \boldsymbol{\alpha}) || p(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{W}, \boldsymbol{\beta}, \mathbf{C}, \mathbf{v}, \boldsymbol{\alpha})] \\
&= \langle \log p(\mathbf{t} | \mathbf{W}, \mathbf{X}^{(T)}) \rangle + \langle \log p(\mathbf{X}^{(C)} | \boldsymbol{\mu}, \boldsymbol{\Lambda}, \mathbf{C}) \rangle - \sum_{m=1}^T \mathbb{KLD} [q(\mathbf{w}_m) || p(\mathbf{w}_m | \boldsymbol{\beta}_m)] \\
&\quad - \sum_{m=1}^T \sum_{p=1}^P \mathbb{KLD} [q(\beta_{md}) || p(\beta_{md})] - \sum_{n=1}^N \mathbb{KLD} [q(\mathbf{c}_n) || p(\mathbf{c}_n | \mathbf{v})] \\
&\quad - \sum_{m=1}^T \mathbb{KLD} [q(\boldsymbol{\rho}_m | \boldsymbol{\Lambda}_m) q(\boldsymbol{\Lambda}_m) || p(\boldsymbol{\rho}_m | \boldsymbol{\Lambda}_m) p(\boldsymbol{\Lambda}_m)] \\
&\quad - \sum_{m=1}^T \mathbb{KLD} [q(v_m) || p(v_m)] - \mathbb{KLD} [q(\alpha) || p(\alpha)] \\
&= \sum_{n=1}^N \sum_{m=1}^T \langle c_{nm} \rangle \left[\log \sigma(\xi_{nm}) + \frac{1}{2} (\langle \gamma_{nm} \rangle - \xi_{nm}) - \lambda(\xi_{nm}) (\langle \gamma_{nm}^2 \rangle - \xi_{nm}^2) \right] \\
&\quad - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T \langle c_{nm} \rangle \left[D \log 2\pi - \langle \log |\boldsymbol{\Lambda}_m| \rangle + \langle (\mathbf{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Lambda}_m (\mathbf{x} - \boldsymbol{\mu}_m) \rangle \right] \\
&\quad - \sum_{m=1}^T \mathbb{KLD} [q(\mathbf{w}_m) || p(\mathbf{w}_m | \boldsymbol{\beta}_m)] - \sum_{m=1}^T \sum_{p=1}^P \mathbb{KLD} [q(\beta_{md}) || p(\beta_{md})] \\
&\quad - \sum_{n=1}^N \mathbb{KLD} [q(\mathbf{c}_n) || p(\mathbf{c}_n | \mathbf{v})] - \sum_{m=1}^T \mathbb{KLD} [q(\boldsymbol{\rho}_m | \boldsymbol{\Lambda}_m) q(\boldsymbol{\Lambda}_m) || p(\boldsymbol{\rho}_m | \boldsymbol{\Lambda}_m) p(\boldsymbol{\Lambda}_m)] \\
&\quad - \sum_{m=1}^T \mathbb{KLD} [q(v_m) || p(v_m)] \\
&\quad - \mathbb{KLD} [q(\alpha) || p(\alpha)]
\end{aligned} \tag{E.72}$$

$$\begin{aligned}
&= \sum_{n=1}^N \langle c_{nm} \rangle \left[\log \sigma(\xi_i) + \frac{1}{2} \left[(2t_n - 1) \mathbf{x}_n^{(T)T} \langle \mathbf{w}_m \rangle - \xi_{nm} \right] \right. \\
&\quad \left. - \lambda(\xi_{nm}) \left(\mathbf{x}_n^{(T)T} \langle \mathbf{w}_m \mathbf{w}_m^T \rangle \phi(\mathbf{x}_n) - \xi_{nm}^2 \right) \right] \\
&\quad - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^T \langle c_{nm} \rangle \left[D \log 2\pi - \langle \log |\mathbf{\Lambda}_m| \rangle + \langle (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m)^T \mathbf{\Lambda}_m (\mathbf{x}_n^{(C)} - \boldsymbol{\mu}_m) \rangle \right] \\
&\quad - \sum_{m=1}^T \text{KLD} [q(\mathbf{w}_m) || p(\mathbf{w}_m | \boldsymbol{\beta}_m)] - \sum_{m=1}^T \sum_{p=1}^P \text{KLD} [q(\beta_{md}) || p(\beta_{md})] \\
&\quad - \sum_{n=1}^N \text{KLD} [q(\mathbf{c}_n) || p(\mathbf{c}_n | \mathbf{v})] - \sum_{m=1}^T \text{KLD} [q(\boldsymbol{\rho}_m | \mathbf{\Lambda}_m) q(\mathbf{\Lambda}_m) || p(\boldsymbol{\rho}_m | \mathbf{\Lambda}_m) p(\mathbf{\Lambda}_m)] \\
&\quad - \sum_{m=1}^T \text{KLD} [q(v_m) || p(v_m)] - \text{KLD} [q(\alpha) || p(\alpha)]
\end{aligned} \tag{E.73}$$

where $\text{KLD} [q(\mathbf{w}_m) || p(\mathbf{w}_m | \boldsymbol{\beta}_m)]$ is a KLD between two Gaussian distributions, $\text{KLD} [q(\beta_{md}) || p(\beta_{md})]$ is a KLD between two Gamma distributions, $\text{KLD} [q(\mathbf{c}_n) || p(\mathbf{c}_n | \mathbf{v})]$ is a KLD between two multinomial distributions, $\text{KLD} [q(\boldsymbol{\rho}_m | \mathbf{\Lambda}_m) q(\mathbf{\Lambda}_m) || p(\boldsymbol{\rho}_m | \mathbf{\Lambda}_m) p(\mathbf{\Lambda}_m)]$ is a KLD between two Normal-Wishart distributions, $\text{KLD} [q(v_m) || p(v_m)]$ is a KLD between two Beta distributions, and $\text{KLD} [q(\alpha) || p(\alpha)]$ is a KLD between two Gamma distributions.

Bibliography

- [1] International Campaign to Ban Landmines, “Fact sheet: Impact of mines/ERW on children,” in *Tenth Meeting of States Parties to the Mine Ban Treaty*, (Geneva), November 2010.
- [2] Joint IED Defeat Organization, *Annual Report: Fiscal Year 2010*. United States Department of Defense, 2010.
- [3] Joint IED Defeat Organization, *Homemade Explosive (HME) / Bulk Explosive (BE) Recognition Guide*. United States Department of Defense, 2006.
- [4] International Campaign to Ban Landmines, “Fact sheet: Victim-activated IED casualties,” in *Intersessional Standing Committee Meetings of the Mine Ban Treaty*, (Geneva), June 2011.
- [5] O. Merrouche, “Economic consequences of wars: Evidence from landmine contamination in Mozambique,” *European University Institute Working Papers*, vol. 22, 2006.
- [6] M. Sato, “GPR evaluation test for humanitarian demining in Cambodia,” in *2010 IEEE International Geoscience and Remote Sensing Symposium (IGARSS '10)*, (Honolulu, HI), pp. 4322–4325, 2010.
- [7] K. Sherbondy, “Status of vehicular mounted mine detection (VMMD) program,” in *Second International Conference on the Detection of Abandoned Land Mines*, pp. 203–207, October 1998.
- [8] United States Department of Agriculture, “Soil map of Afghanistan.” Available: <http://soils.usda.gov/use/worldsoils/mapindex/afghanistan-soil.html>, November November 2001.
- [9] United States Geological Survey, “The Afghanistan agrometeorology monthly/seasonal bulletin (2009-2010 agricultural season).” Available: <http://afghanistan.cr.usgs.gov/agrometeorology-publications-maps>, 2010.

- [10] D. J. Daniels, *Ground Penetrating Radar*. London: Institution of Electrical Engineers, 2004.
- [11] L. Peters Jr., J. J. Daniels, and J. D. Young, "Ground penetrating radar as a subsurface environmental sensing tool," *Proceedings of the IEEE*, vol. 82, no. 12, pp. 1802–1822, 1994.
- [12] L. He, S. Ji, W. Scott, and L. Carin, "Adaptive multimodality sensing of landmines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 6, pp. 1756–1774, 2007.
- [13] Y. Liao, L. W. Nolte, and L. M. Collins, "Decision fusion of ground-penetrating radar and metal detector algorithms: A robust approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 2, pp. 398–409, 2007.
- [14] NIITEK, Inc. personal communication, 2008.
- [15] K. J. Hintz, "SNR improvements in NIITEK ground-penetrating radar," in *Proceedings of the SPIE: Detection and Remediation Technologies for Mines and Minelike Targets IX*, vol. 5415, (Orlando, FL), p. 399, 2004.
- [16] G. C. Topp, J. L. Davis, and A. P. Annan, "Electromagnetic determination of soil water content: measurements in coaxial transmission lines," *Water Resources Research*, vol. 16, no. 3, pp. 574–582, 1980.
- [17] T. W. Miller, B. Borchers, J. M. H. Hendrickx, S. H. Hong, H. A. Lensen, P. B. W. Schwering, and J. B. Rhebergen, "Effect of soil moisture on land mine detection using ground penetrating radar," in *Proceedings of the SPIE: Detection and Remediation Technologies for Mines and Minelike Targets VII*, vol. 4742, (Orlando, FL), pp. 281–290, 2002.
- [18] T. W. Miller, J. M. H. Hendrickx, and B. Borchers, "Radar detection of buried landmines in field soils," *Vadose Zone Journal*, vol. 3, no. 4, p. 1116, 2004.
- [19] B. Borchers, J. M. H. Hendrickx, B. S. Das, and S. H. Hong, "Enhancing dielectric contrast between land mines and the soil environment by watering: modeling, design, and experimental results," in *Proceedings of the SPIE: Detection and Remediation Technologies for Mines and Minelike Targets V*, (Orlando, FL), pp. 993–1000, 2000.
- [20] K. Takahashi, H. Preetz, and J. Igel, "Performance of demining sensors and soil properties," in *Proceedings of the SPIE: Detection and Sensing of Mines*,

- Explosive Objects, and Obscured Targets XVI*, vol. 8017, (Orlando, FL), April 2011.
- [21] C. Ratto, K. Morton, L. Collins, and P. Torrione, “A Bayesian method for discriminative context-dependent fusion of GPR-based detection algorithms,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII*, vol. 8357-76, (Baltimore, MD), April 2012.
 - [22] L. Gurel and U. Oguz, “Simulations of ground-penetrating radars over lossy and heterogeneous grounds,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 6, pp. 1190–1197, 2001.
 - [23] G. M. Milner, “Random GPR antennae height variations and mine detection performance,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVI*, vol. 8017, (Orlando, FL), April 2011.
 - [24] T. Dogaru and L. Carin, “Time-domain sensing of targets buried under a rough air-ground interface,” *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 3, pp. 360–372, 1998.
 - [25] C. Rappaport and M. El-Shenawee, “Modeling GPR signal degradation from random rough ground surface,” in *2000 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2000)*, vol. 7, pp. 3108–3110, 2000.
 - [26] M. El-Shenawee and C. M. Rappaport, “Quantifying the effects of different rough surface statistics for mine detection using the FDTD technique,” in *Proceedings of the SPIE Detection and Remediation Technologies for Mines and Minelike Targets V*, vol. 4038, pp. 966–975, 2000.
 - [27] R. Firoozabadi, E. L. Miller, C. M. Rappaport, and A. W. Morgenthaler, “Subsurface sensing of buried objects under a randomly rough surface using scattered electromagnetic field data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 1, p. 104, 2007.
 - [28] A. Giannopoulos, “Numerical modelling of ground-penetrating radar response from rough subsurface interfaces,” *Near Surface Geophysics*, vol. 6, pp. 357–359, 2008.
 - [29] T. B. Hansen, P. M. Johansen, S. D. Res, and C. T. Ridgefield, “Inversion scheme for ground penetrating radar that takes into account the planar air-soil interface,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 1, p. 496–506, 2000.

- [30] F. Roth, P. van Genderen, and M. Verhaegen, "Convolutional models for buried target characterization with ground penetrating radar," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 11, pp. 3799–3810, 2005.
- [31] O. Lopera, E. C. Slob, N. Milisavljevic, and S. Lambot, "Filtering soil surface and antenna effects from GPR data to enhance landmine detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 3, p. 707, 2007.
- [32] P. Meincke, "Linear GPR inversion for lossy soil and a planar air-soil interface," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 12, p. 2713–2721, 2001.
- [33] S. Lambot, E. C. Slob, I. van den Bosch, B. Stockbroeckx, and M. Vanclooster, "Modeling of ground-penetrating radar for accurate characterization of subsurface electric properties," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 11, p. 2555–2568, 2004.
- [34] M. B. Kowalsky, S. Finsterle, J. Peterson, S. Hubbard, Y. Rubin, E. Majer, A. Ward, and G. Gee, "Estimation of field-scale soil hydraulic and dielectric parameters through joint inversion of GPR and hydrological data," *Water Resources Research*, vol. 41, p. 11425, 2005.
- [35] H. Brunzell, "Detection of shallowly buried objects using impulse radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 875–886, 1999.
- [36] K. Ho and P. Gader, "A linear prediction land mine detection algorithm for hand held ground penetrating radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 6, pp. 1374–1484, 2002.
- [37] M. Yoldemir, A.B.; Sezgin, "Adaptive linear prediction based buried object detection with varying detector height," in *13th International Conference on Ground Penetrating Radar (GPR)*, (Lecce, Italy), pp. 1 – 4, June 2010.
- [38] P. A. Torrione, C. S. Throckmorton, and L. M. Collins, "Performance of an adaptive feature-based processor for a wideband ground penetrating radar system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 2, p. 644, 2006.
- [39] P. Torrione, *Statistical Algorithms for Landmine Detection in Ground-Penetrating Radar Data*. PhD thesis, Duke University, 2008.

- [40] W. Ng, T. Chan, H. So, and K. Ho, "Particle filtering based approach for landmine detection using ground penetrating radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3739–3755, 2008.
- [41] J. N. Wilson, P. Gader, W. H. Lee, H. Frigui, and K. C. Ho, "A large-scale systematic evaluation of algorithms using ground-penetrating radar for landmine detection and discrimination," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 8, pp. 2560–2572, 2007.
- [42] P. D. Gader and M. Y. Zhao, "Landmine detection with ground penetrating radar using hidden Markov models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 6, pp. 1231–1244, 2001.
- [43] H. Frigui and P. Gader, "Detection and discrimination of land mines based on edge histogram descriptors and fuzzy k -nearest neighbors," in *IEEE International Conference on Fuzzy Systems*, pp. 1494–1499, 2006.
- [44] H. Frigui and P. Gader, "Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic k -nearest neighbor classifier," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 185–199, 2009.
- [45] P. Gader, W. H. Lee, and J. N. Wilson, "Detecting landmines with ground-penetrating radar using feature-based rules, order statistics, and adaptive whitening," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 11, pp. 2522–2534, 2004.
- [46] W. H. Lee, P. D. Gader, J. N. Wilson, N. Inc, and V. A. Sterling, "Optimizing the area under a receiver operating characteristic curve with application to landmine detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 2, pp. 389–397, 2007.
- [47] P. Torricione and L. Collins, "Texture features for antitank landmine detection using ground penetrating radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 7, pp. 2374–2382, 2007.
- [48] H. Frigui, K. C. Ho, and P. Gader, "Real-time landmine detection with ground-penetrating radar using discriminative and adaptive hidden Markov models," *EURASIP Journal on Applied Signal Processing*, no. 12, pp. 1867–1885, 2005.
- [49] K. Ho, L. Carin, P. Gader, and J. Wilson, "An investigation of using the spectral characteristics from ground penetrating radar for landmine/clutter discrimination," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 4, pp. 1177–1191, 2008.

- [50] K. C. Ho, P. D. Gader, J. N. Wilson, and H. Frigui, "On improving sub-space spectral feature technique for the detection of weak scattering plastic antitank landmines," in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIV*, vol. 7303, (Orlando, FL), p. 73032D, SPIE, 2009.
- [51] E. Pasolli, F. Melgani, and M. Donelli, "Automatic analysis of GPR images: A Pattern-Recognition approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2206–2217, 2009.
- [52] R. Stanley, P. Gader, and K. Ho, "Feature and decision level sensor fusion of electromagnetic induction and ground penetrating radar sensors for landmine detection with hand-held units," *Information Fusion*, vol. 3, pp. 215–223, 2002.
- [53] K. Ho, L. Collins, L. Huettel, and P. Gader, "Discrimination mode processing for EMI and GPR sensors for hand-held land mine detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 1, pp. 249–263, 2004.
- [54] W. Scott, K. Kim, G. Larson, A. Gurbuz, and J. McClellan, "Combined seismic, radar, and induction sensor for landmine detection," *Journal of the Acoustic Society of America*, vol. 123, no. 5, p. 3042, 2008.
- [55] H. Frigui, P. D. Gader, and A. C. B. Abdallah, "A generic framework for context-dependent fusion with application to landmine detection," in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIII*, vol. 6953, (Orlando, FL), p. 69531F, 2008.
- [56] L. W. Barsalou, "Context-independent and context-dependent information in concepts," *Memory & Cognition*, vol. 10, no. 1, pp. 82–93, 1982.
- [57] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, p. 69–101, 1996.
- [58] A. Tsymbal, "The problem of concept drift: definitions and related work," *Computer Science Department, Trinity College Dublin*, 2004.
- [59] K.-F. Lee, "Context-dependent phonetic hidden Markov models for speaker-independent continuous speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, pp. 599–609, April 1990.
- [60] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 257–265, May 1997.

- [61] J. Fritsch, M. Finke, and A. Waibel, "Context-dependent hybrid HME/HMM speech recognition using polyphone clustering decision trees," in *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, (Munich, Germany), pp. 1759–1762, 1997.
- [62] Q. Jackson and D. Landgrebe, "Adaptive Bayesian contextual classification based on markov random fields," in *2002 IEEE International Geoscience and Remote Sensing Symposium (IGARSS '02)*, vol. 3, pp. 1422–1424 vol.3, 2002.
- [63] L. Bruzzone and L. Carlin, "A multilevel Context-Based system for classification of very high spatial resolution images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 44, no. 9, pp. 2587–2600, 2006.
- [64] H. Frigui, L. Zhang, and P. Gader, "Context-dependent multisensor fusion and its application to land mine detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 6, pp. 2528–2543, 2010.
- [65] J. Bolton and P. Gader, "Random set framework for context-based classification with hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3810–3821, 2009.
- [66] D. Stoyan, W. Kendall, and J. Mecke, *Stochastic Geometry and its Applications*. West Sussex, U.K.: Wiley, 2 ed., 1995.
- [67] D. Blei and M. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121–144, 2006.
- [68] Z. Ghahramani and M. J. Beal, "Variational inference for Bayesian mixtures of factor analyzers," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, pp. 449–455, 2000.
- [69] J. Paisley and L. Carin, "Hidden Markov models with stick-breaking priors," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3905–3917, 2009.
- [70] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.
- [71] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [72] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: John Wiley & Sons, Inc., 2 ed., 2001.

- [73] A. Giannopoulos, “GprMax: A ground penetrating radar simulation tool.” Available: <http://www.gpr-max.org>, 2002.
- [74] A. Giannopoulos, “Modelling ground penetrating radar by GprMax,” *Construction and Building Materials*, vol. 19, no. 10, p. 755–762, 2005.
- [75] M. Skolnik, *Radar Handbook*. New York: McGraw-Hill, 2 ed., 1990.
- [76] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on signal processing*, vol. 41, no. 12, p. 3397–3415, 1993.
- [77] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River: Prentice Hall, 1996.
- [78] C. R. Ratto, P. A. Torrione, and L. M. Collins, “Estimation of soil permittivity through autoregressive modeling of time-domain ground-penetrating radar data,” in *IEEE International Conference on Wireless Information Technology and Systems (ICWITS)*, (Honolulu, HI), pp. 1–4, August 2010.
- [79] C. R. Ratto, K. Morton, P. A. Torrione, and L. M. Collins, “Contextual learning in ground-penetrating radar data using Dirichlet process priors,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVI*, vol. 8017, (Orlando, FL), pp. 8017–64, 2011.
- [80] C. R. Ratto, K. Morton, L. M. Collins, and P. A. Torrione, “Physics-based features for identifying contextual factors affecting landmine detection with ground-penetrating radar,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVI*, vol. 7303, (Orlando, FL), p. 730327, 2011.
- [81] K. Yee, “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media,” *IEEE Transactions on Antennas and Propagation*, vol. 14, pp. 302–307, 1966.
- [82] A. Taflovie and M. E. Brodwin, “Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell’s equations,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 23, no. 8, pp. 623–630, 1975.
- [83] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

- [84] C. M. Bishop and M. E. Tipping, “Variational relevance vector machines,” in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, (Stanford, CA), p. 46–53, 2000.
- [85] A. P. Jaganathan and E. N. Allouche, “Temperature dependence of dielectric properties of moist soils,” *Canadian Geotechnical Journal*, vol. 45, pp. 888–894, 2008.
- [86] C. R. Ratto, P. A. Torrione, and L. M. Collins, “Context-dependent feature selection for landmine detection with ground-penetrating radar,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIV*, vol. 7303, (Orlando, FL), p. 730327, 2009.
- [87] C. Ratto, “A context-dependent approach to landmine detection with ground-penetrating radar,” Master’s thesis, Duke University, Durham, NC, March 2009.
- [88] C. R. Ratto, P. A. Torrione, and L. M. Collins, “Exploiting Ground-Penetrating radar phenomenology in a context-dependent framework for landmine detection and discrimination,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, pp. 1689–1700, May 2011.
- [89] C. Ratto, P. Torrione, and L. Collins, “Context-dependent feature selection using unsupervised contexts applied to GPR-based landmine detection,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XV*, vol. 7664, (Orlando, FL), pp. 7664–2L, 2010.
- [90] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [91] H. Attias, “A variational Bayesian framework for graphical models,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, no. 1-2, pp. 209–215, 2000.
- [92] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1996.
- [93] L. Ayers and E. Rosen, “MIDAS: Mine detection assessment and scoring user’s manual v1.1,” tech. rep., Institute for Defense Analysis, Arlington, VA, 2004.
- [94] C. Wang, X. Liao, L. Carin, and D. B. Dunson, “Classification with incomplete data using Dirichlet process priors,” *Journal of Machine Learning Research*, vol. 11, pp. 3269–3311, 2010.

- [95] R. L. Winkler, *An Introduction to Bayesian Inference and Decision*. Gainesville, FL: Probabilistic, 2 ed., 2003.
- [96] H. Raiffa and R. Schlaiffer, *Applied Statistical Decision Theory*. New York: Wiley Classics Library, 2000.
- [97] T. S. Jaakkola and M. I. Jordan, “Bayesian parameter estimation via variational methods,” *Statistics and Computing*, vol. 10, no. 1, p. 25–37, 2000.
- [98] M. J. Beal, *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London, 2003.
- [99] J. T. Ormerod and M. P. Wand, “Explaining variational approximations,” *The American Statistician*, vol. 2, pp. 140–153, May 2010.
- [100] K. Morton, *Bayesian Techniques for Adaptive Acoustic Surveillance*. PhD thesis, Duke University, 2010.
- [101] K. Morton, P. A. Torriane, and L. M. Collins, “Variational Bayesian learning for mixture autoregressive models with uncertain-order,” *IEEE Transactions on Signal Processing*, vol. 59, pp. 2614–2627, June 2011.
- [102] R. P. Feynman, “Statistical mechanics: A set of lectures,” Reading, MA: Addison-Wesley, 1998.
- [103] T. Ferguson, “A Bayesian analysis of some nonparametric problems,” *The Annals of Statistics*, vol. 1, pp. 209–230, 1973.
- [104] D. Blackwell and J. B. MacQueen, “Ferguson distributions via Pólya urn schemes,” *The Annals of Statistics*, vol. 1, pp. 353–355, 1973.
- [105] J. Sethuraman, “A constructive definition of Dirichlet priors,” *Statistica Sinica*, vol. 1, pp. 639–650, 1994.
- [106] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 2001.
- [107] M. I. Jordan and R. A. Jacobs, “Hierarchical mixtures of experts and the EM algorithm,” *Neural Computation*, vol. 6, pp. 181–214, March 1994.
- [108] C. M. Bishop and M. Svensen, “Bayesian hierarchical mixtures of experts,” in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, (Acapulco, Mexico), p. 57–64, 2003.

- [109] X. Liao, H. Li, and L. Carin, “Quadratically gated mixture of experts for incomplete data classification,” in *Proceedings of the International Conference on Machine Learning*, (Corvallis, OR), pp. 533–560, 2007.
- [110] N. X. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: is a correction for chance necessary?,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, (Montreal, Quebec, Canada), pp. 1073–1080, ACM, 2009.
- [111] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Readings in speech recognition*, vol. 53, no. 3, p. 267–296, 1990.
- [112] Z. Ghahramani and M. I. Jordan, “Factorial hidden Markov models,” *Machine learning*, vol. 29, no. 2, p. 245–273, 1997.
- [113] Y. Qi, J. Paisley, and L. Carin, “Music analysis using hidden Markov mixture models,” *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5209–5224, 2007.
- [114] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, “An HDP-HMM for systems with state persistence,” in *Proceedings of the 25th International Conference on Machine Learning*, (Helsinki, Finland), p. 312–319, 2008.
- [115] J. Hu, M. K. Brown, and W. Turin, “HMM based on-line handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, p. 1039–1045, 1996.
- [116] U.-V. Marti and H. Bunke, “Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 1, 2001.
- [117] P. Runkle, L. Carin, L. Couchman, J. A. Bucaro, and T. J. Yoder, “Multiaspect identification of submerged elastic targets via wave-based matching pursuits and hidden Markov models,” *The Journal of the Acoustical Society of America*, vol. 106, p. 605, 1999.
- [118] T. P. Mann, “Numerically stable hidden Markov model implementation.” Available: <http://bozeman.genome.washington.edu/>, 2006.
- [119] C. Ratto, P. Torrione, K. Morton, and L. Collins, “Context-dependent landmine detection with ground-penetrating radar using a hidden Markov context

- model,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, (Honolulu, HI), pp. 4192–4195, 2010.
- [120] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen, “The infinite hidden markov model,” *Advances in Neural Information Processing Systems*, vol. 1, p. 577–584, 2002.
 - [121] C. R. Ratto, K. D. Morton, L. M. Collins, and P. A. Torrione, “A hidden Markov context model for GPR-based landmine detection incorporating Dirichlet process priors,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, (Vancouver, Canada), pp. 874–877, 2011.
 - [122] J. Benediktsson, J. Sveinsson, and K. Amason, “Classification and feature extraction of AVIRIS data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 5, pp. 1194–1205, 1995.
 - [123] P. Lucey, T. Williams, J. Hinrichs, M. E. Winter, D. Steutel, and E. Winter, “Three years of operation of AHI: the university of Hawaii’s airborne hyperspectral imager,” in *Proceedings of the SPIE: frared Technology and Applications XXVII* (B. F. Andresen, G. F. Fulop, and M. Strojnik, eds.), vol. 4369, (Orlando, FL), pp. 112–120, October 2001.
 - [124] I. Reed and X. Yu, “Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 10, pp. 1760–1770, 1990.
 - [125] L. Kirkland, K. Herr, E. Keim, P. Adams, J. Salisbury, J. Hackwell, and A. Treiman, “First use of an airborne thermal infrared hyperspectral scanner for compositional mapping,” *Remote Sensing of Environment*, vol. 80, pp. 447–459, June 2002.
 - [126] A. Zare, J. Bolton, P. Gader, and M. Schatten, “Vegetation mapping for landmine detection using long-wave hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, pp. 172–178, January 2008.
 - [127] G. Healey and D. Slater, “Invariant recognition in hyperspectral images,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, (Ft. Collins, CO), 1999.
 - [128] M. Chi, Q. Qian, and J. Benediktsson, “Cluster-Based ensemble classification for hyperspectral remote sensing images,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, (Boston, MA), pp. 209–212, July 2008.

- [129] R. Mayer, F. Bucholtz, and D. Scribner, "Object detection by using "whitening/dewhitening" to transform target signatures in multitemporal hyperspectral and multispectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 5, pp. 1136–1142, 2003.
- [130] P. Torricione, C. Ratto, and L. Collins, "Multiple instance and context dependent learning in hyperspectral data," in *First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing WHISPERS*, pp. 1–4, 2009.
- [131] K. D. Morton, Jr., P. A. Torricione, and L. M. Collins, "Dirichlet process based context learning for mine detection in hyperspectral imagery," in *Second Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing WHISPERS*, To appear.
- [132] C. Ratto, K. Morton, L. Collins, and P. Torricione, "A comparison of principal components and endmember-based contextual learning for hyperspectral anomaly classification," in *Third Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing WHISPERS*, (Lisbon, Portugal), pp. 1–4, IEEE, June 2011.
- [133] M. E. Winter, "N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Proceedings of SPIE: Imaging Spectrometry V*, vol. 3753, (Denver, CO), pp. 266–275, 1999.
- [134] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J. Huntington, "ICE: a statistical approach to identifying endmembers in hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 10, pp. 2085–2095, 2004.
- [135] A. Zare and P. Gader, "Sparsity promoting iterated constrained endmember detection in hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, pp. 446–450, July 2007.
- [136] A. Zare and P. Gader, "PCE: piecewise convex endmember detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 6, pp. 2620–2632, 2010.
- [137] USGS Spectroscopy Lab, "USGS digital spectral library." Available: <http://speclab.cr.usgs.gov/spectral-lib.html>, September 2007.
- [138] M. Sato, "Online model selection based on the variational Bayes," *Neural Computation*, vol. 13, pp. 1649–1681, 2001.

- [139] M. Hoffman, D. Blei, and F. Bach, “Online learning for latent Dirichlet allocation,” *Neural Information Processing Systems*, vol. 23, pp. 856–864, 2010.
- [140] R. Harmon, J. Remus, N. McMillan, C. McManus, L. Collins, J. Gottfreid, F. DeLucia, and A. Mizolek, “LIBS analysis of geomaterials: Geochemical fingerprinting for the rapid analysis and discrimination of minerals,” *Applied Geochemistry*, vol. 24, no. 6, pp. 1125–1141, 2009.
- [141] D. Alvey, K. M. Jr., R. Harmon, J. Gottfried, J. Remus, L. Collins, and M. Wise, “Laser-induced breakdown spectroscopy-based geochemical fingerprinting for the rapid analysis and discrimination of minerals: the example of garnet,” *Applied Optics*, vol. 49, no. 13, pp. 168–180, 2010.
- [142] P. Torrione, K. Morton, L. Collins, and C. Ratto, “Mitigating the effects of context on classification in libs spectra: CBRNE applications,” in *International Chemical Congress of Pacific Basin Societies (PACIFICHEM)*, December 2010.
- [143] D. J. Krusienski, E. W. Sellers, D. McFarland, T. Vaughan, and J. Wolpaw, “Toward enhanced P300 speller performance,” *Journal of Neuroscience Methods*, vol. 167, pp. 15–21, 2008.
- [144] K. Colwell, D. Ryan, S. T. and. K.D. Morton, E. Sellers, K. Caves, and L. Collins, “Jumpwise regression for channel selection in BCI: Longitudinal consistency and simulations,” in *2011 Neuroscience Meeting Planner*, vol. 593.04/NN25, (Washington, DC), Society for Neuroscience, 2011.
- [145] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburg, and A. H. Byers, “Big data: The next frontier for innovation, competition, and productivity,” tech. rep., McKinsey Global Institute, May 2011.
- [146] C. Thompson, “If you liked this, you’re sure to love that,” *The New York Times Magazine*, p. MM74, November 21 2008.
- [147] M. Mandel, G. Poliner, and D. Ellis, “Support vector machine active learning for music retrieval,” *Multimedia Systems*, vol. 12, pp. 3–13, 2006.
- [148] C. Ratto, K. Morton, L. Collins, and P. Torrione, “Characterization of the subsurface environment with GPR using feature-based statistical learning,” *IEEE Transactions on Geoscience and Remote Sensing*, In Review.
- [149] C. Ratto, P. Torrione, and L. Collins, “Context-dependent feature selection for classification of simulated ground-penetrating radar data,” in *Quantitative*

Methods in Defense and National Security, (Durham, NC), National Institute for Statistical Sciences, May 2008.

- [150] C. Ratto, P. Torrione, and L. Collins, “Physics-based context identification of ground-penetrating radar data,” in *UXO/Countermines/Range Forum*, (Orlando, FL), August 2009.
- [151] C. Ratto, P. Torrione, and L. Collins, “Applications of context-dependent learning to countermines and IED defeat,” in *Military Sensing Symposium (MSS) Battlefield Survivability and Discrimination (BSD)*, February 2011.
- [152] C. Ratto, K. Morton, L. Collins, and P. Torrione, “Characterization of subsurface environmental factors from ground-penetrating radar data using statistical learning algorithms,” in *Novel Methods for Subsurface Characterization and Monitoring (NovCare)*, (May), 2011.
- [153] P. Torrione, K. Morton, C. Ratto, and L. Collins, “Vehicle mounted video-based change detection for novel anomaly detection,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVI*, vol. 8017-75, (Orlando, FL), April 2011.
- [154] P. Torrione, K. Morton, C. Ratto, and L. Collins, “Change detection for detecting potential threats in forward looking video data,” in *Military Sensing Symposium (MSS) Battlefield Survivability and Discrimination (BSD)*, February 2011.
- [155] C. Ratto, K. Morton, I. McMichael, B. Burns, W. Clark, L. Collins, and P. Torrione, “Integration of LIDAR with the NIITEK GPR for improved performance on rough terrain,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII*, vol. 8357-67, (Baltimore, MD), April 2012.
- [156] K. Morton, C. Ratto, L. Collins, and P. Torrione, “Change based threat detection in urban environments with a forward looking camera,” in *Proceedings of the SPIE: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII*, vol. 8357-59, (Baltimore, MD), April 2012.
- [157] C. Ratto, P. A. Torrione, and L. M. Collins, *Signal and Image Processing for Remote Sensing*, ch. Context Dependent Classification: An Approach for Achieving Robust Remote Sensing Performance in Changing Conditions, pp. 91–114. CRC Press, 2 ed., 2012.

Biography

Christopher Ralph Ratto was born on August 27, 1985 in Mineola, NY. He was raised in Floral Park, NY and graduated from Floral Park Memorial High School in 2003. He graduated *summa cum laude* from The Catholic University of America (CUA) in Washington, DC with a B.E.E. degree in 2007. He then received the M.S. degree in electrical and computer engineering from Duke University in Durham, NC in 2009. His M.S. thesis, “A Context-Dependent Approach to Landmine Detection with Ground-Penetrating Radar” [87], was completed under the supervision of Prof. Leslie M. Collins. Dr. Ratto received the Ph.D. degree from Duke University in 2012. His dissertation, “Nonparametric Bayesian Context Learning for Buried Threat Detection,” was also completed under the supervision of Prof. Leslie M. Collins.

Dr. Ratto’s research interests include machine learning, Bayesian statistics, digital signal/image processing, and electromagnetic modeling. He is particularly interested in applications for remote sensing, pattern recognition, image exploitation, and data mining. At the time of this dissertation’s publication, Dr. Ratto is has authored two refereed journal articles [88, 148], 20 conference proceedings [21, 78–80, 86, 89, 119, 121, 130, 132, 142, 149–156], and a book chapter [157].

Dr. Ratto was awarded a William H. Gardner, Jr. fellowship by the Pratt School of Engineering at Duke University for the 2007-2008 academic year. From 2007-2011, he was also a James B. Duke fellow and a member of the Society of Duke Fellows. Dr. Ratto is a member of the Tau Beta Pi and Phi Eta Sigma honor societies, and

is also an Eagle Scout.

Dr. Ratto is also an accomplished classical musician. While attending CUA, he was awarded a scholarship to pursue a minor at the Benjamin T. Rome School of Music. He concentrated in cello performance, and performed with the CUA Symphony Orchestra and various other ensembles in the Washington, DC metropolitan area. He was also a member of the Duke Symphony Orchestra for all ten semesters while working on his Ph.D.