



**DYNAMIC PROTOTYPE ADDITION IN
GENERALIZED LEARNING VECTOR QUANTIZERS**

DISSERTATION

Jonathon R. Climer, Captain, USAF

AFIT-ENG-DS-17-S-009

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-DS-17-S-009

DYNAMIC PROTOTYPE ADDITION IN
GENERALIZED LEARNING VECTOR QUANTIZERS

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Doctoral of Philosophy in Electrical Engineering

Jonathon R. Climer, B.S.E.E., M.S.E.E.

Captain, USAF

September 2017

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE

AFIT-ENG-DS-17-S-009

DYNAMIC PROTOTYPE ADDITION IN
GENERALIZED LEARNING VECTOR QUANTIZERS

Jonathon R. Climer, B.S.E.E., M.S.E.E.
Captain, USAF

Committee Membership:

Michael J. Mendenhall, PhD
Co-Chair

Gilbert L. Peterson, PhD
Co-Chair

Lieutenant Colonel Ryan D. Kappedal, PhD
Member

ADEDEJI B. BADIRU, PhD
Dean, Graduate School of Engineering and Management

Abstract

The classification accuracy achieved by a statistical machine learning algorithm is determined by selecting the function(s) that most closely match a domain that results in the correct labeling of new samples, while not over-fitting. This is termed the accuracy/generalization trade-off. In Learning Vector Quantization (LVQ) the underlying function is a set of vectors in the domain space that have a relationship with each other and are prototypical of the underlying data. Where the prototypes are and how many there are influences the piece-wise linear decision boundaries they create and hence the final accuracy of the resulting LVQ. This work develops a novel framework that includes the LVQ learning components of Competition, Winner Selection, and Synaptic Adaptation, and adds a new component Network Structure Modification (NSM) that allows for experimentation of four different prototype addition strategies: simple, cost-minimizing, clustering, and cost-minimizing/clustering hybrids. Within these strategies, seven novel methods are tested on data sets using the Generalized Relevance LVQ Improved (GRLVQI) algorithm. Results show that cost-minimizing strategies achieve the highest classification accuracy but do not generalize as well, clustering methods tend to generalize well but are less accurate, and hybrid strategies provide the best trade-off between accuracy and generalization.

Acknowledgements

I would like to thank my research advisor, Dr. Michael Mendenhall, for his unceasing support and confidence in me. I appreciate his dedication and commitment, going above and beyond to guide me throughout the PhD process. I am also thankful for the insight and aid from Dr. Gilbert Peterson and Lt Col Ryan Kappedal to strengthen my document and bring my AFIT experience to a successful conclusion.

My fellow students have been a constant source of encouragement and camaraderie as we have struggled through ups and downs together. I am grateful for their humor and their examples as friends and leaders. There are numerous faculty and staff members who have kept me on track and given me the tools to excel and succeed. Their love for students has been evident, and their help has always been accompanied with a smile.

I am grateful to my parents, Tom and Carol, for raising me to be curious and to work hard. Their love and pride in my efforts has motivated me throughout life. I appreciate the support and prayers from numerous family members and friends. A deep thanks goes to my sister-in-law, Hannah Baird, and friends Jed and Kera Butler who provided sustained support that allowed me to focus on research after the birth of my twins. Their devotion and capacity to give was instrumental during a critical time in my research.

The successful completion of my PhD would not have been possible without my family. The dedication and sacrifice of my wife, Mary Ann, and our five wonderful children have made this journey possible. They have been my biggest cheerleaders and helped me to achieve my dreams. Ultimately, I am grateful to my Creator, who has blessed me with talents, sustained me in body and spirit, and brought people into my path that have helped in unimaginable ways.

Jonathon R. Climer

Table of Contents

	Page
Abstract	iv
Acknowledgements	v
Table of Contents	vi
List of Figures	x
List of Tables	xxv
 I. Introduction	 1
1.1 Problem Statement	2
1.1.1 Contributions	2
1.1.2 Results	3
1.2 Document Outline	3
 II. Background & Literature Review	 5
2.1 Learning Vector Quantization (LVQ)	5
2.1.1 Competitive Learning in LVQ	8
2.1.2 LVQ Variants	9
2.2 Dynamic LVQ (DLVQ)	14
2.2.1 Mean	15
2.2.2 Closest	15
2.2.3 Principal Components	16
2.2.4 Sampling Cost	21
2.3 Generalization in LVQ Networks	21
2.4 Measuring Performance	25
2.5 Summary	26
 III. Methodology	 27
3.1 Framework for DLVQs	27
3.1.1 Region Identification	28
3.1.2 Prototype Insertion	30
3.2 Novel Network Structure Modification (NSM) Methods	31
3.2.1 Simple: Sufficiency and Efficiency (Near-Mean)	31

	Page
3.2.2 Cost-Minimizing	32
3.2.2.1 Misclassified Cost	32
3.2.2.2 Average Misclassified Cost	36
3.2.2.3 Sampling with average Misclassified Cost	38
3.2.3 Clustering	38
3.2.4 Hybrid	42
3.2.4.1 Silhouette-Full	42
3.2.4.2 Silhouette-Lite	46
3.2.5 NSM Discussion	46
3.3 Novel Evaluation and Characterization of DLVQs	49
3.3.1 Accuracy-Generalization Surface (AGS)	49
3.3.2 Prototype Utilization Plane	50
3.3.3 Generalization Plane	51
3.3.4 Empirical Margin Error Plane	52
3.4 Summary	53
IV. Results	54
4.1 Data Sets	54
4.1.1 United States Postal Service (USPS) Data Set	54
4.1.2 Mice Protein Expression (MICE) Data Set	55
4.1.3 Z-Wave wireless protocol (ZWAVE) Data Set	56
4.1.4 Lunar Crater Volcanic Field (LCVF) Data Set	58
4.2 Experimental Setup	59
4.3 Results and Analysis	61
4.3.1 Classification Accuracy	62
4.3.2 Prototype Utilization	62
4.3.3 Margin Analysis	66
4.3.4 AGS Comparisons	69
4.3.4.1 USPS	71
4.3.4.2 MICE	72
4.3.4.3 ZWAVE	74
4.3.4.4 LCVF	75
4.3.4.5 Data Set and NSM Method Interactions	77
4.4 Results Summary	78
V. Summary and Discussion	80
5.1 Contributions	80
5.2 Future Work	82

	Page
Appendix A: Supplemental Data Set Analysis	85
A.1 Combined Charts and Analysis	85
A.1.1 Averaged Time Series Results	85
A.1.2 Performance of NSM Methods over AGS	89
A.2 USPS Generalized Relevance LVQ Improved (GRLVQI) Results and Discussion	90
A.2.1 Parameters and Data	92
A.2.2 Feature Covariance and Correlation	92
A.2.3 Averaged Performance as a Function of Training	94
A.2.4 Baseline and Converged Model Performance	95
A.2.5 Performance with respect to Prototype Utilization	96
A.2.6 Margin Analysis	97
A.2.7 Relative Comparison	97
A.3 MICE GRLVQI Results and Discussion	105
A.3.1 Parameters and Data	105
A.3.2 Feature Covariance and Correlation	105
A.3.3 Averaged Performance as a Function of Training	107
A.3.4 Baseline and Converged Model Performance	107
A.3.5 Performance with respect to Prototype Utilization	108
A.3.6 Margin Analysis	109
A.3.7 Relative Comparison	110
A.4 ZWAVE GRLVQI Results and Discussion	120
A.4.1 Parameters and Data	120
A.4.2 Feature Covariance and Correlation	120
A.4.3 Averaged Performance as a Function of Training	121
A.4.4 Baseline and Converged Model Performance	122
A.4.5 Performance with respect to Prototype Utilization	124
A.4.6 Margin Analysis	124
A.4.7 Relative Comparison	125
A.5 LCVF GRLVQI Results and Discussion	130
A.5.1 Parameters and Data	130
A.5.2 Feature Covariance and Correlation	130
A.5.3 Averaged Performance as a Function of Training	131
A.5.4 Baseline and Converged Model Performance	133
A.5.5 Performance with respect to Prototype Utilization	133
A.5.6 Margin Analysis	134
A.5.7 Relative Comparison	134
Appendix B: Role of Prototype Insertion Phase	146

	Page
Appendix C: Limited M-Cost and Silhouette Performance with USPS	150
Bibliography	153

List of Figures

Figure		Page
2.1	LVQs iteratively adjust prototype positions based off of labeled samples and can be described by three phases, illustrated for an LVQ 2.1-based algorithm. 1) Competition: for a labeled sample, the LVQ algorithm identifies groups of in-class and out-of-class prototypes to compete. 2) Winner Selection: the in-class prototype, w^j, and out-of-class prototype, w^k, that result in the minimum distortion, d^j and d^k respectively, win and are updated. 3) Synaptic Adaptation: the positions of winning prototypes are updated according to Eqn. 2.1, where in-class prototypes are moved closer to the sample and out-of-class prototypes are moved further away (so-called differential shifting) [1, 2]. 	10
2.2	An abbreviated family tree of LVQ algorithms with dynamic variants located in the “leaves”.	11
2.3	Two prototype addition methods are illustrated, where samples are circles and prototypes are stars (new prototypes are four pointed). True class labels are indicated by color (blue, red, or green) and assigned class is determined by Voronoi tessellations of prototype positions (dotted lines). Dotted boxes surround misclassified samples. In both prototype addition methods, the region of red misclassified samples contains the highest proportion of misclassified samples, $M_{c'}$, to true class samples $M_{c^{all}}$ and is identified to receive an additional prototype. The location for new prototype insertion is determined as: (a) Mean: the average position of the red misclassified samples and (b) Closest: the sample with minimum distance between the average misclassified samples and their nearest foreign prototype (distance shown as grey lines).	18

Figure	Page
2.4 The space occupied by samples (points) and related prototypes (stars) are represented in two views. We show true labeled regions corresponding to class labels of $y = c$ or $\mathbf{w}^j$ (which are distinct from piecewise linear decision boundaries formed by labels of $z = c$ or $\mathbf{w}^{net}$). (a) Identification regions typically consist of entire class structures, represented as red, blue, and green shaded backgrounds. However, (b) further subdivides these classes into clusters that share a nearest <i>in-class-prototype</i> , $\mathbf{w}^q$. True cluster membership is represented in corresponding shades of red, blue, and green.	19
2.5 Two prototype addition methods are illustrated where samples are circles and prototypes are stars (new prototypes are four pointed). True class labels are indicated by color (blue, red, or green) and assigned class is determined by Voronoi tessellations of prototype positions (dotted lines). (a) Principal Component: The in-class samples nearest each prototype define potential regions for prototype addition. The red prototype yields the highest split metric and is replaced by two prototypes that are inserted equidistant along the eigenvector of the red samples whose nearest in-class prototype was just split. (b) Sampling Cost: A random subset of samples (surrounded by dotted hexagons) is generated and their cost is calculated over the union of the original prototype network and a candidate prototype position (selected from the subset). A new prototype is inserted at the candidate prototype that leads to the lowest subset cost.	23

Figure	Page
2.6 A simple two class case illustrates alternate margin definitions where blue and red stars are prototypes of opposing classes. (a) The sample margin, θ_s (dotted circle), is drawn around a sample point until it reaches the learned decision boundary (derived from the Voronoi tessellation) (b) The hypothesis margin, θ_h (dotted circle), is drawn around the nearest in-class and out-of-class prototype vectors to the sample and is calculated as half the difference between distance measures d^k and d^j . This margin calculation does not require the explicit locations of all learned decision boundaries. (Figures inspired by [3])	25
3.1 Dynamic Learning Vector Quantization (DLVQ) Framework.	28
3.2 Four region representations correspond to dynamic approaches of: simple, (targeted) cost-minimizing, clustering/hybridized, and (global) cost-minimizing. Prototypes are shown as stars and samples as dots. (a) Regional <i>class membership</i> is shown with shaded red, green, and blue backgrounds to represent the footprint of samples sharing the same class label $y = c$. (b) A Voronoi tessellation quantizes the space according to nearest prototype; shaded regions show predicted class or $z = c$ values. Misclassified samples such that $y \neq z$ populate either side of 2-class boundary regions (dark lines). (c) <i>Clusters</i> are subclass regions defined by samples sharing a common nearest in-class prototype, w . Class membership, $y = c$, is represented by various shades of red, blue, and green backgrounds. (d) A <i>random sampling</i> (dark points) over a global pool shows an abstract region, $\mathcal{X}_{sub}$, that includes multiple class memberships (shaded red, green, and blue backgrounds).	29

Figure	Page
3.3 A simple three class problem illustrates cost contributions of misclassified samples in a confusion matrix. (left) Samples are circles and prototypes are stars, with class membership, y^m , shown by color (blue, red, or green). Assigned or nearest class, z^m , is determined by the decision boundary (dotted lines); three misclassified samples are outlined with a dotted box. (right) Cost contributions (Eqn. 2.2) of the misclassified samples are asterisks in the confusion matrix as a proxy for their actual value. As the red-green boundary represents the greatest total cost for misclassified samples (Eqn. 2.2), they identify region to receive an additional prototype and the misclassified samples form a pool of candidate prototype positions.	34
3.4 An illustration of clustering: assigning labels (purple, orange, green) according to learned natural groupings. Similar techniques can be applied to redistribute prototypes according to natural segmentations within samples of a particular class.	39
3.5 An illustration of cluster metrics: clusters are evaluated based on their “goodness” or reliability. Measures such as inter-cluster (blue bounding geometries) and intra-cluster (black bidirectional arrows) distortions are used to assess the suitability of learned groupings.	43
3.6 An overall view of each NSM’s overall strategy is characterized over the axes of: (1) span of region influence, ranging from local to global and (2) focus of the insertion strategy, ranging from a focus on internal clustering to defining the decision boundary.	48
3.7 An Accuracy-Generalization Surface (AGS) view of NSM method’s characteristic performance over three reference planes represent the entire span of operation.	50

Figure	Page
4.1 Four random samples per class are shown as examples of the United States Postal Service (USPS) data set. Handwritten digits are re-oriented and normalized as 16×16 pixel gray-scale images with real valued intensities on $[0, 1]$	55
4.2 Expression levels in eight classes of mice are shown from the Mice Protein Expression (MICE) data set, with protein and modified protein names listed on the x -axis. Median values are shown with marker type indicating class.	56
4.3 Z-Wave wireless protocol (ZWAVE) signals are characterized by three statistical measures: variance, skewness, and kurtosis. Each column shows a random ZWAVE sample from each class distribution. For ease of viewing, features are plotted according to statistical measure (top, middle, and bottom rows corresponding to variance, skewness, and kurtosis) where the x -axis of each plot represents a different signal region.	57
4.4 Mean spectral reflectances of the 35-class Lunar Crater Volcanic Field (LCVF) data set are stacked to compare characteristic profiles. Spectral gaps correspond to water absorption regions. For historical reasons, classes 1-35 correspond to labels: A-W, X-Z, a-e, and g-j [4]. Image is used with permission from [5].	58
4.5 NSM methods are compared by prototype utilization. Solid horizontal lines show baseline performance with equal or more prototypes. (top) MICE with a maximum of 8×7 prototypes and (bottom) LCVF with a maximum of 3×35 prototypes.	65
4.6 Accuracy of the 11 NSM methods over prototype intervals of size 4 to 5 is shown for the 75th, 50th, and 25th percentiles (top, middle, bottom of vertical boxes) for (top) MICE and (bottom) LCVF.	66

Figure	Page
4.7 Margin error for MICE using Near-Mean, Avg-M-Cost, S-Cost, PC, and Silhouette-F NSM methods: (left) CGNT margin error verses θ_{cp} , (middle) HSV margin error verses θ_{cp} , and (right) HSV margin error versus CGNT margin error.	68
4.8 Margin error for LCVF using Near-Mean, Avg-M-Cost, S-Cost, PC, and Silhouette-F NSM methods: (left) CGNT margin error verses θ_{cp} , (middle) HSV margin error verses θ_{cp} , and (right) HSV margin error versus CGNT margin error.	70
4.9 The composite Accuracy-Generalization Surface (AGS) for the MICE data set.	71
4.10 (left) The 2-dimensional composite AGS for the USPS data set. (right) Trends in NSM methods that perform best for low margin error (green circle) and higher margin error (red circle), where a black arrow shows the transition of best performers between margin error regions.	72
4.11 (left) The 2-dimensional composite AGS for the MICE data set. (right) Trends in NSM methods that perform best for low margin error (green circle) and higher margin error (red circle), where a black arrow shows the transition of best performers between margin error regions.	73
4.12 (left) The 2-dimensional composite AGS for the ZWAVE data set. (right) Trends in NSM methods that perform best for low margin error (green circle) and higher margin error (red circle), where a black arrow shows the transition of best performers between margin error regions.	75
4.13 (left) The 2-dimensional composite AGS for the LCVF data set. (right) Trends in NSM methods that perform best for low margin error (green circle) and higher margin error (red circle), where a black arrow shows the transition of best performers between margin error regions.	77

Figure	Page
A.1 Network Structure Modification (NSM) methods are compared by prototype addition stage for the Mice protein expression (MICE) data set. Solid horizontal lines show non-dynamic baseline performance. By the end of 48 prototype addition stages, each NSM method has the opportunity for the same number of prototypes (8 initial + 48 additional) as the baseline initialized with 7 per each of the 8 classes. Performance is shown for (top) classification accuracy and (bottom) total cost.	87
A.2 NSM methods trained with the Lunar Crater Volcanic Field (LCVF) data set are compared by prototype addition stage. Solid horizontal lines show non-dynamic baseline performance. By the end of 70 prototype addition stages, each NSM method has the opportunity for the same number of prototypes (35 initial + 70 additional) as the baseline initialized with 3 per each of the 35 classes. Performance is shown for (top) classification accuracy and (bottom) total cost.	88
A.3 Classification accuracy (z -axis) of seven representative NSM methods for (left column) MICE and (right column) LCVF are shown against the margin-based empirical error and the number of utilized prototypes (x and y -axes respectively). For ease of viewing and comparison, the data is sphered over all NSM methods for each data set.	91
A.4 The covariance is shown between features within each labeled class over a 256×256 covariance matrix.	93
A.5 The correlation is shown between features within each labeled class over a 256×256 correlation matrix.	93

Figure	Page
A.6 Classification accuracy (a) and Total Cost (b) versus prototype addition stage for seven prototype addition methods over the USPS data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6 One GRLVQI “baseline” trained with 80 fixed prototypes is included for reference. Plots are the average of 150 validation result curves (30 complete runs each consisting of $K = 5$ folds). . . .	98
A.7 Classification accuracy (a) and Total Cost (b) versus utilized number of prototypes for eleven prototype addition methods over the USPS data set with a GRLVQI algorithm. One GRLVQI “baseline” trained with 80 prototypes is included for reference. Validation accuracies from the 150 curves (30 complete runs each consisting of $K = 5$ folds) are pooled for each utilized prototype value then averaged(roughly 18,000 points per prototype value).	99
A.8 Classification accuracy versus utilized number of prototypes for six representative GRLVQI NSM methods for the USPS dataset: Mean, SM-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F are plotted in addition methods over the USPS data set with a GRLVQI algorithm. A non-dynamic GRLVQI baseline trained with 80 prototypes is included for reference. Validation accuracies from the 150 curves (30 complete runs each consisting of $K = 5$ folds) are pooled for each utilized prototype value then averaged(roughly 18,000 points per prototype value).	100
A.9 For USPS-GRLVQI: NSM classification accuracies are compared against number of additional prototypes for a margin test point, θ_{cp} . Percentiles within each NSM method (5 prototype intervals starting with 10) are shown as the top, middle, and bottom of colored vertical boxes (75th, 50th, and 25th percentile). Plots (a), (b), and (c) correspond to θ_{cp} values of 0, $7.89e - 3$, and $1.18e - 2$. .	101

Figure	Page
A.10 NSM methods with 35, 40, 45, and 105 prototypes display CGNT(top) and HSV(bottom) Margin Error as a function of θ_{cp}	102
A.11 Comparison of NSM methods for USPS dataset (top row shows axes for each column: classification accuracy, standard deviation, and relative accuracy (l-r).	103
A.12 (a) This composite of Fig. A.11 (right-column) selects the best performing NSM method over each grid point. (b) This graph is simplified by flattening the z-axis. This flattened view indicates NSM methods with the highest average classification accuracy over a range of utilized prototypes and margin error values.	104
A.13 The covariance between features of the MICE data set are shown within each labeled class over a 71×71 covariance matrix.	106
A.14 The correlation between features of the MICE data set are shown within each labeled class over a 71×71 correlation matrix.	106
A.15 Classification accuracy (a) and Total Cost (b) versus prototype addition stage for seven prototype addition methods over the MICE data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6 Three GRLVQI “Baselines” trained with a fixed number of prototypes are included for reference. Plots are the average of 150 validation result curves (30 complete runs each consisting of $K = 5$ folds).	111

Figure	Page
A.16 Classification accuracy (a) and Total Cost (b) versus utilized number of prototypes for seven prototype addition methods over the Mice data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6 Two GRLVQI baselines trained with a fixed number of prototypes are included for reference. Validation accuracies from the 150 curves (30 complete runs each consisting of $K = 5$ folds) are pooled for each utilized prototype value then averaged (roughly 18750 points per prototype value).	112
A.17 For MICE-GRLVQI: NSM classification accuracies are compared against number of additional prototypes for a margin test point, θ_{cp} . Quartiles within each NSM method (in 4 prototype intervals starting with 8) are shown as the top, middle, and bottom of colored vertical boxes (75%-ile, 50%-ile, and 25%-ile). Plots (a), (b), and (c) correspond to θ_{cp} values of 0, $2.1e - 5$, and $5.3e - 5$.	113
A.18 Simple NSM methods with 35, 40, 45, and 105 prototypes display (left) CGNT Margin Error as a function of θ_{cp} , (middle) HSV Margin Error as a function of θ_{cp} , and (right) HSV Margin Error versus CGNT Margin Error.	114
A.19 Cost Minimizing NSM methods with 35, 40, 45, and 105 prototypes display (left) CGNT Margin Error as a function of θ_{cp} , (middle) HSV Margin Error as a function of θ_{cp} , and (right) HSV Margin Error versus CGNT Margin Error.	115
A.20 Clustering NSM methods with 35, 40, 45, and 105 prototypes display (left) CGNT Margin Error as a function of θ_{cp} , (middle) HSV Margin Error as a function of θ_{cp} , and (right) HSV Margin Error versus CGNT Margin Error.	116

Figure	Page
A.21 Hybridized NSM methods with 35, 40, 45, and 105 prototypes display (left) CGNT Margin Error as a function of θ_{cp} , (middle) HSV Margin Error as a function of θ_{cp} , and (right) HSV Margin Error versus CGNT Margin Error. . .	116
A.22 Comparison of GRLVQI NSM methods for MICE dataset (top row shows axes for each column: classification accuracy, standard deviation, and relative accuracy (l-r).	117
A.23 The surface plot (top) and flattened surface plot (bottom) indicate regions of comparative advantage for eleven NSM methods.	118
A.24 The surface plot (top) and flattened surface plot (bottom) indicate regions of comparative advantage for NSM methods: Avg-M-Cost, M-Cost, S-Cost, PC, Closest, Near-Mean, and Mean.	119
A.25 The covariance is shown between features of the ZWAVE data set within each labeled class over a 189×189 covariance matrix.	121
A.26 The correlation between features of the ZWAVE data set are shown within each labeled class over a 189×189 correlation matrix.	121
A.27 Classification accuracy (a) and Total Cost (b) versus prototype addition stage for seven prototype addition methods over the LCVF data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6 A GRLVQI ‘baseline trained with 30 prototypes is included for reference. Plots are the average of 150 validation result curves (30 complete runs each consisting of $K = 3$ folds).	123

Figure	Page
A.28 Classification accuracy (a) and Total Cost (b) versus utilized number of prototypes for eleven prototype addition methods over the ZWAVE data set with a GRLVQI algorithm. One GRLVQI baseline trained with 30 prototypes is included for reference. Validation accuracies from the 150 curves (30 complete runs each consisting of $K = 5$ folds) are pooled for each utilized prototype value then averaged.	126
A.29 GRLVQI NSM methods for ZWAVE with 3, 4, 9, and 30 prototypes display margin error as a function of θ_{cp} : CGNT(top) and HSV(bottom).	127
A.30 Comparison of GRLVQI NSM methods for ZWAVE dataset. Top row shows axes for each column: classification accuracy, standard deviation, and relative accuracy (l-r).	128
A.31 (a) This composite of Fig. A.30 (right-column) selects the best performing NSM method over each grid point. (b) This graph is simplified by flattening the z-axis. This flattened view indicates NSM methods with the highest average classification accuracy over a range of utilized prototypes and margin error values.	129
A.32 We show the covariance between features of the LCVF data set within each labeled class over a 194×194 covariance matrix.	131
A.33 The correlation between features of the LCVF data set are shown within each labeled class over a 194×194 correlation matrix.	132

Figure	Page
A.34 Classification accuracy (a) and Total Cost (b) versus prototype addition stage for seven prototype addition methods over the LCVF data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6 Two GRLVQI “Baselines” trained with a fixed number of prototypes are included for reference. Plots are the average of 90 validation result curves (30 complete runs each consisting of $K = 3$ folds).	136
A.35 Classification accuracy (a) and Total Cost (b) versus utilized number of prototypes for seven prototype addition methods over the LCVF data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6 Two GRLVQI “Baselines” trained with a fixed number of prototypes are included for reference. Validation accuracies from the 90 curves (30 complete runs each consisting of $K = 3$ folds) are pooled for each utilized prototype value then averaged(roughly 4500 points per prototype value).	137
A.36 Classification accuracy versus utilized number of prototypes for seven representative GRLVQI NSM methods for the LCVF dataset: Closest, M-Cost, Avg-M-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F are plotted in addition methods over the LCVF data set with a GRLVQI algorithm. Two GRLVQI “Baselines” trained with a fixed number of prototypes are included for reference. Validation accuracies from the 90 curves (30 complete runs each consisting of $K = 3$ folds) are pooled for each utilized prototype value then averaged(roughly 4500 points per prototype value).	138

Figure	Page
A.37 For LCVF-GRLVQI: NSM classification accuracies are compared against number of additional prototypes for a margin test point, θ_{cp} . Quartiles within each NSM method (in 5 prototype intervals starting with 35) are shown as the top, middle, and bottom of colored vertical boxes (75%-ile, 50%-ile, and 25%-ile). Plots (a), (b), and (c) correspond to θ_{cp} values of 0, $6.3e - 5$, and $1.5e - 4$	139
A.38 Simple NSM methods with 35, 40, 45, and 105 prototypes display (left) CGNT Margin Error as a function of θ_{cp} , (middle) HSV Margin Error as a function of θ_{cp} , and (right) HSV Margin Error versus CGNT Margin Error.	140
A.39 Cost Minimizing NSM methods with 35, 40, 45, and 105 prototypes display (left) CGNT Margin Error as a function of θ_{cp} , (middle) HSV Margin Error as a function of θ_{cp} , and (right) HSV Margin Error versus CGNT Margin Error.	141
A.40 Clustering NSM methods with 35, 40, 45, and 105 prototypes display (left) CGNT Margin Error as a function of θ_{cp} , (middle) HSV Margin Error as a function of θ_{cp} , and (right) HSV Margin Error versus CGNT Margin Error.	142
A.41 Hybridized NSM methods with 35, 40, 45, and 105 prototypes display (left) CGNT Margin Error as a function of θ_{cp} , (middle) HSV Margin Error as a function of θ_{cp} , and (right) HSV Margin Error versus CGNT Margin Error.	142
A.42 Comparison of NSM methods for LCVF dataset (top row shows axes for each column: classification accuracy, standard deviation, and relative accuracy (l-r).	143
A.43 (a) This composite of Fig. A.42(right-column) selects the best performing NSM method over each grid point. (b) presents a flattened view, indicating NSM methods with the highest average classification accuracy over a range of utilized prototypes and margin error values, representing the generalization error.	144

Figure	Page
A.44 This flattened surface is composed of seven of the best performing NSM methods (Closest, M-Cost, Avg-M-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F). Colored grids indicate NSM methods with the highest average classification accuracy over a range of utilized prototypes and margin error values, representing the generalization error.	145
B.1 LCVF with six selected NSM methods illustrate that choice of prototype insertion approach does play an important role in overall classification accuracy and generalization error.	147
B.2 MICE with six selected NSM methods illustrate that choice of prototype insertion approach does play an important role in overall classification accuracy and generalization error.	149
C.1 NSM plots USPS showing (a) a composite view, (b) a modified boxplot with $\theta_{cp} = 0$, and (c) a modified boxplot with $\theta_{cp} = 7.9e^{-3}$	151

List of Tables

Table	Page
2.1 Foundational LVQ variables.	7
2.2 Variables for Dynamic LVQs.	8
4.1 MICE, USPS, ZWAVE, and LCVF data set attributes.	55
4.2 Generalized Relevance Learning Vector Quantization Improved (GRLVQI) parameters include prototype learn rates, α^w , relevance learn rates, α^l and conscience learning parameters, γ and β for USPS, MICE, ZWAVE and LCVF data sets.	60
4.3 Dynamic parameters for USPS, MICE, ZWAVE, and LCVF independent trials.	61
4.4 Converged model accuracy and number of prototypes is shown for independent trials.	63
A.1 General NSM performance, averaged over $30 \times K$ training profiles, where $K = 5$ for MICE and $K = 3$ for LCVF. Accuracy and corresponding cost performance are shown for peak training profile values.	89
A.2 USPS GRLVQI parameters.	92
A.3 NSM performance for 150 independent trials of USPS is shown when the non-dynamic baseline is chosen as an accuracy performance point. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and prototype utilization (when compared to baseline performance). Statistically significant improvements are shown in bold, statistically similar results are shown in italics.	95

Table	Page
A.4 Converged NSM model performance is shown for $30 \times K$ independent trials ($K = 5$ for USPS, MICE, and ZWAVE, and $K = 3$ for LCVF). Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and corresponding prototype utilization, when compared to baseline performance. Statistically significant improvements are shown in bold, statistically similar results are shown in italics.	96
A.5 MICE GRLVQI parameters.	105
A.6 NSM performance for 150 independent trials is shown when the non-dynamic baseline is chosen as an accuracy performance point. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and prototype utilization (when compared to baseline performance). Statistically significant improvements are shown in bold, statistically similar results are shown in italics.	108
A.7 Converged NSM model performance is shown for 150 independent trials of MICE. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and corresponding prototype utilization, when compared to baseline performance. Statistically significant improvements are shown in bold, statistically similar results are shown in italics.	109
A.8 ZWAVE GRLVQI parameters.	120
A.9 NSM performance for 150 independent trials is shown when the non-dynamic baseline is chosen as an accuracy performance point. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and prototype utilization (when compared to baseline performance). Statistically significant improvements are shown in bold, statistically similar results are shown in italics.	124

Table	Page
A.10 Converged NSM model performance is shown for 150 independent trials of the ZWAVE data set. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and corresponding prototype utilization, when compared to baseline performance. Statistically significant improvements are shown in bold, statistically similar results are shown in italics.	125
A.11 LCVF GRLVQI parameters.	130
A.12 NSM performance for 150 independent trials is shown when the non-dynamic baseline is chosen as an accuracy performance point. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and prototype utilization (when compared to baseline performance). Statistically significant improvements are shown in bold, statistically similar results are shown in italics.	133

DYNAMIC PROTOTYPE ADDITION IN GENERALIZED LEARNING VECTOR QUANTIZERS

I. Introduction

ARTIFICIAL Intelligence (AI) describes a computer system's ability to perform tasks that generally require human intelligence [6]. The cognitive function of problem solving can be mimicked by computers to learn and make predictions from data without being explicitly programmed; this field of research is known as Machine Learning (ML) [6] and is a contributor to the field of Autonomy.

ML can assist the Department of Defense (DoD) with automated processing of traditional and non-traditional sensors (including social media) to drive actionable intelligence and strategic options in real-time [7]; promoting informed decision making for top military commanders and battle field combatants alike [8, 9]. ML can improve user safety and efficiency [10, 11] and perform tedious tasks previously assigned to human operators such as analyzing image or video feeds [12–14]. This shifting of cognitive burdens can promote more efficient tasking of assets and permit operations in new domains [15–17].

There are many different approaches to ML. Artificial Neural Networks (ANNs) [18] are a popular family of approaches (or algorithms). This dissertation employs a special case of ANNs: Learning Vector Quantization (LVQ) [19] because of its ease of implementation and interpretability. Additionally, once trained, LVQ presents a lightweight and computationally efficient algorithm while providing strong classification accuracy [2, 20].

LVQs are prototype-based learners [21, 22] that train codebook vectors existing in the data space (prototypes), to accurately represent labeled training samples. Learning occurs as samples iteratively influence updates to prototype positions based on a distortion measure, attracting (or repelling) a prototype of the same (or another) class. At the conclusion of training, piece-wise linear decision boundaries are drawn between prototype locations to provide classification boundaries [2].

1.1 Problem Statement

Accuracy, which is the percentage of correct predictions for a data set [21], alone is not a sufficient measure of an ML algorithm’s success in classification. *Generalization*, which is the confidence a classifier has in predictions [23], is also critical as it describes an algorithm’s ability to offer continued success over future data samples. Crammer, et al. [23] show that generalization error bounds increase with the number of prototypes used. In other words, too many prototypes can lead to over-fitting.

Unfortunately, the optimal number and position of prototypes among the classes is a non-trivial problem [24, 25]. Often, LVQ networks are initialized indiscriminately with an equal number of prototypes per class, which may not appropriately match the data, leading to the formation of inefficient or under-utilized prototypes [26]. For LVQs, an alternative to distributing a fixed number of prototypes to each class during network initialization is to add them *dynamically* or *incrementally* as part of the learning process (hereinafter *dynamic*) [1]. This prompts the research question: “*Can we achieve improvements in classification accuracy and generalization error in Generalized LVQ classifiers through the dynamic addition of prototypes?*”

1.1.1 Contributions.

Answering the question led to three contributions. First, a novel framework is developed that incorporates Network Structure Modification (NSM) in DLVQ and characterizes four dynamic prototype addition strategies: simple, cost-minimizing,

clustering, and cost-minimizing/clustering hybrids. Second, seven new NSM algorithms are designed for Generalized Relevance LVQ Improved (GRLVQI), a variant of LVQ to remedy potential data set challenges. Third, a unique method is presented to evaluate and characterize the accuracy and generalization of Dynamic GRLVQI (DGRLVQI) across their entire operating range.

1.1.2 Results.

The novel cost minimizing methods presented in the research achieve the best published classification accuracy over both the Z-Wave wireless protocol (ZWAVE) and Lunar Crater Volcanic Field (LCVF) data sets, offering increases from 96.5% [27] to 98.9% and 97.2% [4] to 98.0%, respectively.

In 84% of the novel 7 and existing 4 NSM methods over the United States Postal Service (USPS), Mice protein expression (MICE), ZWAVE, and LCVF data sets, DGRLVQI demonstrated statistically significant (simultaneous) improvements in classification accuracy and prototype reduction when compared to GRLVQI. They achieved 1% - 2% mean accuracy gains, with a mean of 4%-56% fewer prototype resources, leading to improved theoretical bounds on generalization error.

The analysis of classifier margin and margin error offers insight to a learned model's generalizability and expected performance trends in the presence of noise. In relation to other NSMs, cost-minimizing strategies achieve the highest classification accuracy but do not generalize as well. Clustering methods tend to generalize well but are less accurate. The hybridized cost/clustering approach consistently indicates the best accuracy performance when all margin error regions are considered. In all but the United States Postal Service (USPS) data set, it achieves the best balance of accuracy and generalization.

1.2 Document Outline

Chapter 2 summarizes relevant background and publications related to LVQs, dynamic LVQs, and foundational theory unique to the research area. Key advancements of

LVQ variants are summarized, to include the family of Generalized LVQ algorithms (GLVQ). The development of relevant dynamic efforts in the literature are shown, laying the foundation for expanded efforts in the following chapter. Chapter 3 describes the methodologies and novel contributions offered as part of this research. This chapter characterizes the dynamic nature of DLVQs and explores the development of new dynamic approaches. The chapter concludes with a novel method to compare DLVQs on a set of common axes based on aspects of measured, learned, and theoretical performance. Chapter 4 presents the results of experiments over four data sets. Performance is examined in terms of accuracy, prototype utilization, and margin error. Chapter 5 summarizes the key takeaways and contributions of this research effort. This chapter also indicates promising avenues for related future work.

II. Background & Literature Review

THIS chapter summarizes foundational background, theory, and literature pertinent to this research effort. Section 2.1 introduces the Learning Vector Quantization (LVQ) algorithm. Additionally, it outlines the structure of LVQ and its variants to include the family of generalized LVQ (GLVQ) classifiers. Section 2.2 introduces the concept of prototype addition and details the evolution of relevant dynamic methods in the literature. Section 2.3 describes the theoretical and empirical basis of generalization and introduces classifier margins. Section 2.4 discusses typical measures of performance in LVQ.

2.1 Learning Vector Quantization (LVQ)

An important class of pattern recognition algorithms are supervised learners, which use labeled training data to learn decision making models that can be applied to future data. There are many supervised learning approaches such as Bayesian classifiers (BC), support vector machines (SVM), and artificial neural networks (ANN) [14, 28]. ANNs have gained significant popularity with the increasing availability of computational resources [29, 30]. LVQ [31] is a special case of ANNs that has found a niche due to their efficiency (light-weight portability), ease of implementation, and clear interpretability of the training process and resulting classifications [5, 32–34]. LVQ has a large following, now enjoying their 12th biennial workshop since its introduction in 1997 [35].

LVQs are competitive prototype-based learners [21, 22] that train codebook vectors, or prototypes, to accurately represent a field of N -dimensional training data and separate them according to their class labels. The number of prototypes is assigned during LVQ initialization, and prototypes are initialized in each class either at sample locations within the training data or at randomized locations within the data space [21]. Prototypes receive fixed class labels that do not change during training. The competitive learning within

an LVQ algorithm comes from the perspective of a training sample, $\mathbf{x}^m$, where m is the sample index, $m = [1 \dots M]$ and M is the number of samples. Each training sample is a $1 \times N$ feature vector whose values describe a sensed target ($\mathbf{x} = [x(1), x(2), \dots, x(N)]$). Prototype vectors, $\mathbf{w}$, have the same dimensionality of $1 \times N$ and can be similarly expanded. As training progresses, all samples influence updates to prototype positions based on a distortion measure and according to a (typically) time varying learn rate. At the conclusion of training, piece-wise linear decision boundaries are drawn between prototypes locations to provide classification boundaries [2].

This document introduces a notation convention where $\mathbf{x}^m$, has an associated *true* class label y^m and an *assigned* or *classified as* class label z^m . A complete set of samples, prototypes, or labels is represented as $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, respectively. Vectors are signified with a bold font lower-case variable and matrices as a bold font upper-case variable. Functions, scalars, and index values have normal font weight and are either upper or lower-case. Scripted upper-case variables define sets (of either vectors or scalars). Unique class labels, $c = [1 \dots C]$, where C is the number of classes, can be included as sample or prototype subscripts (either individually or as sets). The set $\mathcal{X}_{c^{all}}$ identifies all samples with a common *true* class label ($y^m = c$), $M_{c^{all}}$ samples in all. This set is split between M_c correctly classified samples where $z^m = c$, and $M_{c'}$ misclassified samples where $z^m \neq c$, comprising sets $\mathcal{X}_c$ and $\mathcal{X}_{c'}$ respectively. This document strictly refers to misclassified samples with respect to one class at a time. Samples where $y^m \neq c$ but $z^m = c$ are incorrect assignments and are not identified as a specific set. Key variables presented in this dissertation are shown in Table 2.1 and Table 2.2 for convenience.

Table 2.1: Foundational LVQ variables.

Fundamental variables and symbols				
Indiv	Size	Set	Size	Definition
$\mathbf{x}^m$	$1 \times N$	$\mathcal{X}$	$M \times N$	N -dimensional training sample; M (N -D) training samples
y^m	1×1	$\mathcal{Y}$	$M \times 1$	True class label
z^m	1×1	$\mathcal{Z}$	$M \times 1$	Sample label: assigned class
$\mathbf{w}^q$	$1 \times N$	$\mathcal{W}$	$Q \times N$	Prototype vector
$\lambda(i)$	1×1	λ	$1 \times N$	(Feature) relevance weights
Descriptive cases for training samples with class label $y^m = c$				
Indiv	Size	Set	Size	Definition
$\mathbf{x}_c^m$	$1 \times N$	$\mathcal{X}_c$	$M_c \times N$	Correctly classified samples of class c
$\mathbf{x}_{c'}^m$	$1 \times N$	$\mathcal{X}_{c'}$	$M_{c'} \times N$	Misclassified samples of class c ($z^m \neq c$)
$\mathbf{x}_{c^{all}}^m$	$1 \times N$	$\mathcal{X}_{c^{all}}$	$M_{c^{all}} \times N$	True samples of class c
$\mathbf{x}_{c^{for}}^m$	$1 \times N$	$\mathcal{X}_{c^{for}}$	$M - M_{c^{all}} \times N$	Samples of foreign classes (other than c)
$\mathbf{w}^j$	$1 \times N$	$\mathcal{W}_c$	$Q_c \times N$	In-class prototype vector
$\mathbf{w}^k$	$1 \times N$	$\mathcal{W}_{c'}$	$Q_{c'} \times N$	Out-of-class prototype vector
$\mathbf{w}^{net}$	$1 \times N$	$\mathcal{W}$	$Q \times N$	Net prototype vector
Descriptive cases for samples of class c whose nearest in-class prototypes is $\mathbf{w}^q$; $(y^m = c) \wedge (\mathbf{w}^q = \mathbf{w}^j)$				
Indiv	Size	Set	Size	Definition
$\mathbf{x}_q^m$	$1 \times N$	$\mathcal{X}_q$	$M_q \times N$	Sample where $\mathbf{w}^q$ is nearest in-class prototype
$\mathbf{x}_{q^{net}}^m$	$1 \times N$	$\mathcal{X}_{q^{net}}$	$M_{q^{net}} \times N$	Sample where $\mathbf{w}^q$ is nearest net prototype
$\mathbf{x}_{q'}^m$	$1 \times N$	$\mathcal{X}_{q'}$	$M_{q'} \times N$	Sample where $\mathbf{w}^q$ is nearest in-class prototype, but not the nearest net-prototype ($z^m \neq c$)
$\mathbf{x}_c^m$	$1 \times N$	$\mathcal{X}_{c^{all}}$	$M_{c^{all}} \times N$	Sample belonging to the same class c as $\mathbf{w}^q$
$\mathbf{x}^p$	$1 \times N$	$\mathcal{X}_p$	$P \times N$	Sample in candidate pool
$\mathbf{x}^t$	$1 \times N$	$\mathcal{X}_\tau$	$\tau \times N$	Sample in evaluation pool

Table 2.2: Variables for Dynamic LVQs.

Other		
Symbol	Type	Definition
$S(\mathcal{X}, \mathcal{W})$	Function	Cost contribution of samples in $\mathcal{X}$ over prototype network $\mathcal{W}$
d^j, d^k, d^{net}	Distortion measure	*Distortion between $\mathbf{x}^m$ and $\mathbf{w}^j$, $\mathbf{w}^k$, or $\mathbf{w}^{net}$
$\mathbf{w}^{new}$	Vector	New prototype position
$\mathbf{w}^{old}$	Vector	Redistributed (existing) prototype position
H^q	Counter	Prototype split counter
G^q	Function	Cluster gain function
$a(\mathbf{x}^m)$	Function	Average dissimilarity to samples in the same cluster
$b(\mathbf{x}^m)$	Function	Lowest average dissimilarity to a neighboring cluster
$sil(\mathbf{x}^m)$	Function	Sample silhouette coefficient
$SIL(\mathcal{X}_q)$	Function	Cluster score: average silhouette coefficient over a cluster
$G^{b'}$	Function	Detractive influence of foreign cluster $\mathcal{X}_{q^b}$
HW^q	Function	Relative cluster health score
u	Parameter value	Base for Silhouette H counter
*The distortion measure utilized in this work is the λ weighted squared Euclidean distance in Eqn. 2.5		

2.1.1 Competitive Learning in LVQ.

Competitive learning in LVQ is described through three phases: competition, winner selection, and synaptic adaptation, seen in Fig. 2.1. In the first phase, each LVQ defines the set of prototype vectors that *compete*, commonly selecting one or more from the set of *in-class* prototypes (belonging to the same class as the input sample $\mathbf{x}^m$), *out-of-class* prototypes (belonging to any class other than that of the input sample $\mathbf{x}^m$), or a *net* prototype (chosen from all prototype vectors, regardless of class label) [1]. In the second phase, a *winning* prototype is one which results in the minimum distortion between it and the current sample. When using Euclidean distance as the distortion measure, the winner $\mathbf{w}^j$ represents the prototype from the set of competitors closest to $\mathbf{x}^m$. (Frequently in LVQ, a

second competition selects one more winning prototype, $\mathbf{w}^k$.) After winner selection, many LVQs impose additional conditions in order to apply updates to the prototypes based upon the influence of $\mathbf{x}^m$, such as windowing functions [19]. Where the additional conditions are satisfied, [36] shows the *synaptic adaptation* rule for LVQ algorithms can be generalized as:

$$\mathbf{w}^j \leftarrow \mathbf{w}^j - \alpha \frac{\partial S}{\partial \mathbf{w}^j}; \quad \mathbf{w}^k \leftarrow \mathbf{w}^k - \alpha \frac{\partial S}{\partial \mathbf{w}^k}, \quad (2.1)$$

where α is the (typically time decayed) learn rate and S is a generalized cost function that weights the degree of prototype updates [36]. The completion (or termination) of these three phases for a sample, $\mathbf{x}^m$, constitutes one training step. This process is then repeated T times for all training samples. to constitute one epoch Classification accuracy of the LVQ is reported as the mean of the individual class accuracies.

2.1.2 LVQ Variants.

Since Kohonen’s original LVQ, many variants have arisen that provide differing approaches to better estimate the decision surface. Kohonen’s first instantiation, LVQ1 [31], makes updates based on classification outcome only and behaves like a density estimator. This first approach was successful in providing reasonable class boundary estimates based on prototype locations. Variants of LVQ1 include LVQ2 which performs updates to an in-class and out-of-class prototype vector and LVQ3 which performs updates based on two closest prototypes [19]. The LVQ2 variants are most popular, leading to LVQ2.1 where the prototype updates are only permitted when distances between samples and prototypes fall within a predefined window [1]. Significantly, the authors in [23] showed that LVQ2.1 variants are large margin classifiers, underlying their popularity in LVQ-derived research. Sato and Yamada [36] made one of the most significant breakthroughs in LVQs by generalizing LVQ2.1 (called GLVQ) through the introduction of a formal cost function and minimizing that cost function via gradient descent. This addressed many of the prototype divergence issues seen in LVQ2.1. Throughout this paper,

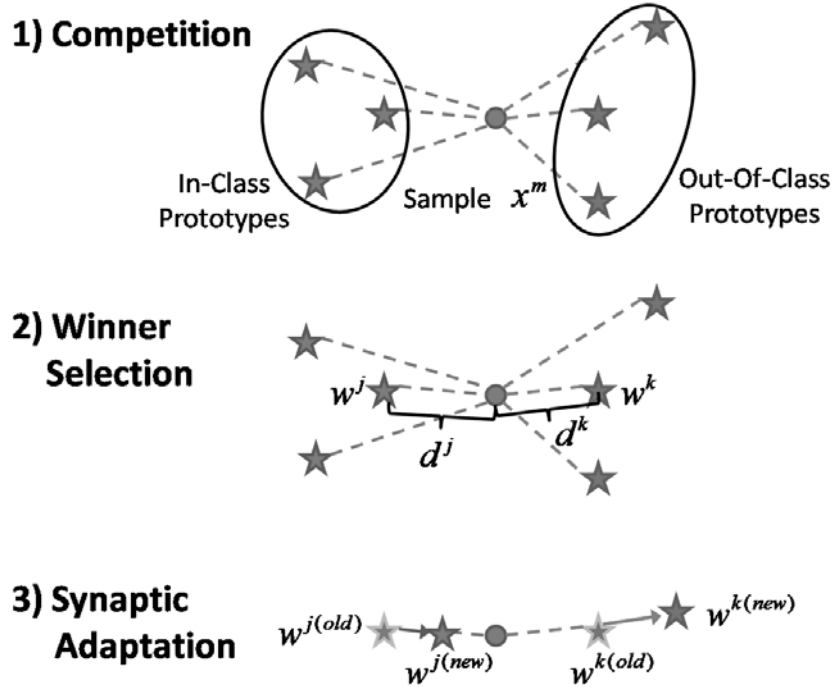


Figure 2.1: LVQs iteratively adjust prototype positions based off of labeled samples and can be described by three phases, illustrated for an LVQ 2.1-based algorithm.

1) Competition: for a labeled sample, the LVQ algorithm identifies groups of in-class and out-of-class prototypes to compete. **2) Winner Selection:** the in-class prototype, w^j , and out-of-class prototype, w^k , that result in the minimum distortion, d^j and d^k respectively, win and are updated. **3) Synaptic Adaptation:** the positions of winning prototypes are updated according to Eqn. 2.1, where in-class prototypes are moved closer to the sample and out-of-class prototypes are moved further away (so-called differential shifting) [1, 2].

the GLVQ family of algorithms provides a basis for our experimental work, as such, greater detail related to this family is given in the following section. An abbreviated LVQ family tree is seen in Fig. 2.2 [1].

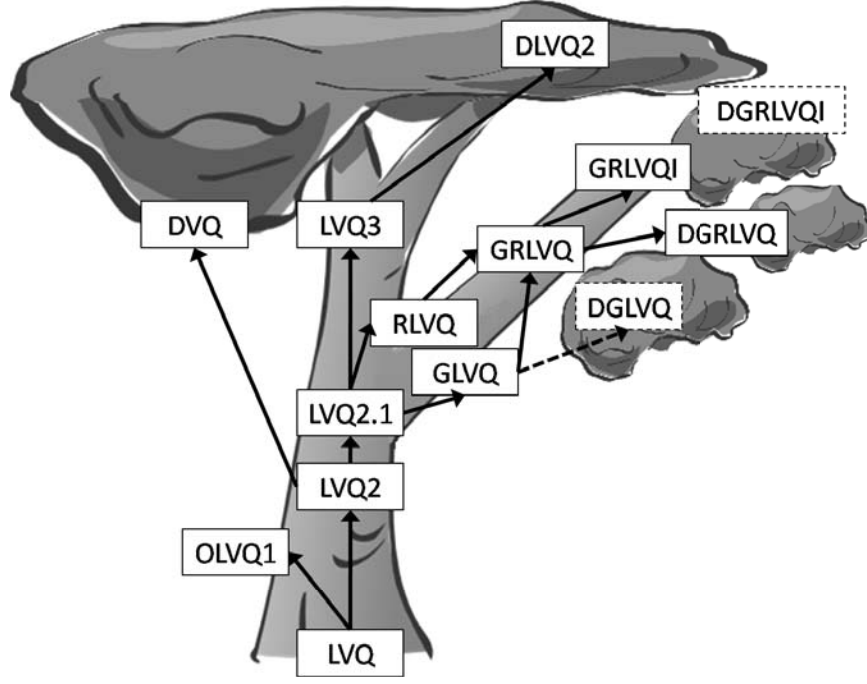


Figure 2.2: An abbreviated family tree of LVQ algorithms with dynamic variants located in the “leaves”.

Following the algorithm structure in Section 2.1.1, the competition and winner selection phases of GLVQ uses both the nearest in-class and nearest out-of-class prototypes ($\mathbf{w}^j$ and $\mathbf{w}^k$ respectively). Instead of explicitly restricting winner selection by a window about the midpoint between $\mathbf{w}^j$ and $\mathbf{w}^k$, GLVQ implicitly does this [36, 37] by employing the cost [36]:

$$S(\mathcal{X}, \mathcal{W}) = \sum_{m=1}^M f(\mu(\mathbf{x}^m)), \quad \mu(\mathbf{x}) = \frac{d^j - d^k}{d^j + d^k}, \quad (2.2)$$

where $\mathcal{X}$ is the set of M training samples, d^j and d^k are squared Euclidean distances between the input sample $\mathbf{x}^m$ and the prototype vectors $\mathbf{w}^j$ and $\mathbf{w}^k$ respectively, and $\mathcal{W}$ is the set of prototype vectors. Consistent with [36], $f(\mu)$ is the sigmoid function $1/(1 + e^{-\mu})$ [1] which yields increased influence in prototype updates to samples near decision boundaries. When $\partial S / \partial \mathbf{w}^j$ and $\partial S / \partial \mathbf{w}^k$ are substituted into Eqn. (2.1), the resulting synaptic adaptation

equations minimize the cost function via gradient descent [1, 36]:

$$\mathbf{w}^j \leftarrow \mathbf{w}^j - \alpha \frac{\partial f}{\partial \mu} \frac{d^k}{(d^j + d^k)^2} [\mathbf{x} - \mathbf{w}^j] \quad (2.3)$$

$$\mathbf{w}^k \leftarrow \mathbf{w}^k + \alpha \frac{\partial f}{\partial \mu} \frac{d^j}{(d^j + d^k)^2} [\mathbf{x} - \mathbf{w}^k]. \quad (2.4)$$

Hammer and Villmann [38] extended Relevance learning from Relevance LVQ into GLVQ. The so-called GRLVQ assigns weighting factors to the input dimension as an adaptive diagonal metric, λ . This N -dimensional vector of relevance weights correspond to the N input feature dimensions, scaling them according to their relevance, or usefulness, in achieving accurate classifications. These weights modify distortion measures, d_λ^q , between an arbitrary sample, $\mathbf{x}^m$, and any prototype $\mathbf{w}^q$ as:

$$d_\lambda^q = \sum_{i=1}^N \lambda_i (\mathbf{x}^m(i) - \mathbf{w}^q(i))^2 \quad (2.5)$$

where i represents feature indices [1]. In each learning step, relevance weights are updated via gradient descent similar to the prototype update rule in Eqns. 2.3, 2.4 such that:

$$\Delta \lambda_i = \max \left\{ -\frac{\alpha}{2} \frac{\partial f}{\partial \mu} \frac{d_\lambda^k}{(d_\lambda^j + d_\lambda^k)^2} [\mathbf{x}(i) - \mathbf{w}^j(i)] + \frac{\alpha}{2} \frac{\partial f}{\partial \mu} \frac{d_\lambda^j}{(d_\lambda^j + d_\lambda^k)^2} [\mathbf{x}(i) - \mathbf{w}^k(i)], 0 \right\}. \quad (2.6)$$

After updates for each feature have been made, the values are re-scaled such that $\|\lambda\|_1 = 1$. This method of assigning and adjusting feature weights allows for a scaling of input dimension according to their usefulness in establishing accurate classifications [1]. This scaling is particularly useful to overcome some of the challenges associated with high dimensional data with varying ranges of input data [39].

Work in [37] addresses the dead neuron problem exhibited by many LVQs while addressing the potential for prototype divergence due to underrepresented classes. Their GRLVQ-Improved (GRLVQI) algorithm includes principals of DeSieno's [40] conscience learning to promote equiprobabilistic prototype learning. For in-class prototype selection,

GRLVQI applies a bias, B^q , to the distortion measure between input sample and prototype under evaluation based on past frequency of prototype selection [1]:

$$B^q = \gamma \left(\frac{1}{Q_c} - U_{old}^q \right), \quad (2.7)$$

where U^q tracks the frequency of past selection and Q_c is the number of prototypes in class c . This bias discourages frequent winners from dominating selection and instead advantages prototypes that are under utilized. Selection frequencies are updated each time a winning in-class prototype is identified as part of the training process:

$$U_{new}^q = U_{old}^{j_c} + \beta (1.0 - U_{old}^q) (\text{winning prototype}), \quad (2.8)$$

$$U_{new}^q = U_{old}^q + \beta (0.0 - U_{old}^q) (\text{remaining prototypes}), \quad (2.9)$$

where β is a conscience learning parameter to control the amount of update to U^q [1].

GRLVQI also counters some of the prototype divergence problems experienced in other LVQ and GLVQ variants [41, 42]. A potential source of divergence results from the standard update rule when a class with a large number of samples shares a boundary with a class containing far fewer samples [37]. When evaluating samples from the larger class, there are more frequent repulsions of foreign prototypes along the boundaries. However classes with fewer samples will not have sufficient power to reverse the action and re-position their respective prototypes. If not dealt with, this condition can lead to substantial drift. To counter this problem, GRLVQI includes an alternate update rule:

- for correctly classified samples: only update the in-class prototype,
- for misclassified samples: update both the in-class and out-of-class prototypes.

While GRLVQI allows for different learn rates when updating the in-class and out-of-class prototype vectors, efforts in this research maintain a single learn rate without deleterious effects [1].

2.2 Dynamic LVQ (DLVQ)

In LVQ networks, multiple prototypes (L per class) are typically initialized to achieve accurate classifications; however, the number of utilized prototypes can have a significant impact on the results. Too few prototypes can lead to inferior class differentiation, whereas too many can lead to over-fitting and poorer generalization. Aspects of both accuracy and generalization must be considered in algorithm selection. Determining the optimal number of prototypes is a non-trivial problem, and it depends on the particular data set and network configuration. Potentially, there will be a different number of prototypes per class. A promising solution is to determine an appropriate configuration through incremental prototype addition [1].

Poirier and Ferrieux [43] first adopted an incremental approach to prototype addition based on Kohonen’s LVQ2 to *dynamically* add prototype vectors as needed to better represent the class distributions and form better decision surfaces. Later works built dynamic LVQ (DLVQ) methods based on LVQ2.1 [44], LVQ3 [45, 46], GLVQ [47], and GRLVQ [48]. In Fig. 2.2, an LVQ family tree shows these dynamic variants as the outermost “leaves”. The variants belonging to the GLVQ family offer improved performance due to their use of a cost function to guide synaptic adaptation [1]. GRLVQI, a promising member of the GLVQ family, is chosen to assess dynamic methods described in this dissertation. While, the DLVQ community is modest in size, multiple approaches have been presented in the literature to achieve the common goal of conserving prototype resources while maintaining strong classification abilities.

Section 3.1 introduces a framework that generalizes the discussion of DLVQs. The portion of the framework that describes prototype addition and/or deletion consists of two parts: (1) a recipient region identification phase and (2) a prototype insertion phase. This two phase process characterizes the dynamic approaches in the literature. Necessary background and key mechanics of these approaches are presented, including adaptations

to normalize their implementation in this work. Qualitative comparison is deferred until Chapter 3.

2.2.1 *Mean.*

Zell, et al. [49] applies dynamic prototype addition to suggest alternate LVQ network configurations. Others, such as Bermejo, et al. [46], have adopted similar approaches in a dynamic context. The work in [49] conducts a broad region identification phase, simultaneously allotting an additional prototype, $\mathbf{w}^{new}$, to each class. The prototype insertion phase in [49] is designed to offer improvements to previously deficient regions. These deficiencies are measured by the number of misclassified samples, $M_{c'}$, for each class, $c = [1 \dots C]$. The position for each additional prototype, $\mathbf{w}^{new}$, is calculated as:

$$\mathbf{w}^{new} = \frac{1}{M_{c'}} \sum_{\mathbf{x}_{c'}^m \in \mathcal{X}_{c'}} \mathbf{x}_{c'}^m, \quad (2.10)$$

which is the mean position of the misclassified samples. Although [49] adds a prototype to each class, this work adds a single prototype to class c which has the largest percentage of misclassified samples:

$$c^{new} = \underset{c}{\operatorname{argmax}} \left\{ \frac{M_{c'}}{M_{call}} \right\}, \quad (2.11)$$

where M_{call} is the number of true class samples of c ($M_{call} = M_c + M_{c'}$). Representative pseudo-code for one iteration of the Mean method is shown in Algorithm 1 and an example is shown in Fig. 2.3(a).

2.2.2 *Closest.*

Kirsten, et al. [50] also use misclassifications to drive the region identification phase. This method identifies all class regions exceeding a threshold value of the percentage of misclassified samples, permitting each to simultaneously receive an additional prototype. While sharing similarities in region identification, Closest varies from Mean in prototype insertion. New prototypes are inserted near class boundaries by initializing them at the

Algorithm 1 One iteration of Mean

```
1: for  $m \in M$  do                                ▶ Loop through all samples
2:   if  $y^m \neq z^m$  then                            ▶ If misclassified
3:      $c \leftarrow y^m$                                 ▶ Identify the true class
4:      $M_{c'} \leftarrow M_{c'} + 1$                     ▶ Increment misclassifications for class  $c$ 
5:      $\mathcal{X}_{c'} \leftarrow m$  if  $z \neq c$             ▶ Save indexes of misclassified samples
6:   end if
7: end for
8: Find class,  $c^{new}$ , with max misclassifications (Eqn. 2.11)
9: Insert  $\mathbf{w}^{new}$  at average position of  $\mathcal{X}_{c'}$  (Eqn. 2.10)
```

same position as the misclassified sample closest to an out-of-class prototype vector:

$$\mathbf{w}^{new} = \underset{\mathbf{x}_{c'}^m \in \mathcal{X}_{c'}}{\operatorname{argmin}} \left\{ d_\lambda(\mathcal{X}_{c'}, \mathcal{W}_{c'}) \right\} \forall \mathbf{w}_{c'}^q \in \mathcal{W}_{c'} \quad (2.12)$$

where d_λ is the relevance weighted distortion measure shown in Eqn. 2.5 and $\mathbf{x}_{c'}^m$ and $\mathbf{w}_{c'}^q$ are misclassified samples and out-of-class prototype vectors for class c , respectively. The Closest method is modified to insert a single prototype in order to normalize the comparison between insertion methods. Algorithm 2 and Fig. 2.3(b) illustrate the operation of Closest.

2.2.3 Principal Components.

Stefano, et al. [45] identifies regions around prototypes in Principal Components (PC)'s region identification phase (seen in Fig. 2.4b). While prototype addition methods have typically identified regions of misclassified samples or a class as a whole (as seen in Fig. 2.4a), [45] tracks the influence or effectiveness of prototypes and frequency of past selection to promote better utilization of additional prototype resources. The influence of each prototype $\mathbf{w}^q$ is described by overlapping sets that describe the association of samples with nearby prototypes:

Algorithm 2 One iteration of Closest

```
1: for  $m \in M$  do                                ▶ Loop through all samples
2:   if  $y^m \neq z^m$  then                            ▶ If misclassified
3:      $c \leftarrow y^m$                                 ▶ Identify the true class
4:      $M_{c'} \leftarrow M_{c'} + 1$                     ▶ Increment misclassifications for class  $c$ 
5:      $\mathcal{X}_{c'}^m \leftarrow m$                     ▶ Save indexes of misclassified samples
6:   end if
7: end for
8: Find class,  $c$ , with max misclassifications (Eqn. 2.11)
9:  $\mathbf{w}^{new} \leftarrow \mathbf{x}_{c'}^m$  with max  $d^k$  (Eqn. 2.12) ▶ Misclassified sample nearest foreign prototype
```

- $\mathcal{X}_q$: samples where $\mathbf{w}^q$ is the in-class winner (potentially also the net winner),
- $\mathcal{X}_{q'}$: samples where $\mathbf{w}^q$ is the in-class winner, but not the net winner,
- $\mathcal{X}_{q^{net}}$: samples where $\mathbf{w}^q$ is the net winner, and
- $\mathcal{X}_{c^{all}}$: samples belonging to class c , which is the same class as $\mathbf{w}^q$.

The total number of samples in each set are M_q , $M_{q'}$, $M_{q^{net}}$, and $M_{c^{all}}$ respectively. In Fig. 2.4, an association by cluster instead of class provides finer segmented regions. *Cluster* regions are seen where prototypes inherit “responsibility” for the correct classification of their samples, $\mathcal{X}_q$. As clusters are defined by their nearest in-class prototypes, the same prototype index variable, q , is adopted to refer to them (*i.e.*, $\mathcal{X}_q$). The comparison of $\mathcal{X}_{q'}$ to $\mathcal{X}_{c^{all}}$ highlights the proportion of misclassifications within a cluster with respect to its class, c , and constitutes the region’s influence. The frequency of past selection, H^q , tracks the number of times a cluster has already received an additional prototype resource. A gain score is assigned to each cluster, $\mathcal{X}_q$, based on region influence and frequency of past

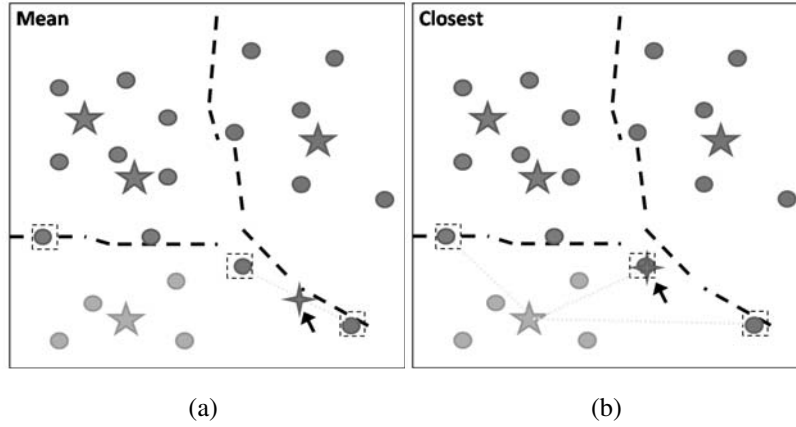


Figure 2.3: Two prototype addition methods are illustrated, where samples are circles and prototypes are stars (new prototypes are four pointed). True class labels are indicated by color (blue, red, or green) and assigned class is determined by Voronoi tessellations of prototype positions (dotted lines). Dotted boxes surround misclassified samples. In both prototype addition methods, the region of red misclassified samples contains the highest proportion of misclassified samples, $M_{c'}$, to true class samples M_{call} and is identified to receive an additional prototype. The location for new prototype insertion is determined as: **(a) Mean**: the average position of the red misclassified samples and **(b) Closest**: the sample with minimum distance between the average misclassified samples and their nearest foreign prototype (distance shown as grey lines).

selection [45]:

$$G^q = \frac{M_{q'}}{M_c \cdot H^q}, \quad (2.13)$$

where the cluster exceeding a threshold for G^q receive an additional prototype. In Fig. 2.5(a), the cluster associated with the singular red prototype is receives a second prototype.

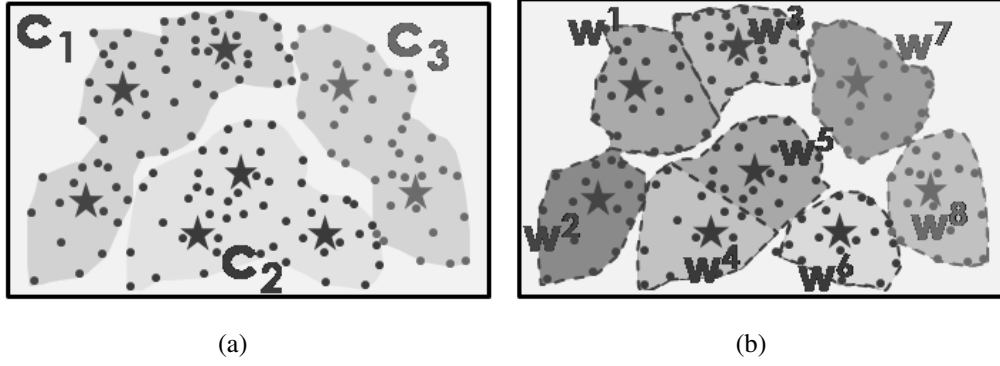


Figure 2.4: The space occupied by samples (points) and related prototypes (stars) are represented in two views. We show true labeled regions corresponding to class labels of $y = c$ or w^j (which are distinct from piecewise linear decision boundaries formed by labels of $z = c$ or w^{net}). **(a)** Identification regions typically consist of entire class structures, represented as red, blue, and green shaded backgrounds. However, **(b)** further subdivides these classes into clusters that share a nearest *in-class-prototype*, w^q . True cluster membership is represented in corresponding shades of red, blue, and green.

PC’s method for prototype insertion involves splitting the prototype, w^q , associated with the identified (cluster) region, $\mathcal{X}_q$, replacing it with two new prototypes that retain the existing class label. Prototypes are inserted along the principal component (eigenvector) direction of $\mathcal{X}_q$. The two new prototypes, w^{old} and w^{new} , are placed equidistant from the cluster mean along the principal component line, shown as a grey line in Fig. 2.5(a). One of the prototypes is randomly selected to be w^{old} , and receives an incremented *split-counter*, H^{old} . The split counter for the new prototype, w^{new} , is initialized as $H^{new} = 1$. Pseudo-code for PC is shown in Algorithm 3.

Algorithm 3 One iteration of PC

```
1: for  $m \in M$  do                                     ▶ Loop through all samples
2:    $c \leftarrow y^m, \mathcal{X}_c \leftarrow \mathbf{x}^m$        ▶ Identify, track class membership
3:   for  $w \in \mathcal{W}$  do                                   ▶ Loop through all prototypes
4:      $\mathcal{X}_{q^{net}} \leftarrow \mathbf{x}^m, \mathcal{X}_q \leftarrow \mathbf{x}^m$    ▶ Track net, in-class winners
5:     if  $(w^j = w^q) \wedge (w^{net} \neq w^q)$  then           ▶ In-class winner, but not net-winner
6:        $\mathcal{X}_{q'} \leftarrow \mathbf{x}^m$                  ▶ Track ownership of misclassified sample
7:     end if
8:   end for
9: end for
10: Total  $M_c$  for all classes  $c$ 
11: for  $w \in \mathcal{W}$  do
12:   Total  $M_{q'}$  and calculate  $G^q$  (Egn. 2.13)           ▶ Gain function score
13:   if Region identified as prototype recipient then
14:     Calculate Eigenvectors and Eigenvalues of  $\mathcal{X}_q$ 's covariance matrix
15:      $w^{old}, w^{new}$  placed equidistant from  $\overline{\mathcal{X}_q}$  along primary Eigenvector
16:      $H^{old} \leftarrow H^q + 1, H^{new} \leftarrow 1$        ▶ Update split counter
17:   end if
18: end for
```

2.2.4 Sampling Cost.

Losing, et al. [47] use a strategy of cost minimization in Sampling Cost (S-Cost) to maximize classification accuracy. This cost is described in Section 2.1.2 and is an underlying mechanism of GLVQ classifiers. S-Cost's region identification phase does not target a specific class or bounded region, but instead, forms a distributed region, $\mathcal{X}_{sub}$, which is a random subset (*e.g.*, a fixed number or percentage) of the sample population, $\mathcal{X}$. S-Cost does not compare multiple random subsets and all further actions such as evaluation and down-selection of the subset occur in the prototype insertion phase.

In the prototype insertion phase, each of the M_{sub} samples in the subset (denoted $\mathbf{x}_{sub}^m$) become candidate prototype positions. Each candidate position (with its associated class label) is temporarily added to the prototype network, $\mathcal{W}_{sub^m} = \mathcal{W} \cup \mathbf{x}_{sub}^m$, and the total cost is evaluated as:

$$S(\mathcal{X}_{sub}, \mathcal{W}_{sub^m}) = \sum_{\mathbf{x}_{sub}^m \in \mathcal{X}_{sub}} f(\mu(\mathbf{x}_{sub}^m)). \quad (2.14)$$

The candidate resulting in the lowest total cost is selected as the new prototype location, and class label of the new prototype. The operation of the S-Cost method is shown in Algorithm 4 and Fig. 2.5(b).

2.3 Generalization in LVQ Networks

In LVQ networks, multiple prototypes per class are typically used to achieve accurate classifications. However, the number of utilized prototypes can have a significant impact on the results. Too few prototypes can lead to inferior class differentiation, whereas too many can lead to over-fitting and poorer generalization. Aspects of both accuracy and generalization must be considered in algorithm selection.

Crammer, et al. [36] defines the generalization error, E , to be upper-bounded by the summation of a margin-based theoretical error term (right hand term) and an empirical

Algorithm 4 One iteration of Sampling Cost

- 1: $\mathcal{X}_{sub} \leftarrow$ simple random sampling of $\mathcal{X}$
 - 2: **for** $\mathbf{x}_{sub}^m \in \mathcal{X}_{sub}$ **do** ▷ Loop through all candidate samples as prototypes
 - 3: $\mathcal{W}_{sub^m} \leftarrow \mathcal{W} \cup \mathbf{x}_{sub}^m$ ▷ Temporarily add candidate pt to prototype network
 - 4: **for** $\forall \mathbf{x}_{sub}^m \in \mathcal{X}_{sub}$ **do** ▷ Samples from candidate pool form test set
 - 5: $\mu(\mathbf{x}_{sub}^m) \leftarrow \frac{d^k - d^j}{d^k + d^j}$ ▷ Quotient of relative distortion between sample and nearest
in-class, out-of-class prototype of network $\mathcal{W}_{sub^m}$
 - 6: $f(\mu)$ ▷ Compute sigmoid cost function of argument $\mu(\mathbf{x})$
 - 7: **end for**
 - 8: $S(\mathcal{X}_{sub}, \mathcal{W}_{sub}) \leftarrow$ Sum costs over temporary prototype network
 - 9: **end for**
 - 10: $\mathbf{w}^{new} \leftarrow \text{argmin}(S(\mathcal{X}_{sub}, \mathcal{W}_{sub}))$ ▷ Select position resulting in smallest cost
-

error term (left hand term) as:

$$E \leq \underbrace{E(\mathcal{X}, \mathcal{W})_\theta}_{\text{empirical}} + \underbrace{\sqrt{\frac{8}{M} \left(v \log^2 \frac{32M}{\theta^2} + \log \frac{4}{\delta} \right)}}_{\text{theoretical}}, \quad (2.15)$$

where $\mathcal{W}$ is the set of prototype vectors, $\mathcal{X}$ is the set of M input samples, $1 - \delta$ is the confidence in the estimate, θ is the margin threshold, and v denotes the Vapnik-Chervonenkis (VC) dimension.

The VC dimension, v , in the theoretical term is derived for the two class case in an LVQ network as [23]:

$$v = \min \left(N + 1, \frac{64R^2}{\theta^2} \right) 2L^C \log(eL^2) \quad (2.16)$$

where N is the dimensionality of the data, $R \geq \|\mathbf{x}^m\| \forall m$, C is the number of classes, e is Euler's constant, and L is the number of prototypes per class. The pertinent relationship from this equation is that theoretical error, and consequently generalization error bounds grow as a result of an increased number of prototypes.

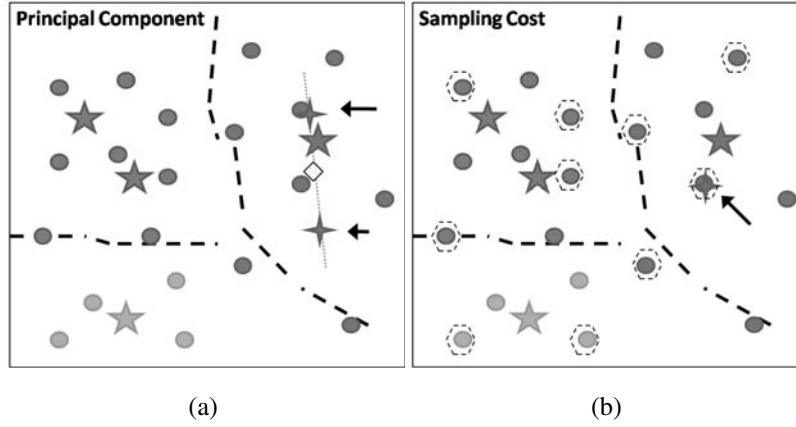


Figure 2.5: Two prototype addition methods are illustrated where samples are circles and prototypes are stars (new prototypes are four pointed). True class labels are indicated by color (blue, red, or green) and assigned class is determined by Voronoi tessellations of prototype positions (dotted lines). **(a)** Principal Component: The in-class samples nearest each prototype define potential regions for prototype addition. The red prototype yields the highest split metric and is replaced by two prototypes that are inserted equidistant along the eigenvector of the red samples whose nearest in-class prototype was just split. **(b)** Sampling Cost: A random subset of samples (surrounded by dotted hexagons) is generated and their cost is calculated over the union of the original prototype network and a candidate prototype position (selected from the subset). A new prototype is inserted at the candidate prototype that leads to the lowest subset cost.

The empirical component captures the degree of margin error, which indicates negative confidence in future predictions from existing classifiers [51]. Crammer, et al. [23] explain two potential approaches to calculate this error. Sample margin (θ_s), which is the distance a sample can move before hitting a decision boundary. Hypothesis margin (θ_h), which is the distance that a classifier's boundary can travel before altering classification results. While sample margins are common in many maximum margin

classifiers, such as Support Vector Machines (SVMs), hypothesis margins are more typical of LVQs [23]. These two margins, seen in Fig. 2.6, are closely related with large hypothesis margins directly translating into large sample margins [23]. Crammer, et al. [23] show the hypothesis margin lower bounds the sample margin and allows for simpler computation in LVQs as:

$$\theta_h(\mathbf{x}) = (d^k - d^j) / 2. \quad (2.17)$$

Hypothesis margin computations are simpler for LVQs (especially for high dimensional spaces with many classes and prototypes) because they do not depend on calculating closed form equations for the piece-wise separating hyper-planes, but for each sample, can be characterized in terms of d^k and d^j which are already calculated as part of LVQ's winner selection phase [1].

Crammer, et al. [23] show that margin error can be calculated from θ_h in an “all or nothing” strategy by counting the number of samples whose hypothesis margin, $\theta_h(\mathbf{x})$, is smaller than a threshold margin, $0 \leq \theta_{cp} \leq 1/2$:

$$E_{CGNT}(\mathcal{X}, \mathcal{W}) = \frac{1}{M} \sum_{\mathbf{x}} \begin{cases} 1, & \text{if } \theta_h(\mathbf{x}) < \theta_{cp} \\ 0, & \text{otherwise} \end{cases}, \quad (2.18)$$

In Eqn. 2.18, $\mathcal{X}$ is the set of M samples and $\mathcal{W}$ is the prototype configuration [1]. This error is named E_{CGNT} to reflect the authors of [23]: Crammer, Gilad-Bachrach, Navot, and Tishby. Hammer, et al. [52] propose an error calculation that assigns fractional error for samples whose hypothesis margins are less than the threshold margin, θ_{cp} . Hypothesis margins that are slightly less than θ_{cp} (near misclassifications) receive a smaller penalty than those that are less than 0 (clear misclassifications). Fractional error is determined by the ratio of hypothesis to threshold margin, θ_{cp} . That fractional error, E_{HSV} is defined

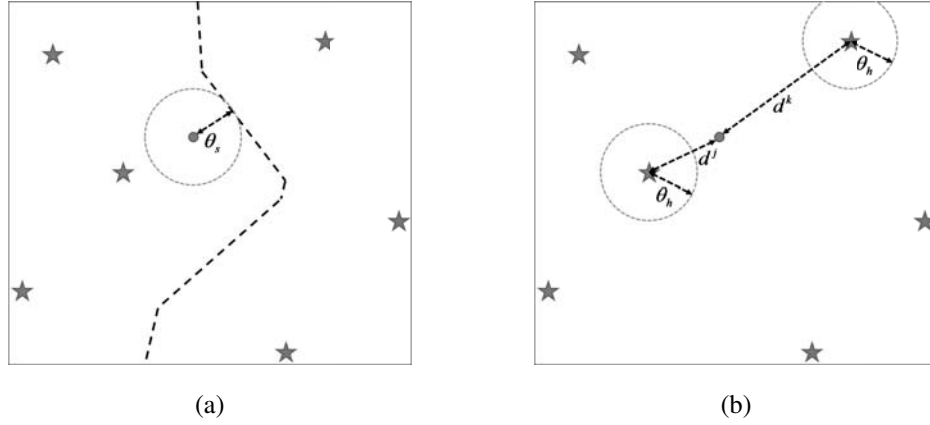


Figure 2.6: A simple two class case illustrates alternate margin definitions where blue and red stars are prototypes of opposing classes. **(a)** The sample margin, θ_s (dotted circle), is drawn around a sample point until it reaches the learned decision boundary (derived from the Voronoi tessellation) **(b)** The hypothesis margin, θ_h (dotted circle), is drawn around the nearest in-class and out-of-class prototype vectors to the sample and is calculated as half the difference between distance measures d^k and d^j . This margin calculation does not require the explicit locations of all learned decision boundaries. (Figures inspired by [3])

as [1, 38]:

$$E_{HSV}(\mathcal{X}, \mathcal{W}) = \frac{1}{M} \sum_{\mathbf{x}} \begin{cases} 1, & \text{if } \theta_h(\mathbf{x}) < 0 \\ 1 - \frac{\theta_h(\mathbf{x})}{\theta_{cp}}, & \text{if } 0 < \theta_h(\mathbf{x}) \leq \theta_{cp} \\ 0, & \text{if } \theta_{cp} < \theta_h(\mathbf{x}). \end{cases} \quad (2.19)$$

The naming of E_{HSV} reflects the authors of [38]: Hammer, Strickert, and Villmann.

2.4 Measuring Performance

The typical way to report classification accuracy is the total number of correct classifications divided by the total number of samples [21]. However, this approach always favors classes with many samples, and those classes with few samples can be completely

misclassified without any real penalty. In response, this dissertation utilizes an equal class weighting scheme where accuracy is the average of the individual class classification accuracies (or recall [53]), ensuring under represented classes get a fair vote in system performance.

2.5 Summary

This chapter introduced LVQ as a competitive, prototype-based learner, and described its initialization and learning strategy. Variants of LVQ were presented, to include the GLVQ family of algorithms, which descend a cost function to influence prototype updates. The GRLVQI algorithm was the final LVQ variant to be introduced. GRLVQI offers benefits for uneven data distributions, promotes better utilization of prototypes, and offers feature reduction capabilities. Due to these pertinent attributes, GRLVQI forms the basis for the work in this dissertation.

The Mean, Closest, PC, and S-Cost DLVQ algorithms were reviewed from the literature. Of these DLVQ algorithms, PC stands out for its use of cluster regions to describe samples in the data space and its use of a split function to redistribute prototypes along a cluster’s direction of largest variance. S-Cost is also noted for its cost minimization approach that focuses on a random sampling of the data.

LVQs generalization error was shown to be upper-bounded by the sum of a VC-dimension theoretical term, and a margin-based empirical term. Equation 2.15 underscores the key theory in this dissertation, and connects both an overall reduction in the number of prototypes, and better learned positions of prototypes to promising strategies for improved LVQ generalization and accuracy.

The next Chapter proposes novel methods for dynamic prototype addition, many of which incorporate promising aspects of cost-minimization or clustering. The E_{CGNT} and E_{HSV} error definitions are utilized to help quantify generalizability of the dynamic methods.

III. Methodology

To Evaluate the bias/generalization trade-off in LVQs, this chapter presents a framework with Network Structure Modification (NSM). Four approaches to NSM are shown through seven algorithms. They are evaluated with a novel *Accuracy-Generalization Surface* (AGS) that offers a holistic conceptualization of accuracy and generalization error.

3.1 Framework for DLVQs

Figure 3.1 presents the framework for Dynamic LVQ (DLVQ), which consists of Initialization, Learning, NSM, and Evaluation. In this framework, NSM captures the dynamic portion of DLVQ. The first stage is initialization. One prototype per class is placed at a random sample location within the class, or randomly over the data space (the latter is chosen in this research effort).

The learning stage incorporates the standard three LVQ learning phases (Competition, Winner Selection, and Synaptic Adaptation) to update prototypes (Section 2.1.1). Prototype learn rates, α^w , are initialized with large coarse values to allow larger weight updates at the onset of training and decreases throughout training until reaching a concluding stable fine learn rate allowing for convergence [54]. After reaching a learning criteria, such as completing a fixed number of training iterations, DLVQs are given an opportunity to add a new prototype as part of the *Network Structure Modification* (NSM) stage.

The NSM stage characterizes the dynamic and adaptive component that distinguish DLVQ approaches [1]. NSM has two phases: *region identification* and *prototype insertion*.

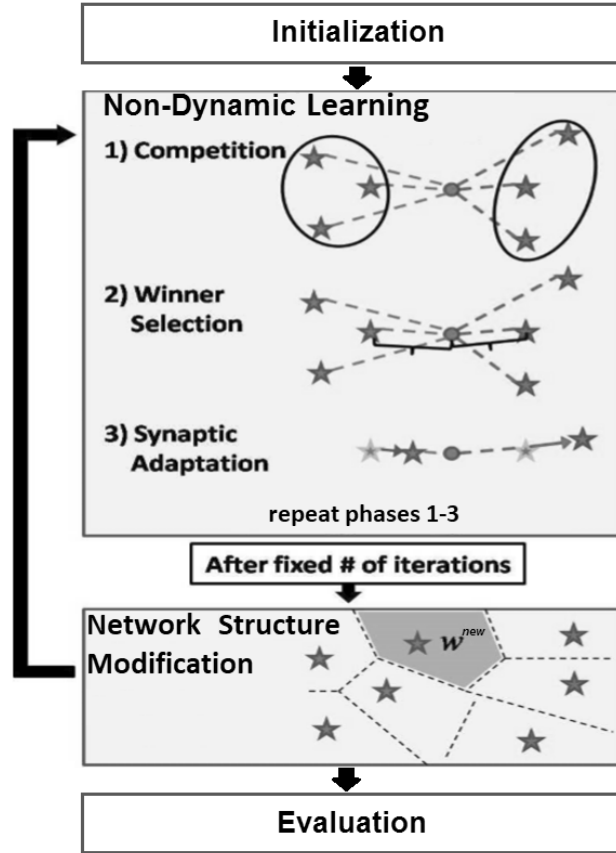


Figure 3.1: Dynamic Learning Vector Quantization (DLVQ) Framework.

3.1.1 Region Identification.

The region identification phase determines which region receives a new prototype. A region may represent a static, contiguous, partitioning of the data space, a collection of samples sharing a common attribute, or any other condition that provides an area of emphasis. Regions are not restricted to a rigid class structure, they may include representation of multiple classes. Regions carry with them implications of size, either as the proportion of the data space that they cover, or the number of samples that they encompass. This research utilizes four different region representations that occur over: (1) labeled classes, (2) 2-class decision boundaries, (3) one or more clusters, and (4) selected

samples across the global sample pool. These four region representations are illustrated in Fig. 3.2.

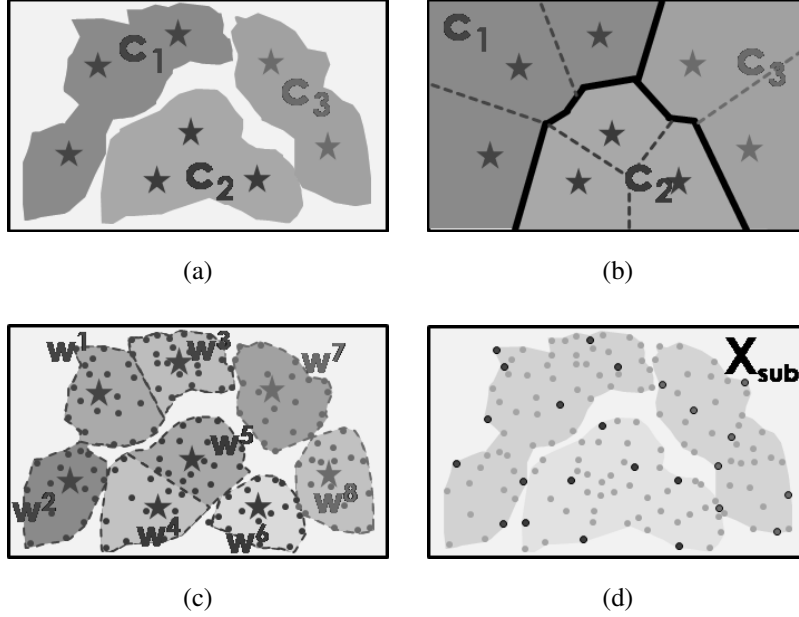


Figure 3.2: Four region representations correspond to dynamic approaches of: simple, (targeted) cost-minimizing, clustering/hybridized, and (global) cost-minimizing. Prototypes are shown as stars and samples as dots. **(a)** Regional *class membership* is shown with shaded red, green, and blue backgrounds to represent the footprint of samples sharing the same class label $y = c$. **(b)** A Voronoi tessellation quantizes the space according to nearest prototype; shaded regions show predicted class or $z = c$ values. Misclassified samples such that $y \neq z$ populate either side of 2-class boundary regions (dark lines). **(c)** *Clusters* are subclass regions defined by samples sharing a common nearest in-class prototype, w . Class membership, $y = c$, is represented by various shades of red, blue, and green backgrounds. **(d)** A *random sampling* (dark points) over a global pool shows an abstract region, X_{sub} , that includes multiple class memberships (shaded red, green, and blue backgrounds).

The critical aspect of the region identification phase is the *identification* of either a problem area, or an area that has the “best” opportunity to benefit from a change in prototype resources, commonly determined through a cost or accuracy performance measure. The structure and size of the identified region may greatly vary by the dynamic method, its underlying measure, heuristic, or general approach that guides the assessment of need or anticipated utility of future prototype allocations or deletions. Depending on the region structure, this phase has the potential to determine, or strongly influence, the class label of the new prototype. For NSMs that do not implement deletions of prototypes, this represents a permanent change to the network.

3.1.2 *Prototype Insertion.*

The prototype insertion phase specifies the position to add (or remove) a prototype, w^{new} , within the identified region. While region identification can be thought of as defining the general problem or opportunity, prototype insertion represents the specific strategy to incorporate a new prototype into the network structure.

Insertion approaches in this research focus on representation of region samples and decision strategies that provide better representation of samples in class regions (simple methods), cluster regions (clustering methods), better definition of decision boundaries (cost-minimizing methods), or a balanced approach targeting clusters and boundaries (hybrids). These approaches can produce smooth, conservative transitions to more aggressive and disruptive modifications that result in strong reactions in subsequent prototype learning.

After a prototype has been added (or deleted), the learning stage restarts as shown in Fig. 3.1. Prototypes are re-initialized in their current positions and relevance weights are maintained, but the prototype learn rates are reset to allow the prototype network to re-adjust to the recent insertion (or deletion).

As the unique aspects of DLVQs occur in the NSM stage, they are hereafter referred to as *NSM methods*.

3.2 Novel Network Structure Modification (NSM) Methods

This research introduces seven novel NSM methods within four broad dynamic approach categories: simple [46], cost-minimizing [47], clustering [45], and hybrid [24].

3.2.1 Simple: Sufficiency and Efficiency (Near-Mean).

It is possible that simplistic dynamic NSM methods may offer sufficient classification accuracy with a reduced number of prototypes for select applications and data sets, representing an *Occam's razor* strategy that when in doubt, the simplest option is preferred [55]. Such an approach is typified by the Mean method that insert prototypes based on misclassified sample locations. However, the Mean method as described, has the potential to inhibit effective prototype insertion in high dimensional data, where few samples with respect to the number of dimensions can lead to a sparsely populated data space. While a distribution of samples may actually lie on a manifold inside this sparsely populated space, the straightforward approach of computing an average position may result in a poorly interpolated location. Poor placement may also result if misclassified samples surround a selected class, the resulting placement would be at the class center, instead of targeting misclassified samples as designed.

The Near-Mean method [24] offers a remedy for poor interpolation, while preserving the simplicity of the approach. Similar to the Mean method, region identification is determined by the class, c , with the highest percentage of misclassified samples. They are inserted at the misclassified sample, $\mathbf{x}_{c'}^m$, that has the smallest distortion from the mean, $\overline{\mathcal{X}_{c'}}$:

$$\mathbf{w}^{new} = \underset{\mathbf{x}_{c'}^m \in \mathcal{X}_{c'}}{\operatorname{argmin}} \left\{ d\left(\mathbf{x}_{c'}^m, \overline{\mathcal{X}_{c'}}\right) \right\}. \quad (3.1)$$

The Near-Mean method is shown in Algorithm 5.

Algorithm 5 One iteration of Near-Mean

```
1: for  $\mathbf{x}^m \in \mathcal{X}$  do                                ▶ Loop through all samples, where  $m = [1 \dots M]$ 
2:   if  $y^m \neq z^m$  then                                ▶ If misclassified
3:      $c \leftarrow y^m$                                 ▶ Identify the true class
4:      $M_{c'} \leftarrow M_{c'} + 1$                         ▶ Increment misclassifications for class  $c$ 
5:      $\mathcal{X}_{c'} \leftarrow \mathbf{x}^m$                     ▶ Add misclassified sample to the set
6:   end if
7: end for
8: Find class,  $c$ , with max misclassifications (Eqn. 2.11)
9: Insert  $\mathbf{w}^{new}$  at average position of  $\mathcal{X}_{c'}$  (Eqn. 3.1)
```

3.2.2 Cost-Minimizing.

Results in [24] show excellent classification performance from dynamic methods that guide new prototype insertions through cost-minimization. Methods with this approach directly target the same cost function utilized in GLVQ's gradient descent. This synergy motivates the introduction of three novel cost-minimizing NSM methods: Misclassified Cost (M-Cost), Average Misclassified Cost (Avg-M-Cost), and Sampling with average Misclassified Cost (SM-Cost).

3.2.2.1 Misclassified Cost.

Misclassified Cost (M-Cost) [24], pursues a targeted cost minimization approach by focusing on misclassified samples. Equation 2.2 shows that all samples, not at the same position as a prototype of the same label, contribute to the total cost. However, misclassified samples represent the largest cost contributions and are the only samples that ultimately detract from classification accuracy. During region identification, M-Cost identifies the most costly two-class boundary region by counting the influence of misclassified samples. Figure. 3.3 illustrates this process with a simple 3-class example with class labels: blue,

red, and green. Since misclassifications along any two-class border can originate from either side (*i.e.* blue classified as green or green classified as blue), the sum of the costs from both sides is used. Conceptually, M-Cost accumulates the cost contribution from each misclassified sample, $S(\mathbf{x}_{c'}^m, \mathcal{W})$ (Eqn. 2.2) in a confusion matrix, Φ , (Fig. 3.3) where row and column entries of Φ are assigned according to “true class”, y^m (row), and “nearest” or assigned class, z^m (column), such that:

$$\Phi(y^m, z^m) = S(\mathbf{x}_{c'}^m, \mathcal{W}). \quad (3.2)$$

Complex relationships may exist between classes, and the confusion matrix is not guaranteed to be symmetric. Complementary confusion matrix entries are merged according to true class and nearest class. For hypothetical class labels, c_1 and c_2 , complementary confusion matrix entries are merged as $\Phi(c_2, c_1) + \Phi(c_1, c_2)$. The merging task can be accomplished by adding Φ with its matrix transpose, Φ' . For example, in Fig. 3.3, the complementary misclassifications over the red-green decision boundary results in the highest total cost.

If c_1 and c_2 represent class labels red and green from Fig. 3.3, the first set of misclassified samples is defined as: $\mathcal{X}_{c_2|c_1}$ where $\mathcal{Y} = c_1$ and $\mathcal{Z} = c_2$ and the second set as: $\mathcal{X}_{c_1|c_2}$ where $\mathcal{Y} = c_2$ and $\mathcal{Z} = c_1$. $\mathcal{X}_P = \mathcal{X}_{c_2|c_1} \cup \mathcal{X}_{c_1|c_2}$ now form a pool of candidate prototype locations, with $p = [1 \dots P]$ as the sample index, with class labels $\mathcal{Y}^P$.

The prototype insertion phase begins as M-Cost temporarily adds each new candidate prototype, $\mathbf{x}^p$, (one by one) to the original prototype network, $\mathcal{W}$. Each merged network is identified as $\mathcal{W} \cup \mathbf{x}^p$, P separate configurations in all. For each configuration, the total cost is calculated, or *evaluated*, over the same set of misclassified samples, $\mathcal{X}_P$. To avoid ambiguity of notation, our evaluation pool is relabeled as $\mathcal{X}_\tau$, with sample $\mathbf{x}^t$ while the candidate pool retains the original $\mathbf{x}^p \in \mathcal{X}_P$. Costs are calculated as:

$$S(\mathcal{X}_\tau, \mathcal{W} \cup \mathbf{x}^p) = \sum_{\mathbf{x}^t \in \mathcal{X}_\tau} f(\mu(\mathbf{x}^t)), \quad (3.3)$$

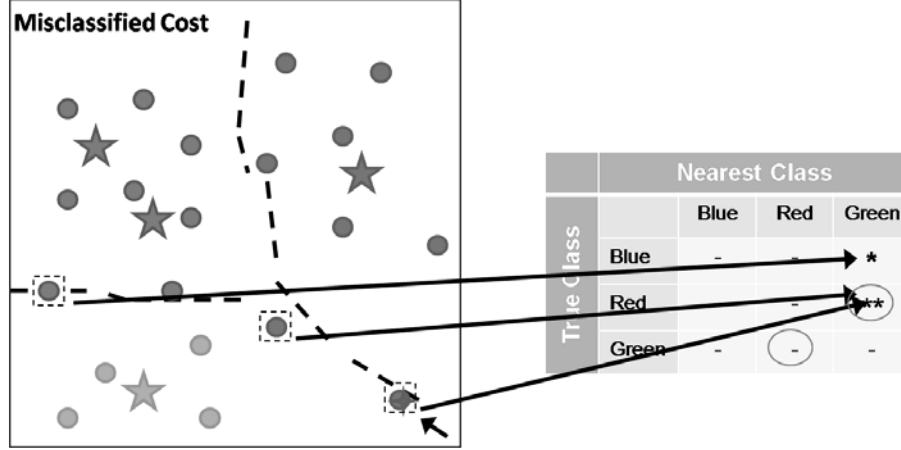


Figure 3.3: A simple three class problem illustrates cost contributions of misclassified samples in a confusion matrix. **(left)** Samples are circles and prototypes are stars, with class membership, y^m , shown by color (blue, red, or green). Assigned or nearest class, z^m , is determined by the decision boundary (dotted lines); three misclassified samples are outlined with a dotted box. **(right)** Cost contributions (Eqn. 2.2) of the misclassified samples are asterisks in the confusion matrix as a proxy for their actual value. As the red-green boundary represents the greatest total cost for misclassified samples (Eqn. 2.2), they identify region to receive an additional prototype and the misclassified samples form a pool of candidate prototype positions.

where $\mathcal{W} \cup \mathbf{x}^p$ are the prototypes used to determine values of $\mu(x)$. The candidate location, $\mathbf{x}^p$, that when combined with $\mathcal{W}$, leads to the lowest overall cost of $\mathcal{X}_\tau$, determines the new prototype resource as:

$$\mathbf{w}^{new} = \underset{\mathbf{x}^p \in \mathcal{X}_p}{\operatorname{argmin}} \left\{ S(\mathcal{X}_\tau, \mathcal{W} \cup \mathbf{x}^p) \right\}, \quad (3.4)$$

and is assigned to the class label $y^{new} = y^p$ associated with the selected prototype $\mathbf{x}^p$. The operation of the M-Cost method is shown in Algorithm 6.

Algorithm 6 One iteration of M-Cost

```
1: for  $\mathbf{x}^m \in \mathcal{X}$  do                                 $\triangleright$  Loop through all samples, where  $m = [1 \dots M]$ 
2:   if  $y^m \neq z^m$  then                                 $\triangleright$  If misclassified
3:      $\mathcal{X}_{c_2|c_1} \leftarrow \mathbf{x}^m$  where  $y^m = c_1, z^m = c_2$    $\triangleright$  Form sets of misclassified samples by
                                                                    assigned, true class membership
4:      $\Phi(y^m, z^m) \leftarrow S(\mathbf{x}^m, \mathcal{W})$    $\triangleright$  Add existing sample costs (prior to NSM phase)
5:   end if
6: end for
7:  $c_1, c_2 \leftarrow \text{max entry of } \Phi + \Phi'$   $\triangleright$  Find costliest 2-class boundary (for misclassifications)
8:  $\mathcal{X}_P \leftarrow \mathcal{X}_{c_2|c_1} \cup \mathcal{X}_{c_1|c_2}$    $\triangleright$  Identify region, candidate prototype locations
9: for  $\mathbf{x}^p \in \mathcal{X}_P, p = [1 \dots P]$  do   $\triangleright$  Test candidates as  $\mathcal{W} \cup \mathbf{x}^p$ 
10:    $S(\mathcal{X}_\tau, \mathcal{W} \cup \mathbf{x}^p)$   $\triangleright$  Cost of evaluated samples when testing candidate  $\mathbf{x}^p$  (Eqn. 3.3)
11: end for
12:  $\mathbf{w}^{new} \leftarrow \mathbf{x}^p$  that yields the lowest total cost   $\triangleright$  New prototype insertion
```

3.2.2.2 Average Misclassified Cost.

Average Misclassified Cost (Avg-M-Cost) balances the cost minimization between classes. The region identification phase utilizes the same mechanics as M-Cost to accumulate the cost contributions from each misclassified sample in the training set. However, after the cost contributions are organized into the confusion matrix Φ , each row is divided by the total number of in-class training samples, $M_{c^{all}}$, such that

$$M_{c^{all}} = \sum_{y^m=c \in \mathcal{Y}} 1. \quad (3.5)$$

The resulting confusion matrix $\tilde{\Phi}$, shows the cost contributions averaged over the number of training samples in each class:

$$\tilde{\Phi} = \begin{bmatrix} 1/M_{c_1^{all}} & \dots & 0 & \dots & 0 \\ \vdots & \ddots & & & \vdots \\ 0 & & 1/M_{c_m^{all}} & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1/M_{c^{all}} \end{bmatrix} \times \Phi. \quad (3.6)$$

Similar to M-Cost, the confusion matrix is combined with its transpose to identify the region that will receive a new prototype. While M-Cost identifies the 2-class decision boundary with the highest total cost, Avg-M-Cost identifies the 2-class decision boundary with the highest average cost. As with M-Cost, the misclassified samples belonging to the identified region define the set, $\mathcal{X}_P$. These samples also define the prototype candidate pool for a new prototype.

Prototype insertion is determined in an identical manner as in M-Cost. The cost for the pool of samples, $\mathcal{X}_P$, is calculated over a merged prototype network of $\mathcal{W} \cup \mathbf{x}^p$, where $\mathbf{x}^p \in \mathcal{X}_P$ is a candidate location. A new prototype, w^{new} , is inserted at the candidate location that results in the smallest cost. The Avg-M-Cost method is shown in Algorithm 7.

Algorithm 7 One iteration of Avg-M-Cost

- 1: **for** $\mathbf{x}^m \in \mathcal{X}$ **do** ▷ Loop through all samples, where $m = [1 \dots M]$
 - 2: **if** $y^m \neq z^m$ **then** ▷ If misclassified
 - 3: $\mathcal{X}_{c_2|c_1} \leftarrow \mathbf{x}^m$ where $y^m = c_1, z^m = c_2$ ▷ Form sets of misclassified samples by
assigned, true class membership
 - 4: $\tilde{\Phi}(y^m, z^m) \leftarrow S(\mathbf{x}^m, \mathcal{W}) / M_{c_1^{all}}$ ▷ Add to confusion matrix average (existing)
costs over # samples/class (Eqn. 3.6)
 - 5: **end if**
 - 6: **end for**
 - 7: $c_1, c_2 \leftarrow \text{max entry of } \tilde{\Phi} + \tilde{\Phi}'$ ▷ 2-class boundary with highest average cost
 - 8: $\mathcal{X}_P \leftarrow \mathcal{X}_{c_2|c_1} \cup \mathcal{X}_{c_1|c_2}$ ▷ Identify region, candidate prototype locations
 - 9: **for** $\mathbf{x}^p \in \mathcal{X}_P, p \in (1 \dots P)$ **do** ▷ Test candidates as $\mathcal{W} \cup \mathbf{x}^p$
 - 10: $S(\mathcal{X}_\tau, \mathcal{W} \cup \mathbf{x}^p)$ ▷ Evaluated sample cost with addition of candidate $\mathbf{x}^p$ (Eqn. 3.3)
 - 11: **end for**
 - 12: $\mathbf{w}^{new} \leftarrow \mathbf{x}^p$ that yields the lowest total cost ▷ New prototype insertion
-

3.2.2.3 Sampling with average Misclassified Cost.

Sampling with average Misclassified Cost (SM-Cost) combines components from S-Cost [47] (in Chapter 2) and the Avg-M-Cost methods. The theoretical backing for S-Cost’s global total cost minimization indicates the potential for local optimal solutions. Experimentally, cases are found where S-Cost rapidly establishes good prototype allocation as indicated by comparatively strong early performance. These initial prototype insertions are reasonable allocations that typically remedy prominent sources of error. However, S-Cost often struggles to provide finer definition of decision boundaries that may be achievable by more local approaches. While Avg-M-Cost may not experience may expend increased training time pursuing more local approaches, its targeted cost minimization of key decision boundaries is anticipated to provide the local definition that S-Cost appears to lack. A combination of these two extremes is proposed with a two stage cost minimization approach: *coarse prototype allocation* with S-Cost, followed by *refined prototype allocation* with Avg-M-Cost. This concept has a similar notion as Kohonen’s Optimized LVQ (OLVQ) [19] that allows for diverse learning strategies to promote faster convergence. The hybrid cost-minimizing method in Algorithm 8.

3.2.3 Clustering.

Clustering is an unsupervised learning approach that groups (un-labeled) samples based on their similarity, or natural grouping (see Fig. 3.4). This contextual understanding of natural groupings within classes, un-aided by *a priori* probability density functions (PDF)s or knowledge of their nearest in-class prototype vector, offers promise in DLVQs. As demonstrated in [24], and in additional experimentation, the adaptation of Principal Components (PC) shows strong classification performance. It is postulated that performance benefits are due to added context PC draws from clustering the samples. While some clustering algorithms, such as DB-Scan [56], attempt to self determine the number of clusters in a data set, they do not necessarily bound the number of prototypes they allocate.

Algorithm 8 One iteration of SM-Cost

```
1:  $T_{max} \leftarrow$  User defined       $\triangleright$  Establish maximum number of prototype addition stages
2:  $T \leftarrow 1$                    $\triangleright$  Initialize counter for prototype addition stages
3: while  $T \leq T_{max}$  do                                 $\triangleright$  Coarse stage
4:   if  $T \leq \frac{T_{max}}{2}$  then
5:     Follow S-Cost method for Network Structure Modification (Algorithm 4)
6:   else if  $T > \frac{T_{max}}{2}$  then                             $\triangleright$  Refinement stage
7:     Follow Avg-M-Cost method for Network Structure Modification (Algorithm 7)
8:   end if
9:    $T \leftarrow T + 1$                                  $\triangleright$  Increment prototype addition stage counter
10: end while
```

The clustering efforts in this dissertation focus on re-allocating an existing prototype, $\mathbf{w}^{old}$, and inserting an additional (new) prototype, $\mathbf{w}^{new}$, within an identified region as part of the prototype insertion phase.

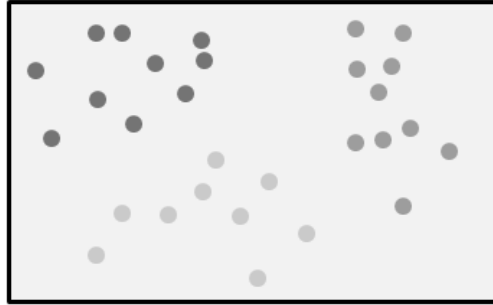


Figure 3.4: An illustration of clustering: assigning labels (purple, orange, green) according to learned natural groupings. Similar techniques can be applied to redistribute prototypes according to natural segmentations within samples of a particular class.

The novel Split-Medoid method adopts the practice of redistributing or splitting prototypes, as performed in PC. Significantly, the *K-Medoids* clustering algorithm [21] is employed to guide the redistributed and insertion of prototypes. Aspects of PC are mirrored, and samples that share common prototypes as net-winner or in-class-winner, are tracked. In Section 2.2.3, a cluster is defined as a group of individual samples $\mathbf{x}_q^m \in \mathcal{X}_q$ that share a nearest in-class prototype, $\mathbf{w}_q^m$ (and adopt the same index q). Split-Medoid tracks the frequency each cluster, $\mathcal{X}_q$, is identified to receive an additional prototype (or whose associated prototype, $\mathbf{w}^q$ is split) as H^q . Clusters are assigned a gain score G^q as seen in Eqn. 2.13 [45], which is replicated below as Eqn. 3.7 for completeness:

$$G^q = \frac{M_{q'}}{M_c \cdot H^q}. \quad (3.7)$$

The cluster $\mathcal{X}_q$ with the largest G^q gain score receives an additional prototype.

Split-Medoid's prototype insertion phase seeks to improve the quality (*i.e.*, natural grouping) of identified sample clusters through the redistribution of prototypes. *K-medoids* clustering [21] is applied to guide the split of prototype, $\mathbf{w}^q$, associated with sample cluster, $\mathcal{X}_q$. *K-medoids* clustering operates by initially fitting samples with K nodes as potential cluster centers. $K = 2$ nodes are chosen (1 redistributed, 1 new) that adopt initial sample positions randomly selected from $\mathcal{X}_q$. These nodes are referred to as $\mathbf{w}^{old}$ and $\mathbf{w}^{new}$. Samples in $\mathcal{X}_q$ are then re-labeled as part of either $\mathcal{X}_{old}$ or $\mathcal{X}_{new}$ reflecting their nearest node. Node positions $\mathbf{x}^{old}$ and $\mathbf{x}^{new}$ are then updated to reflect the medoid or sample position nearest each group average: $\overline{\mathcal{X}_{old}}$ or $\overline{\mathcal{X}_{new}}$. These two steps of (1) relabeling samples according to nearest node, and (2) updating node positions to the medoid position of their labeled samples are repeated until reaching a converged solution. Similar to the Near-Mean algorithm, the use of medoid instead of centroid or mean positions (as in K-means) may present a better approach in high dimensional data, in which distortion measures (such as Euclidean distance) can be dominated by a single dimension if disproportionately large. Besides avoiding potential problems from poor interpolation, a K-medoid approach

minimizes the search space necessary (when compared to K-means), necessitating fewer iterations before achieving convergence. The operation of the Split-Medoid method is shown in Algorithm 9.

Algorithm 9 One iteration of Split-Medoid

- 1: **for** $\mathbf{x}^m \in \mathcal{X}$ **do** ▷ Loop through all samples, where $m = [1 \dots M]$
 - 2: $c \leftarrow y^m, \mathcal{X}_{c^{all}} \leftarrow \mathbf{x}^m$ ▷ Identify, track class membership
 - 3: **for** $\mathbf{w}^q \in W$ **do** ▷ Loop through all prototypes, where $q = [1 \dots Q]$ to form sets
 - 4: $\mathcal{X}_q \leftarrow \mathbf{x}^m$ ▷ If $\mathbf{w}^q$ in-class winner
 - 5: $\mathcal{X}_{q^{net}} \leftarrow \mathbf{x}^m$ ▷ If $\mathbf{w}^q$ net winner
 - 6: $\mathcal{X}_q \leftarrow \mathbf{x}^m$ ▷ If $\mathbf{w}^q$ in-class, but not net winner (misclassified samples)
 - 7: **end for**
 - 8: **end for**
 - 9: $\forall c \in \{1 \dots C\}, \quad M_c = \text{card}(\mathcal{X}_{c^{all}})$ ▷ Cardinality of each class grouping
 - 10: $\forall q \in \{1 \dots Q\}, \quad \text{Calculate } G^q \text{ from } M_c, M_{q'}, H^q$ ▷ Gain function score (Egn. 2.13)
 - 11: $\tilde{q} \leftarrow \underset{q}{\text{argmax}} (G^1, \dots, G^q, \dots, G^Q)$ ▷ Identify region via maximum gain function score
 - 12: $\mathbf{w}^{old}, \mathbf{w}^{new} \leftarrow \mathcal{X}_{\tilde{q}}$ ▷ 2 Initial random assignments from cluster
 - 13: **repeat** ▷ K-Medoid iterations
 - 14: $\mathcal{X}_{old} \leftarrow \text{samples nearest } \mathbf{w}^{old}, \quad \mathcal{X}_{new} \leftarrow \text{samples nearest } \mathbf{w}^{new}$
 - 15: $\mathbf{w}^{old} \leftarrow \text{Medoid of } \mathcal{X}_{old}, \quad \mathbf{w}^{new} \leftarrow \text{Medoid of } \mathcal{X}_{new}$
 - 16: **until** $\mathbf{w}^{old}$ and $\mathbf{w}^{new}$ each converge ▷ Final prototype insertion positions
 - 17: $H^{old} \leftarrow H^{\tilde{q}} + 1, \quad H^{new} \leftarrow 1$ ▷ Update split counter
-

3.2.4 Hybrid.

In the NSM methods presented thus far, cost-minimizers are anticipated to increase classification accuracy by focusing on poorly defined decision boundaries. Clustering methods are expected to better represent data distributions and improve generalization. Aspects of both approaches are combined to create a hybridized approach with potential for improved classification and generalization.

Clustering, as stated in Section 2.2.3, is used to group un-labeled samples, whereas clustering *metrics* are used to evaluate the “goodness” or reliability of clusters, illustrated in Fig. 3.5. While there are a myriad of available clustering metrics, the Silhouette metric is chosen for its comparison of cluster tightness to cluster separation [57]. For a single sample, $\mathbf{x}^m \in \mathcal{X}_q$, a silhouette coefficient contrasts the average distortion between samples in the same cluster, $a(\mathbf{x}^m)$, with the lowest average distortion to a neighboring cluster, $b(\mathbf{x}^m)$, [58]:

$$sil(\mathbf{x}^m) = \frac{b(\mathbf{x}^m) - a(\mathbf{x}^m)}{\max\{a(\mathbf{x}^m), b(\mathbf{x}^m)\}}, \quad (3.8)$$

where $-1 \leq sil(\mathbf{x}^m) \leq 1$. For each sample, silhouette coefficients close to 1 indicate strong similarity to the assigned cluster, while silhouette coefficients close to -1 indicate a stronger similarity to a neighboring cluster. The average of the $sil(\mathbf{x}^m)$ silhouette coefficients in sample cluster, $\mathcal{X}_q$, or *cluster score* is:

$$SIL(\mathcal{X}_q) = \frac{1}{M_q} \sum_{\mathbf{x}^m \in \mathcal{X}_q} sil(\mathbf{x}^m), \quad (3.9)$$

where $m = [1 \dots M_q]$ are the sample indices. The cluster score is influenced by its boundary profile and indicates the appropriateness or sufficiency of the existing sample cluster.

3.2.4.1 Silhouette-Full.

Silhouette-Full (Silhouette-F) evaluates the quality of clusters, and as part of the region identification phase, establishes those that could most benefit from an additional prototype. Each cluster, $\mathcal{X}_q$, is defined by samples sharing a nearest in-class prototype vector, such that

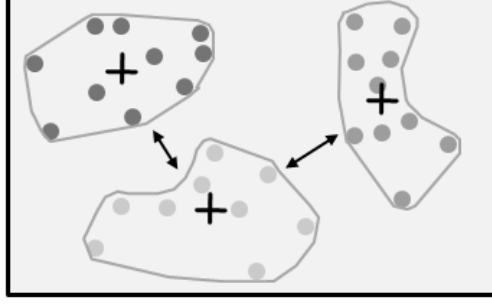


Figure 3.5: An illustration of cluster metrics: clusters are evaluated based on their “goodness” or reliability. Measures such as inter-cluster (blue bounding geometries) and intra-cluster (black bidirectional arrows) distortions are used to assess the suitability of learned groupings.

$\mathbf{w}^j = \mathbf{w}^q$ (seen in Fig. 2.4b). During region identification, the average distortion of a sample, $\mathbf{x}^m$, to other samples in $\mathcal{X}_q$ is calculated as $a(\mathbf{x}^m)$ from Eqn. 3.8. Silhouette-Full is primarily concerned with the exterior boundaries of class regions as in Fig. 2.4a. As such, the $b(\mathbf{x}^m)$ term from Eqn. 3.8 is adapted so as not to penalize clusters for insufficient separation from clusters belonging to the same class. A modified $\widehat{b}(\mathbf{x}^m)$ term instead reflects the distortion between $\mathbf{x}^m$ and the single nearest sample of a different class as:

$$\widehat{b}(\mathbf{x}^m) = \min\{d(\mathbf{x}^m, \mathcal{X} \setminus \mathcal{X}_{c^{all}})\}, \quad (3.10)$$

where $\mathcal{X}_{c^{all}}$ is the set of all samples with class label c . The cluster label of the nearest foreign sample is tracked as $k^m = b$ (or $\mathcal{K}_q$ for all samples $\mathbf{x}^m \in \mathcal{X}_q$) with the same sample index m . The adapted $\widehat{b}(\mathbf{x}^m)$ minimum distortion is potentially obtained through sorted lists of prior computations, whereas Rousseeuw’s [58] original $b(\mathbf{x}^m)$ term requires comparisons with every other sample that cannot be similarly obtained [56]. Similar to PC, Silhouette-F tracks the frequency that cluster, $\mathcal{X}_q$ (with corresponding prototype, $\mathbf{w}^q$) is identified to receive an additional prototype as H^q . In response to configuration differences between PC and *SIL* scores, Silhouette-F uses H^q values as exponents, with a base u to effectively

bias against the frequent selection of any one cluster. Clusters are assigned relative *health scores*, HW , that combine the overall cluster score, SIL , and frequency of past selection as:

$$HW^q = SIL(\mathcal{X}_q) \times u^{H^q-1} \quad (3.11)$$

where the lowest scoring $\mathcal{X}_P$ is the region identified for prototype addition, and

$$P = \underset{q}{\operatorname{argmin}}\{HW^q\}. \quad (3.12)$$

Similar to PC and Split-Medoid, Silhouette-F allocates new LVQ network resources to insufficiently represented clusters instead of being strictly driven by misclassifications.

Silhouette-F's prototype insertion phase mirrors that of M-Cost and uses pools of candidate samples to down-select the specific location for prototype insertion. The location for prototype insertion is the candidate position leading to the smallest total cost as calculated over a set of samples that are the *evaluation pool*. While M-Cost uses the same test and evaluation pool, $\mathcal{X}_P$, Silhouette-F augments the evaluation pool with all samples from $\mathcal{X}_P$'s *neighboring* foreign cluster. This is accomplished by first determining the detractive influence, G' , of each foreign cluster (with cluster label b) as:

$$G'^b = \sum_{\mathbf{x}^m \in \mathcal{X}_q | k^m = b} \left\{ sil(\mathbf{x}^m) \right\}, \quad (3.13)$$

with silhouette coefficients and $\mathcal{K}_q$ labels as determined in region identification. The neighboring foreign cluster is the largest detractor in silhouette coefficient values and results in the lowest G' value. Its label is determined as:

$$B = \underset{b' \in \mathcal{K}_q}{\operatorname{argmin}} \left\{ G'^{b'} \right\}, \quad (3.14)$$

and $\mathcal{X}_B$ is combined with $\mathcal{X}_P$ to form the evaluation pool: $\mathcal{X}_\tau = \mathcal{X}_P \cup \mathcal{X}_B$. As with M-Cost, (Section 3.2.2.1), candidate prototypes, $\mathbf{x}^p$, are individually merged with the prototype network as, $\mathcal{W} \cup \mathbf{x}^p$. For each merged prototype network, total costs, s_p , are calculated as:

$$s_p = \sum_{\mathbf{x}^t \in \mathcal{X}_\tau} S(\mathbf{x}^t, \mathcal{W} \cup \mathbf{x}^p). \quad (3.15)$$

The candidate location, $\mathbf{x}^p$, that leads to the lowest overall cost, determines the new prototype resource as:

$$\mathbf{w}^{new} = \underset{\mathbf{x}^p}{\operatorname{argmin}} \left\{ S(\mathcal{X}_\tau, \mathcal{W} \cup \mathbf{x}^p) \quad \forall \mathbf{x}^p \in \mathcal{X}_P \right\}, \quad (3.16)$$

with corresponding class label y^{new} . This is the label of the candidate location $\mathbf{x}^p$ that minimizes Eqn. 3.16). The operation of the Silhouette-F method is shown in Algorithm 10.

Algorithm 10 One iteration of Silhouette-F

- 1: **for** $\mathbf{x}^m \in \mathcal{X}$ **do** ▷ Loop through all samples $m = [1 \dots M]$ to build sample clusters
 - 2: $c \leftarrow y^m, \mathcal{X}_{c^{all}} \leftarrow \mathbf{x}^m$ ▷ Identify, track class membership
 - 3: $\forall \mathbf{w}^q \in \mathcal{W}, \mathcal{X}_q \leftarrow \mathbf{x}^m$ ▷ Add $\mathbf{x}^m$ to sample cluster when $\mathbf{w}^q$ in-class winner
 - 4: **end for**
 - 5: **for** $\mathbf{w}^q \in \mathcal{W}$ **do** ▷ Loop through all prototype vectors
 - 6: **for** $\mathbf{x}^m \in \mathcal{X}_q$ **do** ▷ Loop through all samples in associated sample cluster
 - 7: $a(\mathbf{x}^m) \leftarrow \overline{d(\mathbf{x}^m, \mathcal{X}_q)}$ ▷ Avg dist between cluster samples
 - 8: $b(\mathbf{x}^m) \leftarrow \min \left\{ d(\mathbf{x}^m, \mathcal{X} \setminus \mathcal{X}_{c^{all}}) \right\}$ ▷ Min dist between to nearest foreign sample
 - 9: $k^m \leftarrow b$ ▷ Cluster label of nearest foreign sample (Eqn. 3.10)
 - 10: $sil(\mathbf{x}^m) \leftarrow$ Calculate according to Eqn. 3.8
 - 11: **end for**
 - 12: $HW^q \leftarrow$ Calculate according to Eqn. 3.11 ▷ Cluster health score
 - 13: **end for**
 - 14: $P \leftarrow \underset{q}{\operatorname{argmin}} \{ HW^q \}$ ▷ Identify for prototype addition/candidate pool
 - 15: $\forall \mathbf{x}^m \in \mathcal{X} | k^m = b, G'^b \leftarrow \sum sil(\mathbf{x}^m)$ ▷ Foreign cluster influence (Eqns. 3.13, 3.14)
 - 16: $\mathcal{X}_B$ selected as $B \leftarrow \underset{b' \in \mathcal{K}}{\operatorname{argmin}} \{ G'^{b'} \}$ ▷ Nearest foreign cluster to augment evaluation pool
 - 17: $\mathcal{X}_\tau = \mathcal{X}_P \cup \mathcal{X}_B$ ▷ $\mathbf{x}^t \in \mathcal{X}_\tau$
 - 18: $\mathbf{w}^{new} \leftarrow \underset{\mathbf{x}^p}{\operatorname{argmin}} \left\{ S(\mathcal{X}_\tau, \mathcal{W} \cup \mathbf{x}^p) \right\}$ ▷ (Eqn. 3.16)
 - 19: $H^{old} \leftarrow H^{old} + 1, H^{new} \leftarrow 1$ ▷ Update selection counter
-

3.2.4.2 *Silhouette-Lite.*

Silhouette-Lite (Silhouette-L) is a variant of Silhouette-F that adopts a more computationally efficient and conservative application of the Silhouette metric to M-Cost. In the region identification phase, Silhouette-L bypasses calculating the pairwise distances between all points as part of $b(\mathbf{x}^m)$ in Eqn. 3.8. Instead of determining the nearest sample from a different class, Silhouette-L utilizes the last known distortion measures and labels to the nearest out-of-class prototype vectors (as calculated prior to the NSM stage). The cluster health score HW^q , and recipient prototype cluster $\mathbf{w}$, are then determined as described for Silhouette-F.

The prototype insertion phase for Silhouette-L involves more conservative candidate placement and evaluation pools than Silhouette-F. The candidate pool, $\mathcal{X}_P$, is determined by the samples from region identification, $\mathcal{X}_q$, that yield negative scores for $sil(\mathbf{x}^m)$ according to Eqn. 3.8. These samples indicate a closer similarity to a different class than their own, bearing a resemblance to the misclassified samples in M-Cost’s candidate pool. The nearest cluster (not of class c) to points in $\mathcal{X}_P$ define $\mathcal{X}_B$ and augments the evaluation sample pool, where $\mathcal{X}_\tau = \mathcal{X}_P \cup \mathcal{X}_B$. As in Silhouette-F, the sample $\mathbf{x}^p \in \mathcal{X}_P$ that leads to the lowest total cost over $\mathcal{X}_\tau$ is the position for the new prototype, $\mathbf{w}^{new}$. The operation of the Silhouette-L method is shown as Algorithm 11.

3.2.5 *NSM Discussion.*

The 7 novel and 4 existing NSM methods presented cover the axes of *region influence* and *prototype insertion*. Region influence portrays the span and variety of samples represented. Prototype insertion portrays the strategic target or focus area for new insertions.

The region identification mechanisms, region representation, and aspects of diversity describe an NSM’s *region influence* in Fig. 3.6. Region influence is seen on a spectrum from local to global. Local, focuses on a selection that represents the best local (region)

Algorithm 11 One iteration of Silhouette-L

```

1: for  $\mathbf{x}^m \in \mathcal{X}$  do                                ▶ Loop through all samples  $m \in [1 \dots M]$  to build clusters
2:    $c \leftarrow y^m, X_{call} \leftarrow \mathbf{x}^m$           ▶ Identify, track class membership
3:    $\forall \mathbf{w}^q \in \mathcal{W}, \quad \mathcal{X}_q \leftarrow \mathbf{x}^m$       ▶ Add  $\mathbf{x}^m$  to sample cluster when  $\mathbf{w}^q$  in-class winner
4:    $b(\mathbf{x}^m) \leftarrow d(\mathbf{x}^m, \mathbf{w}^k)$  ▶ Recall distance and label of nearest out-of-class prototype
5: end for

6: for  $\mathbf{w}^q \in \mathcal{W}$  do                                ▶ Loop through all prototype clusters
7:   for  $\mathbf{x} \in \mathcal{X}_q$  do                                ▶ Loop through all samples in cluster
8:      $a(\mathbf{x}^m) \leftarrow \overline{d(\mathbf{x}^m, \mathcal{X}_q)}$         ▶ Avg dist between cluster samples
9:      $sil(\mathbf{x}^m) \leftarrow$  Calculate according to Eqn. 3.8
10:     $\mathcal{X}_p \leftarrow sil(\mathbf{x}^m) < 0$            ▶ More restrictive conditions reduces candidate pool
11:   end for
12:    $HW^q \leftarrow$  Calculate according to Eqn. 3.11    ▶ Cluster health score
13: end for

14:  $P \leftarrow \operatorname{argmin}_q \{HW^q\}$                 ▶ Identify for prototype addition/candidate pool
15:  $\forall \mathbf{x}^m \in \mathcal{X} | k^m = b, \quad G'^b \leftarrow \sum sil(\mathbf{x}^m)$  ▶ Calculate influence of foreign clusters
16:  $\mathcal{X}_B$  selected as  $B \leftarrow \operatorname{argmin}_{b' \in \mathcal{K}} \{G'^{b'}\}$  ▶ Nearest foreign cluster to augment evaluation pool
17:  $\mathcal{X}_\tau = \mathcal{X}_P \cup \mathcal{B}$                                 ▶  $\mathbf{x}^t \in \mathcal{X}_\tau$ 
18:  $\mathbf{w}^{new} \leftarrow \operatorname{argmin}_{\mathbf{x}^p} \left\{ S(\mathcal{X}_\tau, \mathcal{W} \cup \mathbf{x}^p) \right\}$ 
19:  $H^{old} \leftarrow H^{old} + 1, H^{new} \leftarrow 1$       ▶ Update selection counter

```

decision to reduce the misclassification rate. However, these considerations are for a single class. The other end, global, is characterized by searching across representative samples to determine the type of candidates that may best reduce error.

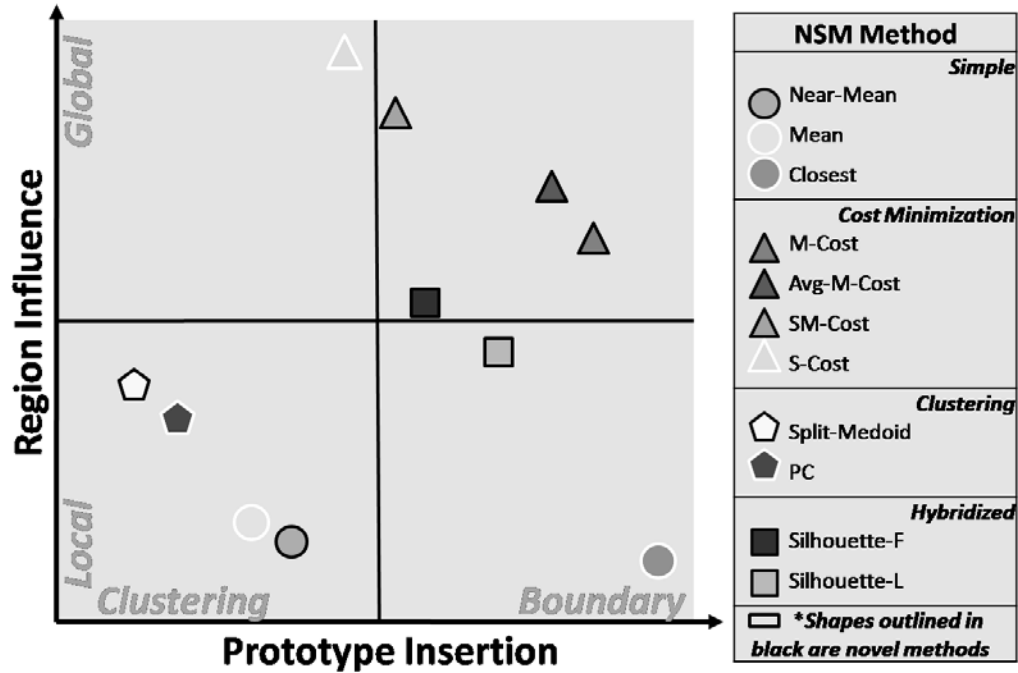


Figure 3.6: An overall view of each NSM's overall strategy is characterized over the axes of: (1) span of region influence, ranging from local to global and (2) focus of the insertion strategy, ranging from a focus on internal clustering to defining the decision boundary.

The 7 novel and 4 existing NSM algorithms vary in how they distribute the new prototypes with respect to the decision boundary. In the prototype insertion phase, PC and Split-Medoid target the interior of identified regions, making insertions to better represent the mode or distribution (representing a clustering insertion strategy). Mean, Near-Mean, and Closest target misclassified samples within their identified (class) region. Of these three methods, Closest takes the most direct approach by inserting a new prototype at the

misclassified sample that is closest to a prototype with a different class label (representing the boundary insertion strategy). Mean and Near-Mean insert a new prototype at the center or sample nearest the center of the misclassified points. The remaining NSM methods insert prototypes at the candidate location that minimizes the total cost of their evaluation pool. S-Cost, SM-Cost, and Silhouette-F do this over a broader region, either over a global subset, or including samples from a neighboring region. NSM methods such as M-Cost, M-Avg-Cost, and Silhouette-L target contested regions. However, they do so based on total cost outcomes and include the perspectives of neighboring regions.

Prototype insertion is represented on the horizontal axis of Fig. 3.6 with the extremes of clustering and boundary insertion strategies. Insertion strategies that focus additional resources to improve the decision boundaries offer higher potential classification accuracy. Clustering insertion strategies, however, promise better generalization as they seek to represent data distributions. A balance of these two strategies may represent a promising option for a more diverse set of environments. Ultimately, the performance of any particular strategy will likely depend on the appropriateness of the match to the specific data set.

3.3 Novel Evaluation and Characterization of DLVQs

Current representations of LVQ classifier performance are often reported in terms of classification accuracy, but ignore the important considerations of generalization. Accuracy is often an insufficient single measure to differentiate the performance of many modern LVQs, as they will often operate in the same range. Fundamentally, an evaluation approach is required that captures the trade-off between accuracy and generalization.

3.3.1 Accuracy-Generalization Surface (AGS).

A new Accuracy-Generalization Surface (AGS) method is presented to evaluate LVQ's accuracy/generalization trade-off. AGS captures the performance of dynamic NSM methods with a performance component, a theoretical component, and a learned component. Figure 3.7 combines the axes of classification accuracy, (empirical) margin

error and the number of utilized prototypes (the last two axes describing generalization error). The combination of the three components (prototype utilization, generalization, and empirical error) results in an embodiment of accuracy and generalization, for an LVQ algorithm. An initialized classifier would be plotted on these axes as a single point. As additional prototypes are added to the classifier, and as it is tested with different margins, it creates a surface that characterizes the performance of the NSM method over its entire operating range. This surface provides a novel approach to consider NSM method trade-space in a concise manner and is utilized in Chapter 4 to provide insights to relative performance. The following sections describe the three reference planes that comprise the AGS.

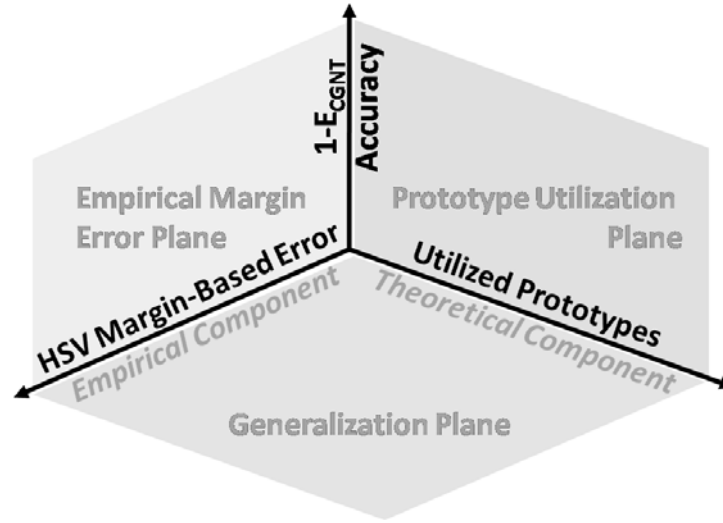


Figure 3.7: An Accuracy-Generalization Surface (AGS) view of NSM method's characteristic performance over three reference planes represent the entire span of operation.

3.3.2 *Prototype Utilization Plane.*

The prototype utilization plane represents trade-offs in accuracy over a range of changing prototype configurations. Depending on an NSM method's approach to region

identification or prototype selection, they may periodically forgo the opportunity to add a new prototype. An alternative to expressing accuracy as a function of prototype addition stage is to re-organize or re-bin the collected data as a function of prototypes that are actually added or *utilized*. While the details of this approach are further explored in Section 4.3.2, suffice it to say that a *prototype utilization* perspective captures how each NSM method is able to utilize current resources and/or capitalize on additional resources. Axes of accuracy and number of prototypes are combined to form a *prototype utilization plane* (Fig. 3.7, right). Significantly, an increased number of prototypes leads to a higher Vapnik-Chervonenkis (VC)-dimension and an associated theoretical error component (Eqn. 2.15 and Eqn. 2.16). Since generalization error bounds grow with an increased number of prototypes, LVQs that operate in a high accuracy, low prototype region of the prototype utilization plane are favored (top left corner of the prototype utilization plane in Fig. 3.7).

3.3.3 Generalization Plane.

The second pair of axes considers generalization or robustness of the classifier to new data sets with a *generalization plane*. Equation 2.15 shows the generalization error is upper-bounded by the sum of an empirical error component (learned component) and the previously mentioned theoretical error component. As such, the number of utilized prototypes is an axis for the generalization plane (as well as the prototype utilization plane). The empirical error component is represented in a second axis by E_{HSV} (Eqn. 2.19).

While the classification process is entirely focused on the current location of this decision boundary, it does not innately consider the size of the margin that those boundaries offer. The E_{HSV} margin error offers this new perspective and describes errors that result for changing margin sizes in relation to a threshold value, θ_{cp} . As the size of θ_{cp} grows, the distance between correctly classified samples and their nearest decision

boundary effectively shrinks resulting in increased E_{HSV} margin error. Classifiers with better generalization will experience less error due to the same sized increases of θ_{cp} .

As LVQs are large margin classifiers [23], this characterization of margin is fundamental to a holistic consideration of their performance. Significantly, this plane represents the entirety of the generalization error (Fig. 3.7 bottom). These combined axes gives a sense of not just how large a margin is, but at what point an additional number of prototypes lead to increased error. An NSM's ideal operating point on the generalization plane is the lowest E_{HSV} (small margin error), and the minimum number of prototypes (lowest possible VC-based theoretical error).

3.3.4 Empirical Margin Error Plane.

The final plane is represented by a pairing of E_{HSV} and accuracy. It is seen that E_{HSV} , changes as a function of θ_{cp} to extrapolate the potential for error based on the learned system configuration. However, these error definitions do not move samples or decision boundaries in actuality. As such, the true classification accuracy remains constant despite changes in the represented error. Instead, the accuracy is *re-factored* based on boundary conditions represented by changing θ_{cp} . This re-factoring captures the *realization* of the error conditions represented by θ_{cp} .

The CGNT Margin error, E_{CGNT} (Eqn. 2.18), has striking similarities with the re-factored accuracy. The E_{CGNT} error is strictly defined by scoring each sample as a 1 if it falls closer to an out-of-class prototype vector (after the margin has been adjusted) or 0 if not. Due to the binary nature of the error calculation, the CGNT margin error is the exact converse of the classification accuracy. Thus as the θ_{cp} value increases and error grows, the difference of one minus the E_{CGNT} serves as a reference classification re-factored accuracy. If θ_{cp} is treated as an independent variable, $1 - E_{CGNT}$ (re-factored accuracy), and E_{HSV} (margin error) can be related on a single pair of axes representing the *empirical margin*

error plane (Fig. 3.7, left). An ideal NSM method would occupy the lowest E_{HSV} (low margin based empirical error) and 100% accuracy ($E_{CGNT} = 0$).

3.4 Summary

This Chapter describes a framework for DLVQs that includes the incorporation of a novel NSM stage to describe the operation of dynamic approaches based on a region identification phase and a prototype insertion phase. The seven NSM methods discussed in this chapter represent four differing approaches to dynamic prototype addition including simple, cost-minimizing, clustering, and a hybridization of cost minimizing and clustering. Cost minimizing approaches (including M-Cost, Avg-M-Cost, and SM-Cost) generally promote improved classification accuracy through a direct targeting of decision boundaries. Clustering approaches (including PC and Split-Medoid) generally promote improved generalization as they better reflect the data distribution. Hybridized methods (including Silhouette-F and Silhouette-L) offer a balance of both approaches.

The final component of the DLVQ framework includes a novel method for evaluation and characterization of NSM methods with an AGS. This characterization matches a performance component with a complete representation of both learned and theoretical components of generalization error, allowing holistic comparisons of NSM methods. This framework, characterizations of NSM methods, and evaluation approaches enable further experimentation and analysis of DLVQs in the following chapter.

IV. Results

THIS Chapter describes the data, experimental setup, and analysis steps to evaluate dynamic Generalized Relevance Learning Vector Quantization Improved (GRLVQI) performance. This chapter tests the hypothesis that dynamic prototype addition leads to improvements in accuracy and generalization over that of non-dynamic GRLVQI. Network Structure Modification (NSM) performance is examined, considering classification accuracy, prototype utilization, and margin-based error. The results validate the hypothesis, showing simultaneous improvements of accuracy and generalization through dynamic methods. Strong classification accuracy is demonstrated from dynamic methods that directly minimize cost. Hybridized methods show the best trade-off between accuracy and generalization.

4.1 Data Sets

Comparison testing of the eleven NSM methods utilizes four data sets that present a mix of conditions and classification challenges. The Lunar Crater Volcanic Field (LCVF), Mice protein expression levels (MICE), United States Postal Service (USPS), and Z-Wave wireless protocol (ZWAVE) data sets are chosen because they represent different data sources and variety in their number of features, classes, and samples per class as seen in Table 4.1.

4.1.1 United States Postal Service (USPS) Data Set.

The USPS data set [59, 60] contains a library of 9,298 handwritten zip code digits, $[0, 1, \dots, 9]$, that the postal service automatically scanned from envelopes. Each digit corresponds to one of 10 classes and the number of samples per class varies from 700 to 1500. Scans of the handwritten digits have been re-oriented into gray-scale 16×16 pixel

Table 4.1: MICE, USPS, ZWAVE, and LCVF data set attributes.

Data Set	Samples	Features	Classes	Samples per Class			Data Type Description
				min	med	max	
USPS	9298	256	10	708	829	1553	Pixel intensity of scanned hand-written digits
MICE	1047	71	8	105	134	150	Protein expression levels in mice cerebral cortex
ZWAVE	690	189	3	230	230	230	RF-DNA statistical features of communication signals
LCVF	1464	194	35	7	37	160	Spectral reflectance of geological regions

images; examples are shown in Fig. 4.1. The (real valued) image scans are vectorized into 256×1 element vectors and re-scaled on $[0, 1]$.

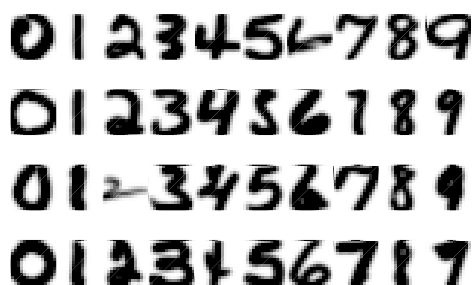


Figure 4.1: Four random samples per class are shown as examples of the United States Postal Service (USPS) data set. Handwritten digits are re-oriented and normalized as 16×16 pixel gray-scale images with real valued intensities on $[0, 1]$.

4.1.2 Mice Protein Expression (MICE) Data Set.

The MICE data set [61, 62], reports protein expression levels in the brains of eight classes of mice split by biology, conditioning, and treatments. The authors of this data

set [61] sought to link proteins or groups of proteins to learning deficits in mice as they relate to Down Syndrome. Due to empty fields in the data, 6 features (columns) each with more than 15 empty fields are omitted as well as 33 samples (rows) with remaining empty fields. The resulting data set contains 71 features (real valued) and 1047 samples. The median expression level of features are shown for each class in Fig. 4.2. For consistency, the protein expression features are linearly scaled on $[0, 1]$.

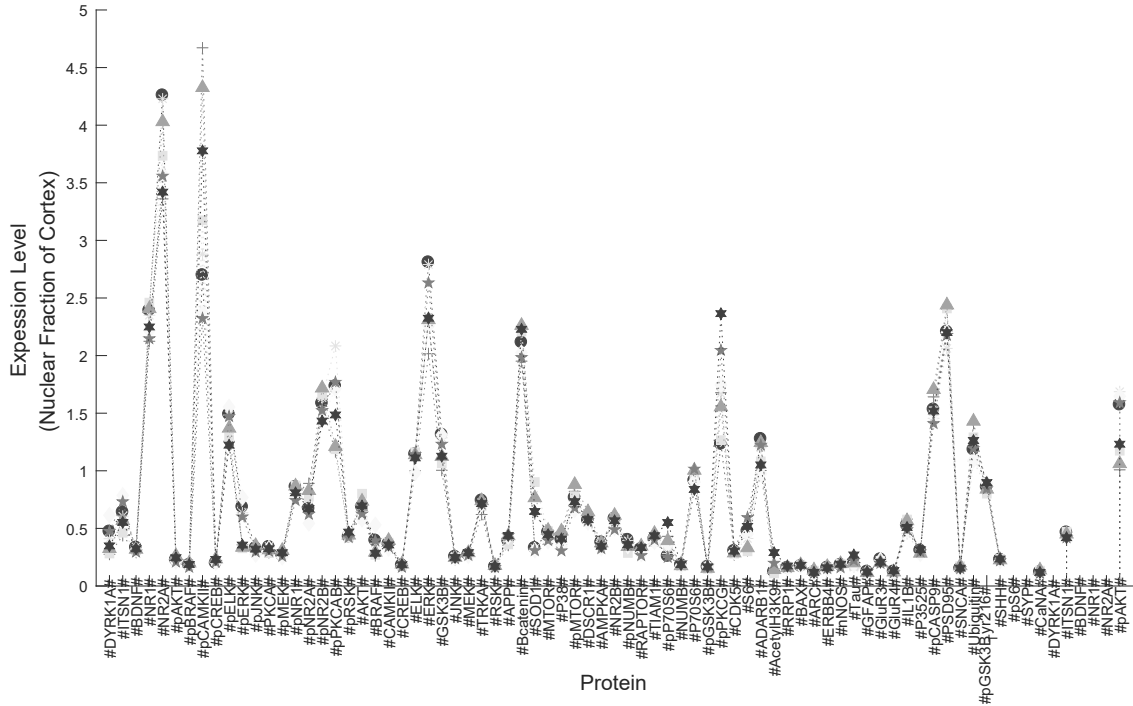


Figure 4.2: Expression levels in eight classes of mice are shown from the Mice Protein Expression (MICE) data set, with protein and modified protein names listed on the x -axis. Median values are shown with marker type indicating class.

4.1.3 Z-Wave wireless protocol (ZWAVE) Data Set.

Z-Wave is a low power, low data rate, wireless communication protocol used in home automation applications, such as wireless lighting control. Due to potential security

vulnerabilities [63], Bihl, et al. [64] investigated the identification of suspicious wireless activity and characterization of signals based on physical layer attributes. The connection between minor hardware deviations and resulting signal differences is captured through a Radio Frequency (RF) Distinct Native Attribute (RF-DNA) feature set [65].

The Z-Wave wireless protocol (ZWAVE) data set is a selection from [64] and includes 690 communication pre-ambls over 3 devices at a 24 decibel signal to noise ratio. Individual regions within each signal are characterized by a grouping of three statistical measures: variance, skewness, and kurtosis. The feature set is composed of a concatenation of these three-element segments (see Fig. 4.3).

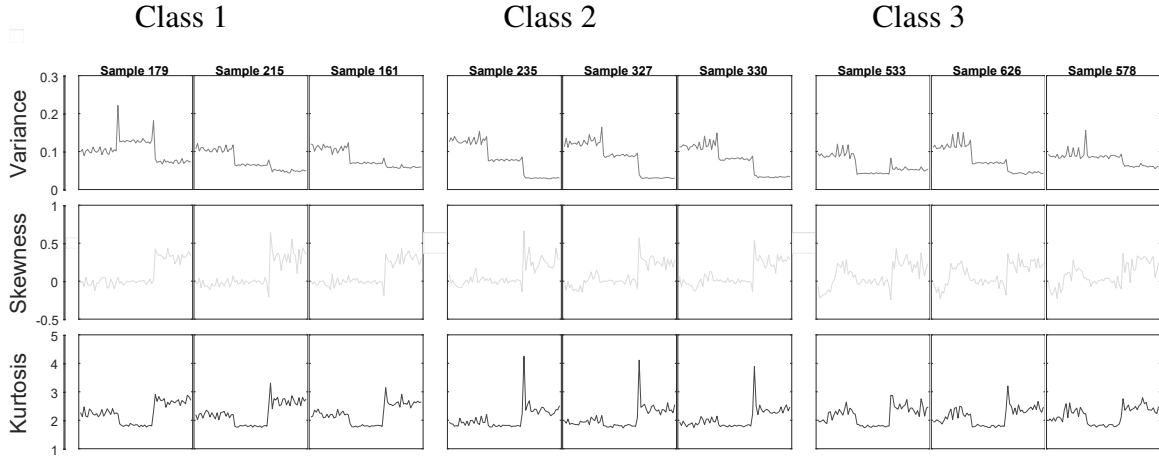


Figure 4.3: Z-Wave wireless protocol (ZWAVE) signals are characterized by three statistical measures: variance, skewness, and kurtosis. Each column shows a random ZWAVE sample from each class distribution. For ease of viewing, features are plotted according to statistical measure (top, middle, and bottom rows corresponding to variance, skewness, and kurtosis) where the x -axis of each plot represents a different signal region.

4.1.4 Lunar Crater Volcanic Field (LCVF) Data Set.

The Lunar Crater Volcanic Field (LCVF) hyperspectral data set [5] captures an aerial scene remotely imaged by the NASA/JPL AVIRIS sensor in 1994 [66]. Spectral responses from survey data are used to detect different material compositions and geological formations. The data set contains 1464 samples from 35 classes, each with 194 spectral dimensions.

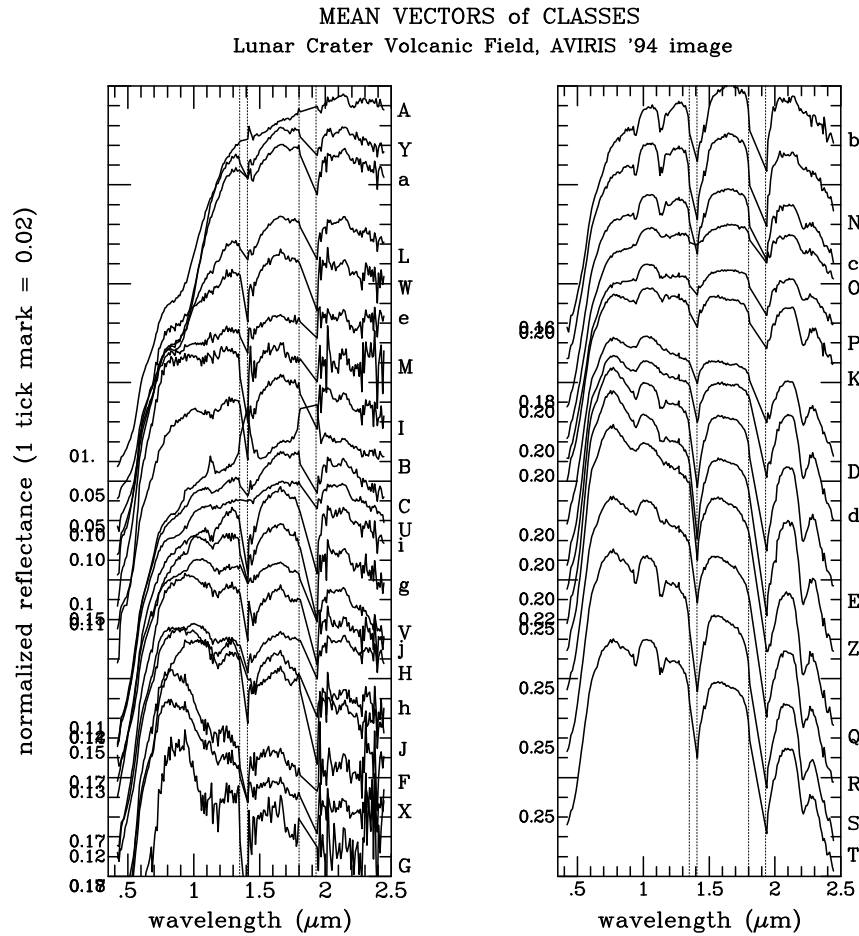


Figure 4.4: Mean spectral reflectances of the 35-class Lunar Crater Volcanic Field (LCVF) data set are stacked to compare characteristic profiles. Spectral gaps correspond to water absorption regions. For historical reasons, classes 1-35 correspond to labels: A-W, X-Z, a-e, and g-j [4]. Image is used with permission from [5].

As recommended in [67], each feature vector is normalized by its ℓ_2 -norm to compensate for the effect of shadowing due to sensor geometry. This data set offers high variation in the number of samples per class ranging from 7 to 160.

4.2 Experimental Setup

This section provides an overview of prototype initialization schemes and key experimental parameters. It describes the formation of training and validation sets, and the implementation of dynamic prototype addition, to include stopping conditions.

LVQ networks for the USPS, MICE, ZWAVE, and LCVF data sets are initialized with 1 prototype per class. Prototypes are (uniformly) randomly initialized on $[0.4, 0.6]$ in each feature dimension. Prototype learn rates, α^w , are initialized with large coarse values to allow larger weight updates at the onset of training. As training continues, the learn rate decreases until reaching a concluding stable fine learn rate allowing for convergence [54]. LCVF and ZWAVE initial and concluding learn rates are established based on documented parameter values [27, 54], USPS and MICE learn rates are determined through experimentation. Generally, it is found that insufficiently small concluding learn rates limit the ability to fine tune placement of prototypes and achieve high classification accuracy. Increased training time at lower learn rates is conducted, through the adoption of a stepped exponential decay between initial and concluding learn-rates. This stepped exponential decay is realized over fixed learn rate intervals, to form a *learn schedule* as shown in Table 4.2.

The relevance learn rate, α^λ , controlling the update size of λ (see Section. 2.1.2), is exponentially decayed in a similar stepped schedule as α^w . Conscience learning parameters, γ and β , are also shown in Table 4.2, with initial and final values adopted from [54]. The cost function utilizes a sigmoid with a slope of 1, and all distortion measures reflect the λ weighted squared Euclidean distance in Eqn. 2.5.

Table 4.2: Generalized Relevance Learning Vector Quantization Improved (GRLVQI) parameters include prototype learn rates, α^w , relevance learn rates, α^l and conscience learning parameters, γ and β for USPS, MICE, ZWAVE and LCVF data sets.

	Training Epochs	Learn Rates		Conscience Parameters			Training Epochs	Learn Rates		Conscience Parameters	
		α^w	α^l	γ	β			α^w	α^l	γ	β
USPS	$1 \leq \text{Ep} \leq 10$	5.00e-2	1.00e-3	2.50	0.35	MICE	$1 \leq \text{Ep} \leq 125$	5.00e-3	1.00e-3	2.50	0.35
	$11 \leq \text{Ep} \leq 20$	3.00e-2	5.01e-4	2.10	0.29		$126 \leq \text{Ep} \leq 250$	3.50e-3	5.24e-4	2.22	0.31
	$21 \leq \text{Ep} \leq 30$	1.70e-2	2.32e-4	1.77	0.24		$251 \leq \text{Ep} \leq 375$	2.20e-3	2.41e-4	1.85	0.26
	$31 \leq \text{Ep} \leq 40$	9.65e-3	1.08e-4	1.49	0.20		$376 \leq \text{Ep} \leq 500$	1.40e-3	1.11e-4	1.54	0.21
	$41 \leq \text{Ep} \leq 50$	5.47e-3	5.01e-5	1.25	0.17		$501 \leq \text{Ep} \leq 625$	9.31e-4	5.12e-5	1.29	0.18
	$51 \leq \text{Ep} \leq 60$	3.11e-3	2.32e-5	1.05	0.14		$626 \leq \text{Ep} \leq 750$	6.01e-4	2.36e-5	1.08	0.15
	$61 \leq \text{Ep} \leq 70$	1.76e-3	1.08e-5	0.90	0.12		$751 \leq \text{Ep} \leq 875$	3.88e-4	1.09e-5	0.90	0.12
	$71 \leq \text{Ep} \leq 80$	1.00e-3	5.00e-6	0.75	0.10		$876 \leq \text{Ep} \leq 1000$	2.50e-4	5.00e-6	0.75	0.10
ZWAVE	$1 \leq \text{Ep} \leq 125$	2.50e-2	5.00e-3	2.50	0.35	LCVF	$1 \leq \text{Ep} \leq 50$	2.50e-2	5.00e-3	2.50	0.35
	$126 \leq \text{Ep} \leq 250$	1.30e-2	2.59e-3	2.10	0.29		$51 \leq \text{Ep} \leq 100$	1.14e-2	2.50e-3	2.22	0.31
	$251 \leq \text{Ep} \leq 375$	6.71e-3	1.34e-3	1.77	0.24		$101 \leq \text{Ep} \leq 150$	3.50e-3	8.89e-4	1.85	0.26
	$376 \leq \text{Ep} \leq 500$	3.47e-3	6.95e-4	1.49	0.20		$151 \leq \text{Ep} \leq 200$	1.10e-3	3.15e-4	1.54	0.21
	$501 \leq \text{Ep} \leq 625$	1.80e-3	3.60e-4	1.25	0.17		$201 \leq \text{Ep} \leq 250$	3.38e-4	1.12e-4	1.29	0.18
	$626 \leq \text{Ep} \leq 750$	9.32e-4	1.86e-4	1.05	0.14		$251 \leq \text{Ep} \leq 300$	1.05e-4	3.97e-5	1.08	0.15
	$751 \leq \text{Ep} \leq 875$	4.83e-4	9.65e-5	0.90	0.12		$301 \leq \text{Ep} \leq 350$	3.23e-5	1.41e-5	0.90	0.12
	$876 \leq \text{Ep} \leq 1000$	2.50e-4	5.00e-5	0.75	0.10		$351 \leq \text{Ep} \leq 400$	1.00e-5	5.00e-6	0.75	0.10

As GRLVQI training is extended to Dynamic GRLVQI (DGRLVQI), classifiers have an opportunity, depending on the NSM method described in Section 2.2 and 3.2, to add prototypes after a fixed number of training epochs and is termed a *prototype addition stage* (PAS). In order to allow coarse adjustments to occur after the addition of new prototypes, the learn schedule resets to its original value and then follows the previous schedule. Classifiers are trained with 1 prototype per class as an initial PAS (I-PAS) and prototypes are added up to a maximum number of PAS's (M-PAS). For each data set, the M-PAS is a multiple of the number of classes to allow balanced comparison with non-dynamic approaches that utilize equal numbers of prototypes per class.

This research utilizes K -Fold cross-validation to form independent training and validation sets, with $K = 5$ for the USPS, MICE, and ZWAVE data sets [21]. A value of $K = 3$ is utilized for the LCVF data set due to the sparsity of several classes [37]. Repeated independent experimentation with 30 random starts of the K -fold cross-validation process

generates $30 \times K$ independent trials. Each of the $30 \times K$ partitions are maintained within each data set to allow pairwise comparisons between NSM methods. Table 4.3 shows configuration parameters for each data set’s independent trials, including the M-PAS’s allowed (the terminating condition) and the total number of iterations for each *experiment* or set of $30 \times K$ trials. Some parameter choices are influenced by practical computational time constraints (due to number of training samples and dimensionality of data sets).

Table 4.3: Dynamic parameters for USPS, MICE, ZWAVE, and LCVF independent trials.

Data Set	Individual Trials		Stages		Learn Schedule		Total Iterations
	Partitions	K-Folds	I-PAS	M-PAS	Intervals	Epoch per Interval	In Experiment
USPS	30	5	1	70	8	10	852,000
MICE	30	5	1	48	8	125	7,200,000
ZWAVE	30	5	1	30	8	125	4,650,000
LCVF	30	3	1	70	8	50	2,556,000

4.3 Results and Analysis

The DGRLVQI classifiers are trained with the 11 NSM methods (the 7 novel in Section 3.2.5 and 4 existing in Section 2.2) over the USPS, MICE, ZWAVE, and LCVF data sets. Additionally, a non-dynamic GRLVQI classifier is trained using the maximum number of prototype vectors that the dynamic versions are allowed (1 per class + M-PAS). The GRLVQI classifier provides a performance baseline for assessing the effectiveness of DGRLVQIs.

Analysis compares dynamic models against non-dynamic baseline models to determine improvements in classification accuracy. In Section 4.3.2, the accuracy of classifiers is assessed with respect to their prototype utilization (prototype utilization plane), describing their capability to benefit from increased numbers of prototypes. An examination of DGR-

LQI classifier margin and related margin error in Section 4.3.3, describes the robustness or flexibility of a classifier’s learned decision boundaries (generalization plane). For brevity the first three analysis steps are illustrated by the MICE and LCVF data sets (corresponding plots for USPS and ZWAVE are in Appendix A). The accuracy, prototype utilization, and margin results combine in the Accuracy-Generalization Surface (AGS) to characterize NSM methods over their accuracy/generalization trade-space. Section 4.3.4 analyzes resulting trends in AGS plots specific to the USPS, MICE, ZWAVE, and LCVF data sets.

4.3.1 Classification Accuracy.

The classification accuracy of trained (converged) dynamic and non-dynamic GRLVQI models is shown in Table 4.4 for $30 \times K$ pairwise trials (where $K = 3$ for MICE and $K = 5$ for LCVF). The central limit theorem permits paired student t-tests [68] to compare the NSM methods against the baseline. Mean improvements and the 99% confidence interval of their improvement are seen in Table 4.4, where greater, similar, or decreased improvement is shown as bold, italics, or normal font (and is determined through multiple hypothesis testing at the 99% significance level). NSM methods on the MICE data set demonstrate a 1.3% – 1.9% statistically significant improvement in accuracy. The only exception is Near-Mean which is statistically similar to the baseline. NSM methods on the LCVF data set achieve 0.9% – 1.1% statistically significant accuracy improvement between trials (with the exception of Silhouette-L).

4.3.2 Prototype Utilization.

Pairwise comparisons against non-dynamic GRLVQI reported in Table 4.4 identify the potential reduction in prototypes (at the accuracy performance reported on the same row). Mean reductions in the number of prototypes and their 99% confidence interval are seen on the right side of Table 4.4, where greater, similar, or decreased improvement is shown as bold, italics, or normal font (and is determined through multiple hypothesis testing at the 99% significance level). With the exception of Near-Mean, NSM methods

Table 4.4: Converged model accuracy and number of prototypes is shown for independent trials.

	NSM Method	Converged Classifier Performance							
		Accuracy				Prototypes			
		μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI
MICE	Near-Mean	95.77	1.88	0.34	(−0.07 – 4.57)	52.41	4.50	3.59	(2.72 – 3.59)
	Mean	97.09	1.12	1.66	(1.37 – 4.57)	52.75	3.38	3.25	(2.60 – 3.25)
	Closest	97.28	1.14	1.85	(1.53 – 4.57)	53.70	2.56	2.30	(1.81 – 2.30)
	M-Cost	97.17	1.07	1.74	(1.42 – 4.57)	53.25	3.41	2.75	(2.09 – 2.75)
	Avg-M-Cost	97.24	1.13	1.80	(1.50 – 4.57)	53.35	3.30	2.65	(2.02 – 2.65)
	SM-Cost	97.31	1.08	1.87	(1.57 – 4.57)	52.85	3.96	3.15	(2.39 – 3.15)
	S-Cost	97.19	1.20	1.76	(1.45 – 4.57)	53.82	2.69	2.18	(1.66 – 2.18)
	Split-Medoid	97.15	1.24	1.72	(1.43 – 4.57)	53.28	3.60	2.72	(2.03 – 2.72)
	PC	97.23	1.16	1.80	(1.50 – 4.57)	53.61	2.85	2.39	(1.84 – 2.39)
	Silhouette-F	96.72	1.30	1.29	(1.00 – 4.57)	53.93	2.85	2.07	(1.53 – 2.07)
	Silhouette-L	96.73	1.31	1.30	(0.98 – 4.57)	53.95	2.67	2.05	(1.54 – 2.05)
	Baseline(7x8p)	95.43	1.77	-	-	56.00	0.00	-	-
LCVF	Near-Mean	97.65	0.94	0.88	(0.66 – 3.23)	82.72	14.57	22.28	(18.64 – 22.28)
	Mean	97.78	0.84	1.01	(0.80 – 3.23)	83.14	13.35	21.86	(18.52 – 21.86)
	Closest	97.81	0.82	1.04	(0.84 – 3.23)	84.10	12.36	20.90	(17.81 – 20.90)
	M-Cost	97.63	0.83	0.87	(0.64 – 3.23)	86.28	10.17	18.72	(16.18 – 18.72)
	Avg-M-Cost	97.73	0.84	0.97	(0.75 – 3.23)	84.50	13.27	20.50	(17.19 – 20.50)
	SM-Cost	97.85	0.85	1.08	(0.86 – 3.23)	87.50	11.50	17.50	(14.63 – 17.50)
	S-Cost	97.95	0.82	1.18	(0.96 – 3.23)	92.78	11.32	12.22	(9.39 – 12.22)
	Split-Medoid	97.79	0.82	1.02	(0.80 – 3.23)	82.10	13.02	22.90	(19.65 – 22.90)
	PC	97.83	0.83	1.07	(0.86 – 3.23)	84.66	13.72	20.34	(16.92 – 20.34)
	Silhouette-F	97.30	0.87	0.53	(0.28 – 3.23)	92.34	11.45	12.66	(9.80 – 12.66)
	Silhouette-L	95.68	1.12	−1.09	(−1.37 – 3.23)	50.46	5.21	54.54	(53.24 – 54.54)
	Baseline(3x35p)	96.77	0.97	-	-	105.00	0.00	-	-

on the MICE data set demonstrate simultaneous statistically significant improvements over non-dynamic forms, utilizing 2.0% – 3.3% fewer prototypes, while maintaining higher classification accuracy. Similarly, NSM methods on the LCVF data set (with the exception of Silhouette-F) utilize 12.2% – 22.9% fewer prototypes, while maintaining greater classification accuracy when compared to the non-dynamic baseline. Over the LCVF data set, some NSM methods have non-overlapping confidence intervals that relate their reduction in prototypes. Split-Medoid shows a mean reduction in prototypes between 19.7 and 22.9 (99% confidence interval), offering potential performance benefit over the

mean reductions from SM-Cost (14.6 – 17.5), and S-Cost (9.4 – 12.7) that appear similar on the basis of classification accuracy alone.

NSM methods are not guaranteed to add a prototype during each Prototype Addition Stage (PAS). They may periodically forgo the opportunity due to their prototype identification or region identification strategy. Increased numbers of prototypes in LVQs lead to growth in the theoretical component of generalization error (see Section 2.3). Consequently, considerations of accuracy versus the number of utilized prototypes provide insight to accuracy-generalization trade-offs.

Classification accuracy and prototype utilization operating points from the last interval of each PAS are pooled together to capture performance over a consistently non-transient region (see Section A.1.1). The resulting pool reflects 919,750 and 319,500 performance points for MICE and LCVF respectively. For each pairing of NSM method and data set, performance points are sorted according to the number of utilized prototypes (as opposed to epoch or PAS). In Fig. 4.5(top), MICE shows smooth increasing trends in classification accuracy as a function of PAS (Fig. A.1). This is likely because NSM methods over the MICE data set utilized most of the available prototypes to achieve their converged performance, rarely skipping opportunities to add them (as seen in Table 4.4). NSM methods with LCVF are more prone to skip a PAS. This trend is observed in Fig. 4.5(bottom), where little performance improvement is seen for most NSM methods beyond 75 prototypes. However, Avg-M-Cost and SM-Cost do the best job of utilizing an increasing number of prototypes. This is especially reflected for Avg-M-Cost between 90 – 105 prototypes.

Performance versus the number of utilized prototypes is also shown as a modified box plot in Fig. 4.6. Instead of showing one averaged value for each bin of pooled data, the accuracy spread is shown as a box with top, middle, and bottom lines corresponding to the 75th, 50th, and 25th percentiles. For MICE in Fig. 4.6(top), slightly higher performance

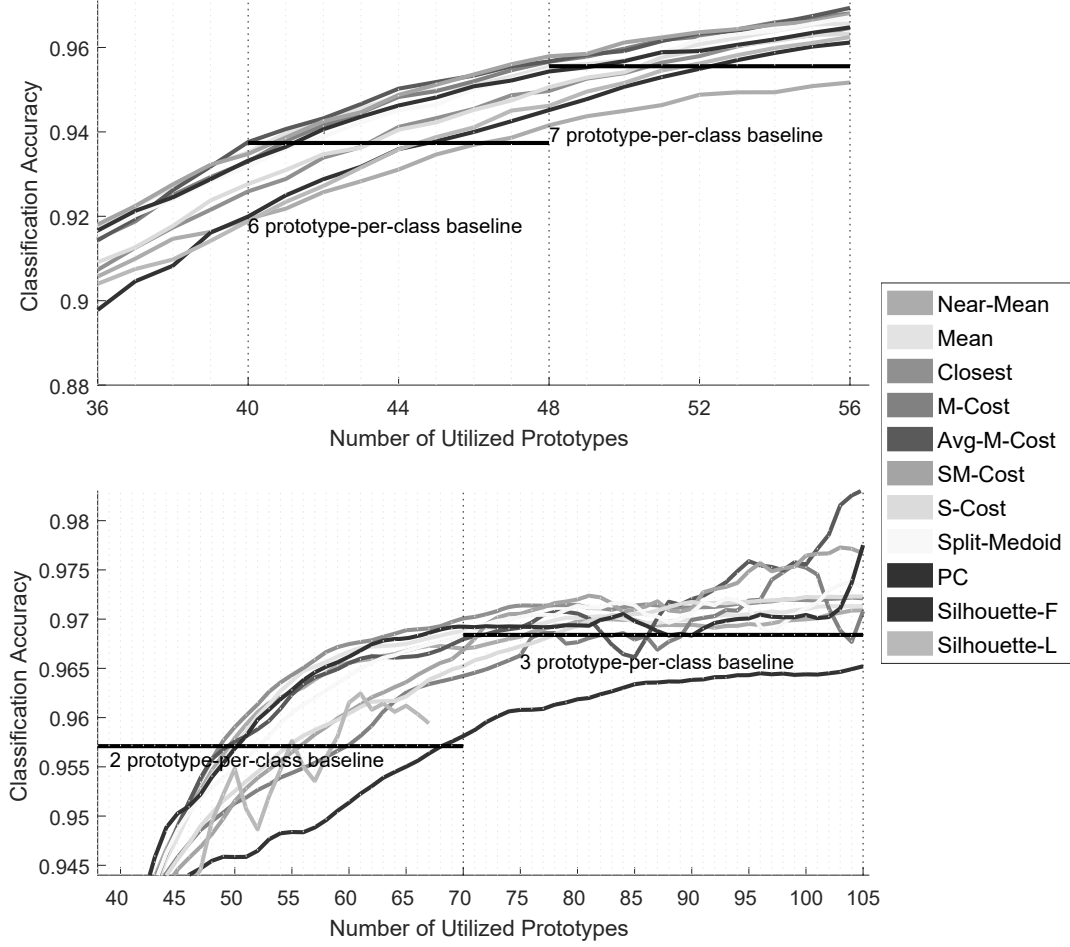


Figure 4.5: NSM methods are compared by prototype utilization. Solid horizontal lines show baseline performance with equal or more prototypes. **(top)** MICE with a maximum of 8×7 prototypes and **(bottom)** LCVF with a maximum of 3×35 prototypes.

is seen for the Avg-M-Cost, M-Cost, and SM-Cost methods past a utilization of 40 (additional) prototypes, however most of these differences are within $\sigma = 0.2$ of their next best alternative and have limited significance. For LCVF, Avg-M-Cost's accuracy trends higher beyond 55 (additional) prototypes and concludes between 98.0% and 98.4% compared to 95.5% to 98.0% for others. In this range of 66 – 70 additional prototypes,

Avg-M-Cost’s performance offers a tight envelope whose median is a standard deviation above the alternative methods.

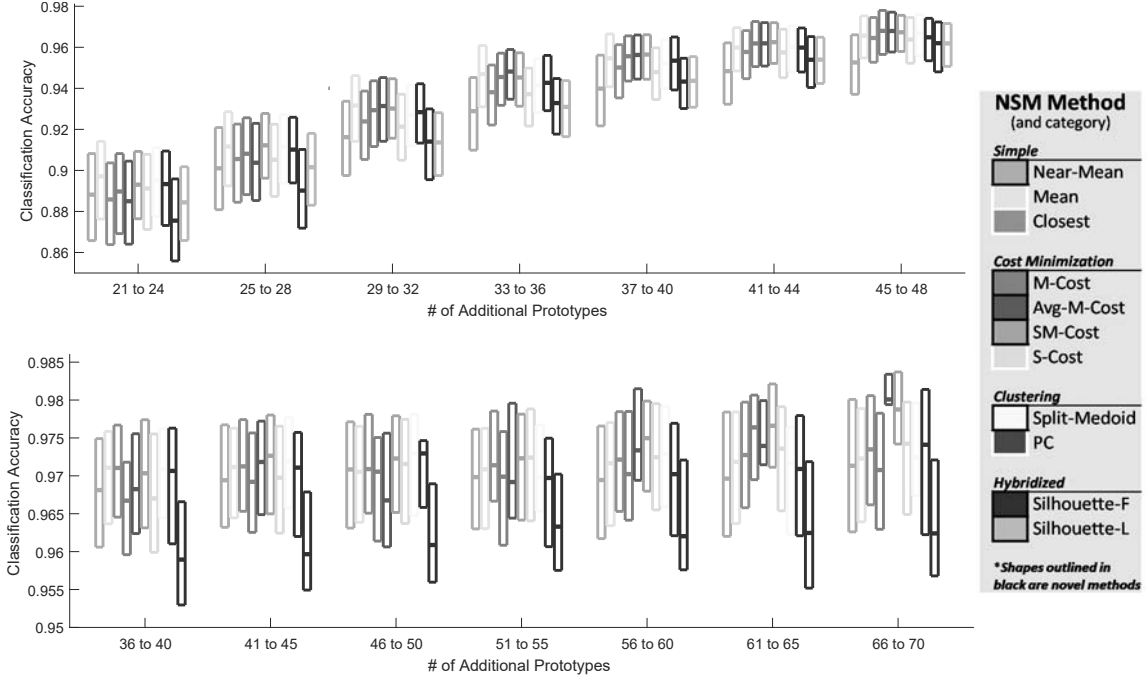


Figure 4.6: Accuracy of the 11 NSM methods over prototype intervals of size 4 to 5 is shown for the 75th, 50th, and 25th percentiles (top, middle, bottom of vertical boxes) for **(top)** MICE and **(bottom)** LCVF.

4.3.3 Margin Analysis.

This section describes classifier robustness through two representations of margin error (with respect to a margin threshold, θ_{cp}). E_{CGNT} is an *all or nothing* assignment, typical of classification. E_{HSV} is a *fractional* assignment, typical of the cost function argument. A combination of the two illustrates the empirical margin error plane.

E_{CGNT} is a function of θ_{cp} (left column of Fig. 4.7 and 4.8) that counts samples whose hypothesis margins are smaller than θ_{cp} , and would be misclassified if their nearest decision boundary were shifted the distance, θ_{cp} , closer to them. The five NSM-based

results in Fig. 4.7(left) show similarly shaped error curves that increase with larger θ_{cp} values. For a constant θ_{cp} value, classifiers with more prototypes yield lower error, indicating benefit from the incremental addition of prototypes. For the MICE data set over $0 \leq \theta_{cp} \leq 5e^{-5}$, Avg-M-Cost and Silhouette-F show the slowest E_{CGNT} growth, suggesting larger hypothesis margins and decision boundaries that are less susceptible to error. In LCVF, for $\theta_{cp} < 5e^{-5}$, Avg-M-Cost consistently achieves the lowest E_{CGNT} of the NSM methods shown; its error reduction with 105 prototypes corresponds to a near 3% increase in classification accuracy at $\theta_{cp} = 3e^{-5}$.

E_{HSV} as a function of θ_{cp} (middle column of Fig. 4.7 and Fig. 4.8) shows *fractional* error based on the distance from the shifted decision boundaries. For the MICE data set, classifiers trained with 8 prototypes yield similar values of $E_{HSV} \sim 0.617$ for $\theta_{cp} = 2e^{-4}$. However, when classifiers are trained with 20-prototypes and evaluated at $\theta_{cp} = 2e^{-4}$, S-Cost and PC yield higher error than the other NSM methods, suggesting the least benefit in error reduction from added prototypes. For the LCVF data set, Avg-M-Cost and Silhouette-F offer lower relative error for networks with 105 prototypes at $\theta_{cp} > 1e^{-4}$. For $\theta_{cp} = 2e^{-4}$, Silhouette-F with 45 prototypes shows a low error of $E_{HSV} = 0.17$ compared to $E_{HSV} = 0.19$ from the other NSM methods with 45 prototypes. Silhouette-F shows a trend of strong generalization over a wide number of utilized prototypes.

The MICE data set results in Fig. 4.7(left, center) show classifiers with more prototypes achieve (non-intersecting) universally lower error curves with respect to θ_{cp} . LCVF is a contrary example where more prototypes do not universally lead to lower error. This is due to complex interaction between a large number of classes with noisy samples, thus making the learned decision boundary more intricate. For classifiers with different numbers of prototypes, error curve intersections indicate transitions between the most beneficial prototype network size. In Fig. 4.8(left), Avg-M-Cost's 105-prototype classifier accommodates larger margin threshold values before an intersection, $\theta_{cp} \approx 6e^{-5}$ compared

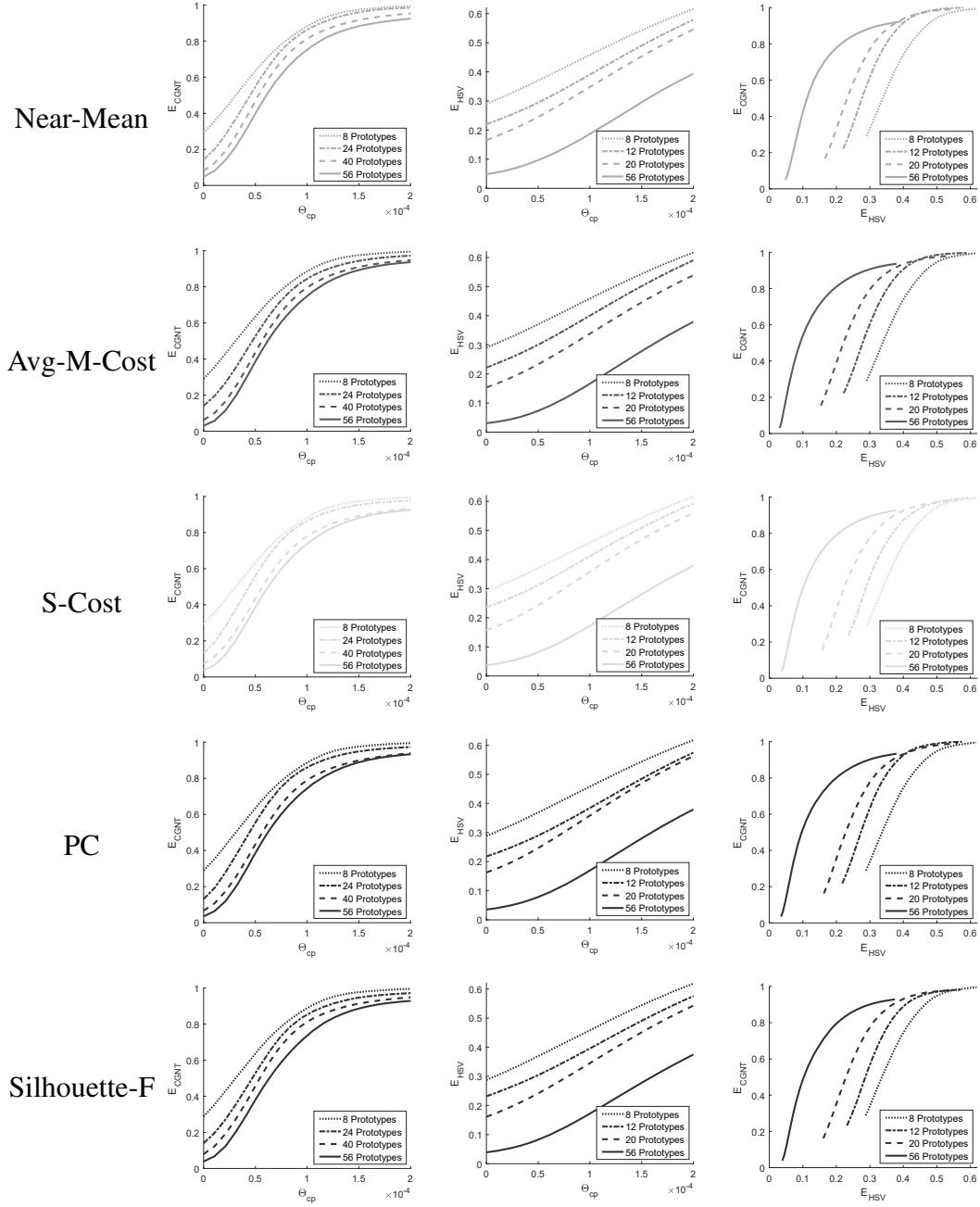


Figure 4.7: Margin error for MICE using Near-Mean, Avg-M-Cost, S-Cost, PC, and Silhouette-F NSM methods: **(left)** CGNT margin error versus θ_{cp} , **(middle)** HSV margin error versus θ_{cp} , and **(right)** HSV margin error versus CGNT margin error.

to $\theta_{cp} \approx 4e^{-5}$ for other NSM methods; this indicates a durable placement strategy for a large number of prototypes.

The right column of Fig. 4.7 and Fig. 4.8 combines E_{CGNT} and E_{HSV} . E_{CGNT} accumulates error as new misclassifications result from shifting decision boundaries, E_{HSV} considers their relative distance past decision boundaries; their combination simultaneously provides insights to impact on classification accuracy and how well the system has maximized the margin. For the LCVF data set, Avg-M-Cost with 105 prototypes shows a pairing of the smallest E_{CGNT} and E_{HSV} for $\theta_{cp} < 3e^{-5}$, achieving high accuracy when future data resembles training data, represented by small or no shifts in decision boundaries (small margin thresholds). Silhouette-F with 40 prototypes shows the lowest E_{CGNT} and E_{HSV} curve for $\theta_{cp} > 8e^{-5}$, achieving strong generalization as it accommodates larger shifts in decision boundaries (large margin thresholds). The relationship between accuracy and generalization is further explored in the following Section.

4.3.4 AGS Comparisons.

Three dimensional (prototype utilization, generalization, and empirical error), AGS plots for each NSM method are overlaid as a composite surface, as shown in Fig. 4.9. Composite AGSs are collapsed across the z - axis to indicate NSM methods with the highest average classification accuracy as a function of utilized prototypes and margin-based error, as shown in Fig. 4.11 for the same data. The statistical significance is not known for all pairings of NSM methods, utilized prototypes, and margin error values. Instead, the flattened composite surfaces convey relative trends in NSM mean accuracy and generalizability over a data set. This analysis is shown for the USPS, MICE, ZWAVE, and LCVF data sets in Sections 4.3.4.1-4.3.4.4.

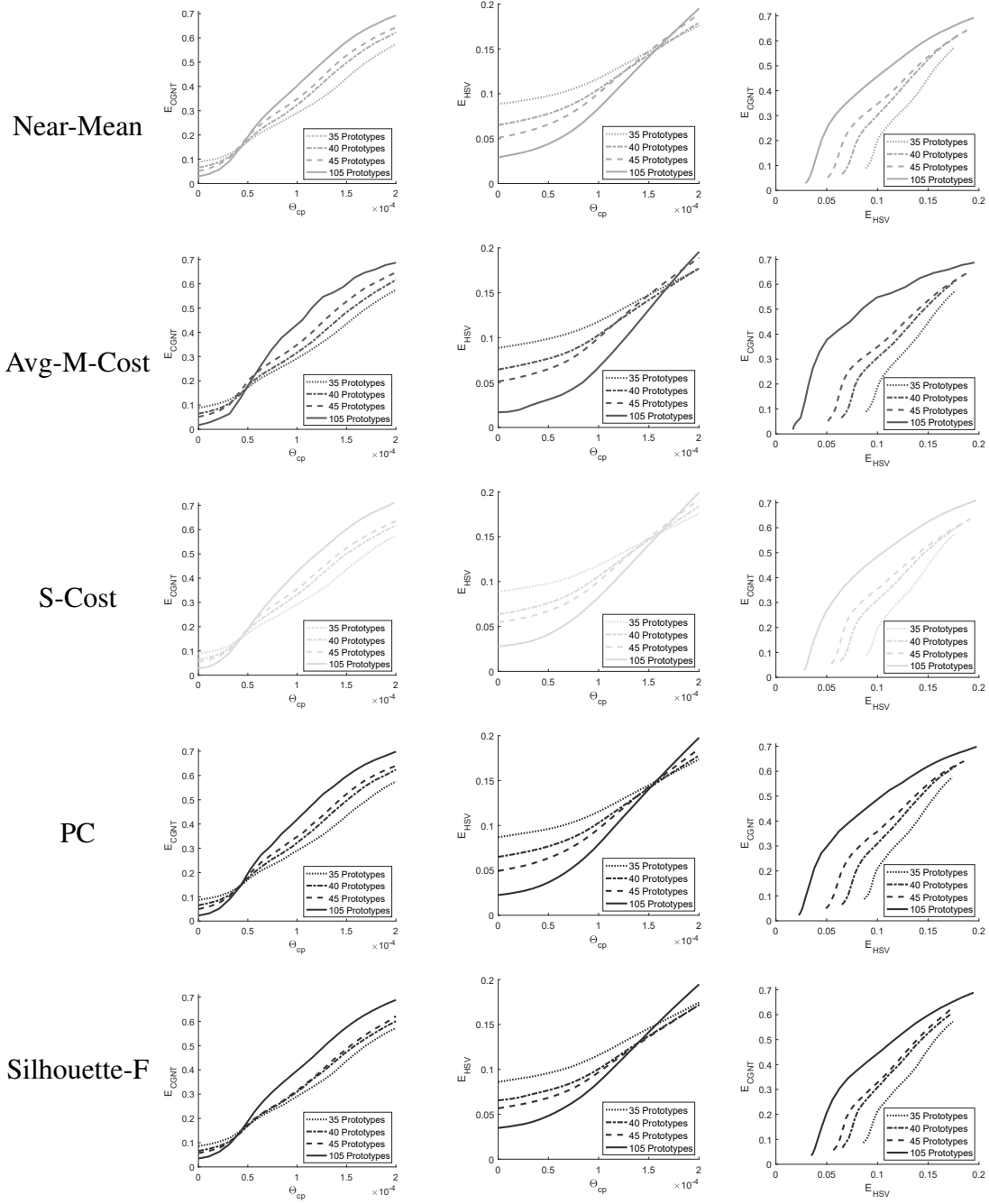


Figure 4.8: Margin error for LCVF using Near-Mean, Avg-M-Cost, S-Cost, PC, and Silhouette-F NSM methods: **(left)** CGNT margin error versus θ_{cp} , **(middle)** HSV margin error versus θ_{cp} , and **(right)** HSV margin error versus CGNT margin error.

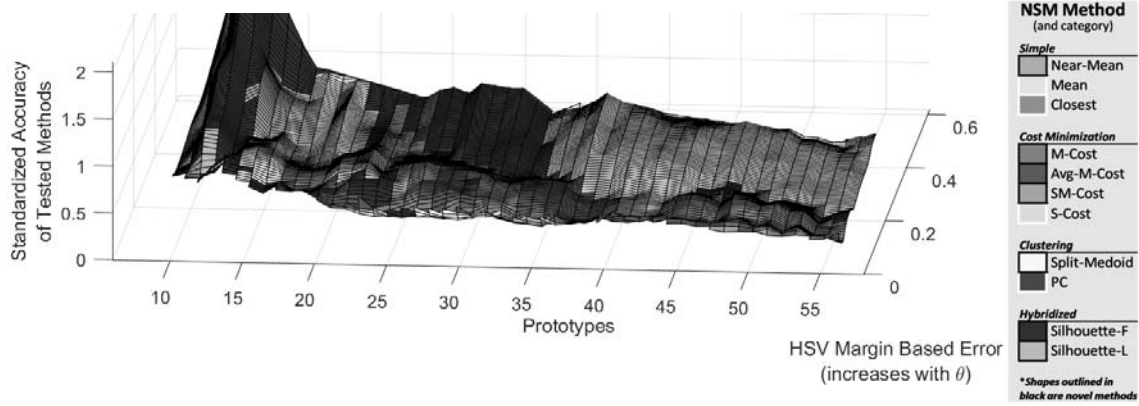


Figure 4.9: The composite Accuracy-Generalization Surface (AGS) for the MICE data set.

4.3.4.1 USPS.

The USPS data set (Fig. 4.10) contains a large number of samples and exhibits high variation within classes. These data characteristics favor generalized approaches that create less sensitive decision boundaries. Split-Medoid and PC have the best mean accuracy in low margin error regions ($E_{HSV} < 0.14$). Both NSM methods are clustering approaches that insert new prototypes near clusters centers to better represent the data space. Their localized focus and memory of past prototype allocations create generalized decision boundaries and result in superior performance.

In Fig. 4.10(right), commonalities between best performing NSMs are related to characteristics of their region influence and prototype insertion strategy. Split-Medoid and PC's best performance in low margin error regions is circled in green. As the margin error increases to approximately $E_{HSV} = 0.16$, Mean, S-Cost, and SM-Cost consistently offer the highest mean classification accuracy (circled in red in Fig. 4.10(right)). These three NSM methods are more typical of clustering than boundary approaches and represent both extremes of local and global region influence. This reinforces the observation that USPS

favors gradual improvements near cluster centers as opposed to decision boundaries, and that this consideration outweighs the regional influence.

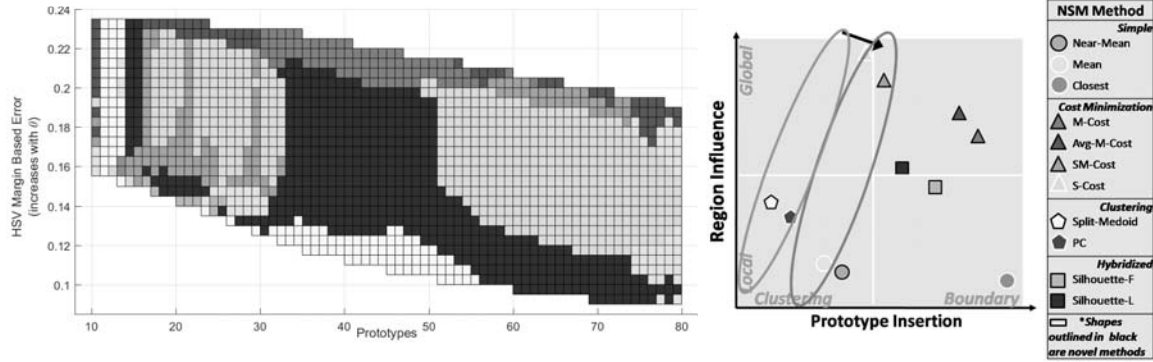


Figure 4.10: **(left)** The 2-dimensional composite AGS for the USPS data set. **(right)** Trends in NSM methods that perform best for low margin error (green circle) and higher margin error (red circle), where a black arrow shows the transition of best performers between margin error regions.

4.3.4.2 MICE.

NSM results for the MICE data set are shown in Fig. 4.11. The absence of strong performance from clustering approaches may suggest weak natural groupings of samples. Instead of allocating prototypes to the centers of sample distributions as performed by clustering approaches, M-Cost and Avg-M-Cost targeted misclassified regions near decision boundaries. This boundary focus, guided by a targeted cost minimization approach, prove successful for the MICE data set, where M-Cost and Avg-M-Cost offer the best average performance for nearly 80% of the low margin error region ($E_{HSV} < 0.225$).

For moderate to high margin error regions ($E_{HSV} > 0.225$), the Mean NSM method provides strong relative classification performance for configurations with fewer than 21 prototypes. While Mean simply targets classes with the highest ratio of misclassified samples and assigns a prototype in their center, it represents a sufficient approach when

operating in moderate to higher margin error regions and (where lower classification accuracies are anticipated).

For high margin error regions ($E_{HSV} > 0.3$), the Silhouette NSM methods offer the highest mean classification accuracy. These balanced approaches incorporate cost minimization to insert prototypes into cluster regions where they will provide the most accuracy benefit (and least disruption). As the number of utilized prototypes exceeds 35, S-Cost offers the highest average classification accuracy for $E_{HSV} > 0.3$. S-Cost's prototype placement via cost minimization using randomly sampled regions is a reasonable approach early in the training process, especially in situations where the sample distribution is roughly equal between classes.

Figure 4.11(right) shows global boundary (M-Cost, Avg-M-Cost) or local clustering approaches (Mean) offer the best relative options for low margin error regions in MICE. Successful NSMs in higher margin error regions are more characteristic of balanced prototype insertion strategies such as Silhouette-F and S-Cost.

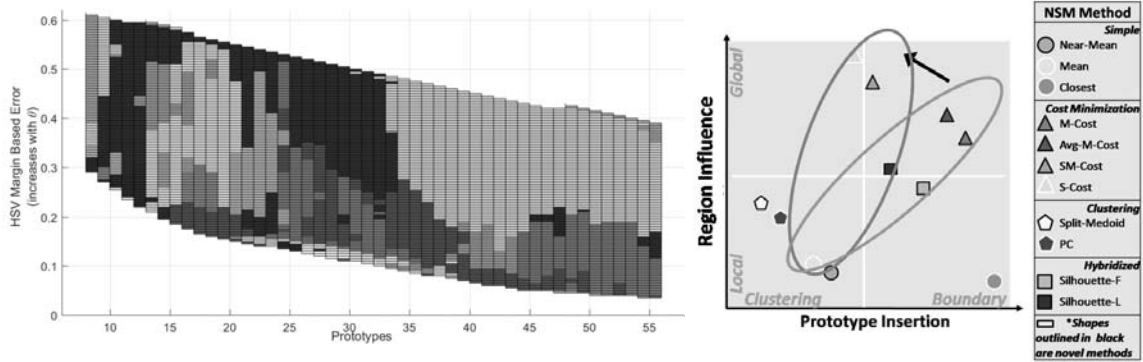


Figure 4.11: **(left)** The 2-dimensional composite AGS for the MICE data set. **(right)** Trends in NSM methods that perform best for low margin error (green circle) and higher margin error (red circle), where a black arrow shows the transition of best performers between margin error regions.

4.3.4.3 ZWAVE.

For the ZWAVE data set, Fig. 4.12(left), Split-Medoid, M-Cost, and Avg-M-Cost show the highest mean classification for low margin error regions ($E_{HSV} < 0.02$). Over this region, Split-Medoid, M-Cost, and Avg-M-Cost appear almost interchangeable despite having very different region influence and prototype insertion strategies. Split-Medoid offers an example of a clustering-based prototype insertion strategy. Its internal K-Medoids function effectively breaks each class into smaller clusters with a prototype initialized in the center of each. The ZWAVE data set consists of only three classes; this small number of classes may represent a comparative advantage for Split-Medoid and other NSMs that represent regions as clusters (instead of classes). The ZWAVE data set is also noisy, which may disadvantage NSMs that interpolate new prototype positions, such as PC (the other clustering NSM). PC's calculation of the covariance matrix may also be negatively impacted by the noise.

M-Cost and Avg-M-Cost present a different approach from Split-Medoid that achieves similar performance results in this same low margin error region ($E_{HSV} < 0.02$), as seen in Fig. 4.12(right). Since M-Cost and Avg-M-Cost represent regions as misclassified samples over two-class boundaries, they have a relatively well targeted, albeit few, set of options that provide strong results in the low margin error region.

For small to very large margin error regions ($0.02 < E_{HSV} < 0.17$), Silhouette-F shows the highest average classification accuracy. Notably, Silhouette-F has the same clustering focus as Split-Medoid and the same mechanism as M-Cost to determine a targeted location for prototype insertion (as seen in Fig. 4.12(right)), supporting the strong observed performance. Additionally, as Silhouette-F begins with one cluster per class, its two-cluster exhaustive comparison, (Section 3.2.4.1), results in a much more global approach (with only 3 classes in ZWAVE) fulfilling the role that S-Cost normally

would. Ultimately, Silhouette-F demonstrates strong generalizability through superior classification performance over a wide margin error range.

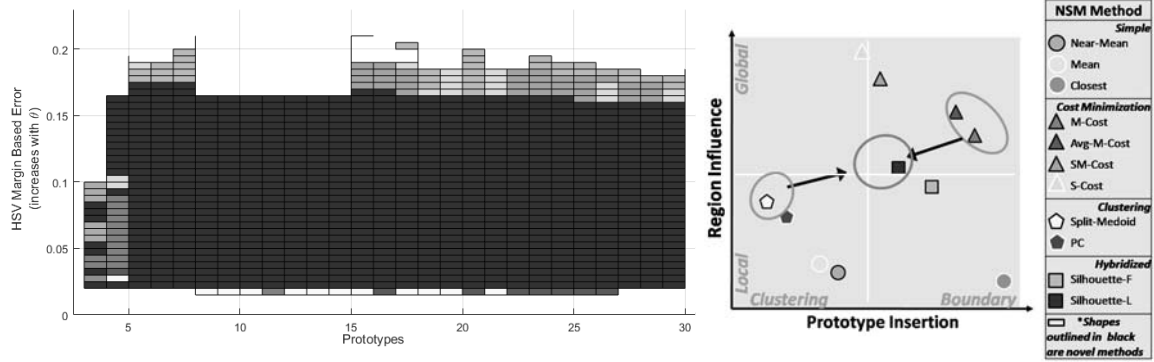


Figure 4.12: **(left)** The 2-dimensional composite AGS for the ZWAVE data set. **(right)** Trends in NSM methods that perform best for low margin error (green circle) and higher margin error (red circle), where a black arrow shows the transition of best performers between margin error regions.

4.3.4.4 LCVF.

In LCVF (Fig. 4.13), Closest, M-Cost, and Avg-M-Cost consistently show the best relative classification in low margin error regions ($E_{HSV} < 0.04$). All three methods (circled in green in Fig. 4.13(right)) initialize new prototypes at actual sample locations instead of interpolated locations. The M-Cost and Avg-M-Cost methods both use cost minimization to guide their insertion of a new prototype, typically near the center of misclassified samples on one side of a two class boundary. However, the Closest method adopts a more aggressive approach, inserting a new prototype at a misclassified sample location closest to a foreign prototype. In LCVF, this focus on providing better boundary definition seems to provide higher mean classification accuracy. It is likely that the complex interactions of LCVF’s 35-classes is best represented by NSM methods that concentrate on defining decision boundaries. Such boundary approaches provide an additional advantage

when dealing with uneven sample distribution, as they allow more targeted fine tuning that is often overshadowed for a more general approach that favors large classes. Unfortunately, an extreme boundary focus runs the risk of producing smaller margins that are highly sensitive and less robust to future variation, as seen by a drop in performance even with small increases of θ_{cp} . M-Cost and Avg-M-Cost achieve the same end-goal as Closest, of providing added definition to the decision boundary. However, their selection of the lowest cost outcomes allows them to maintain a better focus on representing the samples themselves, yielding better generalization.

For slightly larger margin errors ($0.04 < E_{HSV} < 0.06$), the highest mean classification accuracy is seen from SM-Cost and S-Cost. While their prototype insertion strategy is less focused on boundary regions, their adoption of a global approach to minimizing the cost yields high classification accuracies. Their use of randomly sampled evaluation pools also provides for a more general approach. S-Cost may experience a reduced classification ability in LCVF as its method for adding prototypes may overlook weakly represented classes, and result in lower accuracies (when using the average of the individual classification accuracies as the performance measure). The penalty for this global approach is often minimized in areas of higher margin error, where we would typically expect low classification accuracy anyway.

The Silhouette methods dominate classification accuracy for LCVF in regions of moderate to high margin error ($E_{HSV} > 0.055$) as seen in Fig. 4.13(left). In these margin error regions, Silhouette’s more balanced options between global and local boundary approaches provide a more generalized solution while still offering modest classification accuracy. Ultimately as margin error increases, targeted boundary solutions give way to a universally balanced approach with Silhouette-F.

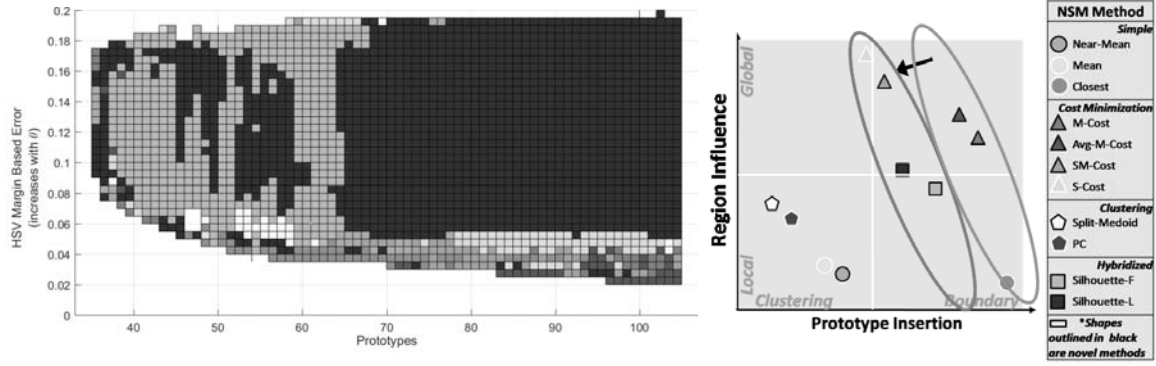


Figure 4.13: **(left)** The 2-dimensional composite AGS for the LCVF data set. **(right)** Trends in NSM methods that perform best for low margin error (green circle) and higher margin error (red circle), where a black arrow shows the transition of best performers between margin error regions.

4.3.4.5 Data Set and NSM Method Interactions.

Comparisons between the USPS, MICE, ZWAVE, and LCVF data sets and their composite AGSs show that the best performing categories of NSM methods often relate to the data set properties. Clustering approaches may offer best performance for noisy data sets with few classes, such as LCVF. They create generalized decision boundaries by allocating prototypes to define primary modes of the data distribution where minimal variation is present. Data sets that offer weak natural groupings of samples, as MICE, may be best served by global boundary approaches that utilize cost minimization to add targeted definition to decision boundaries. Data sets with relatively few classes, such as ZWAVE likely benefit from clustering or hybrid approaches (PC, Split Medoid, Silhouette-F). Their smaller regions provide increased options to determine a recipient for a new prototype, as opposed to NSMs like Mean, Near-Mean, and Closest whose larger regions are rigidly defined by class label. A clustering approach may be similarly fitting for data sets with highly imbalanced numbers of samples in each class. While the LCVF data set features large sample imbalances, clustering approaches do not dominate performance. This is

likely due to LCVF's highly complex hyperspectral data with many classes. In such cases, if high classification accuracy is paramount, a local or global boundary approach such as M-Cost, M-Avg-Cost, SM-Cost, or Closest is typically necessary to achieve high definition around decision boundaries.

For most data sets where current training data may not adequately reflect future samples, a hybridized cost/clustering approach such as Silhouette-F may provide the best balance of accuracy with strong capabilities for generalization, as typified in high margin error regions of MICE, ZWAVE, and LCVF. Recommendations in Appendix C and Chapter 5 may lead to future success with the USPS data set as well.

4.4 Results Summary

This chapter examined NSM method performance in GRLVQI with respect to classification accuracy, prototype utilization, and margin based error performance.

Cost minimization approaches that directly target the decision boundaries, typically showed the highest mean accuracy relative to other NSM methods. These approaches achieved classification accuracy that surpassed the best published performance for the ZWAVE and LCVF data sets, offering increases from 96.5%[27] to 98.9% and 97.2%[4] to 98.0%, respectively. While cost minimization approaches achieve best classification, they do not generalize as well as other approaches.

In 84% of nearly 6,000 runs that include all 11 NSM methods over the USPS, MICE, ZWAVE, and LCVF data sets, DGRLVQI demonstrated statistically significant improvements in classification accuracy and prototype reduction when compared to GRLVQI. They achieved simultaneous 1% - 2% mean accuracy gains, with a mean of 4%-56% fewer prototype resources, ultimately driving down the theoretical component of the upper bounds on the generalization error. These results affirmatively answer the research question and show that DGRLVQIs improve accuracy and generalization.

The analysis of classifier margin and margin error offers insight to a learned model's generalizability and expected performance trends in the presence of noise. For the USPS data set, clustering methods such as Split-Medoid and PC offered lower error and improved generalizability, however in most cases, they are not as accurate as other approaches.

AGS plots were overlaid from competing NSMs in a composite view. This holistic representation of best performing NSM methods over the entire operating region highlighted the regions of relative benefit from different NSM methods. In relation to other NSMs, the hybridized cost/clustering approach consistently indicates the best accuracy performance when all margin error regions are considered. In all but the USPS data set it achieves the best balance of accuracy and generalization.

V. Summary and Discussion

In ML, the balance between achieving high accuracy without over-fitting represents an accuracy/generalization trade-off. Learning Vector Quantization (LVQ) trains a set of prototypical vectors to closely match a known domain and correctly label new samples. The location and number of prototypes influences their resulting piece-wise linear decision boundaries and ultimately classification accuracy.

The challenge of the accuracy/generalization trade-off in LVQ prompts the research question: *“Can we achieve improvements in classification accuracy and generalization error in Generalized LVQ classifiers through the dynamic addition of prototypes?”* This work made three contributions to answer this question: (1) the Dynamic LVQ (DLVQ) framework, (2) 7 Network Structure Modification (NSM) methods over 4 approach categories, and (3) an Accuracy-Generalization Surface.

5.1 Contributions

A novel framework was presented that characterizes the operation of prototype addition across past and present DLVQ efforts. This framework incorporates a unique NSM stage to augment the formal LVQ structure. The design of the NSM stage allows an effective segmentation of NSM methods according to two phases that (1) identify a general region to receive a new prototype resource, and (2) determine its insertion position and class label. This contribution provides a pivotal, unifying construct that lays the foundation for this research and adds value to the DLVQ community at large. Furthermore, it enables a uniform comparison of DLVQ algorithms based solely upon aspects of the specific NSM method employed.

Second, this research designed seven novel methods for dynamic prototype addition in LVQ networks. The NSM methods follow four general approaches: *simple*,

cost-minimizing, *clustering*, and *hybridized* (a combination of clustering and cost-minimization). NSM methods were characterized by *regional influences* describing their use and representation of the data, and *prototype insertion strategy* defining the target placement of prototype additions. Hybridized methods were found to offer a balanced approach across the NSM characterizations. In most cases, the novel methods proposed in this dissertation showed the highest mean accuracy within approach category.

Third, a unique method was introduced to characterize the performance of NSMs in a similar spirit as receiver operator characteristic (ROC) curves [53]. The presented *Accuracy-Generalization Surface* (AGS) reflects the pairing of classification accuracy (performance), empirical margin error (learned), and prototype utilization (theoretical) to provide a holistic view of an NSM method's performance. This construct provides ability to capture the entire span of an NSM method's capabilities across all operating conditions, and provides a mechanism to postulate potential scenario-based impacts.

An examination of NSM methods over four data sets supported several relative performance trends. Cost-minimizing approaches such as Avg-M-Cost and SM-Cost, achieve the best published classification accuracy over both the ZWAVE, and LCVF data sets, offering increases from 96.5% [27] to 98.9% and 97.2% [4] to 98.0%, respectively. These methods benefit from a more targeted approach to define the decision boundary, typically offering the highest mean classification in low margin error regions. In the Mice protein expression levels (MICE) data set, they dominate nearly 80% of operating points with the highest average classification. However, at higher levels of margin error, cost minimizing approaches did not generalize as well as other approaches.

Clustering approaches such as PC and Split-Medoid insert prototypes to represent primary distributions in the data and showed the best trends in generalizability, with strong performance in the presence of high noise (*i.e.* the United States Postal Service (USPS)

data set). While clustering approaches generalize well, they typically do not offer high accuracy.

The Silhouette-F hybridized approach offers exceptionally promising trends in generalizability across all the tested methods. It presents the highest mean classification accuracy for the MICE and Lunar Crater Volcanic Field (LCVF) data sets in regions of moderate to high error, and nearly all of the Z-Wave wireless protocol (ZWAVE) data set. Ultimately, Silhouette-F represent a preferred solutions to balance accuracy-generalization trade-offs and achieve superior classification accuracy over a large range of potential margin error.

In addition, Dynamic GRLVQI (DGRLVQI) demonstrated statistically significant improvements over GRLVQI in 84% of nearly 6,000 runs (over 4 data sets and 11 NSM methods), simultaneously offering mean gains in classification accuracy of 1% to 2%, with 4% – 56% fewer prototypes, driving improved generalization. These results affirmatively answer the research question and show that DGRLVQIs improve accuracy and generalization.

5.2 Future Work

An expanded library of NSM methods can provide additional options for future experimentation. For example, a new Silhouette-based NSM may benefit from a more complete implementation of the original Silhouette algorithm (that removes the simplification step introduced in Silhouette-F). While this would allow the NSM method to be less influenced by a single sample, it may constitute a significant computation increase. However, this may be a promising option for noisy data sets. Another method to potentially improve generalization in dynamic NSM methods, such as M-Cost, would be to augment the sample pools for region identification or evaluation. The inclusion of samples from nearby foreign classes better considers a potential new prototype’s disruptive influence on neighboring classes. Other considerations include new NSM methods based on promising

clustering work, such as crystal clustering [69], density based scanning (DB-Scan) [56], and hierarchical clustering [21].

The current body of NSM methods can be extended by allowing both the addition and deletion of prototypes [70–73]. Deletion is motivated by works in Feed Forward Networks [74], where neural connections between layers are pruned. Related experimentation limiting the growth of prototype networks indicates potential profitability.

Dynamic approaches can be extended to another relevant and promising GLVQ variant, Generalized Matrix LVQ (GMLVQ) [75]. This variant applies a matrix of pairwise relevance weight updates and has been found to produce higher classification accuracies compared to GRLVQ and GRLVQI.

The current form of GLVQ, GRLVQ, and GRLVQI define a cost function that captures total cost, favoring large classes. Modifying that cost function to account for a non-equal class distribution is a logical step and could be accomplished through a class dependent weighting function.

Several aspects of the GRLVQI training process and classifier configuration present opportunities for further investigation. The slope of the sigmoid function as part of cost calculations in Section 2.1.2 strongly influences prototype updates. As larger slopes yield stronger updates for prototypes near decision boundaries, there are instances where they may lead to faster convergence. Experimentation indicates that similar convergence is not easily replicated by simply adjusting learn rates or training time. Further investigation of the slope parameter may yield improved training time, classification accuracy, or better generalization. Also, while it is commonly accepted to decay learn rates for prototype updates from initially coarse (large) to more fine (small) values, [2] suggests that smaller initial learn rates may allow feature relevance weights to adjust before the bulk of training occurs. Closely related to the learn rate values and schedule is the automatic termination of training. In experimentation, all classifiers were allowed to train for the same fixed

number of training epoch to ensure converged solutions. A heuristic such as stability of classification accuracy over an interval, as recommended in [54], could allow a classifier to recognize when it has ceased useful training at a particular learning rate or for a particular number of prototypes. This early exit ability could drastically reduce training time and allow training time to serve as a more compelling evaluation measure. An artificial introduction of noise into training data may allow the creation of more general classifier models.

Appendix A: Supplemental Data Set Analysis

This appendix provides supplemental results and analysis not included in the bulk of the document. Each section investigates the dynamic addition of prototypes in support of the hypothesis that *dynamic approaches will offer improved classification accuracy and generalization*. Section A.1 shows analysis that spans multiple data sets. Sections A.2-A.5 are specifically focused on the United States Postal Service (USPS), Mice protein expression levels (MICE), Z-Wave wireless protocol (ZWAVE), and Lunar Crater Volcanic Field (LCVF) data sets(respectively) and are organized similar to Chapter 4.

A.1 Combined Charts and Analysis

A.1.1 Averaged Time Series Results.

An understanding of NSM method performance, trends in behavior, and links between accuracy and cost are obtained by considering a time-series perspective, averaged over each independent run ($30 \times K$ of them). First, classification accuracy for the MICE and LCVF data sets are examined in Fig. A.1 and A.2. As an alternate to epochs, the x -axis is marked by Prototype Addition Stage (PAS). Periods of transient behavior are commonly seen at the beginning of each new NSM stage as learn rates reset to coarse values and networks adapt to new prototype insertions. Initial transient epochs within each stage are removed from figures to aid viewing. DGRLVQI accuracy is observed to increase as a function of training time. Mean time-averaged performance typically surpasses the accuracy of the non-dynamic baseline before training concludes, indicating similar performance with fewer prototypes.

In the MICE data set, mean improvements in classification accuracy slightly taper with progressively more prototype addition stages. The LCVF data set shows a more dramatic tapering of classification accuracy after approximately 45 prototype addition

stages. Tapering, yet still increasing accuracy, is an indicator of convergence without over-fitting. For both MICE and LCVF data sets, Avg-M-Cost and SM-Cost appear to lead mean classification accuracy; lower accuracies are observed for Silhouette-F and Silhouette-L. While the individual accuracy profiles of each NSM are not evidence of statistical significance between methods (table A.1), they promote a general understanding of mean performance and are relevant to the following discussion of cost.

Since cost drives GRLVQI's gradient descent process to promote efficient prototype updates, NSM methods that best reduce cost are anticipated to offer prototype placements with better boundary definition and classification accuracy. Cost (Eqn. 2.2) is observed with respect to PAS (in Fig. A.1 and Fig. A.2), with relatively smooth downward trends. In the MICE data set, Avg-M-Cost, and M-Cost show low total costs around 25.4. In LCVF, S-Cost and SM-Cost have low total costs of approximately 30.7 and 32.1 (respectively). While specific comparisons between NSM methods is deferred until later discussion, the general trends of NSM cost and accuracy (over the same PAS axis) offers insight to their relationship.

A connection between increasing classification accuracy and decreasing total cost is observed in Fig. A.1 and Fig. A.2. However, methods that result in the lowest total cost do not necessarily yield the highest classification accuracies. While this may seem surprising as the gradient descent of the cost function drives optimization in GRLVQI, there are several reasons that a global cost reduction technique may not offer best classification performance. First, all samples, whether correctly or incorrectly classified, contribute to the total cost. Despite the high costs incurred near the class boundary, classification accuracy is only actually impacted when samples fall on the wrong side of the boundary. A method that specifically considers these misclassified samples when inserting new prototypes, such as M-Cost, appears to provide a better balance of accuracy and total cost as seen in Fig. A.1.

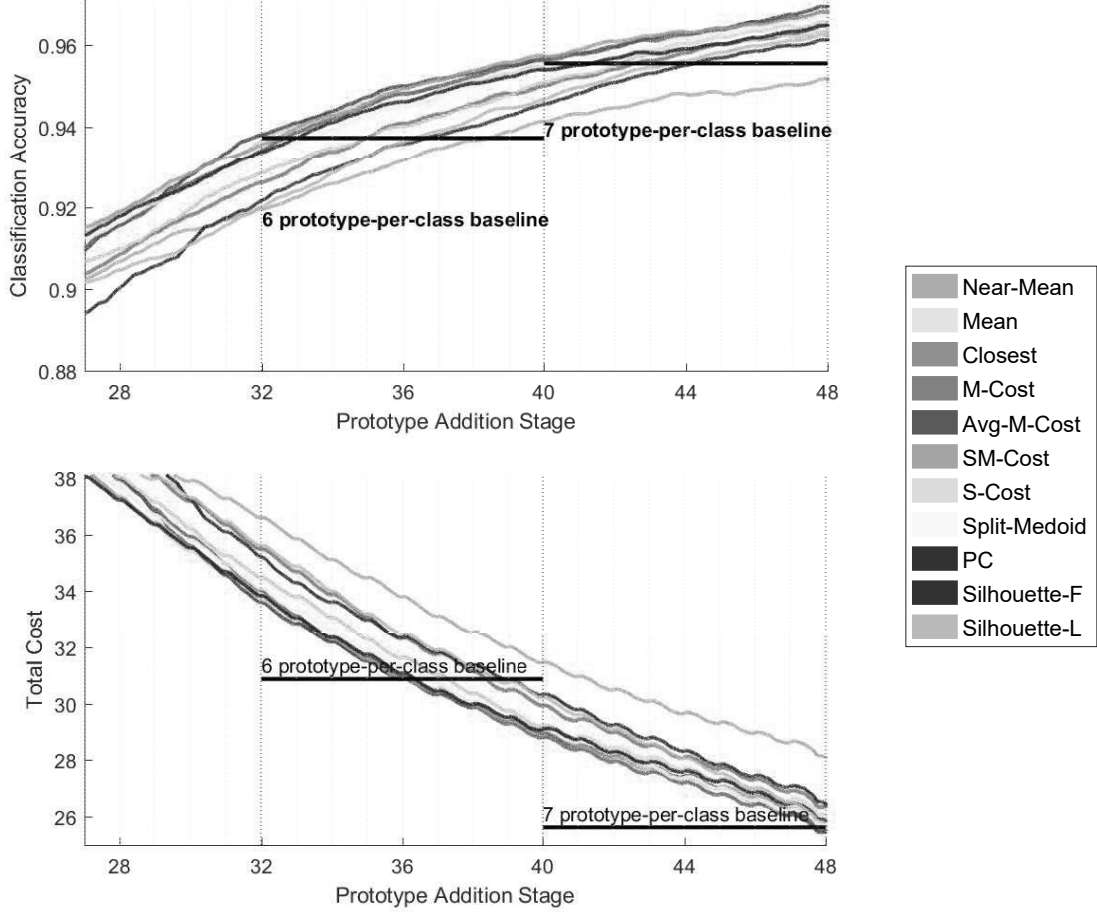


Figure A.1: Network Structure Modification (NSM) methods are compared by prototype addition stage for the Mice protein expression (MICE) data set. Solid horizontal lines show non-dynamic baseline performance. By the end of 48 prototype addition stages, each NSM method has the opportunity for the same number of prototypes (8 initial + 48 additional) as the baseline initialized with 7 per each of the 8 classes. Performance is shown for **(top)** classification accuracy and **(bottom)** total cost.

The pure reliance on global cost reduction is further undermined when the assumptions of classes with similar sample densities (classes with roughly equal numbers of samples) prove not to be true. In these cases, one under-represented (low density) class can easily be overlooked. While the MICE data set maintains roughly equal samples per class, the

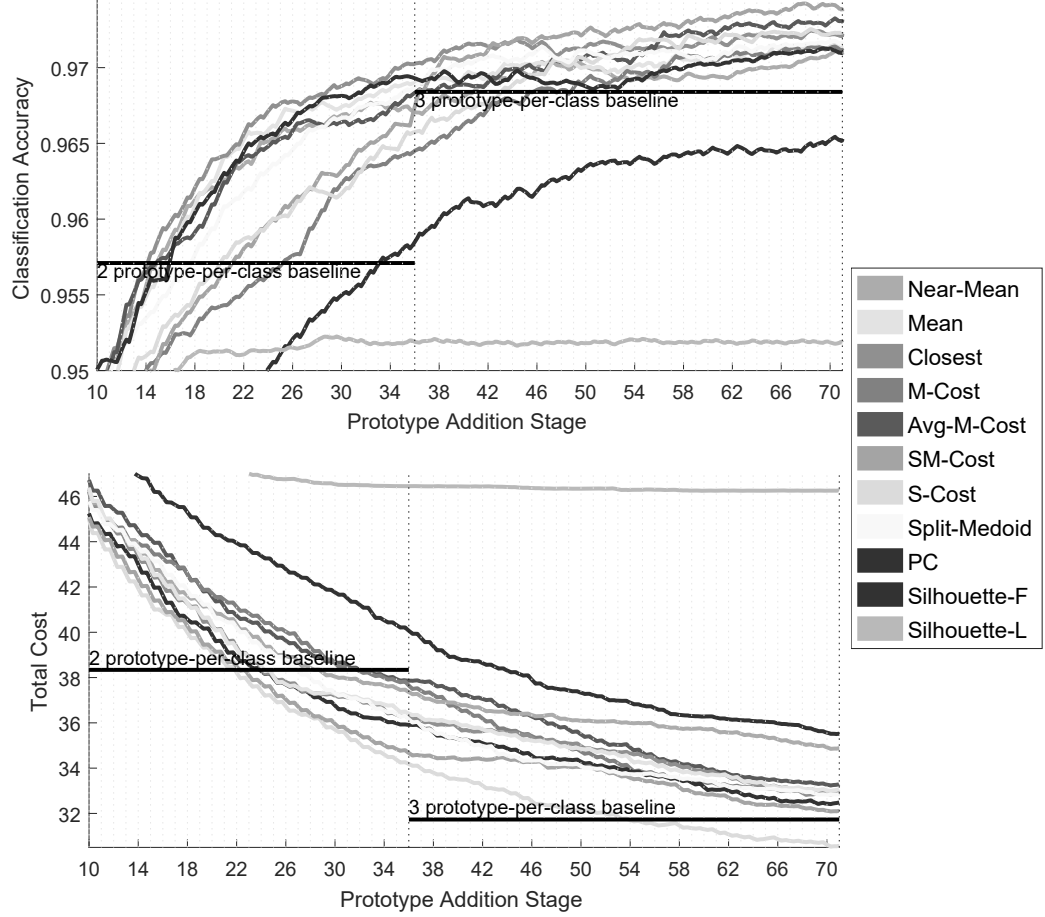


Figure A.2: NSM methods trained with the Lunar Crater Volcanic Field (LCVF) data set are compared by prototype addition stage. Solid horizontal lines show non-dynamic baseline performance. By the end of 70 prototype addition stages, each NSM method has the opportunity for the same number of prototypes (35 initial + 70 additional) as the baseline initialized with 3 per each of the 35 classes. Performance is shown for **(top)** classification accuracy and **(bottom)** total cost.

LCVF data set represents the contrary case, where Avg-M-Cost shows one of the highest classification accuracies (even though it does not offer one of the lowest total costs).

Table A.1: General NSM performance, averaged over $30 \times K$ training profiles, where $K = 5$ for MICE and $K = 3$ for LCVF. Accuracy and corresponding cost performance are shown for peak training profile values.

NSM Method	MICE				LCVF			
	Accuracy		Cost		Accuracy		Cost	
	μ	σ	μ	σ	μ	σ	μ	σ
Near-Mean	95.39	1.88	28.01	2.96	97.13	1.02	34.82	4.30
Mean	96.74	1.50	26.09	2.36	97.19	0.99	32.97	2.61
Closest	96.60	1.56	26.33	2.35	97.28	0.97	32.80	2.57
M-Cost	96.94	1.55	25.46	2.34	97.21	1.01	33.00	2.61
Avg-M-Cost	97.08	1.40	25.39	2.26	97.37	0.89	33.21	3.01
SM-Cost	96.95	1.21	25.72	2.19	97.47	0.94	32.17	2.56
S-Cost	96.47	1.79	25.57	2.31	97.27	1.03	30.71	2.12
Split-Medoid	96.85	1.47	25.68	2.42	97.22	0.93	32.82	2.50
PC	96.67	1.51	25.77	2.27	97.15	0.93	32.31	2.23
Silhouette-F	96.25	1.74	26.42	2.48	96.60	1.00	35.49	2.81
Silhouette-L	96.39	1.46	25.82	2.36	95.25	1.26	46.58	5.39
Baseline(3x35p)	95.76	1.89	25.58	2.85	96.92	0.97	31.70	2.67

A.1.2 Performance of NSM Methods over AGS.

The Accuracy-Generalization Surface (AGS) is applied to seven representative NSM method results from the MICE and LCVF data sets: Closest, M-Cost, M-Avg-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F. For ease of comparison, the three dimensional plots are standardized over the z -axis utilizing the group mean and standard deviation (of all NSM methods) to achieve unit variance. A floor is added to each plot in Fig. A.3 to show the standard deviation present in each NSM surface with respect to the aggregated group.

On a run-by-run basis, variations in training partitions can produce variations in classification accuracy and allocations. Avg-M-Cost consistently displays low standard deviation in accuracy as it targets classes with high average cost, creating a clear order of prototype allocation priorities that also avoids strong influence from training partition disparities. In LCVF, Avg-M-Cost yields some of the lowest classification accuracy standard deviation as the number of utilized prototypes reaches their maximum number.

S-Cost results in some of the highest standard deviation (lowest repeat-ability) when compared to other NSM methods for the MICE data set. S-Cost’s additional random process for defining a subset region causes the large standard deviation. However, in LCVF, S-Cost shows low relative standard deviation, likely because it does not consider LCVF’s different class sizes and gravitates toward larger classes first.

PC demonstrates regions of low relative standard deviation (in accuracy) as indicated by cooler colored plot floors in Fig. A.3. Since PC biases against frequent prototype addition within one region, it can bypass particularly challenging classes that do not show improvement from additional prototypes. This can result in a more distributed placement of prototypes, whereas other NSM methods may endlessly identify and add prototype after prototype to the same region without benefit (and may not be consistent experiment-to-experiment). While other NSM methods add one new prototype, PC replaces an old prototype with two new ones along the maximum variance of its in-class samples. This re-balancing leads to more consistent prototype locations, as the two new position are coordinated to reflect the spread of the cluster.

NSM surfaces heights in Fig. A.3 are the mean classification accuracy (relative to the group), and elevated regions indicate strong relative performance. In the MICE data set, several elevated sections of the S-Cost surface are observed for less than 30 prototypes, corresponding to regions between $0.04 < E_{HSV} < 0.06$. In the LCVF data set, most surfaces appear smooth, with the exception of several portions of M-Cost; Silhouette-F shows relative large surface heights for $E_{HSV} > 0.1$.

A.2 USPS Generalized Relevance LVQ Improved (GRLVQI) Results and Discussion

This section provides supplemental analysis plots for the USPS data set. Section A.2.1 provides experimental parameters. Section A.2.2 shows feature correlation and covariance. Section A.2.3 shows averaged performance (accuracy and cost) as a function of prototype

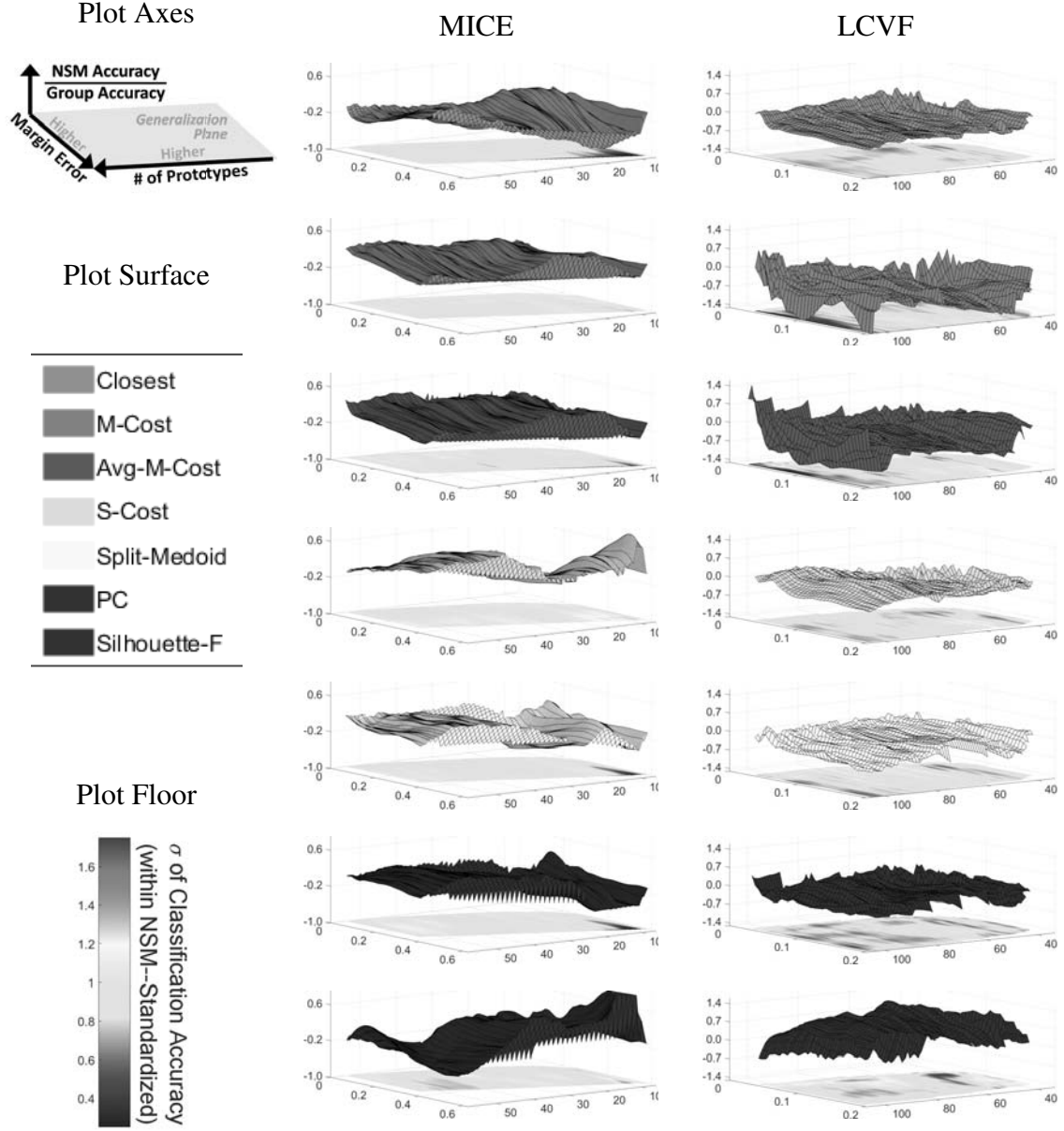


Figure A.3: Classification accuracy (z -axis) of seven representative NSM methods for (**left column**) MICE and (**right column**) LCVF are shown against the margin-based empirical error and the number of utilized prototypes (x and y -axes respectively). For ease of viewing and comparison, the data is sphered over all NSM methods for each data set.

addition stage. Section A.2.4 shows baseline and converged performance from the perspective of individual operating points. Section A.2.5 shows prototype utilization. Section A.2.6 shows margin analysis and plots of E_{HSV} and E_{CGNT} error. Section A.2.7 concludes this chapter of USPS analysis results with a accuracy generalization surface plot.

A.2.1 Parameters and Data.

The learn-rates, training parameters and pertinent notes for dynamic prototype experiments with the GRLVQI algorithm and USPS dataset are provided in Table. A.2.

Table A.2: USPS GRLVQI parameters.

	Training Epochs	Learn Parameters		Conscience Parameters		Data Set:	USPS
		α^w	α^l	γ	β	File Prefix:	
USPS GRLVQI	$1 \leq Ep \leq 10$	5.00e-2	1.00e-3	2.50	0.35	Data Collected:	91049 1/class at random 0.4,0.6 70 1 Relevance Weighted Euclidean None 5
	$11 \leq Ep \leq 20$	3.00e-2	5.01e-4	2.10	0.29	Data Processed:	
	$21 \leq Ep \leq 30$	1.70e-2	2.32e-4	1.77	0.24	Prototype Initialization:	
	$31 \leq Ep \leq 40$	9.65e-3	1.08e-4	1.49	0.20	Prototype Addition Stages:	
	$41 \leq Ep \leq 50$	5.47e-3	5.01e-5	1.25	0.17	Slope of sigmoid:	
	$51 \leq Ep \leq 60$	3.11e-3	2.32e-5	1.05	0.14	Distance measure:	
	$61 \leq Ep \leq 70$	1.76e-3	1.08e-5	0.90	0.12	Thresholding:	
	$71 \leq Ep \leq 80$	1.00e-3	5.00e-6	0.75	0.10	# of Cross-Validation folds:	

A.2.2 Feature Covariance and Correlation.

The covariance and correlation are calculated between feature dimensions and shown in Fig. A.4 and Fig. A.5 respectively. As handwritten digits possess a continuous structure, some joint variability and high correlation between many of the neighboring pixels is observed. Due to the linearization of the original 16×16 image, neighboring pixels are represented in the feature vector as adjacent neighbors or separated by 16 entries. These correlations are mostly between large white spaces that are uninterrupted by writing. While high correlation may represent potentially redundant features, the relevance weighting function inherent in the family of GRLVQ algorithms effectively extracts salient features that assist the task of classification. Ultimately, strong correlation does not serve as a

detractor to the general process or the intent of this research effort as efforts are primarily concerned with making predictions as opposed to interpreting the learned classification boundaries.

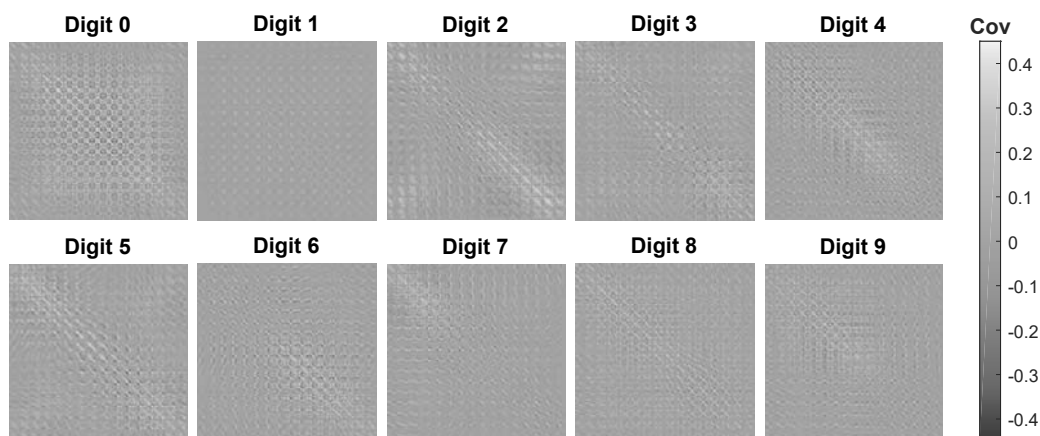


Figure A.4: The covariance is shown between features within each labeled class over a 256×256 covariance matrix.

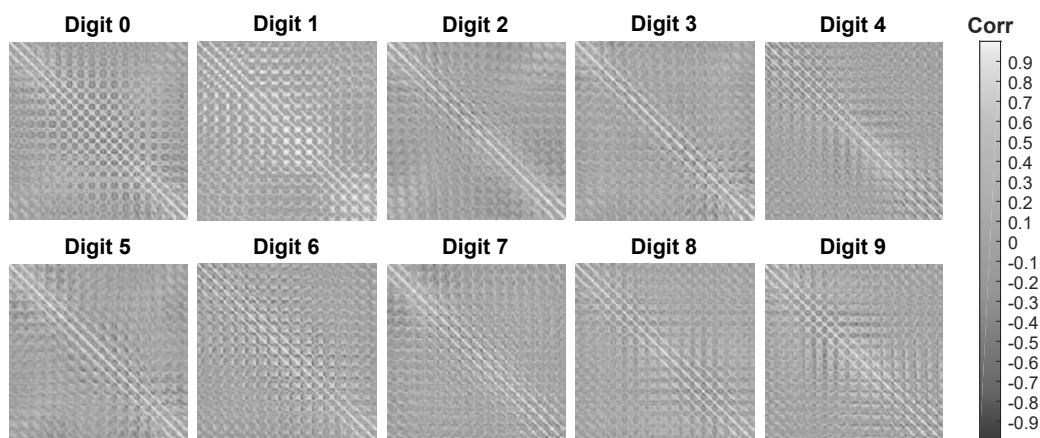


Figure A.5: The correlation is shown between features within each labeled class over a 256×256 correlation matrix.

A.2.3 Averaged Performance as a Function of Training.

Training and validation is conducted with 11 dynamic algorithms representing four categories of approaches: simple (computationally efficient), cost-minimizing, clustering, and hybrid approaches. A fully initialized 80-prototype non-dynamic GRLVQI Baseline is used for comparison. Figure A.6 reports validation accuracy and cost as a function of training time. In lieu of epochs, the x -axis is marked and segmented by prototype addition stage. This illustration of training shows the growth in classification accuracy and reduction in cost as NSM methods have the opportunity to add additional prototypes to their networks. For convenience in plotting, the initial transient regions from each prototype addition stage are removed.

Figure. A.6 presents a training profile view of NSM methods. First, this allows a general top level view of NSM performance, seeing peak accuracies and relative performance between NSM methods. Second, due to the link with time as a variable, trends of growth and convergence are easily observed. Periods of solid increases in classification accuracy, followed by continual declines may indicate over-training. Trends of increasing accuracy that don't taper off may suggest a potential for improved performance with additional prototypes or training time. Third, this view of the data also shows the interaction between cost and accuracy, and shows if the cost based gradient descent is appropriately guiding the non-dynamic components of training. Fourth, it answers the question: "what is the validation response to the training data". Last, the averaged training curve can be easily interpreted in relation to the non-dynamic baseline, by examining crossings.

As the Fig. A.6 training profile is examined, the relationship between increasing classification accuracy and decreasing total cost is observed. However, NSM methods that lead to the greatest reduction in cost do not necessarily provide the best classification accuracy. For example, while S-Cost achieves the lowest cost, it falls behind Split-Medoid, PC, and sometimes others in classification accuracy. When compared against non-dynamic

GRLVQI classifiers initialized with 80 prototypes (the maximum number allowed for the dynamic methods), all but the two Silhouette approaches achieve or surpass the baseline (non-dynamic) classification accuracy within the training time and prototype constraints.

A.2.4 Baseline and Converged Model Performance.

In Table A.3 NSM methods are evaluated against a non-dynamic baseline. The point at which each NSM meets the baseline accuracy and the corresponding number of prototypes required is used as a performance point to identify the potential reduction in prototypes when accuracy is held constant.

Table A.3: NSM performance for 150 independent trials of USPS is shown when the non-dynamic baseline is chosen as an accuracy performance point. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and prototype utilization (when compared to baseline performance). Statistically significant improvements are shown in bold, statistically similar results are shown in italics.

	Match Baseline Performance									
NSM Method	Met	Accuracy					Prototypes			
		μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	
Near-Mean	127	90.74	0.69	0.05	(−0.01 – 9.31)	66.84	9.95	13.16	(11.25 – 13.16)	
Mean	144	90.85	0.73	0.16	(0.12 – 9.31)	62.12	9.41	17.88	(16.07 – 17.88)	
Closest	132	90.79	0.72	0.10	(0.02 – 9.31)	64.73	9.60	15.27	(13.42 – 15.27)	
M-Cost	102	90.57	0.75	−0.12	(−0.22 – 9.31)	73.35	6.61	6.65	(5.38 – 6.65)	
Avg-M-Cost	108	90.62	0.72	−0.07	(−0.17 – 9.31)	72.88	7.01	7.12	(5.77 – 7.12)	
SM-Cost	108	90.64	0.71	−0.04	(−0.13 – 9.31)	71.11	9.13	8.89	(7.13 – 8.89)	
S-Cost	145	90.84	0.73	0.15	(0.11 – 9.31)	66.85	7.37	13.15	(11.73 – 13.15)	
Split-Medoid	142	90.83	0.75	0.14	(0.09 – 9.31)	62.01	9.20	17.99	(16.23 – 17.99)	
PC	144	90.84	0.75	0.16	(0.12 – 9.31)	61.13	9.52	18.87	(17.04 – 18.87)	
Silhouette-F	20	89.47	0.77	−1.21	(−1.38 – 9.31)	78.60	4.55	1.40	(0.53 – 1.40)	
Silhouette-L	16	89.35	0.78	−1.33	(−1.50 – 9.31)	78.51	5.93	1.49	(0.35 – 1.49)	
Baseline(8x10p)	150	90.69	0.73	-	-	80	0.00	-	-	

Table A.4: Converged NSM model performance is shown for $30 \times K$ independent trials ($K = 5$ for USPS, MICE, and ZWAVE, and $K = 3$ for LCVF). Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and corresponding prototype utilization, when compared to baseline performance. Statistically significant improvements are shown in bold, statistically similar results are shown in italics.

NSM Method	Converged							
	Accuracy				Prototypes			
	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI
Near-Mean	91.36	0.64	0.68	(0.55 – 9.31)	74.97	5.49	5.03	(3.98 – 5.03)
Mean	91.75	0.62	1.06	(0.94 – 9.31)	76.11	4.47	3.89	(3.03 – 3.89)
Closest	91.39	0.61	0.70	(0.59 – 9.31)	73.55	5.44	6.45	(5.40 – 6.45)
M-Cost	91.02	0.72	0.34	(0.19 – 9.31)	76.97	4.03	3.03	(2.26 – 3.03)
Avg-M-Cost	91.01	0.74	0.32	(0.19 – 9.31)	76.27	4.63	3.73	(2.84 – 3.73)
SM-Cost	91.10	0.70	0.41	(0.28 – 9.31)	76.51	3.97	3.49	(2.72 – 3.49)
S-Cost	91.56	0.63	0.87	(0.77 – 9.31)	76.41	3.75	3.59	(2.87 – 3.59)
Split-Medoid	91.62	0.69	0.93	(0.83 – 9.31)	75.43	5.22	4.57	(3.57 – 4.57)
PC	91.71	0.66	1.02	(0.90 – 9.31)	75.72	4.84	4.28	(3.35 – 4.28)
Silhouette-F	89.80	0.69	−0.89	(−1.04 – 9.31)	72.62	9.04	7.38	(5.64 – 7.38)
Silhouette-L	89.72	0.68	−0.97	(−1.11 – 9.31)	70.76	9.63	9.24	(7.39 – 9.24)
Baseline(8x10p)	90.69	0.73	-	-	80.00	0.00	-	-

A.2.5 Performance with respect to Prototype Utilization.

Similar classification performances are observed in Figure A.8. In order to de-clutter the space and allow for better viewing, well performing NSM methods are selected, representative of the simple, cost-minimizing, clustering, and hybrid approaches. A balance is maintained of novel NSM methods and methods from the literature. Consequently Closest, M-Cost, Avg-M-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F are shown in Fig. A.8. In this simplified view, general trends are more easily observed, with novel clustering approaches appearing to lead performance, followed by S-Cost and SM-Cost.

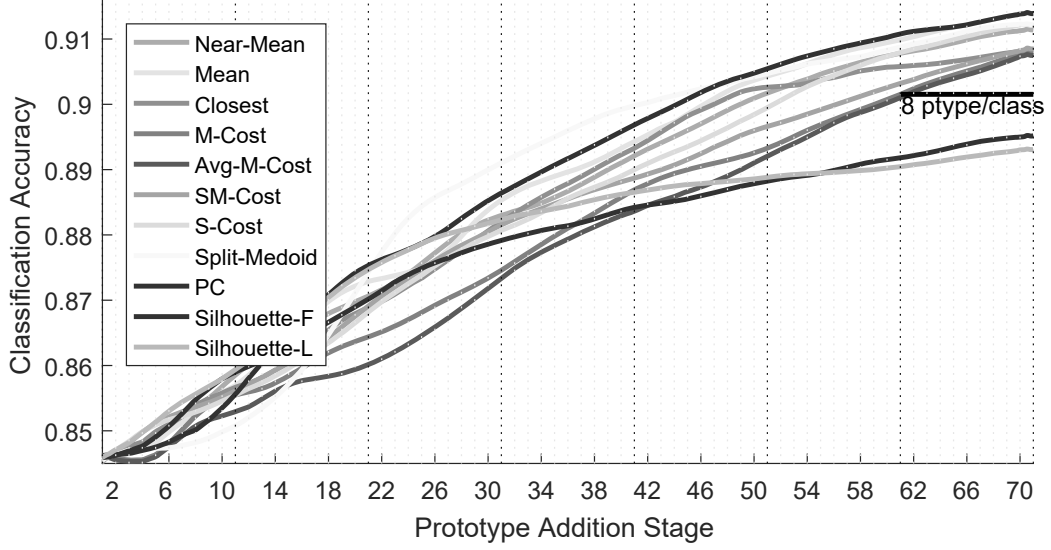
A.2.6 Margin Analysis.

After examining how total cost and classification accuracy change as a function of the number of prototypes in Section A.2.4 and Section A.2.5, the relationship between margin error and the number of prototypes is examined.

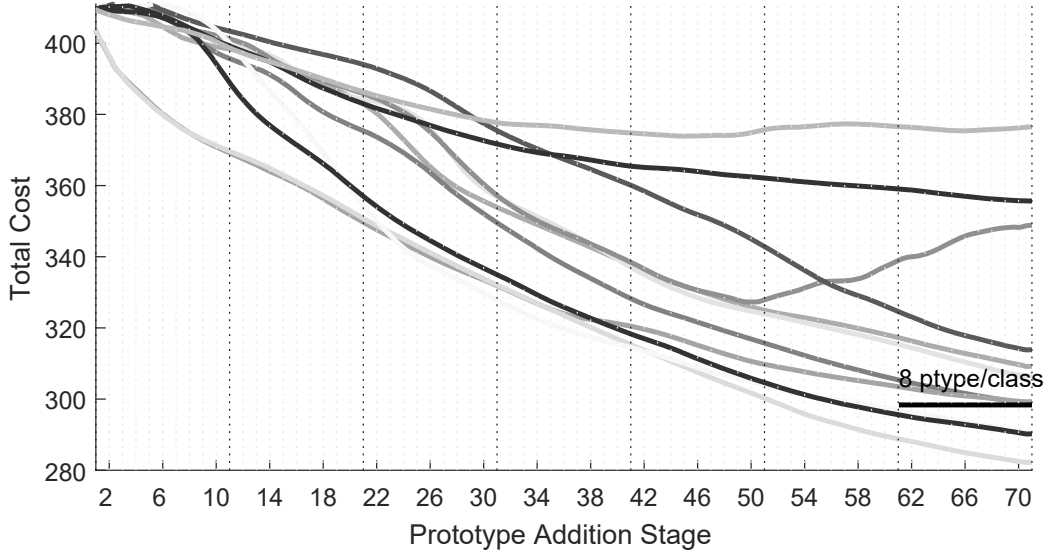
A.2.7 Relative Comparison.

The performance of GRLVQI based NSM methods are shown for the USPS dataset in Fig. A.11. Three columns of plots are compared over the axes of margin error and the number of prototypes, representing empirical and theoretical components of the generalization error as described in Section 3.3 . The z-axis depicts **(left)** classification accuracy, **(middle)** standard deviation in classification accuracy, and **(right)** classification accuracy, standardized over group the as $\sim N(0, 1)$.

PC and Split-Medoid NSM methods demonstrate such an ability as they tracked the past assignments of prototypes. Often prototypes are assigned to regions with high rates of misclassification or error. PC and Split-Medoid make similar determinations of need, but bias these decisions with the number of resources that have already been assigned to that region. This inherently promotes improved prototype utility, as it bypasses regions that likely will not benefit from the added resource. While M-Cost and Avg-M-Cost target decision boundaries, PC and Split-Medoid take a more conservative approach and concentrate on representing the distribution of the data. It was found that this often did not produce the highest overall classification accuracies, but was a more stable and generalizable solution. In the analysis of the USPS data set, Split-Medoid offered performance (between 20–55 utilized prototypes) that yielded over 1% improvement when compared against NSM methods from the three different approach categories.



(a)



(b)

Figure A.6: Classification accuracy **(a)** and Total Cost **(b)** versus prototype addition stage for seven prototype addition methods over the USPS data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6. One GRLVQI “baseline” trained with 80 fixed prototypes is included for reference. Plots are the average of 150 validation result curves (30 complete runs each consisting of $K = 5$ folds).

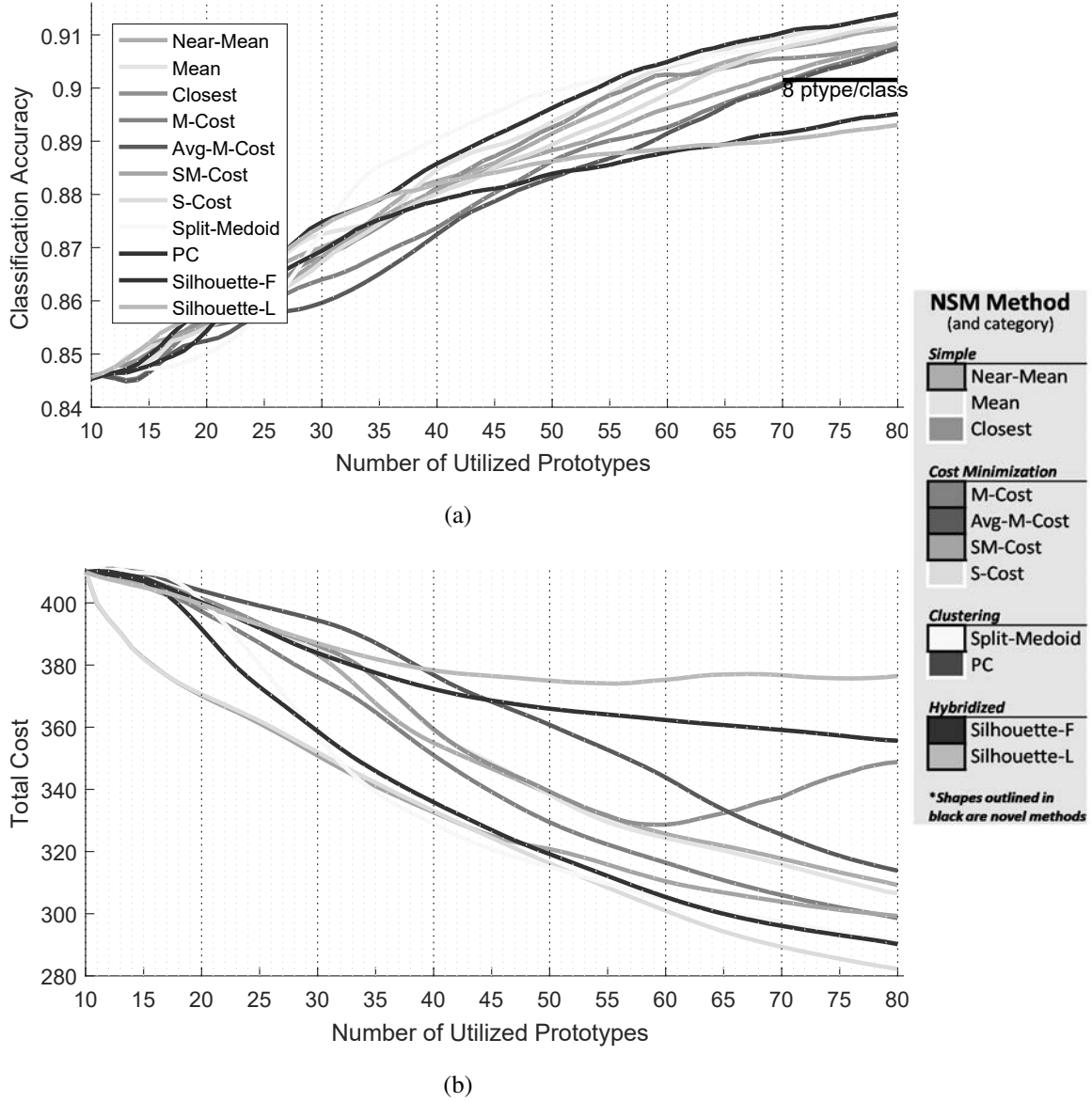


Figure A.7: Classification accuracy **(a)** and Total Cost **(b)** versus utilized number of prototypes for eleven prototype addition methods over the USPS data set with a GRLVQI algorithm. One GRLVQI “baseline” trained with 80 prototypes is included for reference. Validation accuracies from the 150 curves (30 complete runs each consisting of $K = 5$ folds) are pooled for each utilized prototype value then averaged (roughly 18,000 points per prototype value).

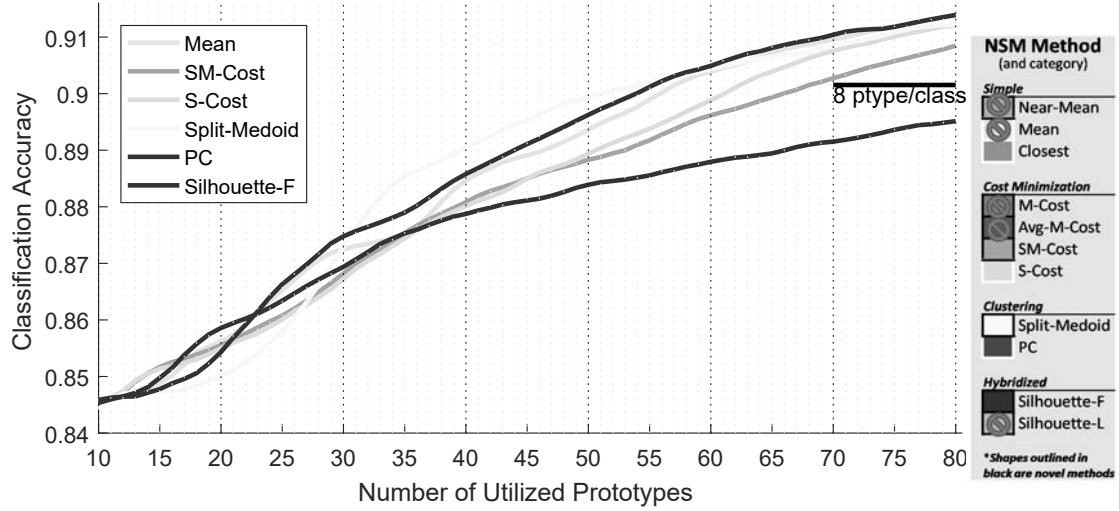


Figure A.8: Classification accuracy versus utilized number of prototypes for six representative GRLVQI NSM methods for the USPS dataset: Mean, SM-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F are plotted in addition methods over the USPS data set with a GRLVQI algorithm. A non-dynamic GRLVQI baseline trained with 80 prototypes is included for reference. Validation accuracies from the 150 curves (30 complete runs each consisting of $K = 5$ folds) are pooled for each utilized prototype value then averaged(roughly 18,000 points per prototype value).

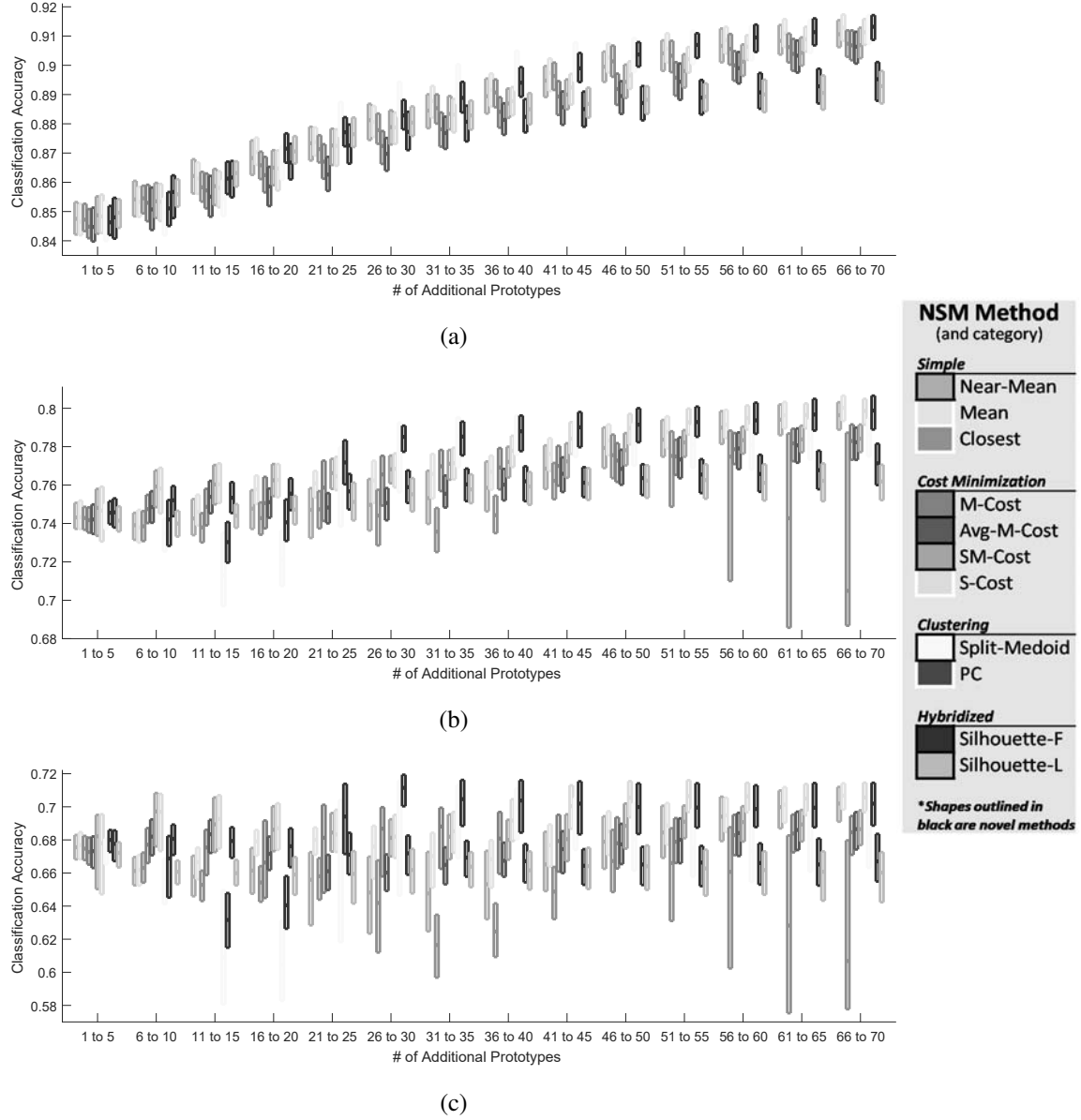


Figure A.9: For USPS-GRLVQI: NSM classification accuracies are compared against number of additional prototypes for a margin test point, θ_{cp} . Percentiles within each NSM method (5 prototype intervals starting with 10) are shown as the top, middle, and bottom of colored vertical boxes (75th, 50th, and 25th percentile). Plots (a), (b), and (c) correspond to θ_{cp} values of 0, $7.89e - 3$, and $1.18e - 2$.

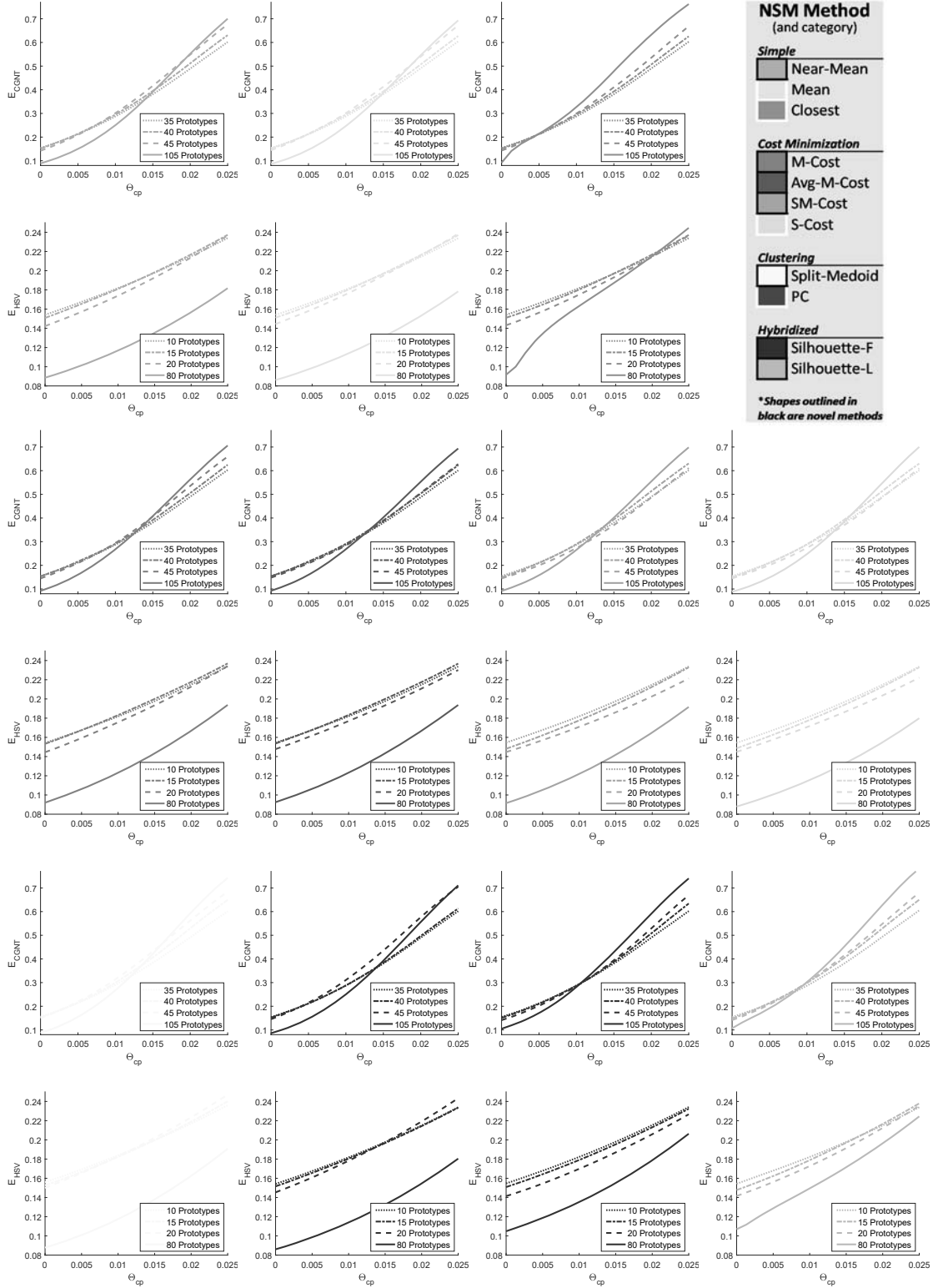


Figure A.10: NSM methods with 35, 40, 45, and 105 prototypes display CGNT(top) and HSV(bottom) Margin Error as a function of θ_{cp} .

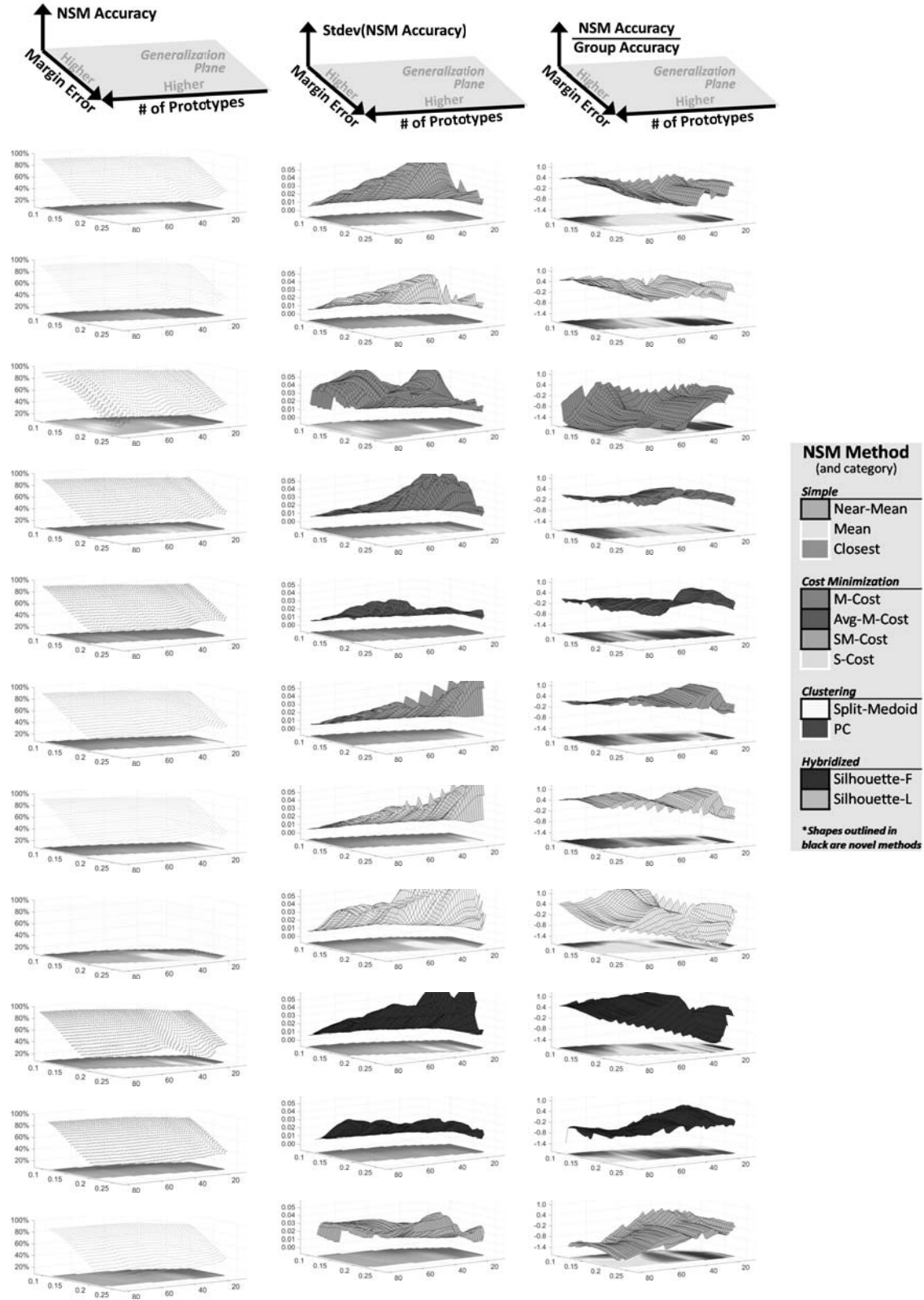


Figure A.11: Comparison of NSM methods for USPS dataset (top row shows axes for each column: classification accuracy, standard deviation, and relative accuracy (l-r).

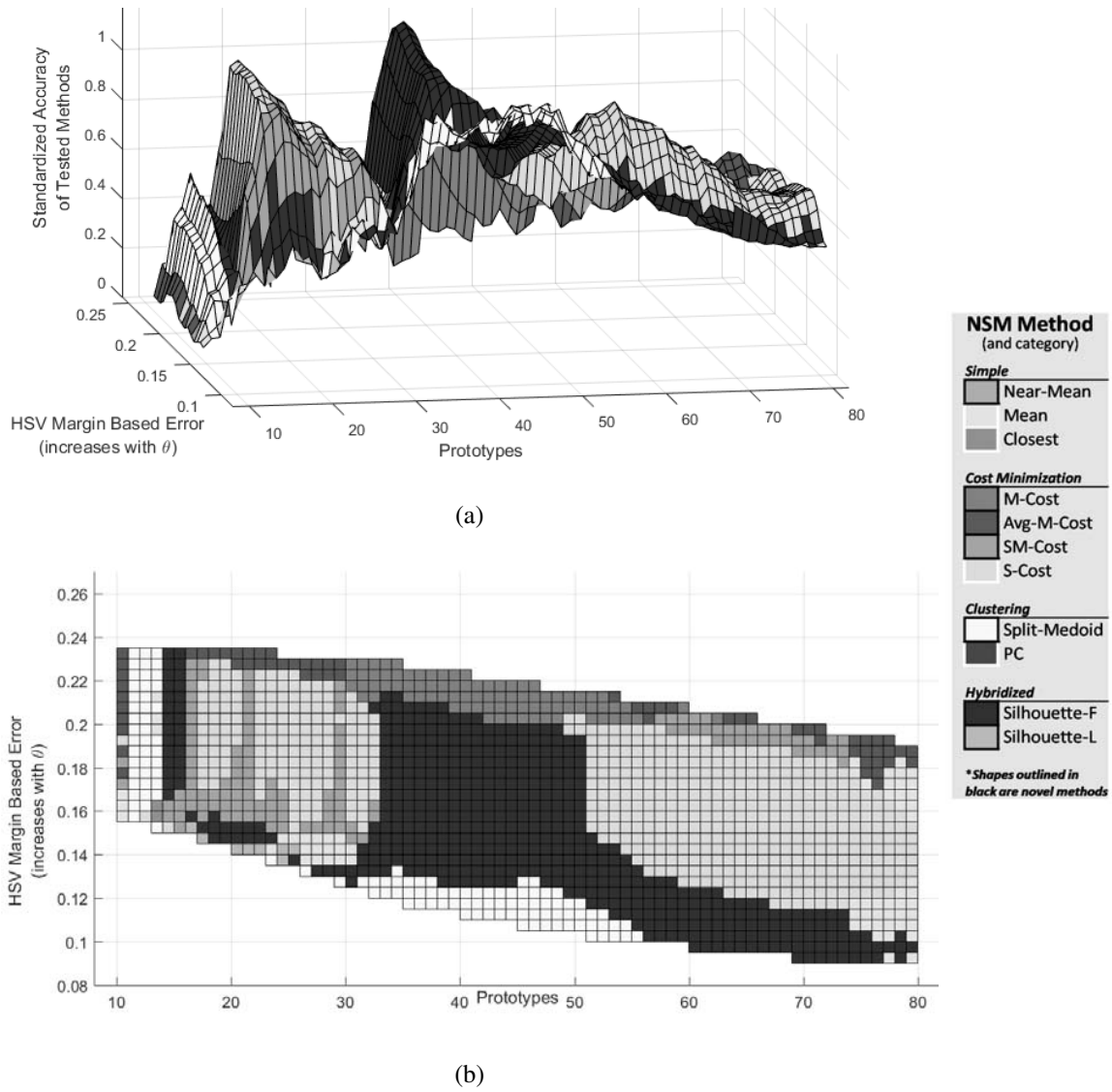


Figure A.12: (a) This composite of Fig. A.11 (right-column) selects the best performing NSM method over each grid point. (b) This graph is simplified by flattening the z-axis. This flattened view indicates NSM methods with the highest average classification accuracy over a range of utilized prototypes and margin error values.

A.3 MICE GRLVQI Results and Discussion

This section provides supplemental analysis plots for the MICE data set. Section A.3.1 provides experimental parameters. Section A.3.2 shows feature correlation and covariance. Section A.3.3 shows averaged performance (accuracy and cost) as a function of prototype addition stage. Section A.3.4 shows baseline and converged performance from the perspective of individual operating points. Section A.3.5 shows prototype utilization. Section A.3.6 shows margin analysis and plots of E_{HSV} and E_{CGNT} error. Section A.3.7 concludes this chapter of MICE analysis results with a accuracy generalization surface plot.

A.3.1 Parameters and Data.

Table A.5 shows the learning parameters for use in experimentation.

Table A.5: MICE GRLVQI parameters.

	Training Epochs	Learn Parameters		Conscience Parameters		Data Set:	MICE
		α^w	α^l	γ	β	File Prefix:	
MICE GRLVQI	$1 \leq Ep \leq 125$	2.50e-2	5.00e-3	2.50	0.35	Data Collected:	859
	$126 \leq Ep \leq 250$	8.21e-3	1.87e-3	2.10	0.29	Data Processed:	
	$251 \leq Ep \leq 375$	2.68e-3	6.97e-4	1.77	0.24	Prototype Initialization:	
	$376 \leq Ep \leq 500$	8.77e-4	2.59e-4	1.49	0.20	Prototype Addition Stages:	
	$501 \leq Ep \leq 625$	2.86e-4	9.67e-5	1.25	0.17	Slope of sigmoid:	
	$626 \leq Ep \leq 750$	9.37e-5	3.60e-5	1.05	0.14	Distance measure:	
	$751 \leq Ep \leq 875$	3.06e-5	1.34e-5	0.90	0.12	Thresholding:	
	$876 \leq Ep \leq 1000$	1.00e-5	5.00e-6	0.75	0.10	# of Cross-Validation folds:	

A.3.2 Feature Covariance and Correlation.

The covariance and correlation of the un-scaled MICE data set is shown in Fig. A.32 and Fig. A.33 respectively. For most classes, the highest positive joint variability is seen between proteins “NR2A” and “pCAMKII” due to their high expression levels relative to the other proteins. High positive correlations are seen, especially between rough groupings in the first, middle, and last third of protein markers. These correlations must be accepted as another aspect of the data set and the LVQ classification process will likely not be strongly

impacted, since these features are utilized to make final predictions instead of establish individual causal links.

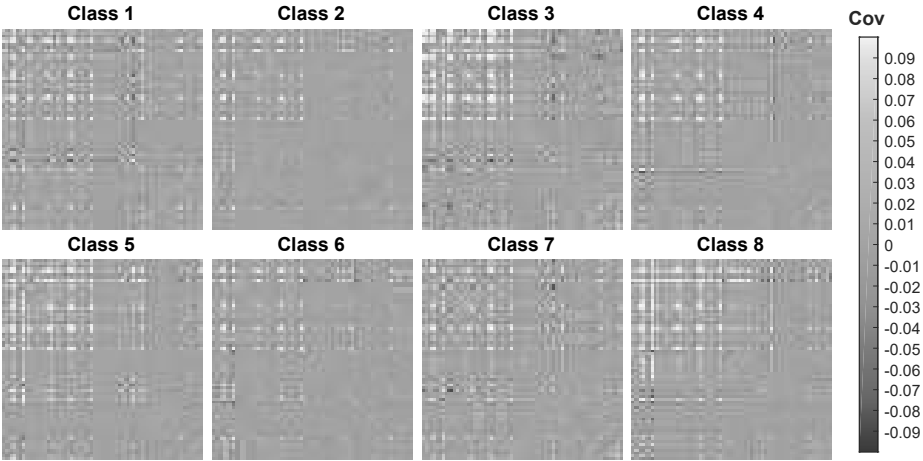


Figure A.13: The covariance between features of the MICE data set are shown within each labeled class over a 71×71 covariance matrix.

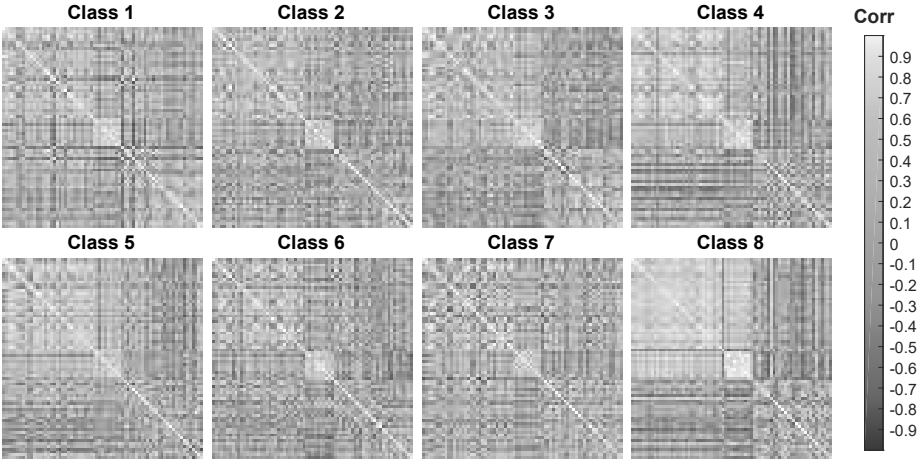


Figure A.14: The correlation between features of the MICE data set are shown within each labeled class over a 71×71 correlation matrix.

A.3.3 Averaged Performance as a Function of Training.

Figure A.15 displays both the classification accuracy and cost as a function of prototype addition stage.

It is noted in Fig. A.15 that the 2-prototype per class baseline shows a higher classification accuracy than the 3-prototype per class baseline (as well as a lower total cost). The 2-prototype per class baseline values were generated at a later date than the other data, and it is not clear if the differences are due to normal variation or a slight change in code or other parameter value that is not accounted for.

A.3.4 Baseline and Converged Model Performance.

The discussion of Table 4.4 in Section ?? is continued, replicating results pertinent to MICE as Table A.7. As matured models utilize near the maximum possible 56 prototypes, it is possible that training does not plateau or converge prior to the final prototype addition stage. As such, the positive slope in Fig. A.1 may suggest potential for additional growth, given more prototype resources. However, based on the decreasing trends observed uniformly over the NSM methods, it does not appear that additional training will constitute a material change to the relative comparison.

Next, performance of each NSM's matured (converged) classification models are compared against the non-dynamic model. An application of the central limit theorem permits the use of a paired student t-test [68]. Table A.7, shows mean classification accuracy and number of prototypes for the 150 paired trials. The mean improvement over the non-dynamic baseline and confidence interval of improvement are additionally shown for accuracy and the number of prototypes. In MICE, the converged models of Near-Mean achieve a statistically similar classification accuracy as the baseline. All other NSM methods show statistically significant improvements in accuracy with gains of 1.29%–1.87%, with Silhouette-F and M-Cost respectively forming the extreme cases (with

Table A.6: NSM performance for 150 independent trials is shown when the non-dynamic baseline is chosen as an accuracy performance point. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and prototype utilization (when compared to baseline performance). Statistically significant improvements are shown in bold, statistically similar results are shown in italics.

	Match Baseline Performance								
NSM Method	Accuracy					Prototypes			
	Met	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI
Near-Mean	78	94.85	1.92	-0.58	(-0.92 - 4.57)	50.73	6.67	5.27	(3.99 - 5.27)
Mean	130	95.72	1.61	0.29	(0.12 - 4.57)	46.41	6.61	9.59	(8.32 - 9.59)
Closest	136	95.84	1.62	0.40	(0.27 - 4.57)	47.61	6.20	8.39	(7.20 - 8.39)
M-Cost	128	95.74	1.58	0.31	(0.16 - 4.57)	46.22	6.31	9.78	(8.57 - 9.78)
Avg-M-Cost	131	95.73	1.63	0.29	(0.16 - 4.57)	45.93	6.44	10.07	(8.84 - 10.07)
SM-Cost	137	95.78	1.55	0.35	(0.20 - 4.57)	45.27	6.38	10.73	(9.50 - 10.73)
S-Cost	132	95.80	1.65	0.37	(0.21 - 4.57)	47.93	6.32	8.07	(6.86 - 8.07)
Split-Medoid	128	95.68	1.62	0.25	(0.12 - 4.57)	47.40	6.18	8.60	(7.41 - 8.60)
PC	136	95.81	1.64	0.38	(0.24 - 4.57)	46.76	6.82	9.24	(7.93 - 9.24)
Silhouette-F	122	95.61	1.59	0.18	(0.03 - 4.57)	49.56	5.73	6.44	(5.34 - 6.44)
Silhouette-L	121	95.59	1.56	<i>0.16</i>	(-0.02 - 4.57)	49.81	5.76	6.19	(5.08 - 6.19)
Baseline(7x8p)	150	95.43	1.77	0.00	(0.00 - 0.00)	56	0.00	0.00	(0.00 - 0.00)

mean accuracies of 96.72% and 97.3%). These achievements coincide with simultaneous improvements in the number of prototypes, offering statistically significant decreases of 2.0 – 3.6 prototypes on average. As matured models utilize near the maximum possible 56 prototypes, it is possible that training does not plateau or converge prior to the final prototype addition stage. As such, the positive slope in Fig. A.1 may suggest potential for additional growth, given more prototype resources. However, based on the decreasing trends observed uniformly over the NSM methods, it does not appear that additional training will constitute a material change to the relative comparison.

A.3.5 Performance with respect to Prototype Utilization.

Performance of NSM methods is next considered from the perspective of resource management. The validation data from Figure. A.15 is re-arranged according to the number

Table A.7: Converged NSM model performance is shown for 150 independent trials of MICE. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and corresponding prototype utilization, when compared to baseline performance. Statistically significant improvements are shown in bold, statistically similar results are shown in italics.

NSM Method	Converged								
	Accuracy					Prototypes			
	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI		μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI
Near-Mean	95.77	1.88	<i>0.34</i>	(−0.07 – 4.57)		52.41	4.50	3.59	(2.72 – 3.59)
Mean	97.09	1.12	1.66	(1.37 – 4.57)		52.75	3.38	3.25	(2.60 – 3.25)
Closest	97.28	1.14	1.85	(1.53 – 4.57)		53.70	2.56	2.30	(1.81 – 2.30)
M-Cost	97.17	1.07	1.74	(1.42 – 4.57)		53.25	3.41	2.75	(2.09 – 2.75)
Avg-M-Cost	97.24	1.13	1.80	(1.50 – 4.57)		53.35	3.30	2.65	(2.02 – 2.65)
SM-Cost	97.31	1.08	1.87	(1.57 – 4.57)		52.85	3.96	3.15	(2.39 – 3.15)
S-Cost	97.19	1.20	1.76	(1.45 – 4.57)		53.82	2.69	2.18	(1.66 – 2.18)
Split-Medoid	97.15	1.24	1.72	(1.43 – 4.57)		53.28	3.60	2.72	(2.03 – 2.72)
PC	97.23	1.16	1.80	(1.50 – 4.57)		53.61	2.85	2.39	(1.84 – 2.39)
Silhouette-F	96.72	1.30	1.29	(1.00 – 4.57)		53.93	2.85	2.07	(1.53 – 2.07)
Silhouette-L	96.73	1.31	1.30	(0.98 – 4.57)		53.95	2.67	2.05	(1.54 – 2.05)
Baseline(7x8p)	95.43	1.77	-	-		56.00	0.00	-	-

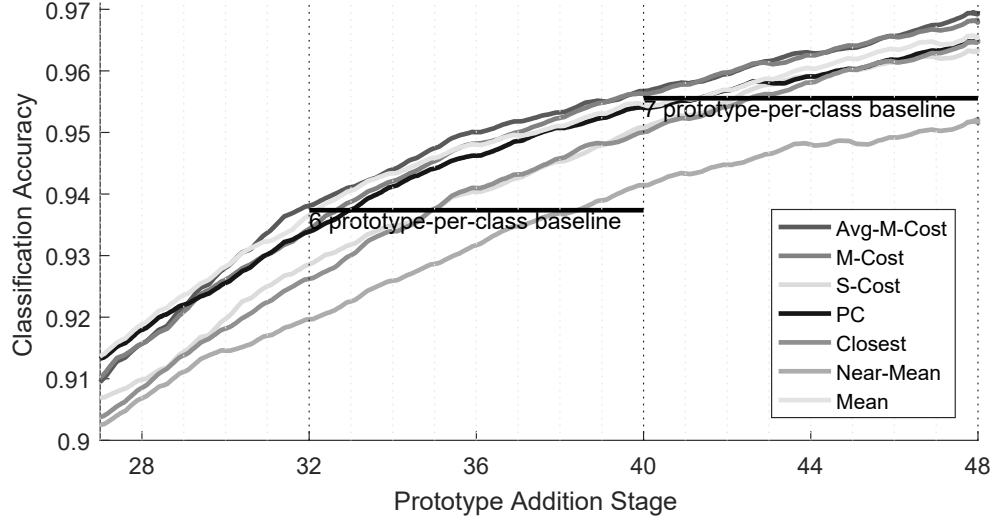
of prototypes that the configuration possessed at the time the data was taken. This number of prototypes then becomes an explicit axis. This new view of the data is seen in Fig. A.16.

A.3.6 Margin Analysis.

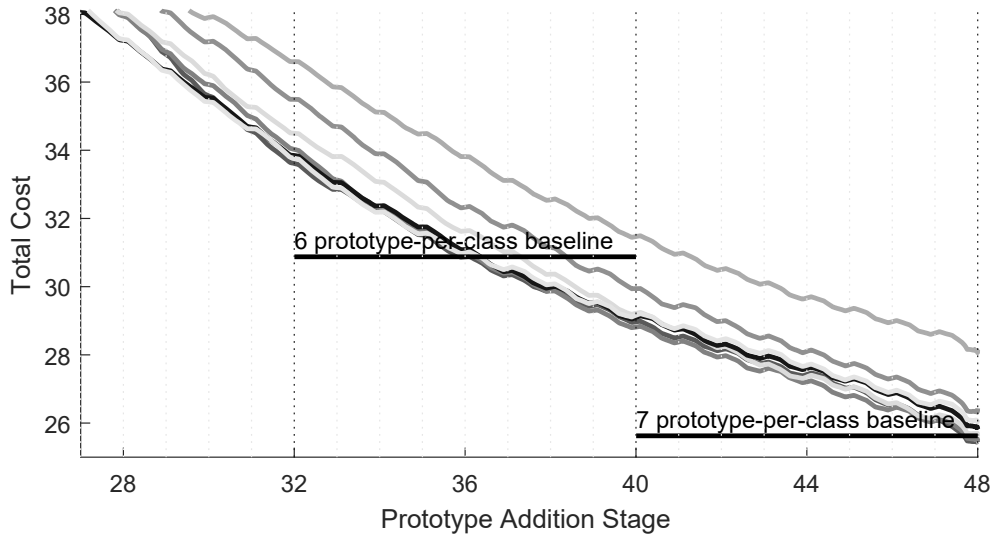
After examining how total cost and classification accuracy change as a function of the number of prototypes in Section A.3.4 and Section A.3.5, the relationship between margin error and the number of prototypes is considered. First, the margin error is calculated for operating points at each prototype addition stage. In Fig. A.18 a grid of plots is seen, two with the same color: E_{CGNT} (top) and E_{HSV} (bottom) versus θ_{cp} for each NSM.

A.3.7 Relative Comparison.

Accuracy Generalization surfaces are provided in Fig. A.22 to allow comparison between NSM methods over the entire performance operating range.

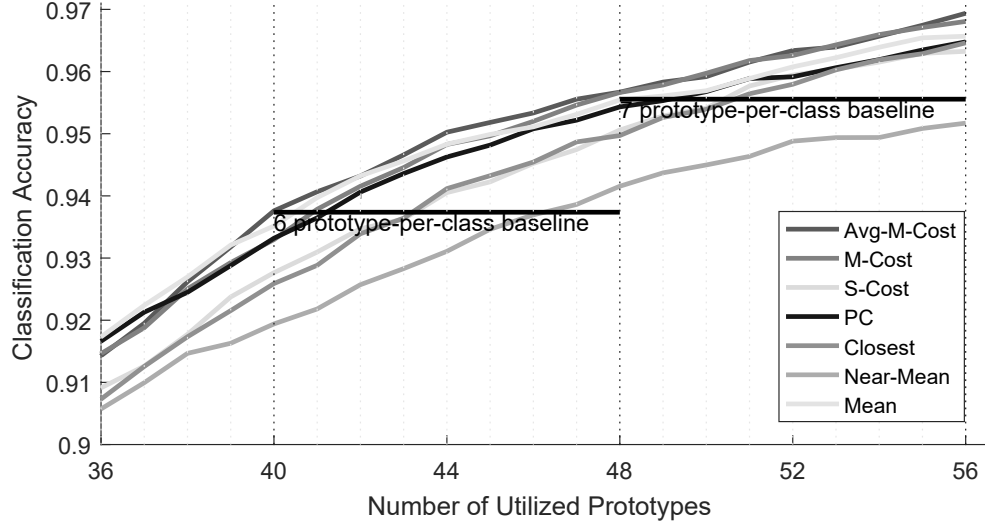


(a)

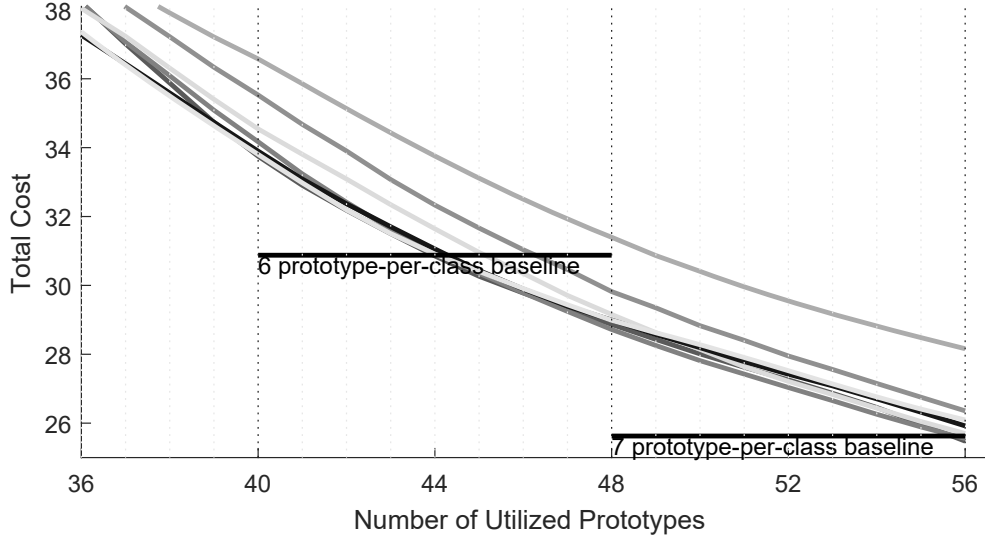


(b)

Figure A.15: Classification accuracy **(a)** and Total Cost **(b)** versus prototype addition stage for seven prototype addition methods over the MICE data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6. Three GRLVQI “Baselines” trained with a fixed number of prototypes are included for reference. Plots are the average of 150 validation result curves (30 complete runs each consisting of $K = 5$ folds).



(a)



(b)

Figure A.16: Classification accuracy **(a)** and Total Cost **(b)** versus utilized number of prototypes for seven prototype addition methods over the Mice data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6 Two GRLVQI baselines trained with a fixed number of prototypes are included for reference. Validation accuracies from the 150 curves (30 complete runs each consisting of $K = 5$ folds) are pooled for each utilized prototype value then averaged (roughly 18750 points per prototype value).

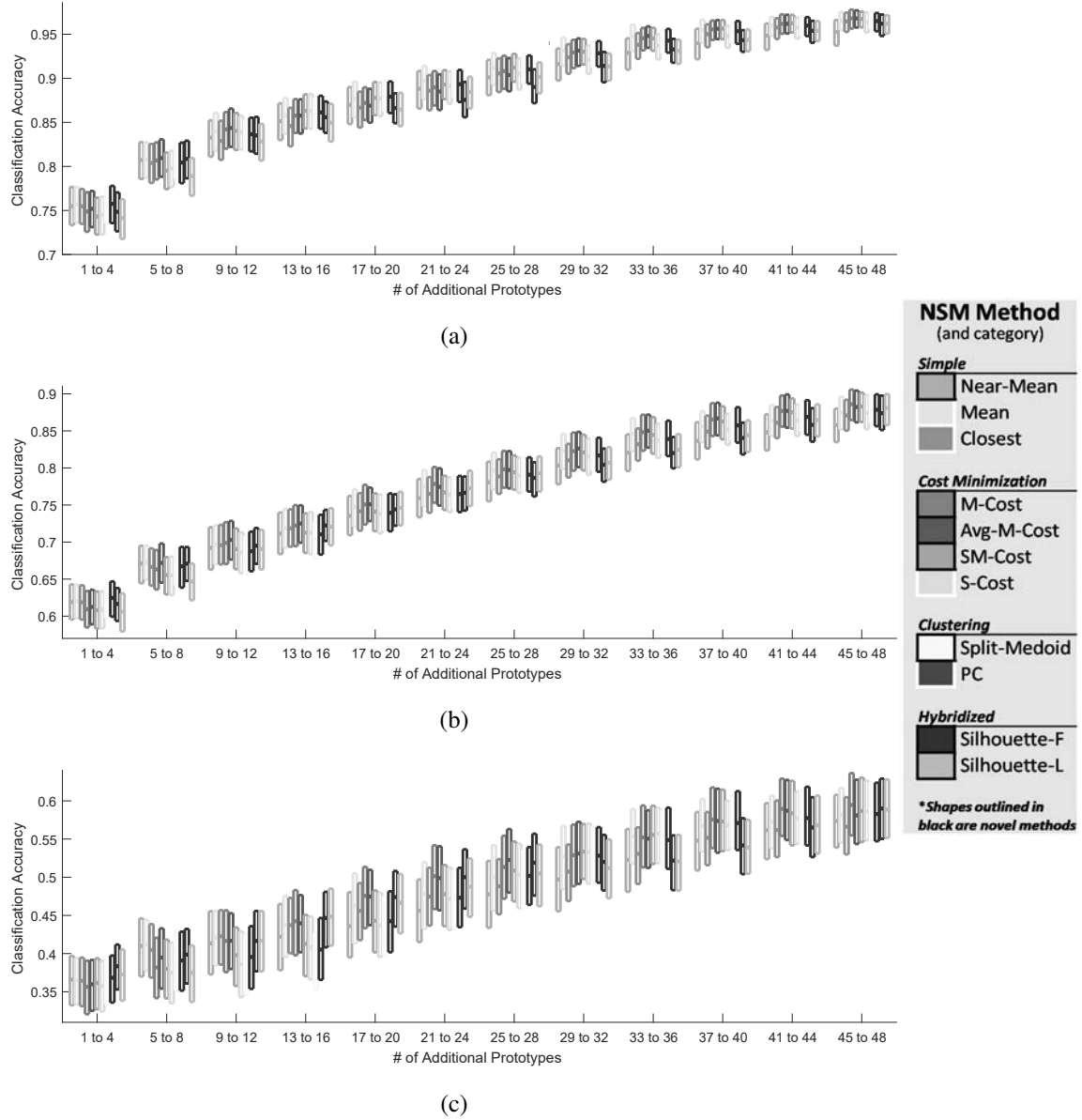


Figure A.17: For MICE-GRLVQI: NSM classification accuracies are compared against number of additional prototypes for a margin test point, θ_{cp} . Quartiles within each NSM method (in 4 prototype intervals starting with 8) are shown as the top, middle, and bottom of colored vertical boxes (75%-ile, 50%-ile, and 25%-ile). Plots (a), (b), and (c) correspond to θ_{cp} values of 0, $2.1e - 5$, and $5.3e - 5$.

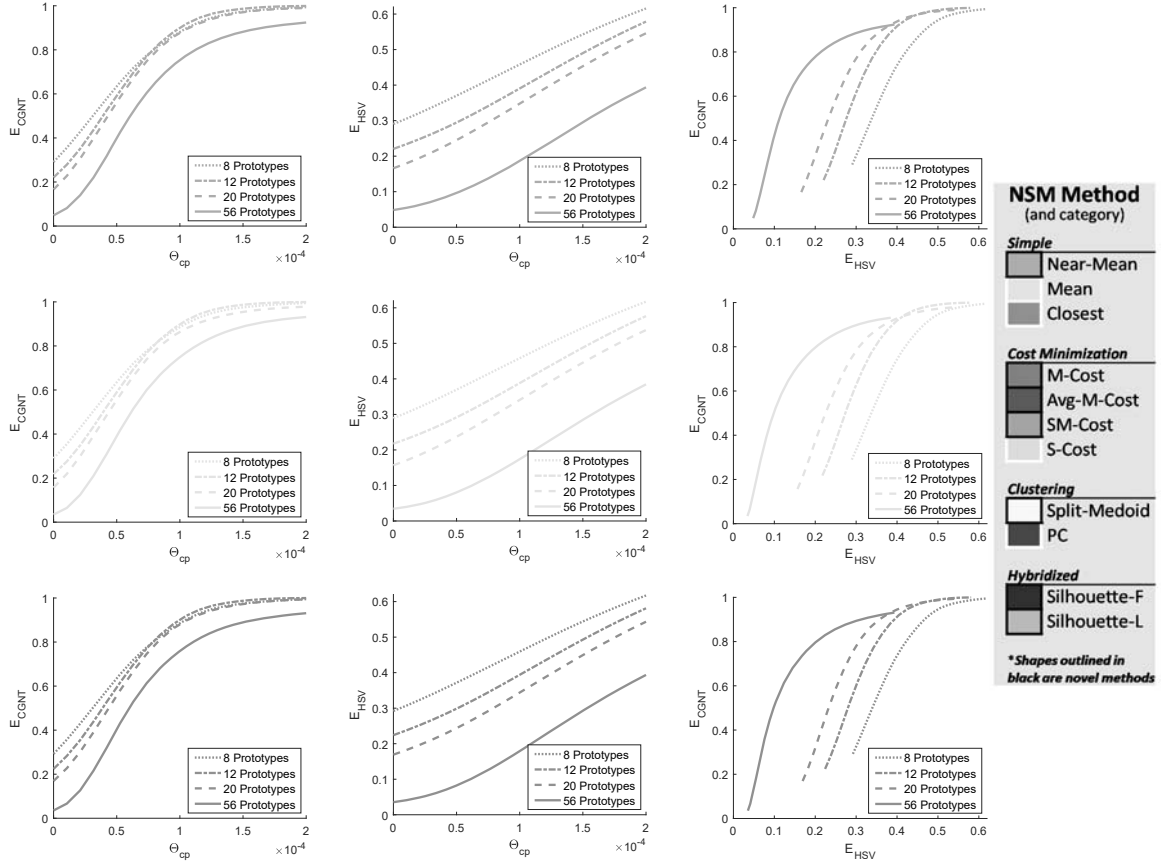


Figure A.18: Simple NSM methods with 35, 40, 45, and 105 prototypes display **(left)** CGNT Margin Error as a function of θ_{cp} , **(middle)** HSV Margin Error as a function of θ_{cp} , and **(right)** HSV Margin Error versus CGNT Margin Error.

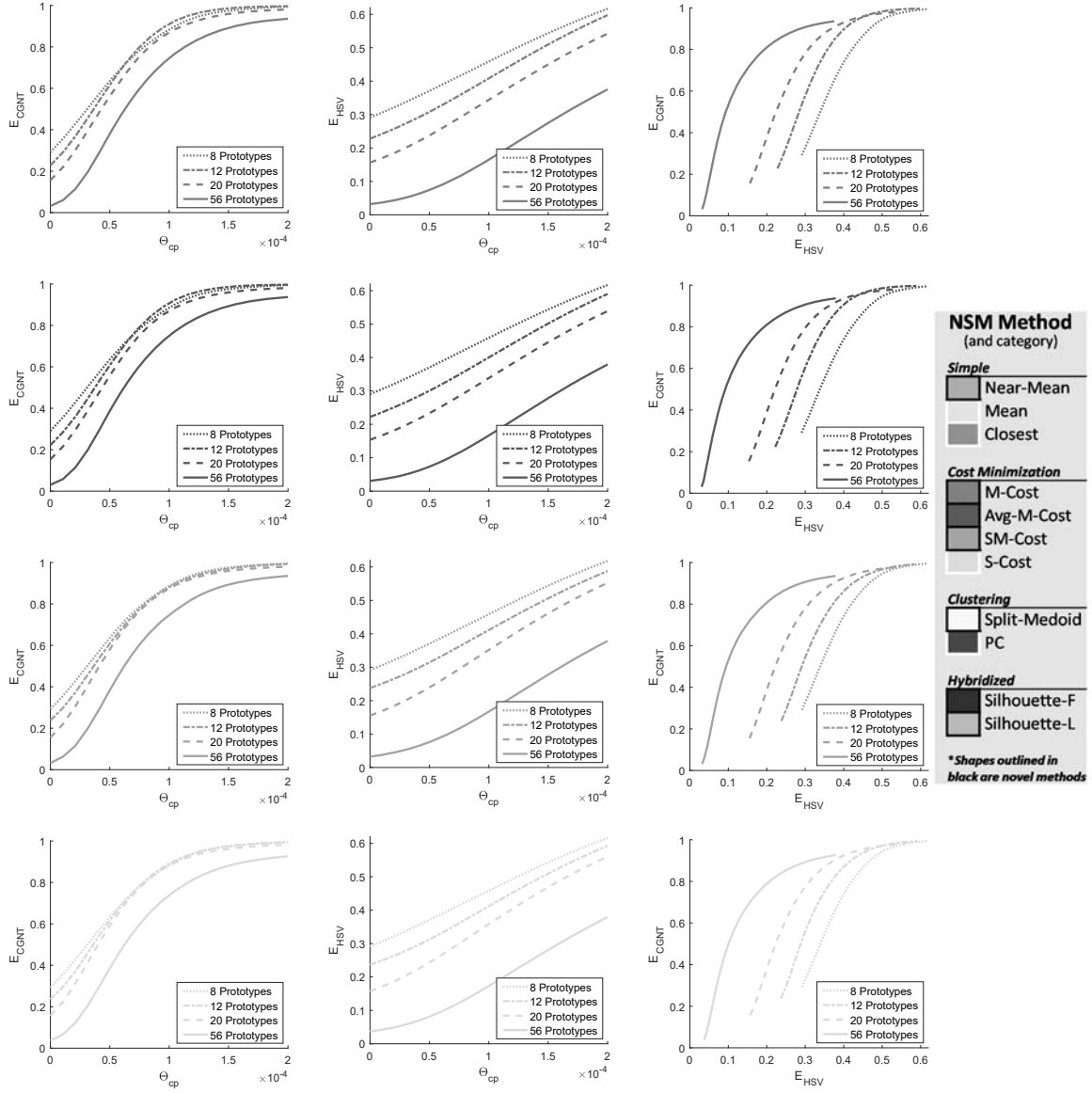


Figure A.19: Cost Minimizing NSM methods with 35, 40, 45, and 105 prototypes display **(left)** CGNT Margin Error as a function of θ_{cp} , **(middle)** HSV Margin Error as a function of θ_{cp} , and **(right)** HSV Margin Error versus CGNT Margin Error.

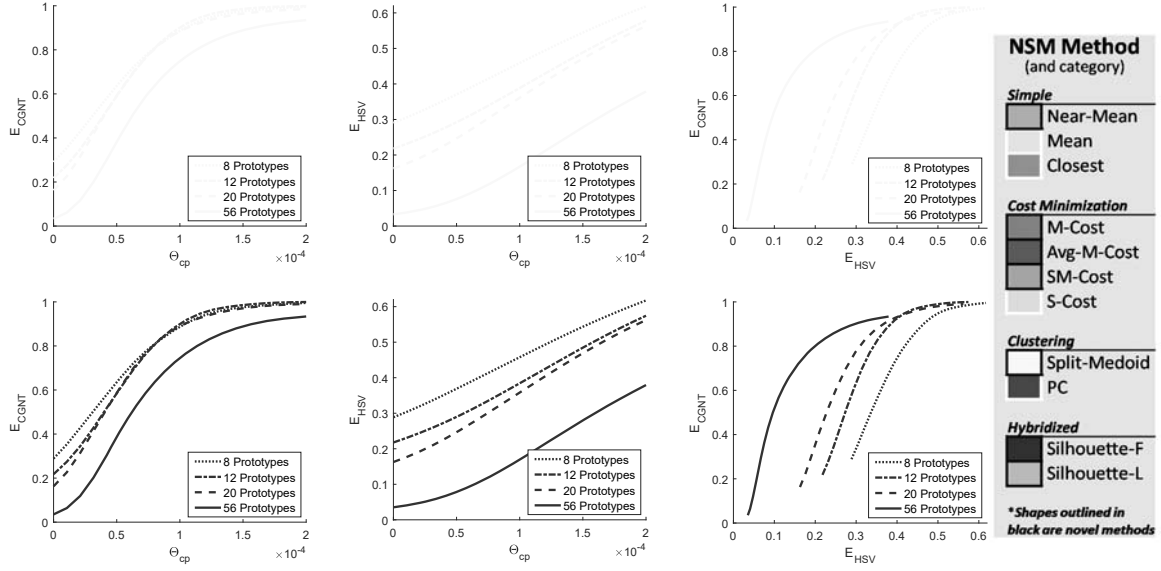


Figure A.20: Clustering NSM methods with 35, 40, 45, and 105 prototypes display **(left)** CGNT Margin Error as a function of θ_{cp} , **(middle)** HSV Margin Error as a function of θ_{cp} , and **(right)** HSV Margin Error versus CGNT Margin Error.

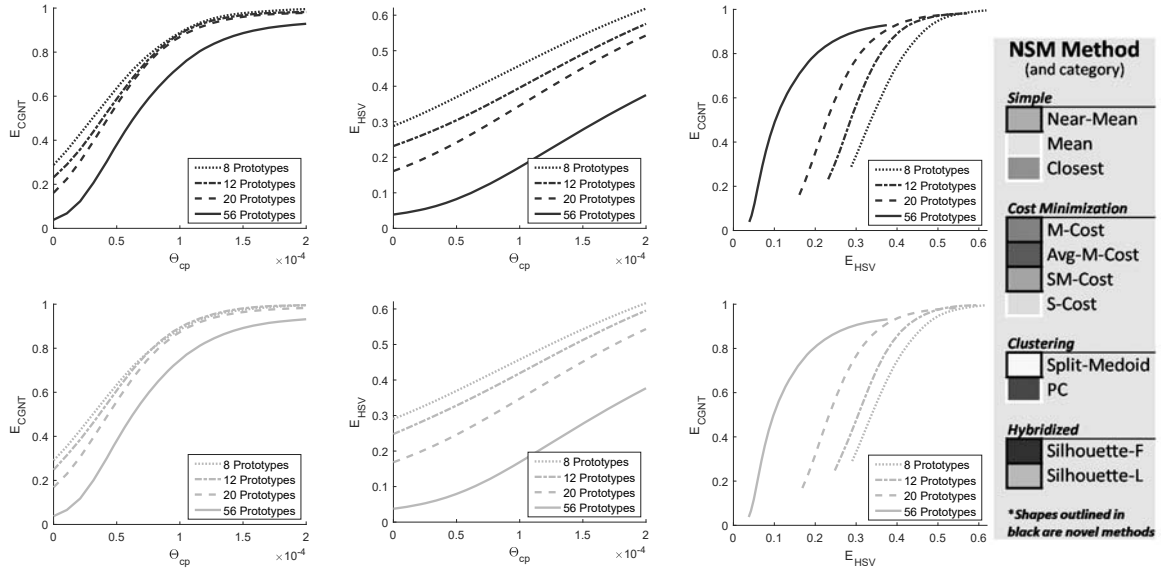


Figure A.21: Hybridized NSM methods with 35, 40, 45, and 105 prototypes display **(left)** CGNT Margin Error as a function of θ_{cp} , **(middle)** HSV Margin Error as a function of θ_{cp} , and **(right)** HSV Margin Error versus CGNT Margin Error.

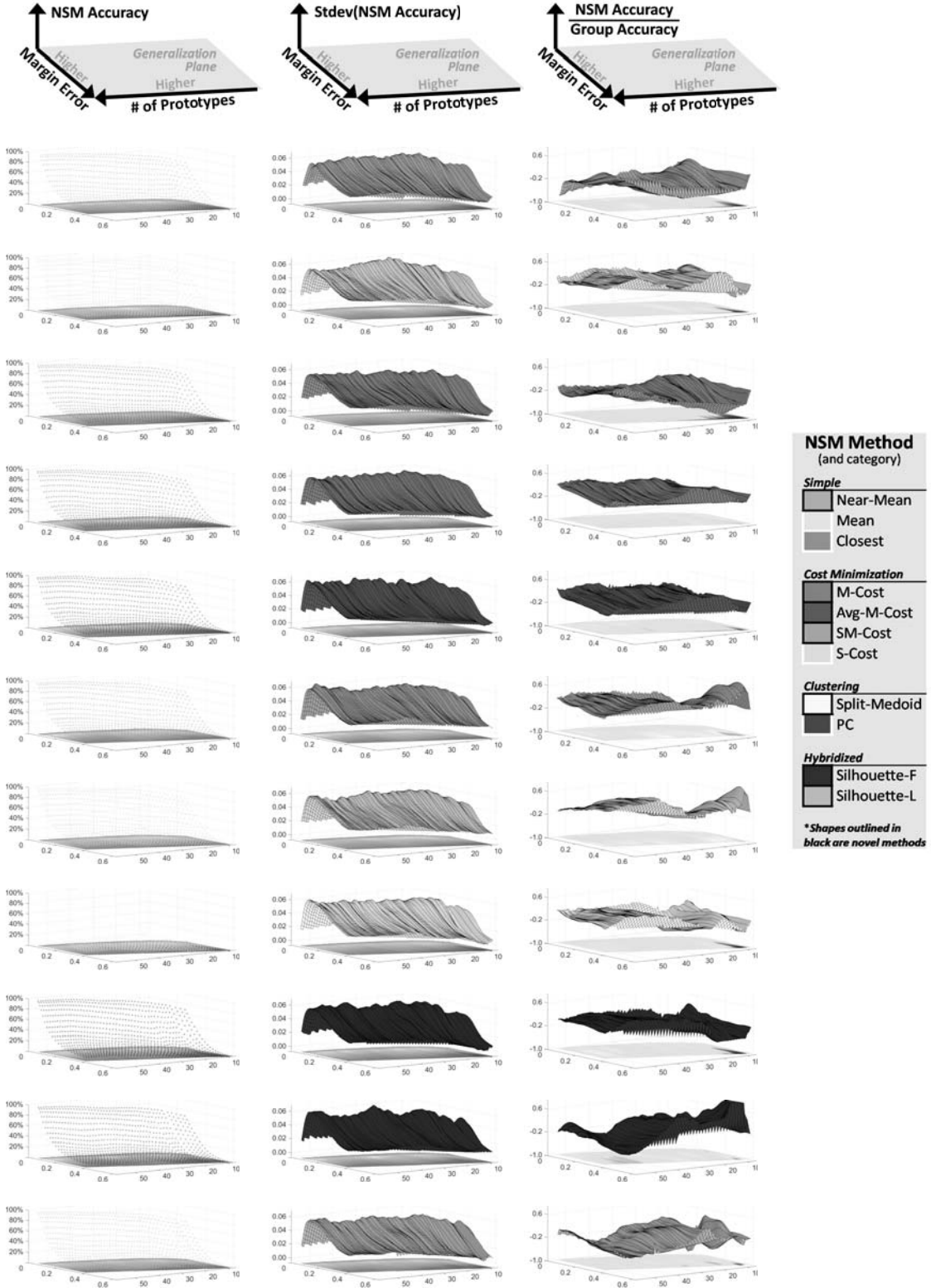


Figure A.22: Comparison of GRLVQI NSM methods for MICE dataset (top row shows axes for each column: classification accuracy, standard deviation, and relative accuracy (l-r).

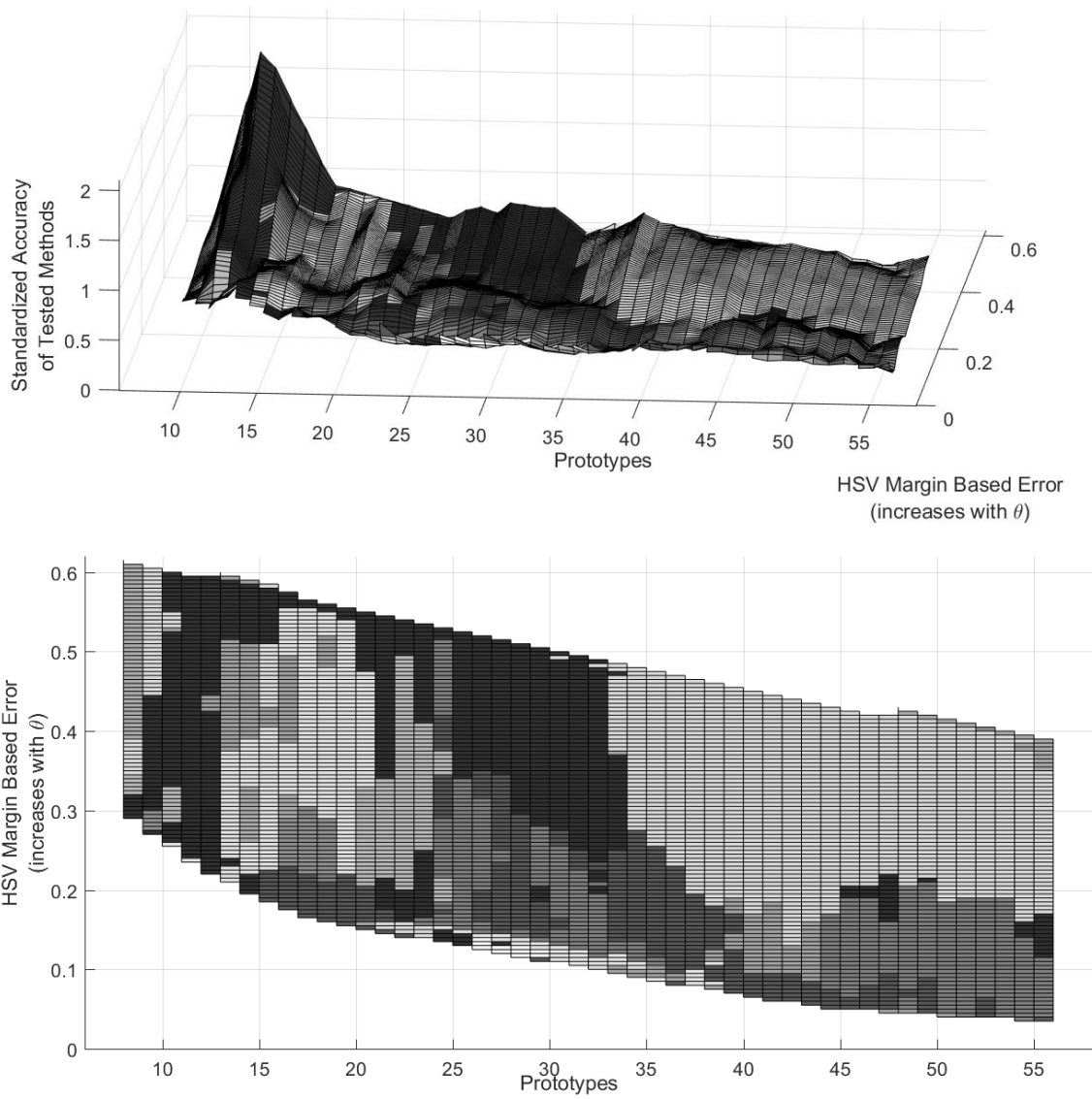


Figure A.23: The surface plot (top) and flattened surface plot (bottom) indicate regions of comparative advantage for eleven NSM methods.

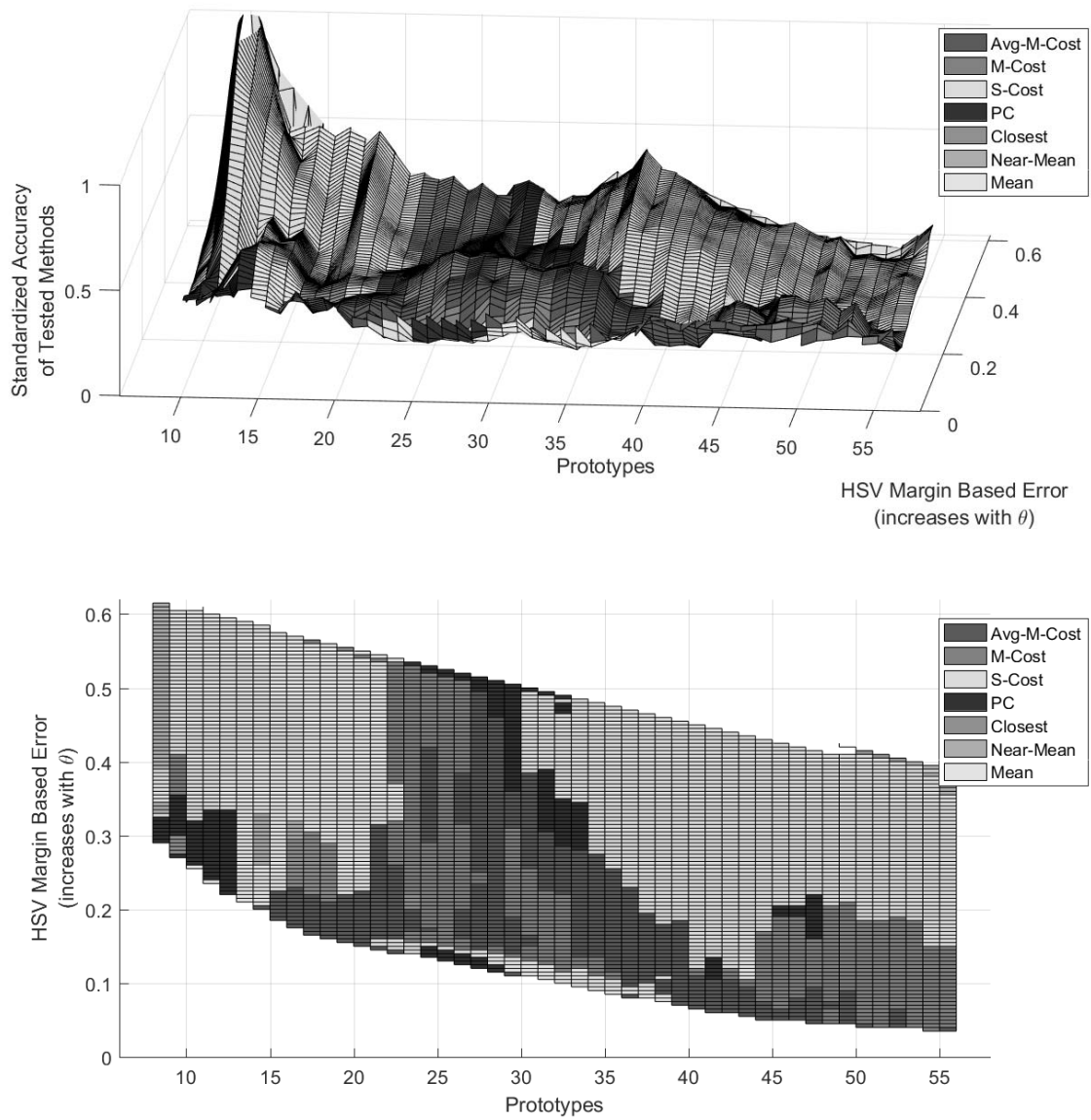


Figure A.24: The surface plot (top) and flattened surface plot (bottom) indicate regions of comparative advantage for NSM methods: Avg-M-Cost, M-Cost, S-Cost, PC, Closest, Near-Mean, and Mean.

A.4 ZWAVE GRLVQI Results and Discussion

This Chapter provides supplemental analysis plots for the ZWAVE data set. Section A.4.1 provides experimental parameters. Section A.4.2 shows feature correlation and covariance. Section A.4.3 shows averaged performance (accuracy and cost) as a function of prototype addition stage. Section A.4.4 shows baseline and converged performance from the perspective of individual operating points. Section A.4.5 shows prototype utilization. Section A.4.6 shows margin analysis and plots of E_{HSV} and E_{CGNT} error. Section A.4.7 concludes this chapter of ZWAVE analysis results with a accuracy generalization surface plot.

A.4.1 Parameters and Data.

The learn-rates, training parameters and pertinent notes for dynamic prototype experiments with the GRLVQI algorithm and ZWAVE data set are provided in Table. A.8.

Table A.8: ZWAVE GRLVQI parameters.

	Training Epochs	Learn Parameters		Conscience Parameters		Data Set:	ZWAVE
		α^w	α^l	γ	β	File Prefix:	91079
ZWAVE GRLVQI	$1 \leq \text{Ep} \leq 125$	2.50e-2	5.00e-3	2.50	0.35	Data Collected:	7-1-2017
	$126 \leq \text{Ep} \leq 250$	1.30e-2	2.59e-3	2.10	0.29	Data Processed:	7-4-2017
	$251 \leq \text{Ep} \leq 375$	6.71e-3	1.34e-3	1.77	0.24	Prototype Initialization:	1/class at random 0.4,0.6
	$376 \leq \text{Ep} \leq 500$	3.47e-3	6.95e-4	1.49	0.20	Prototype Addition Stages:	27
	$501 \leq \text{Ep} \leq 625$	1.80e-3	3.60e-4	1.25	0.17	Slope of sigmoid:	1
	$626 \leq \text{Ep} \leq 750$	9.32e-4	1.86e-4	1.05	0.14	Distance measure:	Relevance Weighted Euclidean
	$751 \leq \text{Ep} \leq 875$	4.83e-4	9.65e-5	0.90	0.12	Thresholding:	None
	$876 \leq \text{Ep} \leq 1000$	2.50e-4	5.00e-5	0.75	0.10	# of Cross-Validation folds:	5

A.4.2 Feature Covariance and Correlation.

In keeping with the process for our other data sets, the covariance and correlation of features is observed within each class. In terms of covariance and correlation, all three classes offer similar results. Due to relative magnitude, the highest joint variability occurs within the first and last thirds of the signal, and is concentrated between the kurtosis terms

as seen in Fig. A.25. The highest correlation between features occurs within the middle of the signal between variance terms and is seen in Fig. A.26.

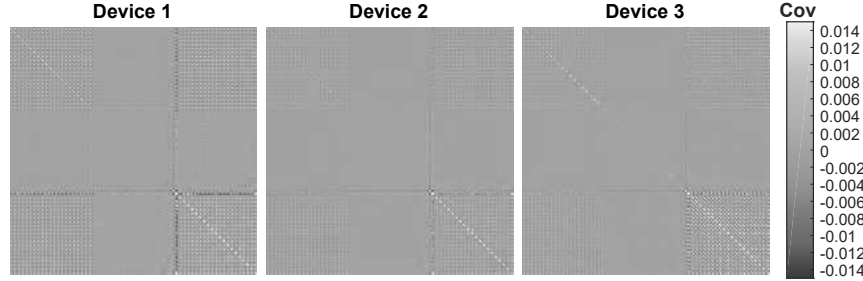


Figure A.25: The covariance is shown between features of the ZWAVE data set within each labeled class over a 189×189 covariance matrix.

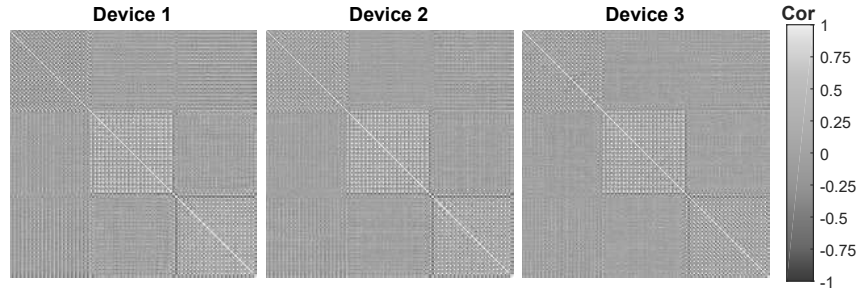


Figure A.26: The correlation between features of the ZWAVE data set are shown within each labeled class over a 189×189 correlation matrix.

A.4.3 Averaged Performance as a Function of Training.

Training and validation is conducted with 11 dynamic algorithms representing 4 categories of approaches: simple (computationally efficient), cost-minimizing, clustering, and hybrid approaches. A fully initialized 30-prototype non-dynamic GRLVQI baseline is used for comparison. Figure A.27 reports validation accuracy and cost as a function

of training time. In lieu of epochs, the x-axis is marked and segmented by prototype addition stage. This illustration of training shows the growth in classification accuracy and reduction in cost as NSM methods have the opportunity to add additional prototypes to their networks. For convenience in plotting, the initial transient regions from each prototype addition stage are removed.

A.4.4 Baseline and Converged Model Performance.

The performance of each NSM is evaluated with respect to a non-dynamic baseline. First, (in Table. A.9) the accuracy of NSM methods is set to match or slightly exceed that of the non-dynamic method to observe the reduction in prototype resources. For the ZWAVE data set, virtually all pairwise trials meet the non-dynamic baseline accuracy 98% – 99% for most NSM methods, 91% for Silhouette-F). Instead of merely achieving a statistically similar classification accuracy, each of these methods is shown to offer a statistically significant improvement or around 0.4%. Significantly each NSM method reduces the number of prototypes by 23.5 to 25.5, representing a (statistically significant) 78% to 85% savings.

Second, NSM methods are allowed to achieve their best or converged model performance as shown in Table A.10. The resulting performance continues to reflect statistically significant improvements in classification accuracy and reductions in prototypes for all NSM methods when compared to the non-dynamic baseline. The methods commonly achieve around 0.7% improvements in classification accuracy. While improvements between classification accuracy are within similar confidence intervals, there are several groupings of NSM methods that can be differentiated based on the confidence intervals of mean reduction in prototypes. S-Cost and Silhouette-F have a mean reduction of prototypes that is 99% likely to fall within the ranges of 15.08 - 16.71 and 14.78-16.73 respectively outside the upper bands of the other NSM methods.

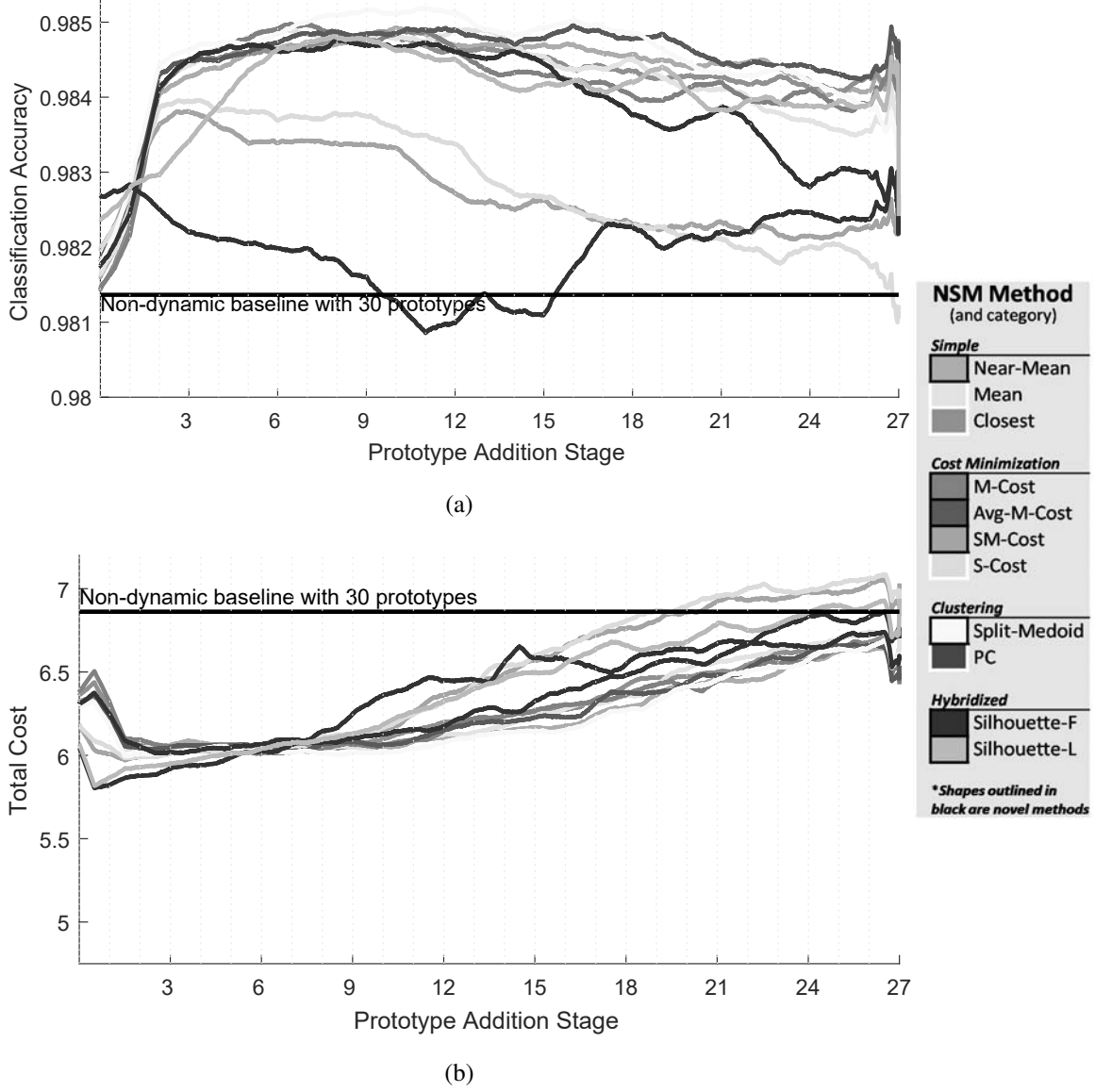


Figure A.27: Classification accuracy **(a)** and Total Cost **(b)** versus prototype addition stage for seven prototype addition methods over the LCVF data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6. A GRLVQI ‘baseline trained with 30 prototypes is included for reference. Plots are the average of 150 validation result curves (30 complete runs each consisting of $K = 3$ folds).

Table A.9: NSM performance for 150 independent trials is shown when the non-dynamic baseline is chosen as an accuracy performance point. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and prototype utilization (when compared to baseline performance). Statistically significant improvements are shown in bold, statistically similar results are shown in italics.

	Match Baseline Performance									
NSM Method	Met	Accuracy					Prototypes			
		μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	
Near-Mean	148	98.57	0.80	0.40	(0.34 – 1.84)	4.85	4.37	25.15	(24.31 – 25.15)	
Mean	149	98.56	0.80	0.40	(0.33 – 1.84)	4.52	3.59	25.48	(24.79 – 25.48)	
Closest	149	98.56	0.80	0.39	(0.33 – 1.84)	4.67	4.13	25.33	(24.53 – 25.33)	
M-Cost	147	98.54	0.80	0.38	(0.31 – 1.84)	4.59	4.42	25.41	(24.56 – 25.41)	
Avg-M-Cost	148	98.56	0.80	0.40	(0.33 – 1.84)	4.81	4.27	25.19	(24.37 – 25.19)	
SM-Cost	148	98.53	0.80	0.37	(0.29 – 1.84)	4.63	4.19	25.37	(24.57 – 25.37)	
S-Cost	147	98.55	0.81	0.38	(0.31 – 1.84)	5.02	4.99	24.98	(24.02 – 24.98)	
Split-Medoid	149	98.56	0.80	0.40	(0.33 – 1.84)	4.58	4.23	25.42	(24.61 – 25.42)	
PC	149	98.55	0.80	0.38	(0.31 – 1.84)	4.99	4.89	25.01	(24.07 – 25.01)	
Silhouette-F	137	98.51	0.83	0.34	(0.26 – 1.84)	6.54	8.33	23.46	(21.86 – 23.46)	
Silhouette-L	148	98.55	0.81	0.38	(0.31 – 1.84)	4.60	4.16	25.40	(24.60 – 25.40)	
Baseline(10x3p)	150	98.16	0.00	-	-	30	0.00	-	-	

A.4.5 Performance with respect to Prototype Utilization.

Valuable insight is gained from a prototype utilization perspective for the ZWAVE data in Fig. A.28 as each NSM’s capability to make use out of an additional prototype is seen. The Avg-M-Cost, M-Cost, and Silhouette-L methods generally trend higher than the other methods.

A.4.6 Margin Analysis.

After examining how total cost and classification accuracy change as a function of the number of prototypes in Section A.4.4 and Section A.4.5, the relationship between margin error and the number of prototypes is considered.

Table A.10: Converged NSM model performance is shown for 150 independent trials of the ZWAVE data set. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and corresponding prototype utilization, when compared to baseline performance. Statistically significant improvements are shown in bold, statistically similar results are shown in italics.

NSM Method	Converged Classifier Performance							
	Accuracy				Prototypes			
	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI
Near-Mean	98.88	0.70	0.71	(0.63 – 1.84)	16.87	8.65	13.13	(11.47 – 13.13)
Mean	98.88	0.70	0.71	(0.63 – 1.84)	16.37	8.70	13.63	(11.96 – 13.63)
Closest	98.87	0.71	0.70	(0.62 – 1.84)	17.27	8.89	12.73	(11.03 – 12.73)
M-Cost	98.86	0.71	0.69	(0.61 – 1.84)	16.63	8.96	13.37	(11.65 – 13.37)
Avg-M-Cost	98.87	0.70	0.71	(0.62 – 1.84)	16.23	8.50	13.77	(12.13 – 13.77)
SM-Cost	98.85	0.70	0.68	(0.60 – 1.84)	15.36	8.71	14.64	(12.97 – 14.64)
S-Cost	98.84	0.71	0.68	(0.60 – 1.84)	13.29	8.49	16.71	(15.08 – 16.71)
Split-Medoid	98.89	0.69	0.73	(0.65 – 1.84)	16.35	8.79	13.65	(11.97 – 13.65)
PC	98.88	0.69	0.71	(0.63 – 1.84)	16.09	9.03	13.91	(12.17 – 13.91)
Silhouette-F	98.71	0.80	0.55	(0.46 – 1.84)	13.27	10.16	16.73	(14.78 – 16.73)
Silhouette-L	98.90	0.70	0.74	(0.66 – 1.84)	16.19	8.45	13.81	(12.18 – 13.81)
Baseline(10x3p)	98.16	0.82	-	-	30.00	0.00	-	-

A.4.7 Relative Comparison.

The performance of GRLVQI based NSM methods are shown for the ZWAVE dataset in Fig. A.30. Three columns of plots are compared over the axes of margin error and the number of prototypes, representing empirical and theoretical components of the generalization error as described in Section 3.3. The z-axis depicts **(left)** classification accuracy, **(middle)** standard deviation in classification accuracy, and **(right)** classification accuracy, standardized over group the as $\sim N(0, 1)$.

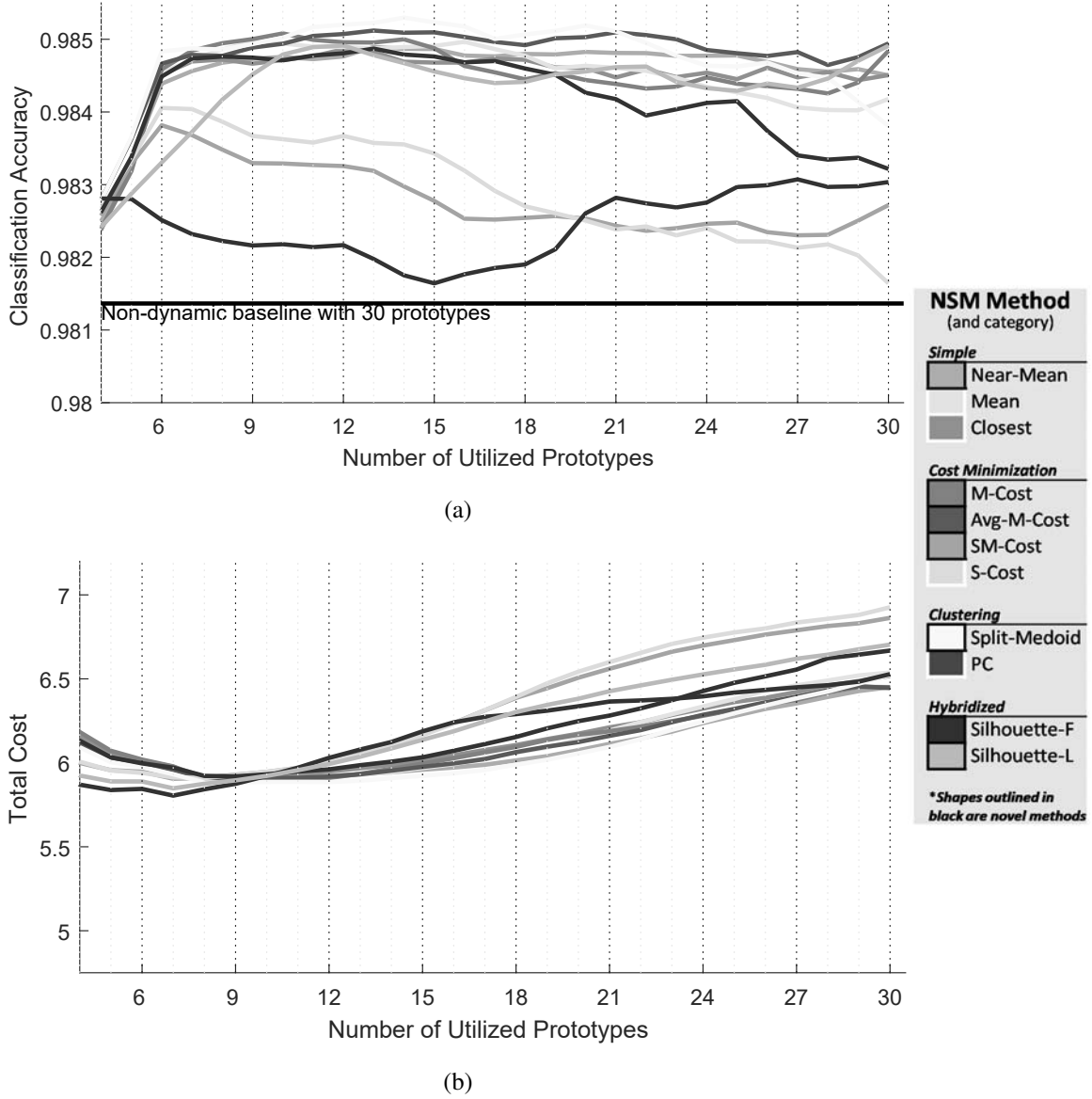


Figure A.28: Classification accuracy **(a)** and Total Cost **(b)** versus utilized number of prototypes for eleven prototype addition methods over the ZWAVE data set with a GRLVQI algorithm. One GRLVQI baseline trained with 30 prototypes is included for reference. Validation accuracies from the 150 curves (30 complete runs each consisting of $K = 5$ folds) are pooled for each utilized prototype value then averaged.

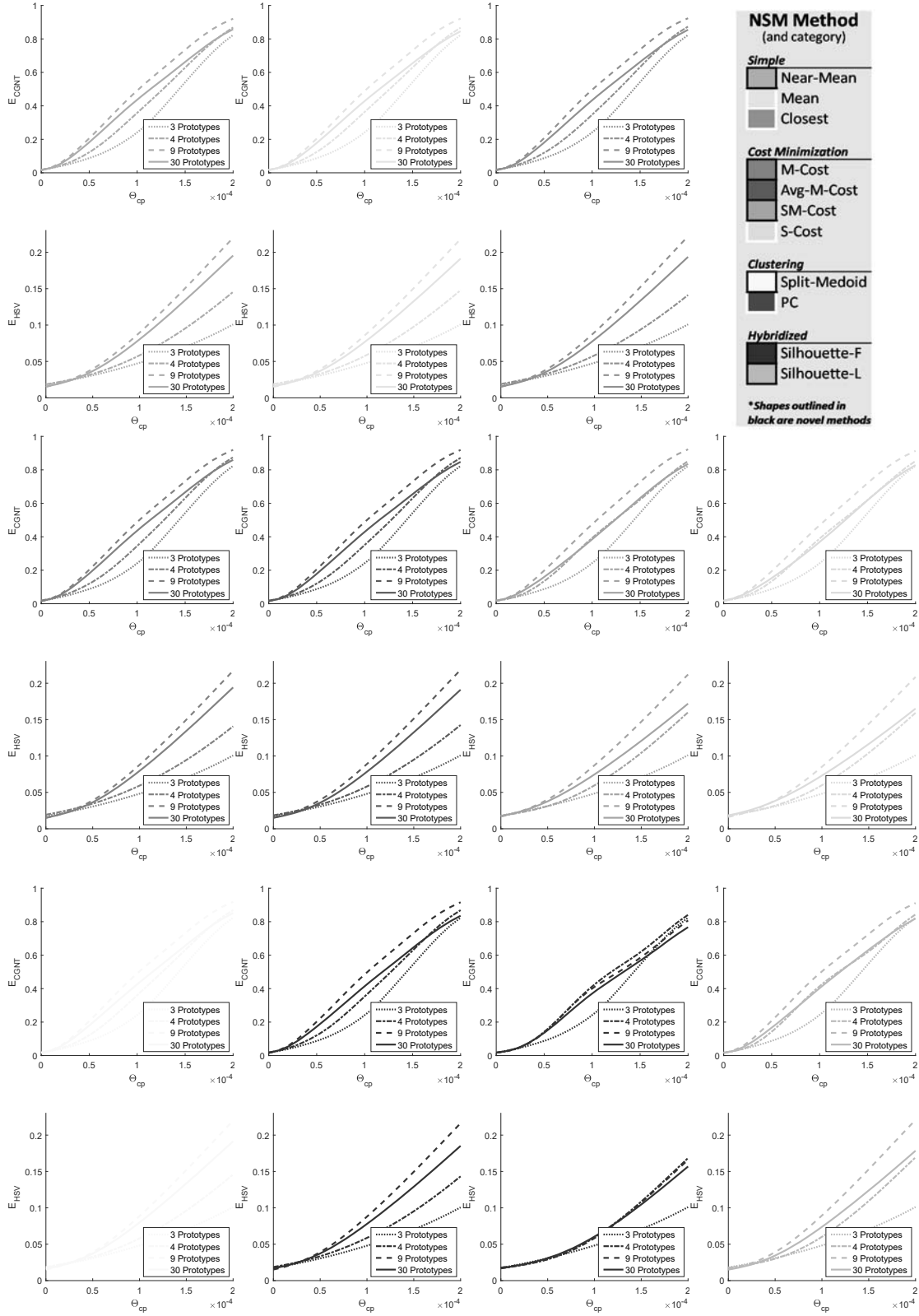


Figure A.29: GRLVQI NSM methods for ZWAVE with 3, 4, 9, and 30 prototypes display margin error as a function of θ_{cp} : CGNT(top) and HSV(bottom).

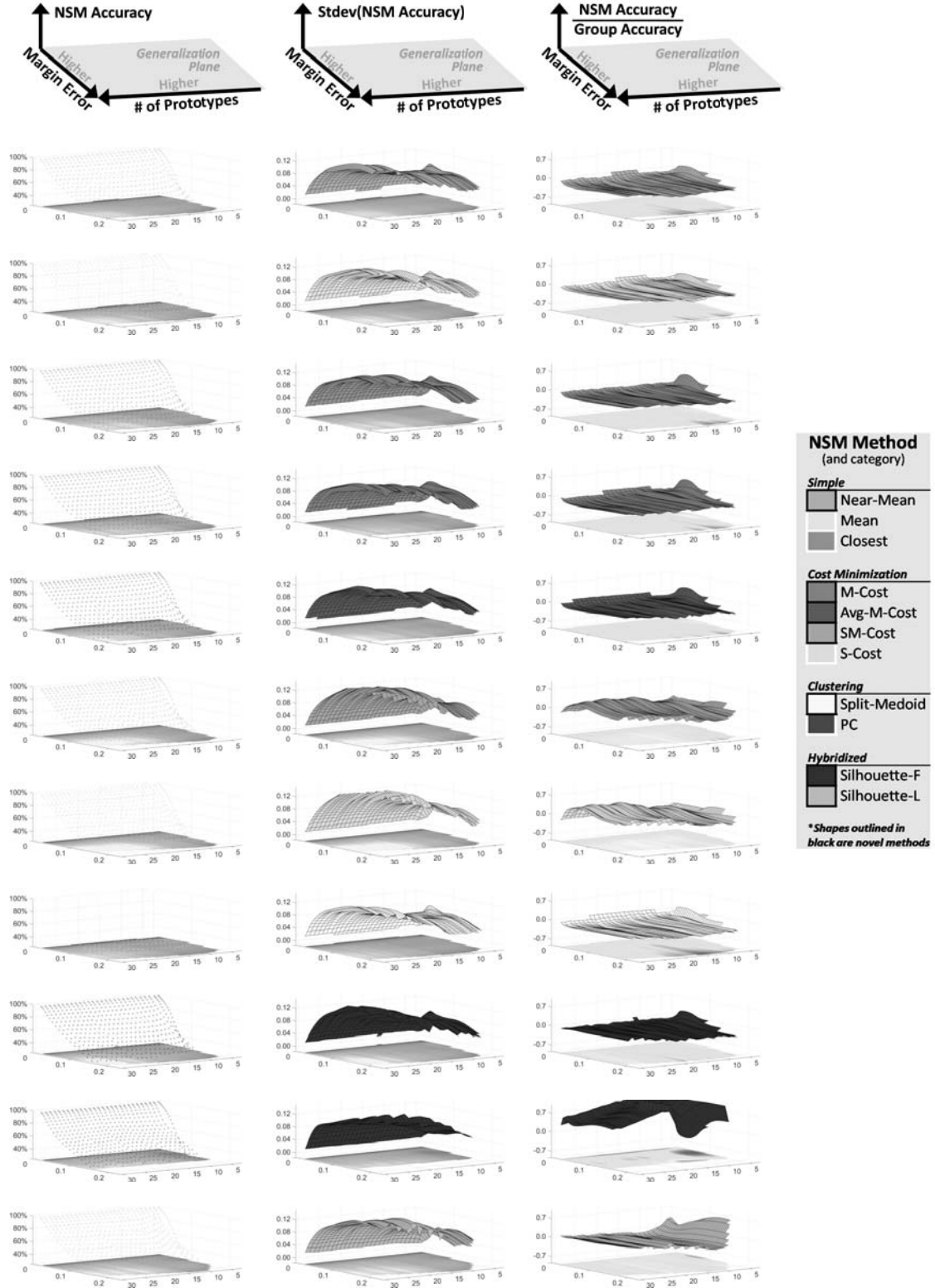


Figure A.30: Comparison of GRLVQI NSM methods for ZWAVE dataset. Top row shows axes for each column: classification accuracy, standard deviation, and relative accuracy (l-r).

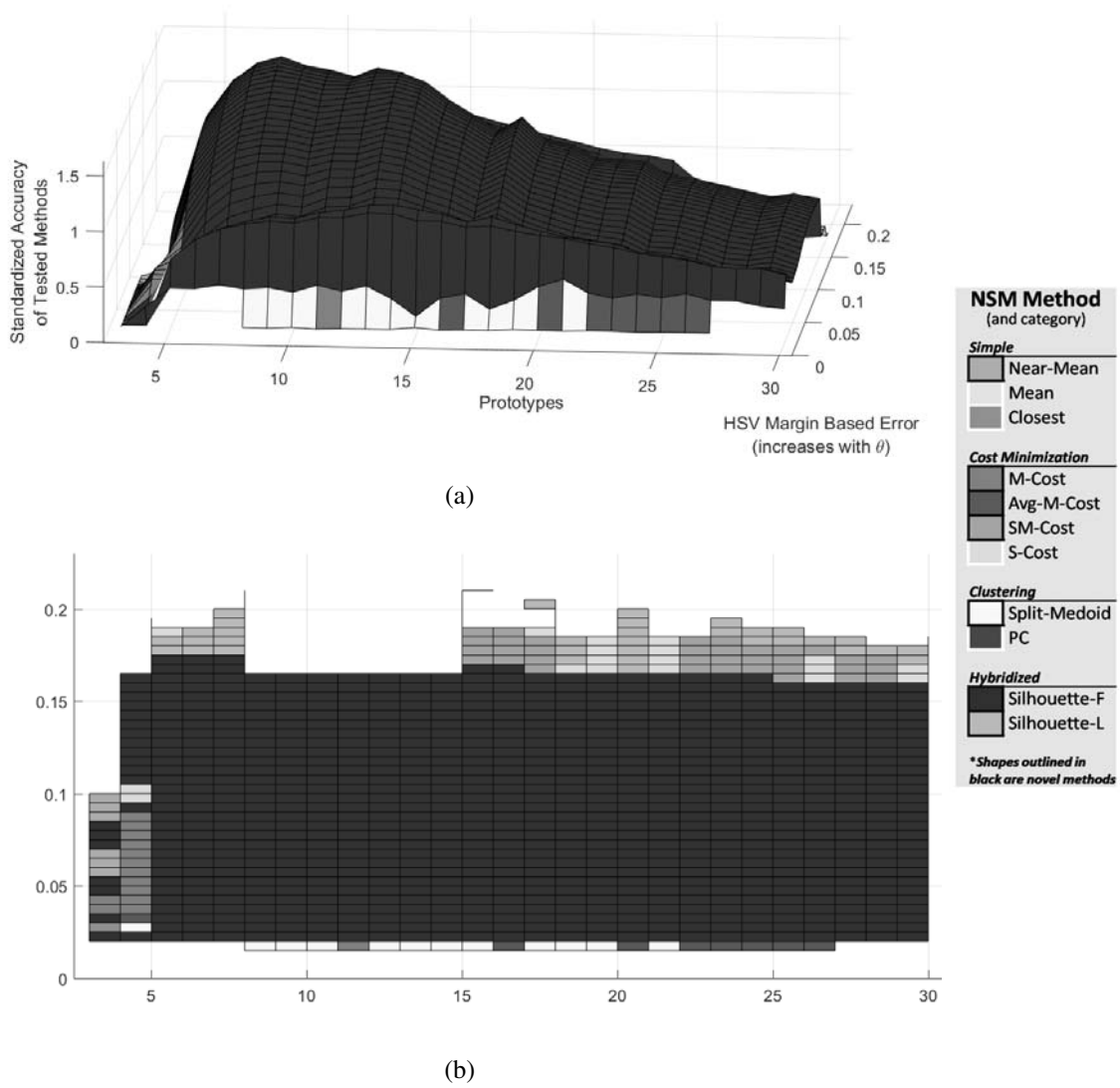


Figure A.31: (a) This composite of Fig. A.30 (right-column) selects the best performing NSM method over each grid point. (b) This graph is simplified by flattening the z-axis. This flattened view indicates NSM methods with the highest average classification accuracy over a range of utilized prototypes and margin error values.

A.5 LCVF GRLVQI Results and Discussion

A.5.1 Parameters and Data.

The learn-rates, training parameters and pertinent notes for dynamic prototype experiments with the GRLVQI algorithm and LCVF dataset are provided in Table. A.11.

Table A.11: LCVF GRLVQI parameters.

	Training Epochs	Learn Parameters		Conscience Parameters		Data Set:	LCVF
		α^w	α^l	γ	β	File Prefix:	769, 6699
LCVF GRLVQI	$1 \leq \text{Ep} \leq 50$	2.50e-2	5.00e-3	2.50	0.35	Data Collected:	
	$51 \leq \text{Ep} \leq 100$	8.21e-3	1.87e-3	2.10	0.29	Data Processed:	
	$101 \leq \text{Ep} \leq 150$	2.68e-3	6.97e-4	1.77	0.24	Prototype Initialization:	1/class at random 0.4,0.6
	$151 \leq \text{Ep} \leq 200$	8.77e-4	2.59e-4	1.49	0.20	Prototype Addition Stages:	70
	$201 \leq \text{Ep} \leq 250$	2.86e-4	9.67e-5	1.25	0.17	Slope of sigmoid:	1
	$251 \leq \text{Ep} \leq 300$	9.37e-5	3.60e-5	1.05	0.14	Distance measure:	Relevance Weighted Euclidean
	$301 \leq \text{Ep} \leq 350$	3.06e-5	1.34e-5	0.90	0.12	Thresholding:	None
	$351 \leq \text{Ep} \leq 400$	1.00e-5	5.00e-6	0.75	0.10	# of Cross-Validation folds:	3

A.5.2 Feature Covariance and Correlation.

After normalizing the LCVF data set with its ℓ_2 -norm to compensate for the effect of shadowing due to sensor geometry, the feature covariance and correlation are calculated as shown in Fig. A.32 and Fig. A.33 respectively. Classes 1, 5, 6, 7, 12, and 24 show the highest joint variability between spectra. Classes 1-5, 12, 19, 20, and 29 show the strongest correlations between features. These correlations however are anticipated as spectral responses are structural in nature. Additionally, atmospheric effects further spread material responses across neighboring wavelength bins.

These correlations are accepted as another aspect of the data set. The LVQ classification process will likely not be strongly impacted if these features are accepted, since these features are utilized to make final predictions instead of establish individual causal links.

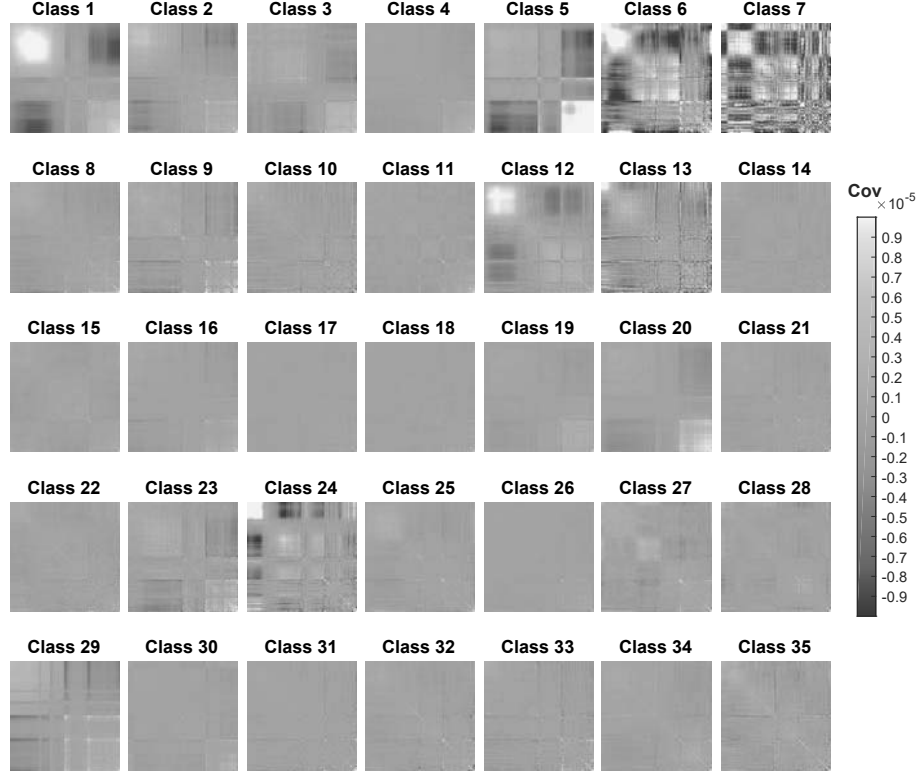


Figure A.32: We show the covariance between features of the LCVF data set within each labeled class over a 194×194 covariance matrix.

A.5.3 Averaged Performance as a Function of Training.

Training and validation is conducted with 11 dynamic algorithms representing 4 categories of approaches: simple (computationally efficient), cost-minimizing, clustering, and hybrid approaches. A fully initialized 70-prototype and 105-prototype non-dynamic GRLVQI Baseline are used for comparison. Figure A.34 reports validation accuracy and cost as a function of training time. In lieu of epochs, the x-axis is marked and segmented by prototype addition stage. This illustration of training shows the growth in classification accuracy and reduction in cost as NSM methods have the opportunity to add additional prototypes to their networks. For convenience in plotting, the initial transient regions from each prototype addition stage are removed.

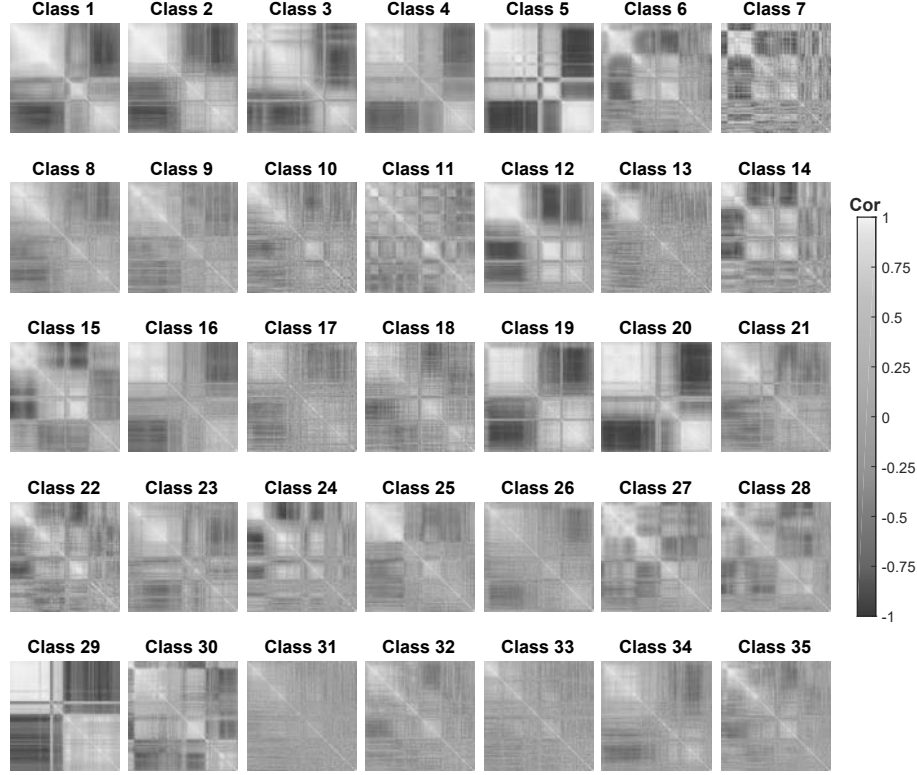


Figure A.33: The correlation between features of the LCVF data set are shown within each labeled class over a 194×194 correlation matrix.

Throughout the course of training, the relative performance between NSM methods vary, however a general rank ordering of achieved NSM accuracy is observed. SM-Cost and Avg-M-Cost appear to offer leading classification accuracies at 97.4% and 97.3% respectively. Lowest accuracies are seen with Silhouette-F and Silhouette-L at 96.5% and 95.2% respectively.

As introduced in Section A.1.1, Fig. A.34 presents a training profile view of our NSM methods. This view provides a general top level view of NSM performance and shows growth, convergence, and trends in performance (patterns evidence of over training, more potential for training). In Fig. A.34, a relationship between increasing classification

accuracy and decreasing total cost is observed. However, NSM methods that lead to the greatest reduction in cost do not necessarily provide the best classification accuracy.

A.5.4 Baseline and Converged Model Performance.

Table A.12: NSM performance for 150 independent trials is shown when the non-dynamic baseline is chosen as an accuracy performance point. Mean, standard deviation, mean improvement, and confidence interval for improvement are shown for accuracy and prototype utilization (when compared to baseline performance). Statistically significant improvements are shown in bold, statistically similar results are shown in italics.

	Match Baseline Performance									
	Accuracy						Prototypes			
NSM Method	Met	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	μ	σ	$\Delta\mu$	$\Delta\mu$ 99% CI	
Near-Mean	77	96.82	1.01	0.06	(−0.06 – 3.23)	67.93	19.05	37.07	(32.31 – 37.07)	
Mean	79	96.84	0.93	0.07	(−0.03 – 3.23)	63.71	17.99	41.29	(36.80 – 41.29)	
Closest	79	96.86	0.91	0.09	(−0.02 – 3.23)	63.36	16.06	41.64	(37.63 – 41.64)	
M-Cost	75	96.80	1.00	0.03	(−0.10 – 3.23)	72.46	15.44	32.54	(28.69 – 32.54)	
Avg-M-Cost	77	96.79	0.95	0.03	(−0.08 – 3.23)	66.01	17.35	38.99	(34.66 – 38.99)	
SM-Cost	79	96.88	0.95	0.11	(0.03 – 3.23)	69.20	13.04	35.80	(32.54 – 35.80)	
S-Cost	85	96.94	0.99	0.18	(0.10 – 3.23)	69.60	16.49	35.40	(31.28 – 35.40)	
Split-Medoid	79	96.88	0.97	0.11	(−0.00 – 3.23)	65.32	15.51	39.68	(35.81 – 39.68)	
PC	81	96.87	0.96	0.10	(0.01 – 3.23)	64.78	15.47	40.22	(36.36 – 40.22)	
Silhouette-F	60	96.60	0.91	−0.17	(−0.33 – 3.23)	82.32	20.19	22.68	(17.64 – 22.68)	
Silhouette-L	13	95.35	1.14	−1.42	(−1.69 – 3.23)	52.41	5.42	52.59	(51.24 – 52.59)	
Baseline(3x35p)	90	96.77	0.97	-	-	105	0.00	-	-	

A.5.5 Performance with respect to Prototype Utilization.

Many NSM methods offer similar performance in Figure 4.5. To compare general approach categories, well performing NSM methods are selected representative of the simple, cost-minimizing, clustering, and hybrid approaches. [Note: Closest, M-Cost, Avg-M-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F are plotted in Fig. A.36].

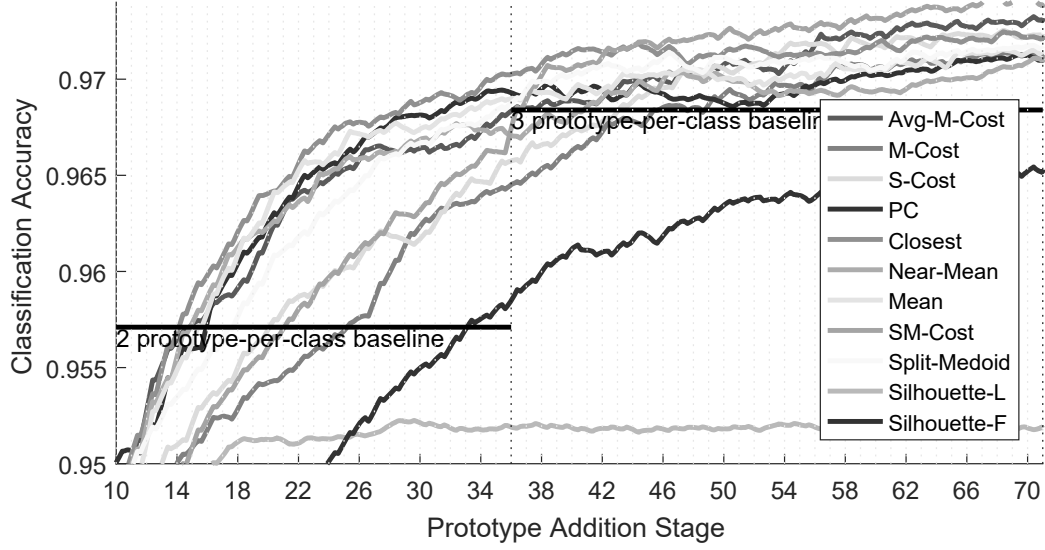
A.5.6 Margin Analysis.

After examining how total cost and classification accuracy change as a function of the number of prototypes in Section ?? and Section A.5.5, Fig. A.38 considers the relationship between margin error and the number of prototypes.

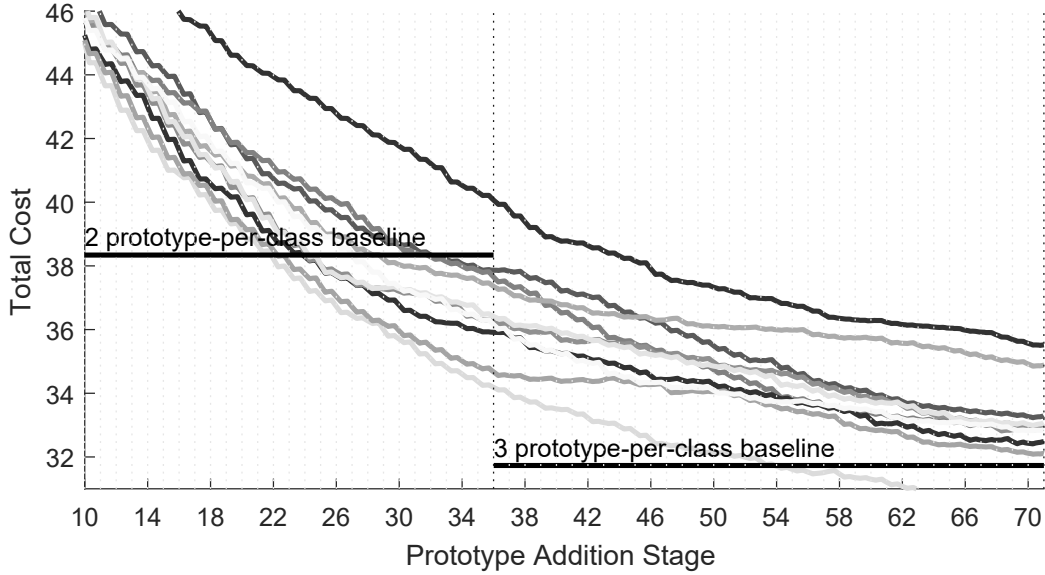
A.5.7 Relative Comparison.

Figure A.42 shows the performance of NSM methods with the LCVF data set. Three columns of plots are compared over the axes of margin error and the number of prototypes, representing empirical and theoretical components of the generalization error as described in Section 3.3. The z-axis depicts **(left)** classification accuracy, **(middle)** standard deviation in classification accuracy, and **(right)** classification accuracy, standardized over group the as $N(0, 1)$.

A less cluttered version of Fig. A.43b is achieved by down selecting the same seven top performers as mentioned in Section. A.5.5. The chosen methods are representative of our four dynamic approaches: simple, cost-minimizing, clustering, and hybrid.



(a)



(b)

Figure A.34: Classification accuracy **(a)** and Total Cost **(b)** versus prototype addition stage for seven prototype addition methods over the LCVF data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6. Two GRLVQI “Baselines” trained with a fixed number of prototypes are included for reference. Plots are the average of 90 validation result curves (30 complete runs each consisting of $K = 3$ folds).

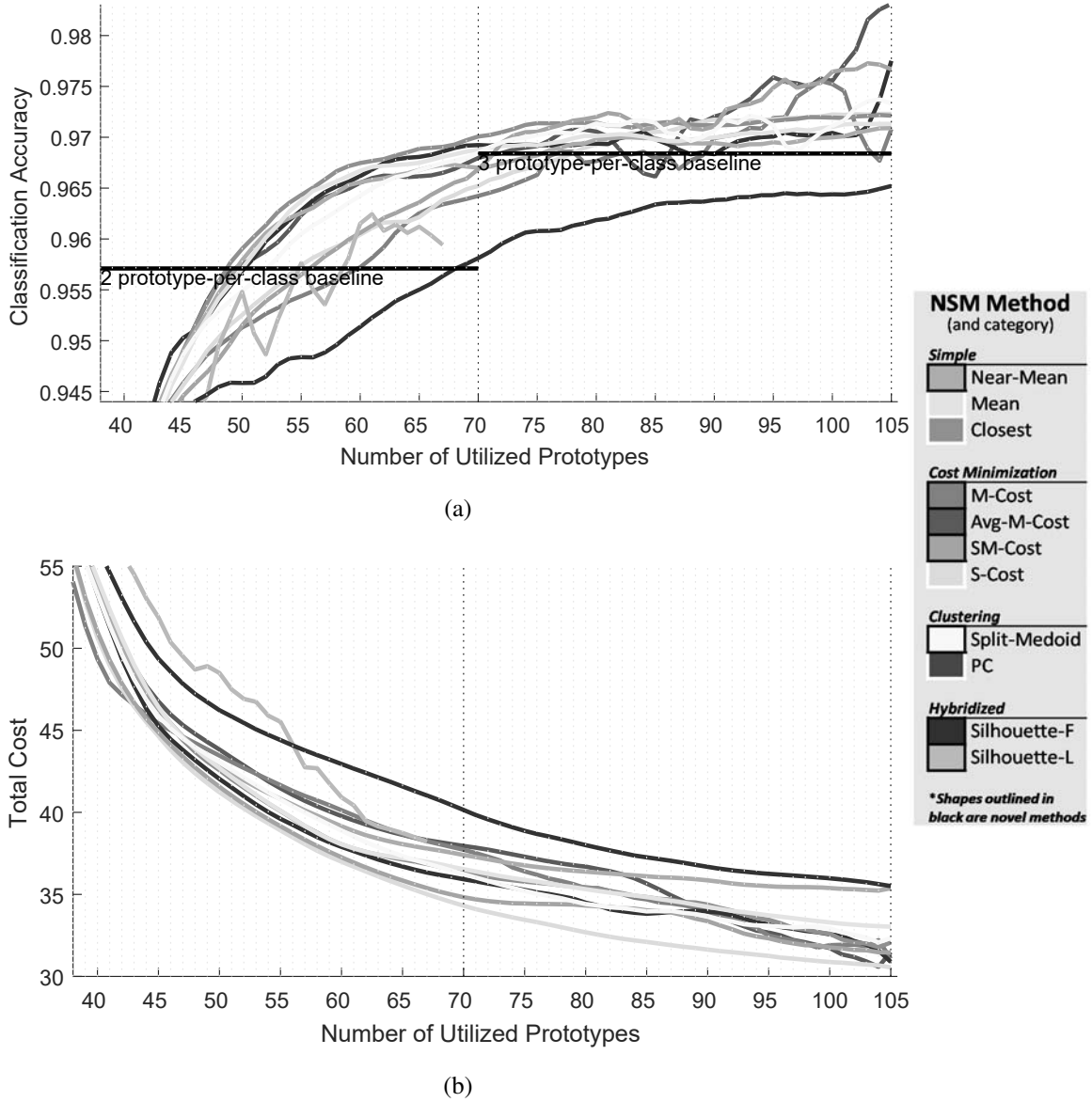


Figure A.35: Classification accuracy **(a)** and Total Cost **(b)** versus utilized number of prototypes for seven prototype addition methods over the LCVF data set with a GRLVQI algorithm. NSM methods are initialized with 1 prototype per class distributed randomly over 0.4,0.6 Two GRLVQI “Baselines” trained with a fixed number of prototypes are included for reference. Validation accuracies from the 90 curves (30 complete runs each consisting of $K = 3$ folds) are pooled for each utilized prototype value then averaged(roughly 4500 points per prototype value).

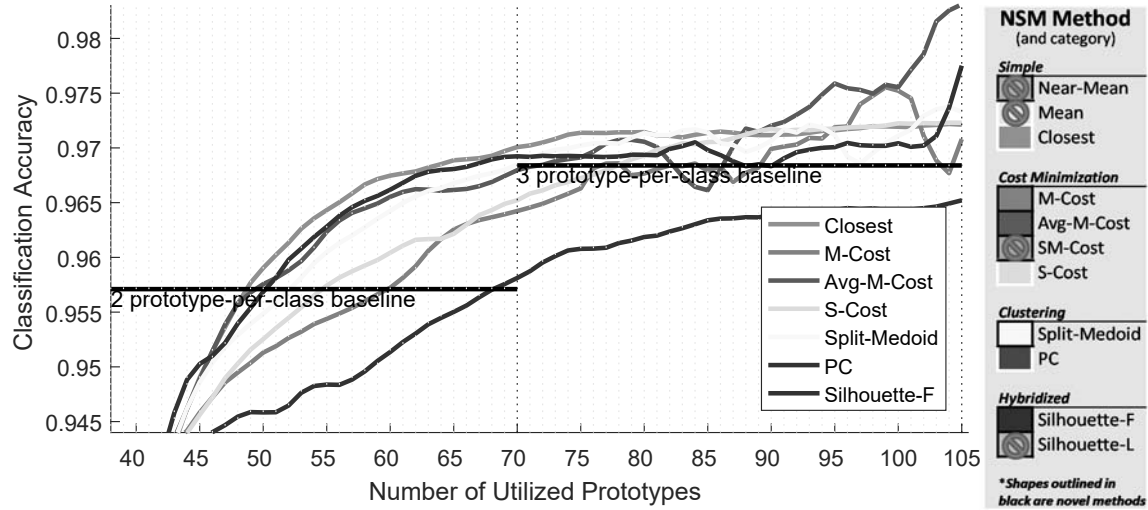


Figure A.36: Classification accuracy versus utilized number of prototypes for seven representative GRLVQI NSM methods for the LCVF dataset: Closest, M-Cost, Avg-M-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F are plotted in addition methods over the LCVF data set with a GRLVQI algorithm. Two GRLVQI “Baselines” trained with a fixed number of prototypes are included for reference. Validation accuracies from the 90 curves (30 complete runs each consisting of $K = 3$ folds) are pooled for each utilized prototype value then averaged(roughly 4500 points per prototype value).

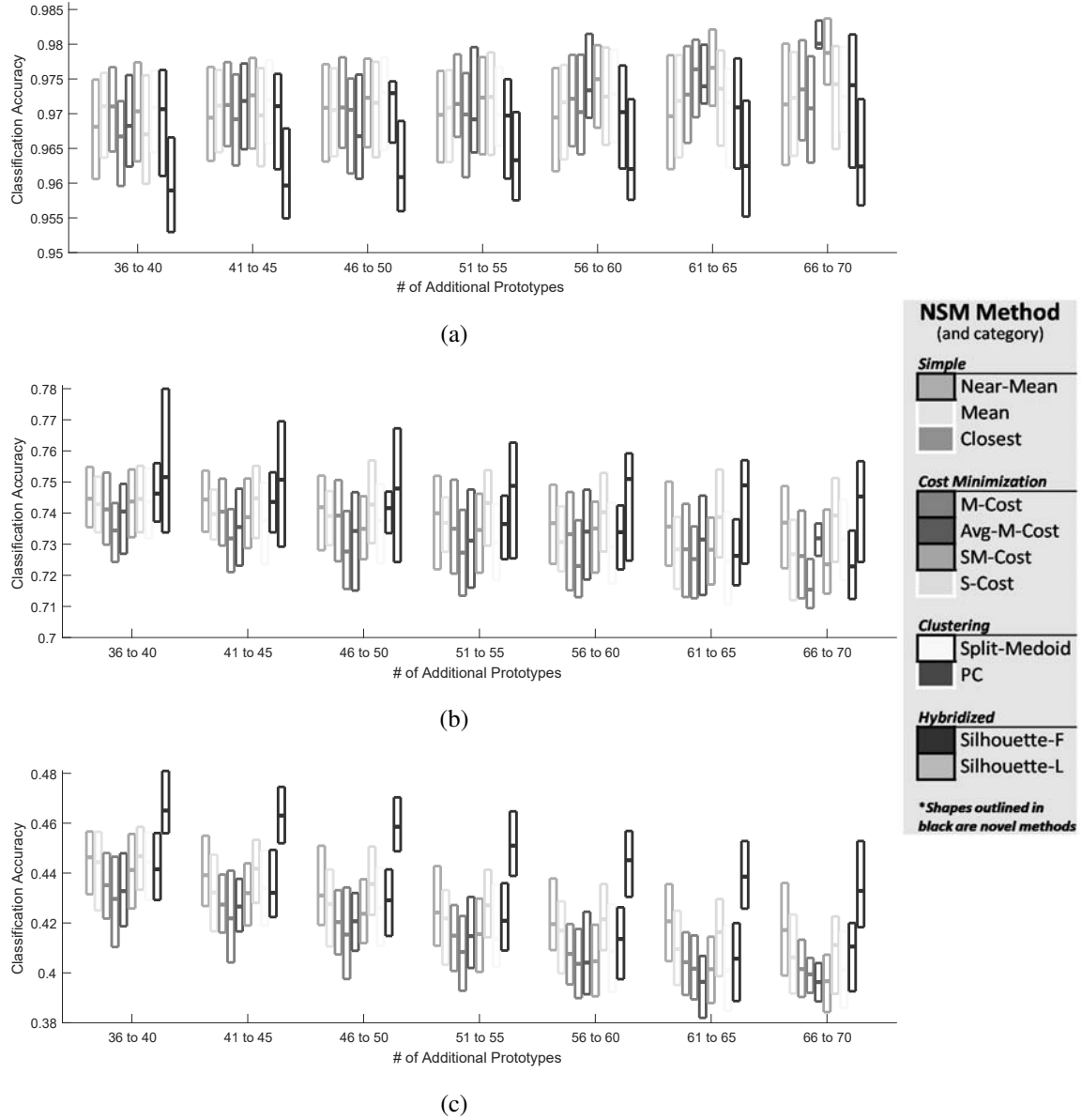


Figure A.37: For LCVF-GRLVQI: NSM classification accuracies are compared against number of additional prototypes for a margin test point, θ_{cp} . Quartiles within each NSM method (in 5 prototype intervals starting with 35) are shown as the top, middle, and bottom of colored vertical boxes (75%-ile, 50%-ile, and 25%-ile). Plots (a), (b), and (c) correspond to θ_{cp} values of 0, $6.3e - 5$, and $1.5e - 4$.

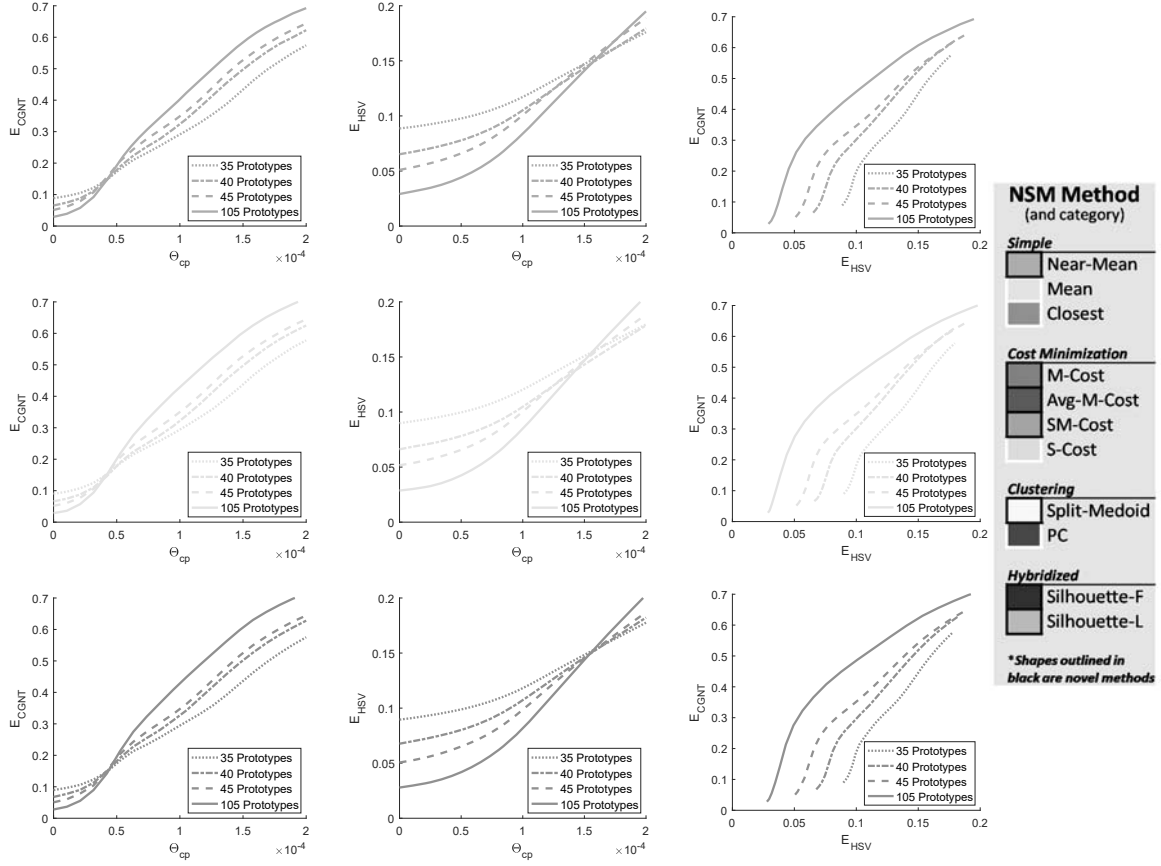


Figure A.38: Simple NSM methods with 35, 40, 45, and 105 prototypes display **(left)** CGNT Margin Error as a function of θ_{cp} , **(middle)** HSV Margin Error as a function of θ_{cp} , and **(right)** HSV Margin Error versus CGNT Margin Error.

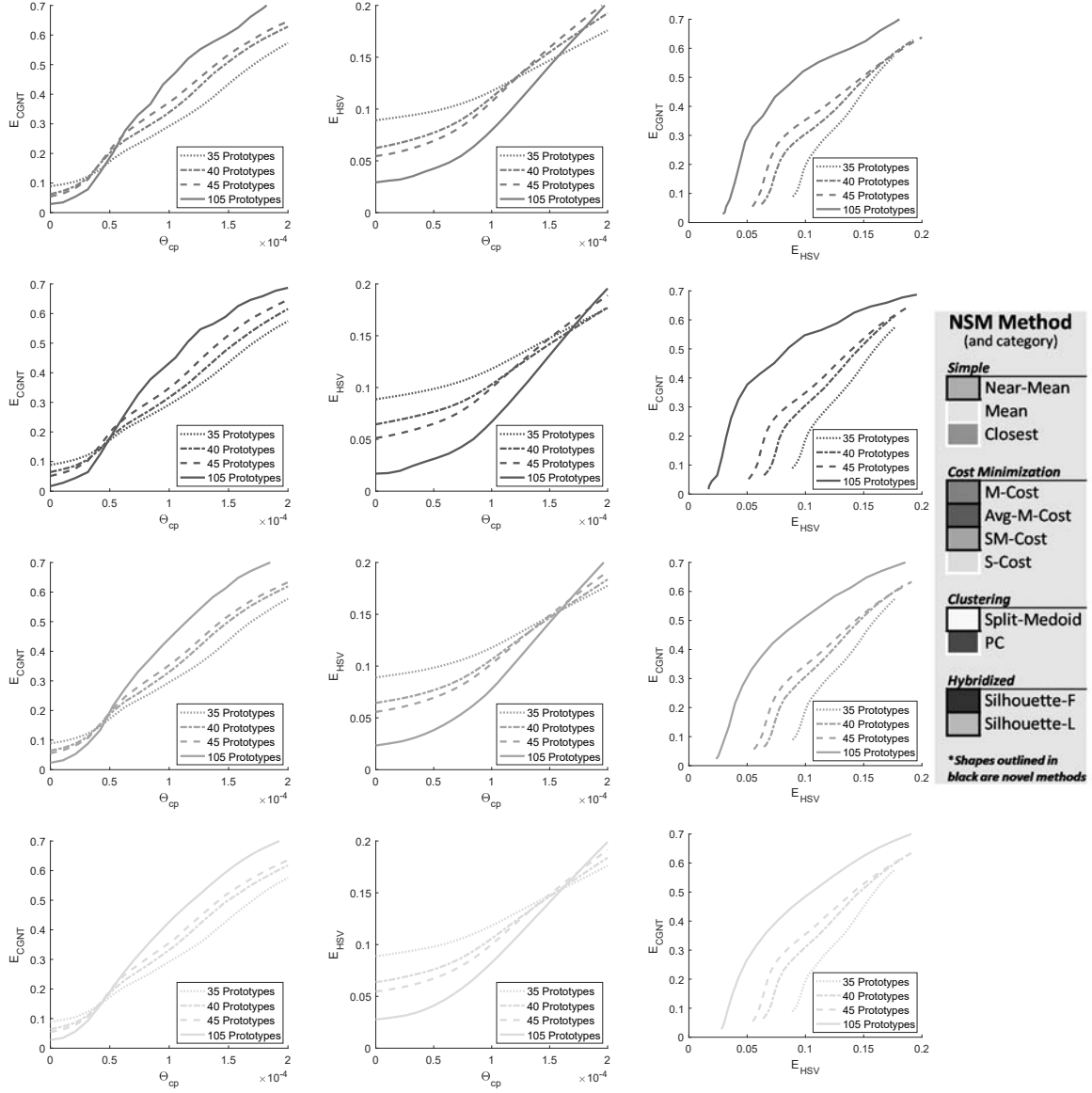


Figure A.39: Cost Minimizing NSM methods with 35, 40, 45, and 105 prototypes display **(left)** CGNT Margin Error as a function of θ_{cp} , **(middle)** HSV Margin Error as a function of θ_{cp} , and **(right)** HSV Margin Error versus CGNT Margin Error.

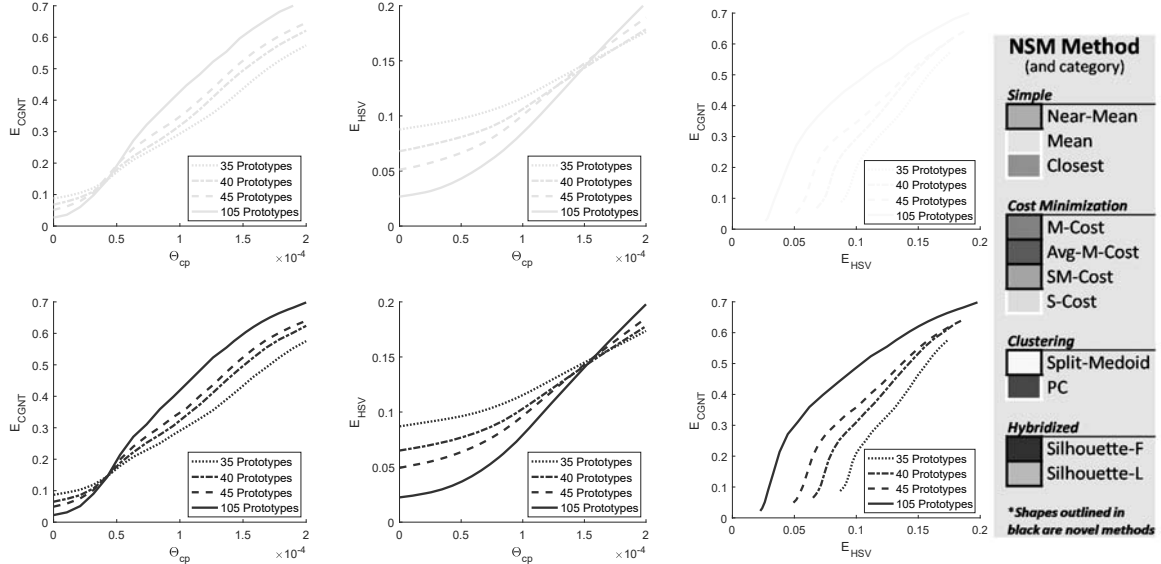


Figure A.40: Clustering NSM methods with 35, 40, 45, and 105 prototypes display **(left)** CGNT Margin Error as a function of θ_{cp} , **(middle)** HSV Margin Error as a function of θ_{cp} , and **(right)** HSV Margin Error versus CGNT Margin Error.

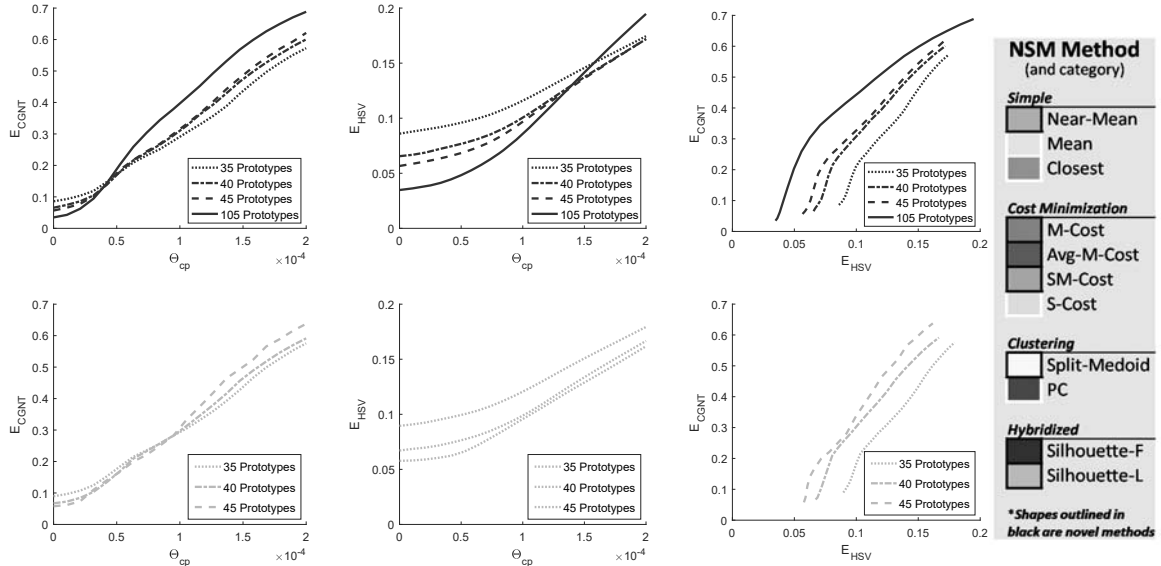


Figure A.41: Hybridized NSM methods with 35, 40, 45, and 105 prototypes display **(left)** CGNT Margin Error as a function of θ_{cp} , **(middle)** HSV Margin Error as a function of θ_{cp} , and **(right)** HSV Margin Error versus CGNT Margin Error.

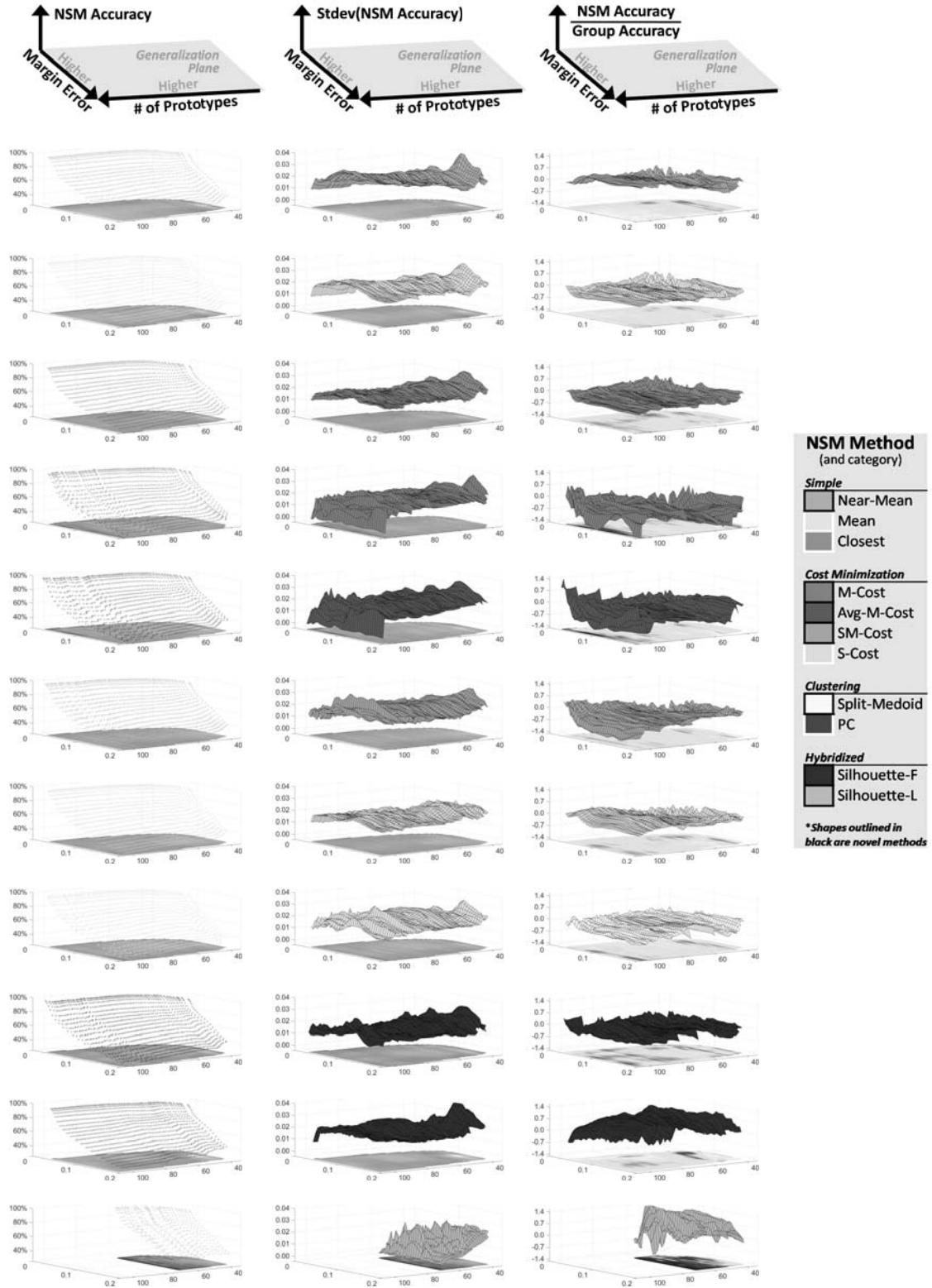


Figure A.42: Comparison of NSM methods for LCVF dataset (top row shows axes for each column: classification accuracy, standard deviation, and relative accuracy (l-r)).

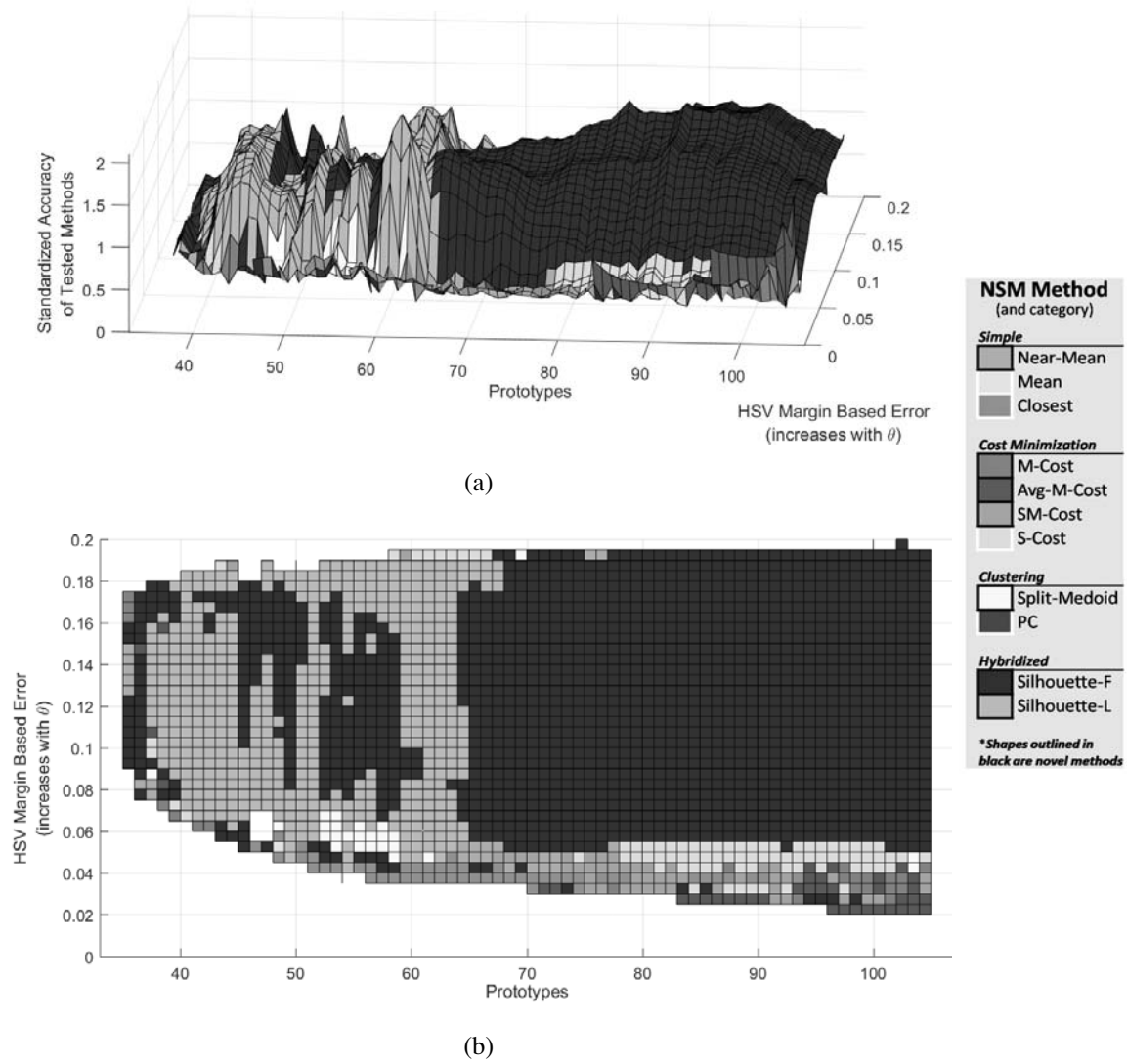


Figure A.43: (a) This composite of Fig. A.42(right-column) selects the best performing NSM method over each grid point. (b) presents a flattened view, indicating NSM methods with the highest average classification accuracy over a range of utilized prototypes and margin error values, representing the generalization error.

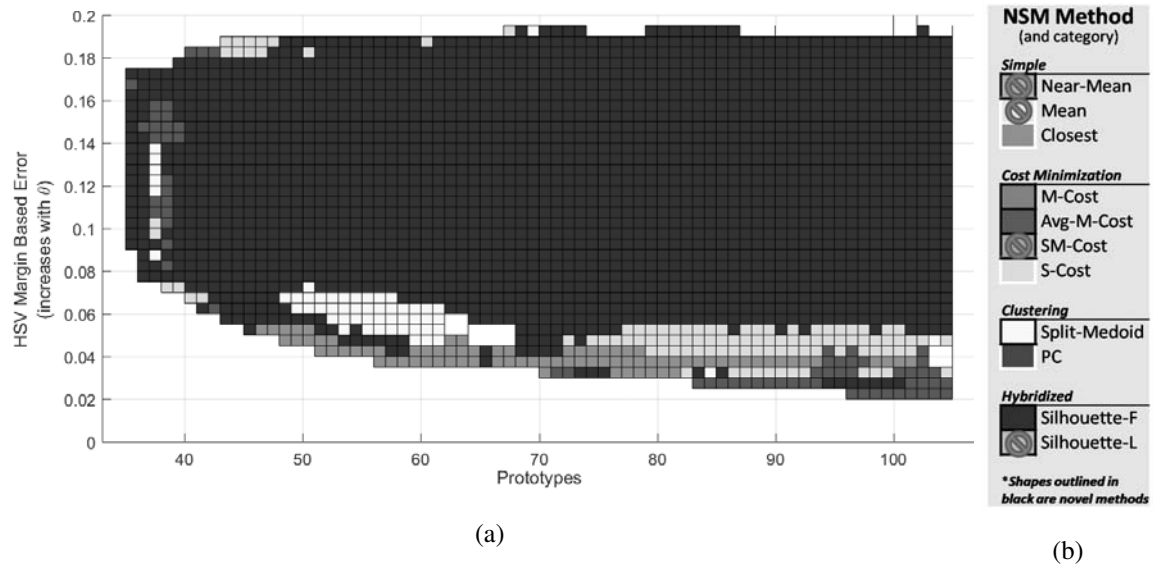


Figure A.44: This flattened surface is composed of seven of the best performing NSM methods (Closest, M-Cost, Avg-M-Cost, S-Cost, Split-Medoid, PC, and Silhouette-F). Colored grids indicate NSM methods with the highest average classification accuracy over a range of utilized prototypes and margin error values, representing the generalization error.

Appendix B: Role of Prototype Insertion Phase

This Appendix demonstrates application of the analysis methodologies established in this research and challenges the significance of the NSM prototype insertion phase. Several case studies are presented that illustrate NSM performance is not principally determined through region selection alone.

Network Structure Modification (NSM) (as introduced in Section 3.1) is composed of two phases: (1) region identification and (2) prototype insertion. Intuition suggests that of these two phases, region identification would largely dominate classification outcome as it often defines the class label. It is possible additional training can overcome a poor prototype position, however, a prototype label is permanent. To examine this assumption, both MICE and LCVF are investigated over a limited set of NSM methods, that includes Avg-M-Cost, M-Cost, S-Cost, PC, Closest, Near-Mean, and Mean. To develop the intended flow we begin the investigation with LCVF.

Mean, Closest, and Near-Mean share a common approach for region identification (the class with the largest percentage of misclassified samples). Based upon the assumption that performance is dominated by region identification, similar performance would be expected from these three methods, as any differences in placement would be largely overcome through the continued training process following each new prototype insertion. Fig. B.1(top), shows a composite view of best performing NSMs for different regions of margin error and number of utilized prototypes (from 30×3 runs). Three circles are drawn on the composite plot, identifying operating regions where Mean, Closest, and Near-Mean offer the highest relative performance. These circles correspond to portions of the individual surface plots in Fig. B.1(bottom), which reflect the performance differences through shape and height (especially Near-Mean).

Mean offers best performance in limited regions ($0.07 < E_{HSV} < 0.09$ and 42 – 45 utilized prototypes), whereas Closest appears dominant across most of the low margin error region ($E_{HSV} < 0.05$ and 46 – 93 utilized prototypes). Near-Mean also offers strong performance, but specifically in regions of medium to high margin error ($0.06 < E_{HSV}$ and 65 – 105 utilized prototypes). Of the three NSMs, Closest’s approach of boundary bombardment produces the highest overall classification accuracies (low error region) and appears to have an advantage for this data set. Near Mean however, behaves very different from the other two NSM methods for LCVF, which has dramatically different class densities. It is possible that misclassified samples are widely spread, giving Near-Mean an advantage in generalization since prototype placement occurs in known locations, guaranteed to provide benefit. These differences in performance outcomes illustrate the influence of the prototype insertion phase.

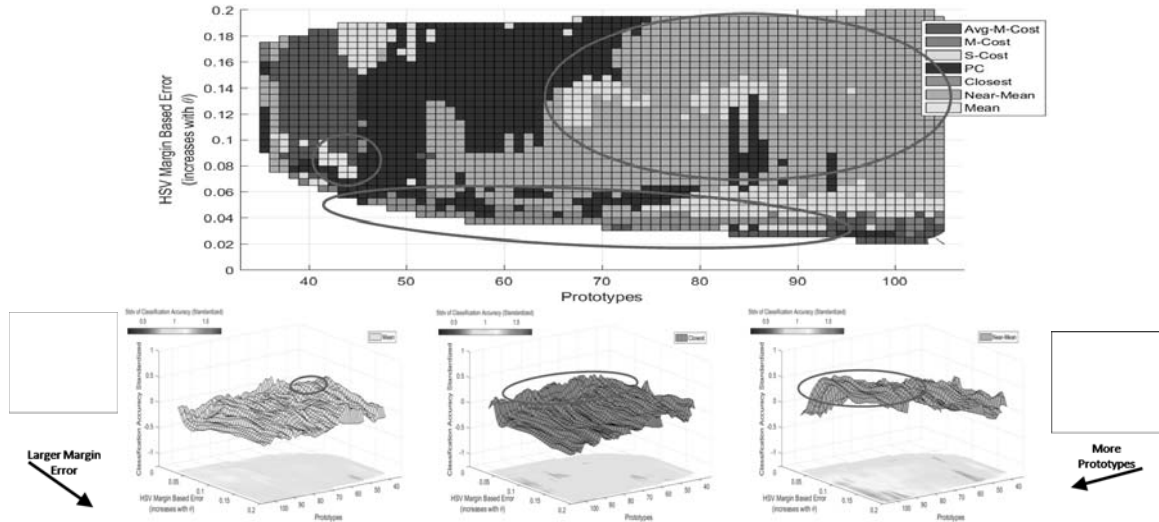


Figure B.1: LCVF with six selected NSM methods illustrate that choice of prototype insertion approach does play an important role in overall classification accuracy and generalization error.

This investigation of the prototype insertion phase is continued with the MICE data set for the same six NSM methods. In Fig. B.2, two operating regions are observed where Mean offers the best relative performance: ($E_{HSV} > 0.2$ and 13 – 22 utilized prototypes) and ($E_{HSV} < 0.15$ and 25 – 40 utilized prototypes). Closest and Near Mean on the other hand, show much lower relative accuracies as indicated by surface heights in Fig. B.2(bottom). As neither of the other two NSMs offer high relative performance, we re-affirm the conclusion that choice of prototype insertion strategy Interestingly, while Near-Mean offered a clear advantage over Mean in LCVF, the reverse is true for MICE. In Fig. B.2(top) two circles are drawn that correspond to best performance from Mean Fig. B.2(bottom-left).

This suggests the choice between inserting prototypes at interpolated positions (Mean) or at known sample location (Closest, Near-Mean) depends on the data set. Mean's previous disadvantage with LCVF could be attributed to less useful interpolation between potentially multi-modal groups of misclassified samples. This risk grows with increasing numbers and complexity of classes. Disproportional sample numbers per class may also have posed a strong disadvantage for Mean with LCVF data. For the MICE data set, however, we see that for fewer classes of roughly equal size, Mean performs superior to NSM methods with the same region identification approach.

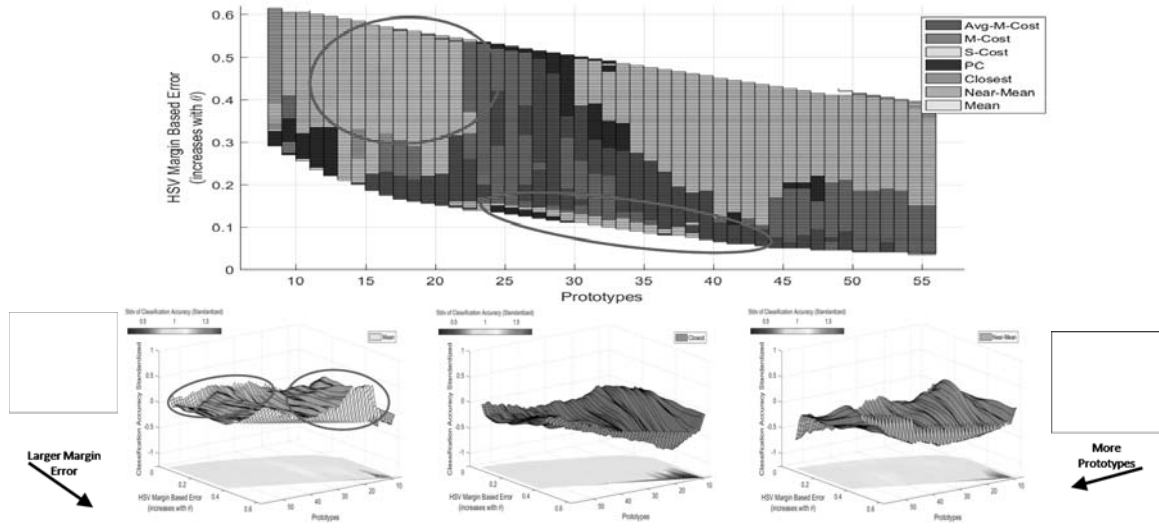


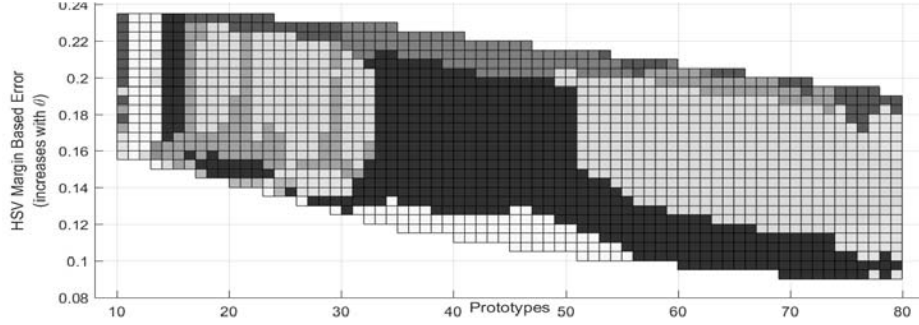
Figure B.2: MICE with six selected NSM methods illustrate that choice of prototype insertion approach does play an important role in overall classification accuracy and generalization error.

Appendix C: Limited M-Cost and Silhouette Performance with USPS

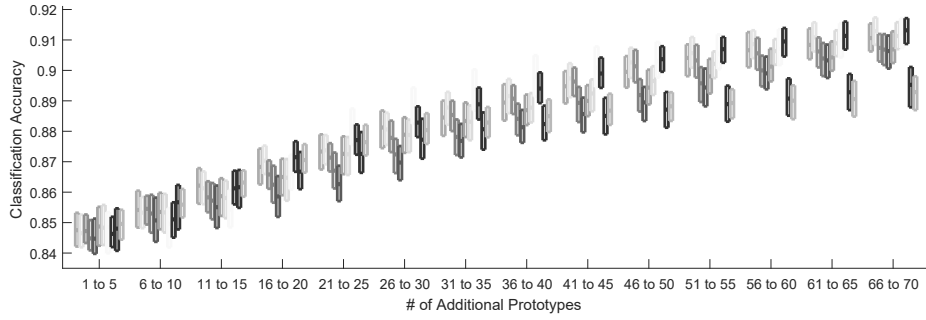
This Appendix demonstrates application of the analysis methodologies established in this research and investigates different performance trends of NSM methods on the USPS data set. We examine these trends in context of data set characteristics, and trace potential links between USPS attributes and NSM design.

In prior analysis (Section 4.3.4), M-Cost, Avg-M-Cost, and Silhouette-F, were among the best performers with the MICE, ZWAVE, and LCVF data sets, but struggle to perform well with USPS. To investigate these results, several modified box plots for USPS are considered, showing prototype utilization results for values of $\theta_{cp} = 0$ and $\theta_{cp} = 7.9e^{-3}$. For perspective, a maximum value of $\theta_{cp} = 2.5e^{-2}$ is utilized to form the composite view in Fig. C.1. Fig. C.1 suggests that neither M-Cost, Avg-M-Cost, or Silhouette-F is a near competitor within this low to moderate margin error region $E_{HSV} < 0.16$, especially for more than 30 utilized prototypes. Silhouette-L and Closest are among the other poor performers. As ineffective prototype utilization could be due to poor region identification or prototype insertion, we examine commonalities between these NSM methods. As seen in Fig. 4.10(right), all five of these methods exhibit behaviors more closely aligned with a boundary prototype insertion approach. In fact, M-Cost, Avg-M-Cost, and both Silhouette NSM methods utilize the same cost minimization mechanism during their prototype insertion phase.

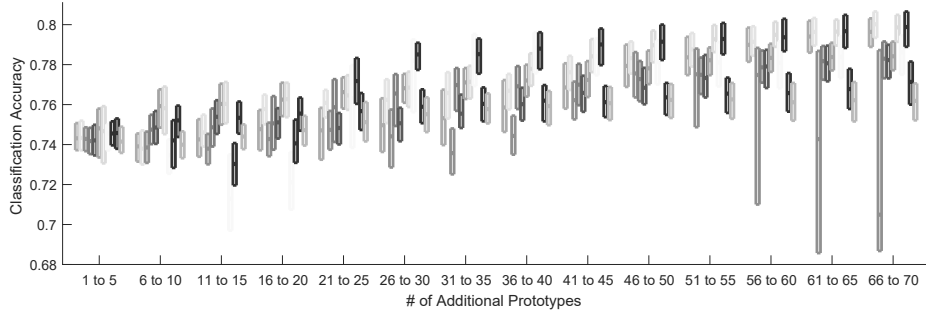
While a more cluster-centric prototype insertion strategy is reasonable, Silhouette-F is expected to be included due to its balanced or multi-role capacity. The source of Silhouette-F's difficulty with the USPS data may be traced to the implementation of the silhouette coefficient. Silhouette-F adopted a simplification step to compute the distance to the nearest out-of-class sample (as opposed to average distance to out-of-class samples). As a result, the adapted form may be sensitive to uncharacteristically close out-of-class samples. USPS



(a)



(b)



(c)

Figure C.1: NSM plots USPS showing **(a)** a composite view, **(b)** a modified boxplot with $\theta_{cp} = 0$, and **(c)** a modified boxplot with $\theta_{cp} = 7.9e^{-3}$.

offers by far the most samples per class (a median of 829) compared to the other three data sets (with medians of 37-230 samples per class), creating increased opportunities to disrupt Silhouette-F. The original silhouette coefficient, as described by [58], considers all out of class samples, making it less sensitive to outliers. Unfortunately, a full implementation

presents an increased computational burden due to the increased number of necessary calculations. However, adopting the original silhouette coefficient formulation may provide better accuracy for data sets that have high probabilities of containing outliers (either through in-class variation or sheer quantity of samples).

Bibliography

- [1] J. Climer, M. J. Mendenhall, and G. L. Peterson, “Dynamic prototype addition in generalized relevance learning vector quantization improved,” *Transactions on Neural Networks*, 2017. (In review).
- [2] D. Nova and P. A. Estévez, “A review of learning vector quantization classifiers,” *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 511–524, 2014.
- [3] R. Gilad-bachrach, A. Navot, and N. Tishby, “Large margin principals for feature selection: A tutorial.” http://www.cs.huji.ac.il/labs/learning/code/feature_selection/tutorial/distBased.html, 2004. Accessed: 2017-05-16.
- [4] M. J. Mendenhall, *A Neural Relevance Model for Feature Extraction from Hyperspectral Images, and Its Application in the Wavelet Domain*. Dissertation, Rice University, 2006.
- [5] E. Merenyi, W. Farrand, J. Taranik, and T. Minor, “Classification of hyperspectral imagery with neural networks: comparison to conventional tools,” *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 71–90, 2014.
- [6] S. Russell and P. Norvig, *Artificial Intelligence: A modern approach*. Upper Saddle River, NJ: Prentice Hall, 2nd ed., 2002.
- [7] General Mark A. Welsh III, “Data-to-decision: a campaign of experiments,” tech. rep., Office of the Air Force Chief of Staff, May 2016.
- [8] General David L. Goldfein, “CSAF letter to airmen: Enhancing multi-domain command and control.” <http://www.af.mil/News/Article-Display/Article/1108931/csaf-letter-to-airmen/>, March 2017. Online; posted 6-Mar-2017.
- [9] Defense Science Board, “Summer study on autonomy.” <http://www.acq.osd.mil/dsb/reports/2010s/DSBSS15.pdf>, 2016. Accessed: 2017-06-13.
- [10] M. R. Endsley, “Supporting situation awareness in aviation systems,” in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 5, pp. 4177–4181, IEEE, 1997.
- [11] S. W. Harshman, B. A. Geier, M. Fan, S. Rinehardt, B. S. Watts, L. A. Drummond, G. Preti, J. B. Phillips, D. K. Ott, and C. C. Grigsby, “The identification of hypoxia biomarkers from exhaled breath under normobaric conditions,” *Journal of breath research*, vol. 9, no. 4, p. 047103, 2015.
- [12] Office of the Deputy Secretary of Defense, “Establishment of an algorithmic warfare cross-functional team (project maven).” <https://www.govexec.com/media/gbc/>

docs/pdfs_edit/establishment_of_the_awcft_project_maven.pdf, March 2017. Online; posted 26-Apr-2017.

- [13] J. Climer, “Overcoming pose limitations of a skin-cued histograms of oriented gradients dismount detector through contextual use of skin islands and multiple support vector machines.” Masters Thesis. Air Force Institute of Technology.
- [14] R. Mash, B. Borghetti, and J. Pecarina, *Improved aircraft recognition for aerial refueling through data augmentation in convolutional neural networks*, pp. 113–122. Cham: Springer International Publishing, 2016.
- [15] M. D. Williams, S. A. Munns, M. A. Temple, and M. J. Mendenhall, “RF-DNA fingerprinting for airport wimax communications security,” in *Network and System Security (NSS), 2010 4th International Conference on*, pp. 32–39, IEEE, 2010.
- [16] J. B. Fraley and J. Cannady, “The promise of machine learning in cybersecurity,” in *SoutheastCon, 2017*, pp. 1–6, IEEE, 2017.
- [17] United States Air Force Office of the Chief Scientist, “Autonomous horizons: system autonomy in the Air Force—a path to the future.” <http://www.af.mil/Portals/1/documents/SECAF/AutonomousHorizons.pdf>, 2015. Accessed: 2017-07-28.
- [18] D. Graupe, *Principles of Artificial Neural Networks*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2nd ed., 2007.
- [19] T. Kohonen, M. R. Schroeder, and T. S. Huang, eds., *Self-Organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 3rd ed., 2001.
- [20] T. Villmann, A. Bohnsack, and M. Kaden, “Can learning vector quantization be an alternative to svm and deep learning?-recent trends and advanced variants of learning vector quantization for classification learning,” *Journal of Artificial Intelligence and Soft Computing Research*, vol. 7, no. 1, pp. 65–81, 2017.
- [21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2009.
- [22] T. Villmann, B. Hammer, and M. Strickert, “Supervised neural gas for learning vector quantization,” in *Proc. of the 5th German Workshop on Artificial Life (GWAL-5)*, pp. 9–16, 2002.
- [23] K. Crammer, R. Gilad-bachrach, A. Navot, and N. Tishby, “Margin analysis of the lvq algorithm,” in *Advances in Neural Information Processing Systems 2002*, pp. 462–469, MIT press, 2002.
- [24] J. Climer and M. J. Mendenhall, “Dynamic prototype addition in generalized learning vector quantization,” in *Advances in Self-Organizing Maps and Learning Vector Quantization*, pp. 355–368, Springer International Publishing, 2016.

- [25] M. Blachnik and W. Duch, "Improving accuracy of lvq algorithm by instance weighting," in *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III, ICANN'10*, (Berlin, Heidelberg), pp. 257–266, Springer-Verlag, 2010.
- [26] D. D. Wang and J. Xu, "Fault detection based on evolving LVQ neural networks," in *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No.96CH35929)*, vol. 1, pp. 255–260 vol.1, Oct 1996.
- [27] T. J. Bihl, M. A. Temple, and K. W. Bauer, "An optimization framework for generalized relevance learning vector quantization with application to Z-Wave device fingerprinting," in *HICSS*, 2017.
- [28] C. M. Bishop, *Pattern recognition and machine learning*. New York, NY: Springer-Verlag, 2006.
- [29] N. Singh and S. Mohanty, "Short term price forecasting using adaptive generalized neuron model," in *Advances in Computer and Computational Sciences*, pp. 419–428, Springer, 2017.
- [30] D. C. CireÅan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, 2010. PMID: 20858131.
- [31] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, pp. 1464–1480, Sep 1990.
- [32] M. Biehl, A. Ghosh, and B. Hammer, "Dynamics and generalization ability of LVQ algorithms," vol. 8, pp. 323–360, 2007.
- [33] M. Swiercz, J. Kochanowicz, J. Weigele, R. Hurst, D. S. Liebeskind, Z. Mariak, E. R. Melhem, and J. Krejza, "Learning vector quantization neural networks improve accuracy of transcranial color-coded duplex sonography in detection of middle cerebral artery spasm preliminary report," *Neuroinformatics*, vol. 6, no. 4, p. 279, 2008.
- [34] R. Fusco, M. Sansone, S. Filice, G. Carone, D. M. Amato, C. Sansone, and A. Petrillo, "Pattern recognition approaches for breast cancer dce-mri classification: A systematic review," *Journal of Medical and Biological Engineering*, vol. 36, no. 4, pp. 449–459, 2016.
- [35] M. Cottrell, "WSOM+ 2017 12th international workshop on self-organizing maps and learning vector quantization, clustering and data visualization." Accessed: 2017-07-07.
- [36] A. Sato and K. Yamada, "Generalized learning vector quantization," in *Advances in Neural Information Processing Systems*, pp. 423–429, The MIT Press, 1996.

- [37] M. J. Mendenhall and E. Merenyi, "Relevance-based feature extraction for hyperspectral images," *IEEE Transactions on Neural Networks*, vol. 19, pp. 658–672, April 2008.
- [38] B. Hammer and T. Villmann, "Generalized relevance learning vector quantization," *Neural Netw.*, vol. 15, pp. 1059–1068, Oct. 2002.
- [39] C. J. Burges *et al.*, "Dimension reduction: A guided tour," *Foundations and Trends® in Machine Learning*, vol. 2, no. 4, pp. 275–365, 2010.
- [40] D. DeSieno, "Adding a conscience to competitive learning," in *Neural Networks, 1988., IEEE International Conference on*, pp. 117–124 vol.1, July 1988.
- [41] A. Sato, "An analysis of initial state dependence in generalized LVQ," in *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, vol. 2, pp. 928–933, 1999.
- [42] T. Villmann and J. C. Claussen, "Magnification control in self-organizing maps and neural gas," *Neural Comput.*, vol. 18, pp. 446–469, Feb. 2006.
- [43] F. Poirier, "DVQ: dynamic vector quantization application to speech processing," in *Second European Conference on Speech Communication and Technology, EUROSPEECH 1991, Genova, Italy, September 24-26, 1991*, 1991.
- [44] M. Grbovic and S. Vucetic, "Learning vector quantization with adaptive prototype addition and removal," in *Proceedings of the 2009 International Joint Conference on Neural Networks, IJCNN'09*, (Piscataway, NJ, USA), pp. 911–918, IEEE Press, 2009.
- [45] C. De Stefano, C. D'Elia, A. Marcelli, and A. di Frecac, "Improving dynamic learning vector quantization," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2, pp. 804–807, 2006.
- [46] S. Bermejo, J. Cabestany, and M. Payeras-Capell, "A new dynamic LVQ-based classifier and its application to handwritten character recognition.," in *ESANN*, pp. 203–208, 1998.
- [47] V. Losing, B. Hammer, and H. Wersing, "Interactive Online Learning for Obstacle Classification on a Mobile Robot," in *Neural Networks (IJCNN), 2015 International Joint Conference on*, pp. 1–8, IEEE, 2015.
- [48] T. C. Kietzmann, S. Lange, and M. Riedmiller, "Incremental GRLVQ: Learning relevant features for 3d object recognition," *Neurocomputing*, vol. 71, no. 1315, pp. 2868 – 2879, 2008. Artificial Neural Networks (ICANN 2006) / Engineering of Intelligent Systems (ICEIS 2006).
- [49] A. Zell, G. Mamier, M. Vogt, N. Mache, R. Hübner, S. Döring, K.-U. Herrmann, T. Soye, M. Schmalzl, T. Sommer, A. Hatzigeorgiou, D. Posselt, T. Schreiner,

- B. Kett, G. Clemente, and J. Wieland, *Stuttgart Neural Network Simulator (SNNS) User Manual*. University of Stuttgart, 1995.
- [50] S. Kirstein, H. Wersing, and E. Körner, “A biologically motivated visual memory architecture for online learning of objects,” *Neural Netw.*, vol. 21, pp. 65–77, Jan. 2008.
 - [51] R. Gilad-Bachrach, A. Navot, and N. Tishby, “Margin based feature selection - theory and algorithms,” in *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML ’04, (New York, NY, USA), pp. 43–51, ACM, 2004.
 - [52] B. Hammer, M. Strickert, and T. Villmann, “On the generalization ability of GRLVQ networks,” *Neural Processing Letters*, vol. 21, pp. 109–120, Apr 2005.
 - [53] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006.
 - [54] S. Bischoff, M. J. Mendenhall, A. C. Rice, and J. R. Vasquez, “Adapting learning parameter transition in the generalized learning vector quantization family of classifiers,” in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*, pp. 1–4, IEEE, 2010.
 - [55] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, “Occam’s razor,” *Information processing letters*, vol. 24, no. 6, pp. 377–380, 1987.
 - [56] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, “Density-based clustering in spatial databases: The algorithm GDBscan and its applications,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 169–194, 1998.
 - [57] R. Layton, P. Watters, and R. Dazeley, “Authorship attribution for twitter in 140 characters or less,” in *Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second*, pp. 1–8, IEEE, 2010.
 - [58] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987.
 - [59] J. J. Hull, “A database for handwritten text recognition research,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 550–554, May 1994.
 - [60] C.-C. Chang and C.-J. Lin, “A library for support vector machines: Classification (multi-class).” <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps>, 2015. Accessed: 2015-08-19.
 - [61] C. Higuera, K. J. Gardiner, and K. J. Cios, “Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome,” *PloS one*, vol. 10, no. 6, p. e0129126, 2015.

- [62] M. Lichman, "UCI machine learning repository." <http://archive.ics.uci.edu/ml>. Accessed: 2015-05-30.
- [63] M. Knight, "Wireless security - how safe is Z-wave?," *Computing Control Engineering Journal*, vol. 17, pp. 18–23, Dec 2006.
- [64] T. J. Bihl, K. W. Bauer, M. A. Temple, and B. Ramsey, "Dimensional reduction analysis for physical layer device fingerprints with application to ZigBee and Z-Wave devices," in *MILCOM 2015 - IEEE Military Communications Conference*, pp. 360–365, Oct 2015.
- [65] R. W. Klein, M. A. Temple, and M. J. Mendenhall, "Application of wavelet-based RF fingerprinting to enhance wireless network security," *Communications and Networks, Journal of*, vol. 11, no. 6, pp. 544–555, 2009.
- [66] R. O. Green, "Summaries of the 6th annual JPL airborne geoscience workshop," *J. AVIRIS Workshop*, Mar 1996.
- [67] E. Mernyi, R. B. Singer, and J. S. Miller, "Mapping of spectral variations on the surface of mars from high spectral resolution telescopic images," *Icarus*, no. 124, pp. 280–295, 1996.
- [68] P. Billingsley, *Probability and Measure*. New York, NY, USA: John Wiley and Sons, 3rd ed., 1995.
- [69] P. Bhattacharya and M. L. Gavrilova, "Crystal - a new density-based fast and efficient clustering algorithm," in *2006 3rd International Symposium on Voronoi Diagrams in Science and Engineering*, pp. 102–111, July 2006.
- [70] J. Li, M. T. Manry, C. Yu, and D. R. Wilson, "Prototype classifier design with pruning," *International Journal on Artificial Intelligence Tools*, vol. 14, no. 01n02, pp. 261–280, 2005.
- [71] Y. Xu, S. Furao, O. Hasegawa, and J. Zhao, *An Online Incremental Learning Vector Quantization*, pp. 1046–1053. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
- [72] Y. Xu, F. Shen, and J. Zhao, "An incremental learning vector quantization algorithm for pattern classification," *Neural Computing and Applications*, vol. 21, no. 6, pp. 1205–1215, 2012.
- [73] J. Merelo and A. Prieto, "G-LVQ, a combination of genetic algorithms and LVQ," in *Artificial Neural Nets and Genetic Algorithms*, pp. 92–95, Springer, 1995.
- [74] G. Castellano, A. M. Fanelli, and M. Pelillo, "An iterative pruning algorithm for feedforward neural networks," *IEEE transactions on Neural networks*, vol. 8, no. 3, pp. 519–531, 1997.

- [75] F.-M. Schleif, T. Villmann, and B. Hammer, “Local metric adaptation for soft nearest prototype classification to classify proteomic data,” in *International Workshop on Fuzzy Logic and Applications, volume 3849 of Lecture Notes in Computer Science*, pp. 290–296, Springer, 2006.

REPORT DOCUMENTATION PAGE					<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this 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 Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 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 any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.						
1. REPORT DATE (DD-MM-YYYY) 12-09-2017		2. REPORT TYPE Doctoral Dissertation			3. DATES COVERED (From — To) Oct 2014-Sep 2017	
4. TITLE AND SUBTITLE Dynamic Prototype Addition in Generalized Learning Vector Quantizers					5a. CONTRACT NUMBER	
					5b. GRANT NUMBER	
					5c. PROGRAM ELEMENT NUMBER	
					5d. PROJECT NUMBER	
6. AUTHOR(S) Climer, Jonathon R., Captain, USAF					5e. TASK NUMBER	
					5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB, OH 45433-7765					8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENG-DS-17-S-009	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally Left Blank					10. SPONSOR/MONITOR'S ACRONYM(S)	
					11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE						
13. SUPPLEMENTARY NOTES This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.						
14. ABSTRACT The classification accuracy achieved by a statistical machine learning algorithm is determined by selecting the function(s) that most closely match a domain that results in the correct labeling of new samples, while not over-fitting. This is termed the accuracy/generalization trade-off. In Learning Vector Quantization (LVQ) the underlying function is a set of vectors in the domain space that have a relationship with each other and are prototypical of the underlying data. Where the prototypes are and how many there are influences the piece-wise linear decision boundaries they create and hence the final accuracy of the resulting LVQ. This work develops a novel framework that includes the LVQ learning components of Competition, Winner Selection, and Synaptic Adaptation, and adds a new component Network Structure Modification (NSM) that allows for experimentation of four different prototype addition strategies: simple, cost-minimizing, clustering, and cost-minimizing/clustering hybrids. Within these strategies, seven novel methods are tested on data sets using the Generalized Relevance LVQ Improved (GRLVQI) algorithm. Results show that cost-minimizing strategies achieve the highest classification accuracy but do not generalize as well, clustering methods tend to generalize well but are less accurate, and hybrid strategies provide the best trade-off between accuracy and generalization.						
15. SUBJECT TERMS LVQ, RF-DNA						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT		18. NUMBER OF PAGES	
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	 UU		 188	
					19a. NAME OF RESPONSIBLE PERSON Gilbert Peterson (ENG)	
					19b. TELEPHONE NUMBER (include area code) (937) 255-3636 x4281 gilbert.peterson@afit.edu	