BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

ARTMAP AND ORTHONORMAL BASIS FUNCTION NEURAL NETWORKS FOR PATTERN CLASSIFICATION

by

BYRON MITCHELL SHOCK

B.S., Albertson College of Idaho, 1994

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2006

Approved by

First Reader

Michael A. Cohen, Ph.D.
Associate Professor of Cognitive and Neural Systems and Computer Science

Second Reader

Eric Schwartz, Ph.D.
Professor of Cognitive and Neural Systems; Electrical, Computer and Systems Engineering; and Anatomy and Neurobiology

Third Reader

Eric D. Kolaczyk, Ph.D.
Associate Professor of Mathematics and Statistics

Acknowledgments

I would like to thank my wife, Ann Walker, for her moral support and financial support as I worked to complete this dissertation. To find an adequate adjective for my thanks could take as many years as completing this dissertation has.

To the members of my dissertation committee, and especially to Dr. Michael Cohen, thank you for giving me the chance to choose my own path, to make mistakes, and to learn from both my mistakes and your guidance.

Very special thanks to Robert Ajemian, Albert Ler, Jay Bohland, and Dr. Kenneth Kraft for being there not only for many years of graduate school but also for the times I required medical attention.

I wish I could list my many officemates and teammates, as well as many others in the Department of Cognitive and Neural Systems, but there are so many of you who have made my time here better. To each and every person I've had the pleasure of sharing an office with, playing on an intramural team with, or having a conversation with about image processing, single-neuron computation, baseball, or poker, thank you. I have always looked forward to visiting with you and sharing this time.

I am grateful to colleagues at the Boston University Center for Remote Sensing for assembling the Nile River delta land use change dataset used herein.

This dissertation is based upon work supported under a National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions or recommendations expressed in this dissertation are those of the author and do not

necessarily reflect the views of the National Science Foundation. This work has also been partially supported under a Presidential University Graduate Fellowship and grants from the National Science Foundation (NSF SBR 95-13889), the Office of Naval Research (ONR N00014-95-I-409 and ONR N00014-95-0657), and the Air Force Office of Scientific Research (AFOSR F49620-01-1-0397 and AFOSR F49620-01-1-0423).

ARTMAP AND ORTHONORMAL BASIS FUNCTION NEURAL NETWORKS FOR PATTERN CLASSIFICATION

(Order No.)

BYRON MITCHELL SHOCK

Boston University Graduate School of Arts and Sciences, 2006

Major Professor: Michael A. Cohen, Associate Professor of Cognitive and Neural Systems and Computer Science

ABSTRACT

This dissertation investigates neural network approaches to pattern classification. One application considered is the classification of land use change in the Nile River delta between 1984 and 1993 from ten Landsat Thematic Mapper (Landsat TM) images acquired during this period. Other applications, including image segmentation, letter recognition, and prediction of variables from census data, are represented by the standardized DELVE (Data for Evaluating Learning in Valid Experiments) machine learning database.

An ARTMAP (Adaptive Resonance Theory Map) neural network system is developed for the land use change classification task. Cross-validation is used to enable design decisions and to enable model fitting to be done without regard to data in test partitions. The training of voting ARTMAP systems on brightness-greenness-wetness (BGW) data for multiple dates and location data results in performance competitive with previously used expert systems.

Orthonormal basis function classification methods are extended to make them appropriate for multidimensional problems. These methods share the multilayer perceptron architecture common to many neural networks. A layer of basis functions transforms the data prior to classification. Stopping rules are used to determine which basis functions to include in a model to minimize the expected mean integrated squared error (MISE). To perform stopping when using the discriminant function of Devroye et al. (1996), an appropriate MISE estimator is developed. Linear transformations to rotate data and improve multiple classification results are investigated using development benchmarks from the DELVE suite. Orthonormal basis function neural network classifiers using these principles are developed and tested along with standard pattern classification techniques on the DELVE suite. Orthonormal basis function systems appear to be well suited for some multidimensional problems. These systems, along with benchmark classifiers, are also applied to the Nile River delta dataset. Although orthonormal basis function systems are an appropriate choice for this task, the best performance observed on this dataset is that of linear discriminant analysis (LDA) applied to multitemporal data.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AIC	Akaike Information Criterion
ANOVA	Analysis of Variance
ART	Adaptive Resonance Theory
ARTMAP	Adaptive Resonance Theory Map
ASR	Average Squared Residual
Backprop	Backpropagation Neural Network
CCA	Canonical Correlation Analysis
CVA	Canonical Variates Analysis
CVAq	First q Canonical Variates
DAGSVM	Directed Acyclic Graph Support Vector Machine
DELVE	Data for Evaluating Learning in Valid Experiments (University of Toronto machine learning benchmark suite)
GCV	Generalized Cross-validation
i.i.d.	Independent, Identically Distributed
KNN	K-Nearest Neighbors
Landsat TM	Landsat Thematic Mapper
LDA	Linear Discriminant Analysis
MARS	Multivariate Adaptive Regression Splines
MISE	Mean Integrated Squared Error
PCA	Principal Components Analysis
PCAq	First q Principal Components

POLYMARS	Polynomial Multivariate Adaptive Regression Splines
RMISE	Relative Mean Integrated Squared Error
SMO	Sequential Minimal Optimization
SVD	Singular Value Decomposition
SVM	Support Vector Machine
ZC order	Zero-crossing order

Chapter 1

Introduction

1.1 ARTMAP neural networks for land use change classification

The ability to detect and monitor changes in conditions at the Earth's surface is essential for understanding human impact on the environment and for assessment of the sustainability of development. Advances in remote sensing technology are making vast multitemporal databases available to researchers. *Multitemporal* refers to data collected at multiple times; the multitemporal data studied in this dissertation are *multi-date* data, having been collected on multiple dates. Such databases, which contain multiple images of a given region acquired over a period of time, may yield important information about environmental changes. This information needs to be extracted from high-dimensional multispectral and multitemporal data. Automated change classification based on sequences of large satellite images requires new, appropriate pattern recognition methods. These methods should detect subtle long-term changes from high-dimensional data.

A novel land use change classification methodology that employs an ARTMAP neural network classifier has been developed as part of this dissertation. This methodology allows the identification of changes across a sequence of images of a given area. These images need not be taken under the same seasonal, atmospheric, or illumination conditions, and sensor calibration need not be consistent across the sequence. The ARTMAP system can overcome these inconsistencies by learning to identify the spectral patterns across multiple dates. This methodology was developed and

evaluated on a multi-date database of Landsat TM images of the Nile River delta region showing land use changes from 1984 to 1993.

The ARTMAP land use change classifier system employs a cross-validation scheme that allows the system to be evaluated on subsets of an available database while being developed using other subsets. The system was developed for and applied to the problem of classifying land use changes in the Nile River delta over a period of ten years.

1.2 Orthonormal basis function neural networks for pattern classification

The current understanding of pattern classification using orthonormal basis functions is insufficient for application to multidimensional classification problems. Existing basis function selection methods as mainly used in the literature do not scale well to multidimensional problems. Moreover, there are few serious evaluations of orthonormal basis function classifiers available for the practitioner who wishes to implement such a method.

This dissertation develops statistical tools based on analysis of the mean integrated squared error (MISE) measure of goodness-of-fit applied to classification models. These tools are necessary to allow stopping rule basis set selection methods to be applied to multidimensional classification problems in a novel way. The MISE-based tools lead to methods for simplifying orthonormal basis function neural network models by single-term exclusion and methods for comparing models that use different bases. Multidimensional classification models have been constructed and optimized using these

new methods and have been tested on a number of benchmark classification tasks including databases from the DELVE (Rasmussen, Neal et al. 1996) and UCI machine learning (Hettich, Blake et al. 1998) archives. These tests were examined for further insights into the characteristics of multidimensional orthonormal basis function classifiers.

Orthonormal basis function neural networks are based on the multilayer perceptron architecture. They are closely related to radial basis function neural networks. Both orthonormal basis function neural networks and radial basis function neural networks consist of an input layer, a basis function layer that maps the input vector into a high-dimensional space, and an output layer composed of additive neurons.

Radial basis function networks employ a set of basis functions of identical form that can be described by the locations of their centers and the variables that control their spread. These parameters specify a transformation from the problem space into a radial basis space. In this type of basis, data are transformed into a system in which the parameters associated with each basis function covary with the parameters associated with every other basis function. Determining optimal parameters requires a nonlinear optimization, which takes more time to compute than a linear optimization.

Orthonormal basis function networks, on the other hand, employ a set of basis functions that are necessarily orthogonal. In models of this type, the parameters associated with each basis function can be determined independently of the parameters associated with every other basis function. Determining optimal parameters requires only

fast linear computations. For a two-class problem, this training process has computational complexity

$$O(MND), \tag{2.1}$$

where M is the number of training exemplars, N is the number of basis functions under consideration, and D is the dimensionality of the input space. The complexity is uniform under all assumptions about the data and scales linearly in each of M, N, and D. For comparison, training of a fast support vector regularization algorithm has computational complexity

$$O(M^2 D) \tag{2.2}$$

(Platt 1999). Redundancy in the data can improve computation speed by up to an order of M. Fitting a radial basis function neural network or similar model with fixed, nonorthogonal basis functions requires regularization, carried out through computation of a matrix pseudoinverse (Haykin 1994; Bishop 1995). The pseudoinverse is equivalent computationally to obtaining the singular value decomposition (SVD). The SVD can be computed in $4MN^2 + 8N^3$ operations, in the case of the Golub-Reinsch SVD algorithm, or $2MN^2 + 11N^3$ operations, in the case of the Chan or R-SVD algorithm (Golub and Van Loan 1989). Thus, regularization of a network of fixed, nonorthogonal basis functions typically has computational complexity

$$O(MN^2 + N^3). \tag{2.3}$$

Orthonormal basis function models are expected to be faster to fit than similar models employing other bases or regularization methods.

Orthonormal basis function networks are known in the statistical literature, in which they are referred to as the method of orthonormal series expansions (Devroye, Györfi et al. 1996). Examples given in the literature demonstrate the use of orthonormal series expansions for one-dimensional and two-dimensional classification problems. However, it appears that this method has not been successfully applied to relevant multidimensional problems of dimension greater than two and that further methodology needs to be developed for these problems.

This dissertation develops and formalizes a novel methodology for constructing orthonormal basis function classifiers. The methodology differs from those currently available in that it combines all of the elements required for applicability to multidimensional problems. The mean integrated squared error (MISE) measure of goodness-of-fit is well known within the pattern recognition literature (Tarter and Lock 1993). In this dissertation, it forms the foundation for a number of statistical decisions that are required for determining an orthonormal basis function model. The development of the relative MISE (RMISE) measure in this dissertation leads to new statistical tests for selecting a model from a set of potential models, determining the optimum complexity of a model, and selecting individual terms that can be removed from a model without adverse impact. All of these procedures may be conducted rapidly within a pattern recognition algorithm. This is important since a key reason for choosing orthonormal basis function networks over other multilayer perceptron models is the greater speed of fitting an orthonormal model. However, this simplicity translates into a

less parsimonious parameterization for nonlinear as opposed to linear models (Barron 1993; Barron 1994).

Another aspect of this work is benchmarking orthonormal basis function neural network models to identify their strengths and limitations with respect to existing classification methods. Currently there is insufficient scientific evidence to guide practitioners in determining whether orthonormal basis function models should be considered for particular types of problems. One reason for this is that current orthonormal basis function methodologies are not generally useful for applications to practical problems of more than two dimensions. Further investigation of the theoretical methodology should lead to greater applicability to classification models. For example, tools such as the University of Toronto's DELVE statistical suite for machine learning performance assessment (Rasmussen, Neal et al. 1996) and the University of California Irvine's repository of machine learning datasets (Hettich, Blake et al. 1998) provide standardized processes for testing machine learning methods. They enable standardized comparisons of test results for methods that are applicable to datasets similar to those in the repositories. Experiments employing these tools have been conducted to evaluate the new orthonormal basis function methodology. Results are discussed in this dissertation.

Chapter 2

ARTMAP Neural Networks for Land Use Change Classification

2.1 Introduction

Detecting and monitoring changes in conditions at the Earth's surface are essential for understanding human impact on the environment and for assessing the sustainability of development. In the next decade, NASA will gather high-resolution multispectral and multitemporal data, which could be used for analyzing long-term changes, provided that available methods can keep pace with the accelerating flow of information. This chapter introduces an automated technique, based on the ARTMAP neural network, for change identification. In addition to classifying land use changes from multitemporal, multispectral data, the system produces a measure of confidence in classification accuracy. Landsat thematic mapper (TM) imagery of the Nile River delta provides a testbed for these land use change classification methods. This dataset consists of a sequence of ten images acquired between 1984 and 1993 at various times of year. Field observations and photo interpretations have identified 358 sites as belonging to eight classes, three of which represent changes in land use over the ten-year period. A particular challenge posed by this database is the unequal representation of various land use categories: three classes, *urban*, *agriculture in delta*, and *other*, comprise 95% of pixels in labeled sites. A two-step sampling method enables unbiased training of the neural network system across sites.

ARTMAP systems belong to the adaptive resonance theory (ART) family of neural networks, which feature fast and stable learning. They benefit from an architecture that differentiates them from other neural networks such as multilayer perceptrons (MLPs). This architecture enables ARTMAP systems to retain memories without forgetting them when other data are presented (Carpenter, Gopal et al. 1999). These systems' parameters can converge to completely code an input vector in a single presentation without forgetting previously presented data. This fast learning limits the number of iterations required to fully train an ARTMAP neural network.

In addition to fast training, ARTMAP networks incorporate control mechanisms that enable them to create internal category representations that allow generalization within classes while ensuring that each training pixel is correctly classified (Carpenter, Gopal et al. 1999). This adaptive method of constructing category representations allows these neural networks to be applied to extended areas in which it is not known how well results for one site will generalize to other sites. Where generalization is not possible within the training set, multiple internal representations will be constructed to incorporate dissimilar sites.

ARTMAP neural networks have previously been shown to be effective tools for land cover classification of individual images (Carpenter, Gjaja et al. 1997; Carpenter, Gopal et al. 1999; Gopal, Woodcock et al. 1999). A straightforward extension of these networks to land cover change classification might first establish categorical classifications for each date. Postclassification comparisons of single-date class labels would then show how land cover had changed during the study period. Unfortunately,

such a straightforward method gives poor results, since errors in single-date classifications are compounded when multiple images are considered (Singh 1989). Abuelgasim et al. (Abuelgasim, Ross et al. 1999) introduced a Change Detection Adaptive Fuzzy (CDAF) network for environmental change detection and classification to monitor land cover changes resulting from the Persian Gulf War. This ARTMAP-based neural network compares images from multiple dates by assessing quantitative change in class likelihood or class intensity, rather than directly comparing class labels.

Multi-date classification combines spectral information from a series of dates to form multitemporal feature vectors. This method does not rely on single-date classifications, but rather differentiates constant land use from changing land use by direct application of a classifier algorithm to the multitemporal data. Multi-date classification has previously been implemented to detect land use change using the K-means technique (Abuelgasim, Ross et al. 1999). Muchoney and Williamson (2001) used multitemporal NDVI data as inputs to a Gaussian ARTMAP neural network that classifies land cover.

An advantage of multi-date classification is that images need not be taken under uniform seasonal, atmospheric, or illumination conditions, and sensor calibration need not be consistent across the sequence. Images are not compared directly to one another; rather, they are combined to form a rich database from which land use change patterns can be discovered. This is both a blessing and a curse, as the dimension of the feature vectors in the database increases with each date represented. The ARTMAP system is designed to deal effectively with high-dimensional data. It has been applied successfully

to generate models with less classification error than those previously available for problems involving hundreds of input features (Caudell, Smith et al. 1994; Rubin 1995), as well as to a number of remote sensing land cover classification problems (Carpenter, Gjaja et al. 1997; Abuelgasim, Ross et al. 1999; Carpenter, Gopal et al. 1999; Carpenter, Gopal et al. 1999; Gopal, Woodcock et al. 1999; Muchoney and Williamson 2001).

The multi-date ARTMAP classification method developed here and in Carpenter, Gopal et al. (2001) and Shock, Carpenter et al. (2002) extends single-date neural network land cover classification methods by using multitemporal, multispectral feature vectors derived from a sequence of ten satellite images as inputs to the neural network system. The ARTMAP change classification system overcomes inconsistencies by learning to identify the multi-date spectral signatures of image pixels. Using internal measures, it estimates confidence in classification accuracy. This is similar to decision trees that can give classification probability estimates (McIver and Friedl 2001).

2.2 Data

Ten Landsat TM images of the Nile River delta region and surrounding areas were taken at various times of year between 1984 and 1993. The images form the dataset used by Lenney et al. (Lenney, Woodcock et al. 1996) to classify land use changes based on characteristics of the multi-date NDVI vegetation index feature vector. The images were geometrically registered and normalized as described in that study. Field data were collected during the summer of 1993 at 88 sites in the study area. Ground truth labels for 270 additional sites were determined by expert image analysis at the Boston University Center for Remote Sensing. This information was combined to form a database of 358

sites. In order to make full use of the limited number of labeled sites, the present study employs four-fold cross-validation. To this end, the database was partitioned into four subsets, each containing 89, 90, or 91 sites. Each of the four subsets was then used, in turn, as a test set to evaluate the performance of an ARTMAP classifier which had been trained on the sites in the other three subsets. Carpenter et al. (1999) describe the use of such a cross-validation method to evaluate machine learning systems for remote sensing applications.

2.3 Method

2.3.1 Data preprocessing

Prior to performing model selection, input vectors were preprocessed. This preparation consisted of computing transformations and scaling each input component to the interval [0,1], which is the domain of Fuzzy ARTMAP inputs.

In order to investigate which input variables would be most useful for ARTMAP neural network identification of land use change categories, several feature sets were prepared using different transformations of the spectral data. Results of prior ARTMAP remote sensing applications suggested that auxiliary variables (pixel location coordinates and geographic zone designations) might also contribute to classification performance (Carpenter, Gjaja et al. 1997; Carpenter, Gopal et al. 1999). The transformed spectral data from multiple dates and auxiliary variables were concatenated to create multitemporal, multimodal input vectors for the neural network classifier.

2.3.2 Model selection

Cross-validation was used to select both a linear transformation of the input data and certain parameters of the ARTMAP model based on the transformed data. For each of the four training/testing partitions, input variable transformations were selected by cross-validated evaluation of three potential transformations. The Tasseled Cap transformation applied to each image (Table 2.1) gave the best performance of the transformations under consideration. This fixed transformation is desirable for many remote sensing tasks because of its similarity to PCA performed on Landsat TM images and dimension reduction to three variables that correspond closely to features of interest. Performance also improved when the Brightness, Greenness and Wetness (BGW) coefficients of this feature set were supplemented with geographic zone information and image pixel locations.

Input vector element	Description
1-3	Brightness, Greenness, and Wetness coefficients from June 7, 1984 image
4-6	Brightness, Greenness, and Wetness coefficients from September 11, 1984 image
7-9	Brightness, Greenness, and Wetness coefficients from June 10, 1985 image
10-12	Brightness, Greenness, and Wetness coefficients from December 22, 1986 image
13-15	Brightness, Greenness, and Wetness coefficients from August 21, 1988 image
16-18	Brightness, Greenness, and Wetness coefficients from August 3, 1990 image
19-21	Brightness, Greenness, and Wetness coefficients from February 19, 1991 image
22-24	Brightness, Greenness, and Wetness coefficients from June 13, 1992 image
25-27	Brightness, Greenness, and Wetness coefficients from April 29, 1993 image
28	Pixel *x*-coordinate
29	Pixel *y*-coordinate
30	Geographic region indicator for *delta* (Boolean)
31	Geographic region indicator for *desert* (Boolean)
32	Geographic region indicator for *coast* (Boolean)
33	Geographic region indicator for *wetlands* (Boolean)

Table 2.1: Inputs to the ARTMAP neural network classification system. Although ten images were available, the Brightness, Greenness, and Wetness (BGW) coefficients of the Tasseled Cap transformation could only be computed for nine dates due to a missing spectral band in one image.

Similarly, most parameters of the four neural network systems were determined by evaluation on the respective training sets (Table 2.2).

	Partition 1	Partition 2	Partition 3	Partition 4
$\bar{\rho}$ (baseline vigilance parameter)	0	0	0	0
α (choice parameter)	.0025	.001	.01	.001
V (number of voters)	3	2	5	4
Average number of training presentations of each site via representative pixels	190	70	106	62

Table 2.2: ARTMAP parameters determined by cross-validated evaluation on the training data for four partitions of the dataset. All systems used *a priori* the learning rate parameter β=1.0, match tracking control ε=-.001, and CAM decision rule power *p*=1.0. Instance counting (IC) was not enabled.

2.3.3 Training

Each ARTMAP network was trained by presenting a random sequence of pixels from the training subset. A major challenge encountered with this database was that the number of pixels in individual sites varied considerably, with training sites ranging in size from 4 to 3,440 pixels. It seemed that adequate representation required that small sites be adequately represented in the neural network training set while still exploiting information contained in all pixels of large sites. This goal was achieved via a two-step pixel sampling process. Each training pixel was determined by first selecting a random training site and then selecting a random pixel from that site to produce a sample unbiased with respect to the available sites. The duration of training was determined during the model selection phase.

2.3.4 Model testing (validation)

Multiple trained ARTMAP networks were combined to form a committee voting system to improve classification performance and stability (Bishop 1995). Combining

two or more networks in a committee and making a classification decision on the basis of the average output of these committee members improves the expected performance of neural network systems (Bishop 1995). The number of voting networks (V) was determined during parameter selection, with each voter weighted equally. The net vote for each class k was taken to be the average analog output across the V voters. A classification decision was made by selecting the class with maximum average output value.

The analog values assigned to pixels by the voting system may be thought of as estimates of their fuzzy membership in various classes. Averaging these values across all the pixels within a site gives membership estimates for the site. The system labels a four-pixel testing (validation) site as belonging to the class to which it attaches the greatest fuzzy membership value.

2.4 Results and discussion

The present analysis shows how an ARTMAP system can automate the classification of land use change from remote sensing data, producing the map shown in Figure 2.1. The *user's accuracy*, defined as the rate of correct classification of test set sites in the ground truth database, averaged 84.6% for the four systems (Table 2.3), compared to user's accuracy of 87.55% reported by Lenney et al. (1996). The *producer's accuracy*, which adjusts classification rates in proportion to the estimated true fractions of land use change categories in the map, averaged 86.4%, as estimated using the method of Card (Card 1982). During training, each neural network attempts to optimize user's accuracy, without knowledge of underlying class probabilities that might enable higher

producer's accuracy, such as the 95.85% obtained by Lenney et al. Note that Lenney et al. used a different, overlapping assessment dataset and different testing methodology, so these results are not directly comparable.

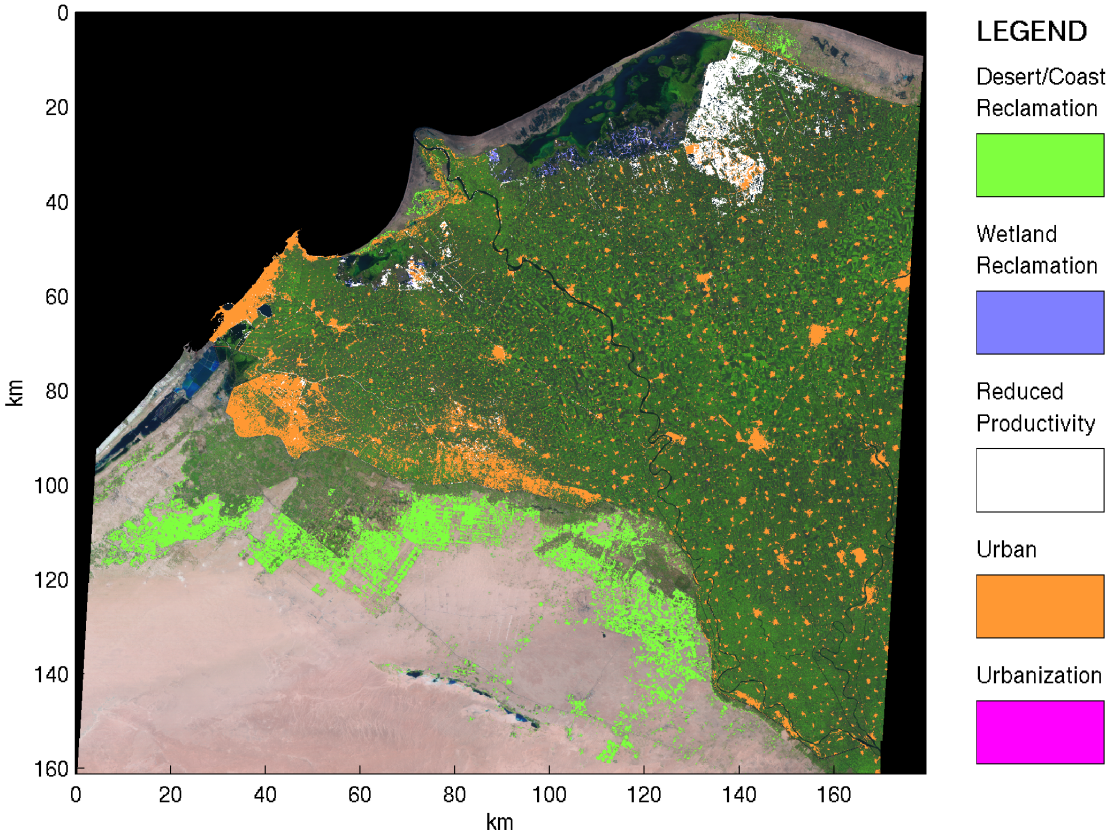

Figure 2.1: Composite map showing ARTMAP classifications of land use changes, after water had been separated from land via a linear threshold mask. Classes are superimposed on a false color image acquired in 1993. Four systems, each of whose performance has been determined by cross-validated testing, were combined to create this map.

	Partition 1	Partition 2	Partition 3	Partition 4	Mean of four partitions
User's accuracy (%)	89.9	85.4	84.3	79.1	**84.6**
Producer accuracy (%)	88.5	86.9	90.2	80.1	**86.4**

Table 2.3: Performance of the ARTMAP land use change classifier on four cross-validation partitions. The variability reflects sampling bias in the selection of training and validation sets. Four different systems were determined from the respective training sets using the same methodology; however, some parameters of these systems varied widely. Furthermore, the validation accuracy was limited by the number of sites available for this purpose, approximately 90 for each partition.

Confusion matrices (Table 2.4 and Table 2.5) provide details of system predictive accuracy for each of the nine output classes. Two of the land use change classes, *urbanization* and *wetlands reclaimed*, had insufficient data for training the neural network. In particular, the entire ground truth dataset included only three *wetlands reclaimed* sites. Not surprisingly, the learning systems consistently failed to identify these sites when they had not been seen at all during training. Like the NDVI-based classification system developed by Lenney et al. (1996), the ARTMAP classifier had substantial difficulty distinguishing between *urban* and *reduced productivity* classes. These classes have similar spectral signatures which are easily confused. The separability characteristics of these data are revisited in Chapter 6 of this dissertation.

		Field assessments								
Land use classifications	Sites	Urban	Urbanization	Reduced productivity	Agriculture in delta	Agriculture in desert/coast	Reclamation	Wetlands reclaimed	Other	User's accuracy
Urban	83	**63**	2	10	3		2		3	*75.9%*
Urbanization	1		**1**							*100.0%*
Reduced productivity	20	2	1	**17**						*85.0%*
Agriculture in delta	147	4	6	4	**132**				1	*89.8%*
Agriculture in desert/coast	15					**12**	1		2	*80.0%*
Reclamation	13					2	**10**		1	*76.9%*
Wetlands reclaimed	1								1	*0.0%*
Other	78					1	6	3	**68**	*87.2%*
Total	358	69	10	31	135	15	19	3	76	**Overall** ***84.6%***

Table 2.4: User's Accuracy Assessment: a composite of the performance of the ARTMAP land use change classifier on the four cross-validation partitions.

		Field assessments								
Land use classifications	Sites	Urban	Urbanization	Reduced productivity	Agriculture in delta	Agriculture in desert/coast	Reclamation	Wetlands reclaimed	Other	Map proportions
Urban	25	**5.055 %**		0.689 %						*5.744%*
Urbanization	0									
Reduced productivity	4			**1.804 %**						*1.804%*
Agriculture in delta	35		2.619 %		**43.214 %**					*45.833%*
Agriculture in desert/coast	4					**4.411 %**				*4.411%*
Reclamation	2					2.223 %	**2.223 %**			*4.446%*
Wetlands reclaimed	0									
Other	19						3.968 %	1.984 %	**31.742 %**	*37.694%*
True proportions		*5.055 %*	*2.619 %*	*2.493 %*	*43.214 %*	*6.634 %*	*6.191 %*	*1.984 %*	*31.742 %*	
Producer's accuracy		100.0 %	0.00 %	72.36 %	100.0 %	66.49 %	35.91 %	0.00 %	100.0 %	**Overall 88.45%**

Table 2.5: Producer's Accuracy Assessment: This performance assessment is for the system developed for the first cross-validation partition. Table 2.3 indicates that the performance on this partition is typical.

A benefit of using ARTMAP neural networks to generate land use change classification maps is that the confidence of classification decisions is readily available via the variables σ_k, which provide the system's class probability estimates. A map of classification confidence similar to Figure 2.2 thus accompanies each primary map of land use changes. Note in Figure 2.1 that large areas in the southwest quadrant of the study area are incorrectly classified by the ARTMAP system as *urban*. Figure 2.2 shows

that the ARTMAP system is least certain of its predictions in these regions. Identification of the areas in which the network's classifications are most likely to be incorrect could guide manual editing of a land use change map. These areas could also be used to guide collection of additional ground truth data.

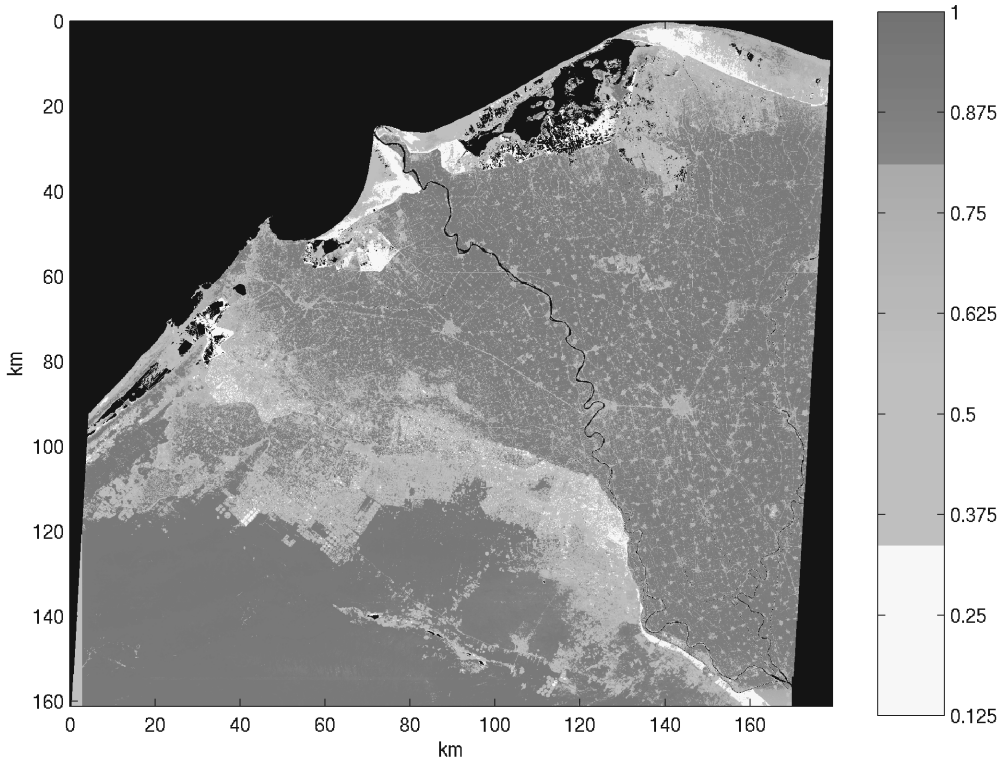

Figure 2.2: Composite map showing confidence of ARTMAP land use change classifications, with red indicating regions of lowest confidence. Four systems, each of whose performance has been determined by cross-validated testing, were combined to create this map. The confidence measure, which is based on ARTMAP output values, reflects the degree of system confusion between two or more classes.

A key feature of ARTMAP neural network classifiers is that large-scale datasets can be analyzed rapidly and automatically once enough sample field identifications have

been made to form the training set. No ARTMAP system in this study required more than 18,000 input vector presentations during training.

2.5 Conclusions

Like other change classification methods, the ARTMAP system presented in this chapter has attributes that recommend it for certain types of problems. In particular, the multi-date ARTMAP neural network classifier accepts high-dimensional spectral signatures containing features from a number of different dates. It produces both a land use change classification map and a confidence map, based on internal parameters, which can be used to evaluate the quality of the land use change classifications.

The methods described in this chapter are useful for identifying pixels that correspond to known types of land use and land use change in the image database. A second type of categorical change detection is the identification of new land cover classes, as discussed by Abuelgasim et al. (1999). The latter type of detection was not within the scope of this study but is a promising area for further application and analysis of multi-date neural network change detection systems.

Chapter 3

Orthonormal basis function neural networks for pattern classification

3.1 Introduction

The rest of this dissertation explores how organization of artificial neurons into an orthonormal frame simplifies the computations required to perform pattern classification tasks. To this end, this and subsequent chapters develop computational procedures that enable orthonormal basis function neural networks to be applied to a wide range of classification problems.

Classifiers that utilize orthonormal bases are known within the statistical literature, but their applicability is limited. Much work remains to be done to make orthonormal basis function classifiers viable options for multidimensional classification problems. Current methods for selecting a set of basis functions from a multidimensional tensor product basis typically require selection of a cutoff frequency *n*-tuplet (Devroye, Györfi et al. 1996). Such methods are inappropriate for problems of dimension greater than two.

A key contribution of this dissertation is the adaptation of rules that rely upon single-parameter cutoff determination to the selection of a set of basis functions from a multidimensional tensor product basis. This allows one-dimensional stopping methods to be applied to the construction of a pool of neurons that exploit the property of orthonormality to represent a multidimensional problem space. The stopping methods employed for basis function selection need to utilize a goodness-of-fit measure. For

reasons of convenience, an estimate of the mean integrated squared error (MISE), a common goodness-of-fit measure, is used for this work. It is important to note that there are methods for model selection that do not require a stopping point, such as the stepwise forward and backward regression used in certain additive spline fitting procedures (Friedman and Silverman 1989; Friedman 1991; Stone, Hansen et al. 1997). These were not pursued in this dissertation, although they provide an interesting direction for future work (7.2.1).

A second, related novel aspect of this dissertation is the combination of a stopping rule to determine an optimal set of basis function neurons to represent a particular problem with a single-term exclusion rule to remove from the set neurons that do not contribute sufficiently to system performance. Tarter and Lock (1993) have demonstrated that the use of a single-term inclusion rule, which individually selects terms to be included in a the orthonormal basis function model of a problem, results in an excess of basis functions. Given a preselected set of basis functions, however, single-term exclusion rules can be used to determine which of these basis functions do not individually reduce the error of a system. Excluding such terms reduces the number of parameters in a model and decreases the expected error. This is a new use of single-term criteria based on measures of goodness-of-fit such as the MISE.

The MISE is frequently used to fit and evaluate orthonormal basis function systems. Estimators of the MISE exist for density estimation using orthonormal systems, and similar computations can be used for classification using the orthonormal discriminant method presented by Devroye et al. (1996). Although other discriminant

functions and estimators may be used, the method of Devroye was selected for this work because of its simplicity and known asymptotic properties.

3.2 Background

Cencov (1962) introduced the use of Fourier series to represent probability density estimates. Other orthonormal series density estimation and classification methods use the same fundamental model. They differ in the bases used, such as the Daubechies wavelet bases and the discrete cosine basis; the functions being estimated, discriminant functions instead of density functions, for example; and the ways in which coefficients are modified or eliminated to control the complexity of a model. Tarter and Lock (1993) and Devroye et al. (1996) review many of these methods.

Specht (1971) made use of polynomial bases for probability density estimation. Greblicki (1978) proved the asymptotic efficiency of Fourier series density estimates and extended this result (Greblicki 1981) to the Hermite polynomial basis. Hall (1981) advocated a cosine series (DCT) estimator, and Diggle and Hall (1986) also mention the Legendre series as an option with similar properties to the trigonometric bases. Devroye et al. (1996) suggest other orthonormal bases that might be appropriate for pattern classification, including the standard trigonometric, Laguerre, Haar, Rademacher, and Walsh bases. Recent advances in wavelets offer such possibilities as the Daubechies D4 basis (Daubechies 1992; Strang 1993).

A problem that must be addressed to use orthonormal series density estimators as well as related classification methods is how to determine which of the infinitely many terms of a series estimator to include. A single-term inclusion rule was proposed by

Tarter et al. (1967; Kronmal and Tarter 1968; Tarter and Lock 1993). Using this rule, each term is considered individually for inclusion in an orthonormal series model based on its contribution to the overall mean integrated squared error (MISE). The insufficiencies of this rule were addressed by Hart (1985) and Diggle and Hall (1986), who independently developed stopping rules to determine the term at which a series estimator should be truncated. Efromovich (1999) uses a different estimator that is similar to these. All of the stopping rules select a stopping term for orthonormal series models by minimizing an estimate of the MISE. The orthonormal basis function classifiers investigated in this dissertation use truncation methods (stopping rules and a single-term exclusion rule) to select the basis functions that contribute to a model. It would also be possible to use coefficient shrinkage methods (Tibshirani 1996; Hastie, Tibshirani et al. 2001) to achieve this end. Such a use of shrinkage methods is discussed briefly as a direction for future work (Section 7.2.2).

Many of these authors have studied classification by taking ratios of orthonormal series probability density estimates for each class (Specht 1971; Greblicki 1978; Greblicki 1981; Greblicki and Pawlak 1981; Greblicki and Pawlak 1982; Greblicki and Pawlak 1983; Efromovich 1999). Devroye et al. (1996) introduced an alternative classification approach in which a discriminant function is directly estimated using orthonormal series expansions. For two-class problems, this has the potential to be more accurate than taking a ratio of two density estimates as this discriminant combines positive and negative class information in a single function.

3.3 Orthonormal basis function neural network architecture

Orthonormal basis function neural networks are based on the feedforward multilayer perceptron architecture. A three-layer architecture is employed by radial basis function networks and orthonormal basis function networks, among many systems. In this architecture, an input vector x is represented by corresponding nodes in an *input layer*. These nodes output the values of the input vector elements. A *hidden layer* of nodes implements a nonlinear mapping of the input vector elements. Typically a hidden layer node represents a nonlinear function $\varphi_j(x)$ of the input layer values. This function may be fixed, as it is for certain radial basis function neural networks (Haykin 1994) and for orthogonal basis function neural networks, or the function may be adaptive, as it is for backpropagation neural networks (Bishop 1995). An *output layer* consists of one or more nodes that compute network output functions by combining the results of the computations performed by the hidden layer nodes. Typically an output layer node implements a weighted sum

$$y_k(\mathbf{x}) = \sum_{j=1}^{n} w_{jk}\varphi_j(\mathbf{x}) \tag{3.1}$$

to perform this combination (Bishop 1995). In this equation, w_{jk} is the weight assigned to the connection between hidden layer node j and output layer node k. It is common for the output layer nodes to pass the result of the weighted sum through a nonlinear function, in which case Equation (3.1) is instead written

$$y_k(\mathbf{x}) = \phi_k\left[\sum_{j=1}^{n} w_{jk}\varphi_j(\mathbf{x})\right]. \tag{3.2}$$

A common problem with models such as (3.1) and (3.2) is that the high-dimensional nonlinear transformation $\{\varphi_j(\mathbf{x}), j = 1, 2, \ldots\}$ may make fitting the models a computationally intensive and time-consuming task. This is especially true of backpropagation neural networks, which can require tens of thousands of iterations to converge to a stable model (Kooperberg and Stone 1999); however, radial basis function neural networks and generalized additive models also suffer from computational difficulties imposed by nonlinear fitting.

A promising approach to building feedforward models is to use orthonormal basis functions for hidden layer nodes. Special properties of orthonormal bases enable all parameters of a model of this type to be determined independently of other parameters using fast linear computations provided that the model fits an appropriate objective function, such as the MISE. Other models using the same architecture require iterative computations or matrix inversion operations to solve equations of many dependent parameters in the model fitting process. From a computational efficiency standpoint, orthonormal basis function neural networks offer significant speed improvements over other similar models.

The next section discusses the properties of orthonormal series expansions in general, including the properties that lead to this favorable result.

3.4 Orthonormal series expansions

3.4.1 Properties of orthonormal bases on the domain of x

An orthonormal basis $\{\varphi_j(\mathbf{x}),\ j=1,2,\ldots\}$ has three fundamental properties on the domain Ξ over which $\mathbf{x}$ is defined. These properties hold whether Ξ is unidimensional or multidimensional and whether Ξ is bounded or unbounded. First, the norm of every function $\varphi_j(\mathbf{x})$ is unity. For a basis in L_2, this means that

$$\int_{\Xi} \varphi_j(\mathbf{x})^2\, d\mathbf{x} = 1 \quad (\forall j). \tag{3.3}$$

Second, the functions $\{\varphi_j(\mathbf{x})\}$ form an orthogonal set. In L_2, this implies that

$$\int_{\Xi} \varphi_j(\mathbf{x})\varphi_k(\mathbf{x})\, d\mathbf{x} = 0 \quad (\forall j,k \mid j \neq k). \tag{3.4}$$

Finally, if $\psi(\mathbf{x})$ is a function in L_2 then there exists a sequence of weights $\{a_j\}$ such that

$$\lim_{j\to\infty} \int_{\Xi} \left[\psi(\mathbf{x}) - \sum_{k=1}^{j} a_j \varphi_j(\mathbf{x}) \right]^2 d\mathbf{x} = 0 \tag{3.5}$$

3.4.2 Orthonormal bases considered for pattern classification

3.4.2.1 Discrete cosine basis

The univariate orthonormal discrete cosine basis on the interval $0 \le x \le 1$ is given by the sequence:

$$\begin{aligned} &\varphi_1(x) = 1 \\ &\varphi_j(x) = \sqrt{2}\cos[(j-1)\pi x], \quad j \ge 2 \end{aligned} \tag{3.6}$$

The first nine terms of this sequence are shown in Figure 3.1.

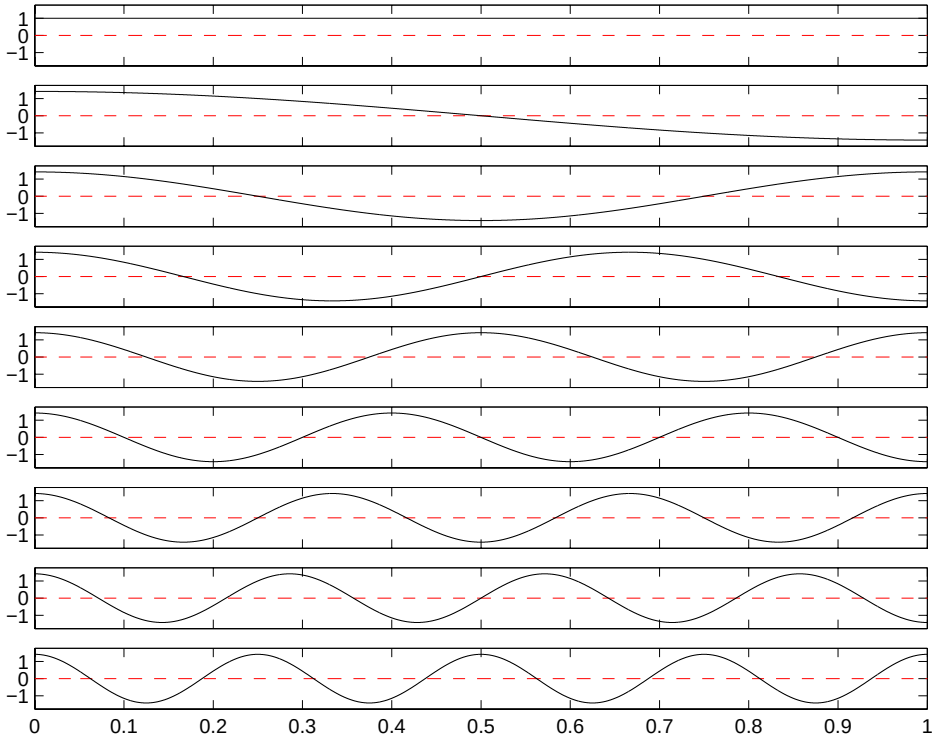

Figure 3.1: First nine functions of the discrete cosine basis

3.4.2.2 Legendre polynomial basis

The Legendre polynomial basis offers a representation in which linear, quadratic, and higher-order polynomial relationships require a finite number of nonzero coefficients. This is intuitively useful for data that may have strong linear or quadratic trends. Like the cosine basis, the Legendre basis has support over the entire interval.

The orthonormal series of Legendre polynomials on $[-1,1]$ is given by the Rodrigues representation:

$$P_l(x) = \frac{1}{2^l l!}\frac{d^l}{dx^l}\left(x^2 - 1\right)^l \tag{3.7}$$

(Gradshteyn, Rhyzik et al. 1994; Weinstein 1999). To conform to the conventions of this dissertation, the orthonormal Legendre polynomials on the unit interval may be constructed by translation and rescaling of this series, and the constant term may be included:

$$\begin{aligned} \varphi_1(x) &= 1 \\ \varphi_j(x) &= \sqrt{2j+1}P_{j-1}(2x-1), \quad j = 2,3,4,\ldots \end{aligned} \tag{3.8}$$

The first nine terms of this sequence are shown in Figure 3.2.

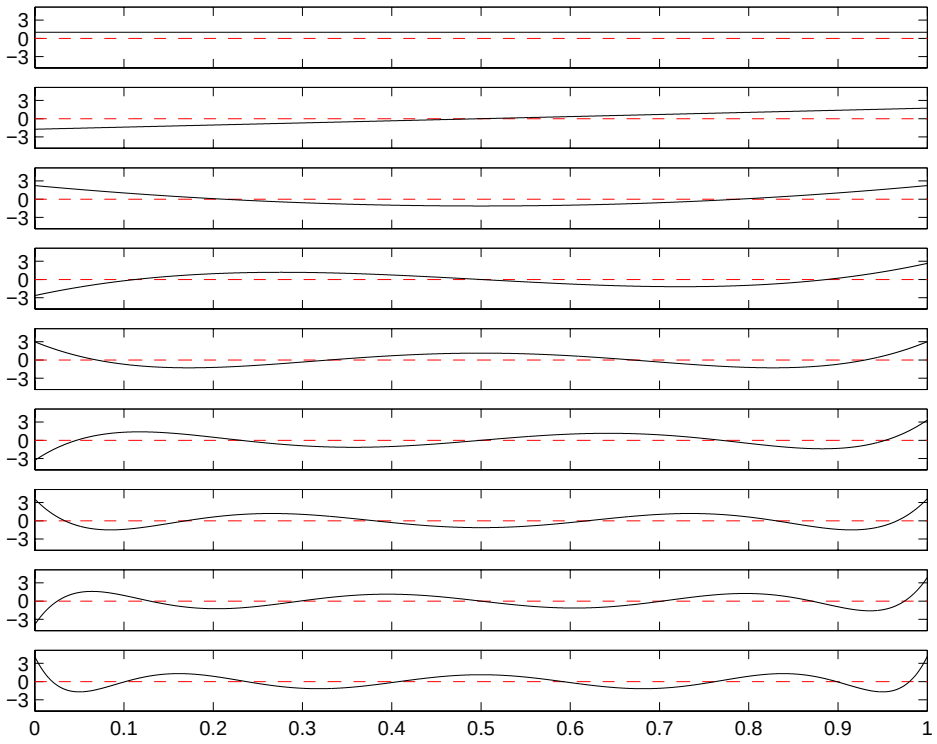

Figure 3.2: First nine functions of the Legendre polynomial basis

3.4.2.3 Haar wavelet basis

The Haar wavelets are shifted and rescaled variants of the function

$$\phi(x) = \begin{cases} 1 & 0 \le x \le \frac{1}{2} \\ -1 & \frac{1}{2} < x \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

Let

$$\phi_{k,l}(x) = \phi\left(2^k x - l\right), \quad k = \{0,1,2,\ldots\}, \quad l = \{0,1,\ldots,2^k - 1\} \tag{3.10}$$

(Strang 1993; Weinstein 1999). These and the constant function form the univariate Haar basis on the interval $[0,1]$. Normalizing, converting the double subscript k,l to a single subscript j, and including the constant function as the first term yields the sequence:

$$\begin{aligned} &\varphi_1(x) = 1 \\ &\varphi_j(x) = \left(\sqrt{2}\right)^k \phi_{k,l}(x), \quad j \ge 2, \quad \text{where} \\ &k = \lfloor \log_2(j-1) \rfloor \\ &l = (j-1) - 2^k \end{aligned} \tag{3.11}$$

The first nine terms of this univariate Haar basis on $[0,1]$ are shown in Figure 3.3.

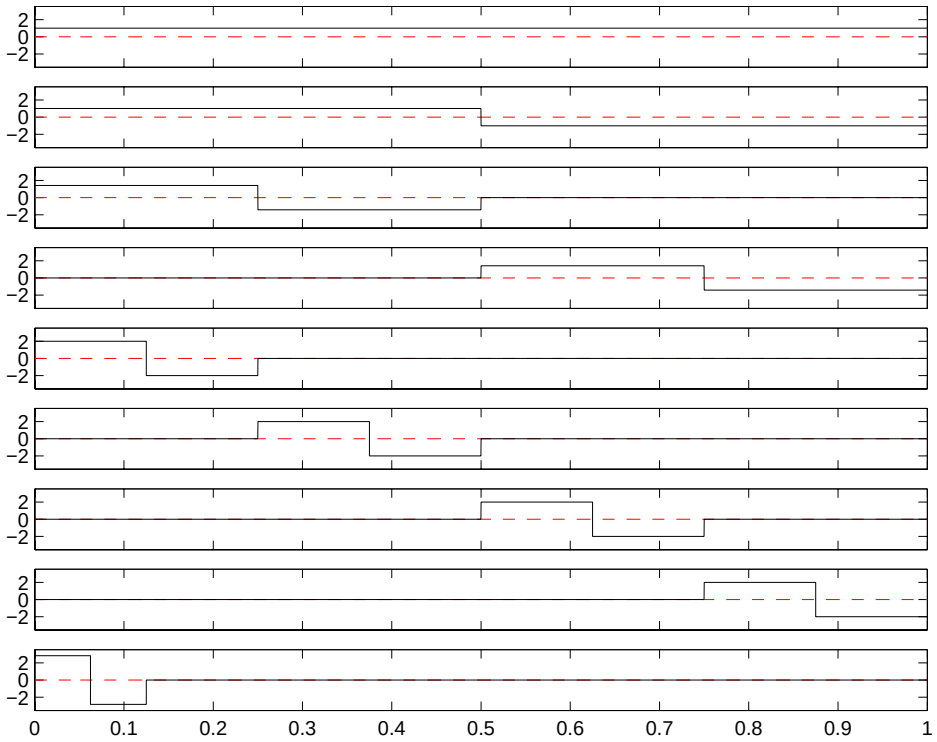

Figure 3.3: First nine functions of the Haar wavelet basis

As the Haar wavelets have local support, they are intuitively best for representing functions with local variations. In a classification problem, this might be the case if local regions of the feature space need to be subdivided between multiple classes.

3.4.2.4 Daubechies D4 wavelet basis

Another orthonormal wavelet basis with local support is the Daubechies D4 wavelet basis (Daubechies 1992). Like the Haar basis, this should be well suited for tracking classification functions that vary locally; however, the D4 basis presents a potential advantage. The Daubechies wavelet has two vanishing moments (Strang and Nguyen 1997):

$$\int_{-\infty}^{\infty} w(x)dx = 0$$
$$\int_{-\infty}^{\infty} xw(x)dx = 0 \tag{3.12}$$

All of the wavelet functions are orthogonal to both a constant and a linear term, so linear global characteristics of a classification function may be combined with the local characteristics represented by wavelets.

These wavelets are *multiresolution*, meaning that a basis is formed by a single mother wavelet function $w(x)$ and the set of all dilated and translated copies of this function $\left\{w\left(2^j x-k\right) \mid \forall j,k\right\}$. Let

$$W_{j,k}(x) = w\left(2^j x-k\right) \tag{3.13}$$

The *mother wavelet* function $W_{0,0}(x)$ is normally defined on the interval $[-1,2]$. To form an orthonormal basis on the interval $[0,1]$, let

$$\psi_{0,0}(x) = \frac{\sqrt{3}W_{0,0}(3x-1)}{\sqrt{\int_{-1}^{2}\left[W_{0,0}(3x-1)\right]^2 dx}} \tag{3.14}$$

be the mother wavelet $W_{0,0}$ translated to $[0,1]$ and normalized. The wavelets with support on this interval are then

$$\psi_{j,k}(x) = \psi_{0,0}\left(2^j\left(x-\frac{1}{3\cdot 2^j}k\right)\right), \tag{3.15}$$

where $j = 0,1,2,\ldots$ and $k = 0,1,\ldots,3\cdot 2^j - 3$.

The Daubechies D4 wavelet basis (Daubechies 1992), along with its two vanishing moments, is incorporated in the following sequence of orthonormal basis functions on the interval $[0,1]$:

$$\begin{aligned}
&\varphi_1(x) = 1 \\
&\varphi_2(x) = 2\sqrt{3}\left(x - \frac{1}{2}\right) \\
&\varphi_3(x) = \psi_{0,0}(x) \\
&\varphi_4(x) = \psi_{1,0}(x) \\
&\varphi_5(x) = \psi_{1,1}(x) \\
&\varphi_6(x) = \psi_{1,2}(x) \\
&\varphi_7(x) = \psi_{1,3}(x) \\
&\varphi_8(x) = \psi_{1,4}(x) \\
&\varphi_9(x) = \psi_{2,1}(x) \\
&\vdots
\end{aligned} \qquad . \tag{3.16}$$

The first nine functions of this sequence are shown in Figure 3.4.

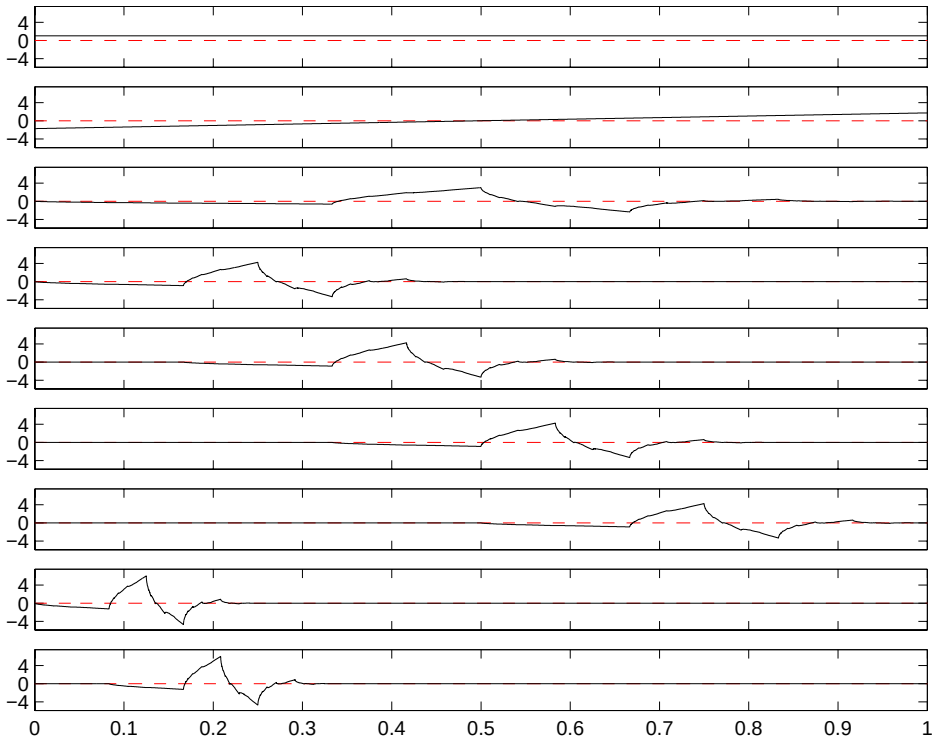

Figure 3.4: First nine functions of the second-order (D4) Daubechies wavelet basis in one dimension

3.4.3 Tensor product construction of a multivariate orthonormal basis

A multivariate orthonormal basis can be constructed from univariate bases by taking the tensor product of the vectors of univariate basis functions. A bivariate tensor-product basis $\{\varphi_{m,n}(x,y), \quad m,n=1,2,\ldots\}$, for example, can be made from the univariate bases $\{\phi_m(y), m=1,2,\ldots\}$ and $\{\psi_n(z), n=1,2,\ldots\}$ as follows (Efromovich 1999):

$$\{\varphi_{m,n}(y,z) \coloneqq \phi_m(y)\psi_n(z), \quad m,n=1,2,\ldots\} \tag{3.17}$$

It becomes difficult to index subscripts for elements of bases in more than two dimensions. Although it is possible to denote the elements of a tensor-product basis by

their multiple subscripts, this notation is often cumbersome. It is possible to map the functions in a tensor-product basis to univariate indices in a one-to-one manner.

$$\{\varphi_{m,n,\ldots}(y,z,\ldots),\quad m,n,\ldots=1,2,\ldots\} \tag{3.18}$$

can simply be written

$$\{\varphi_j(\boldsymbol{x}),\quad j=1,2,\ldots\}, \tag{3.19}$$

where

$$\boldsymbol{x}=\langle y,z,\ldots\rangle .$$

Employing the vector notation $\boldsymbol{x}$ for the multivariate function arguments simplifies the mathematical notation and highlights the parallels with univariate orthonormal series expansions. The notation in (3.19) is not dependent on the dimensionality of the problem and can represent orthonormal bases for problems of arbitrary dimensionality. Such a mapping of indices from a vector to a scalar does not specify the ordering of basis functions with respect to the scalar index. Because the series will be truncated, the actual ordering of terms is an important consideration. The goal of ordering the components is to assign low indices to components that are likely to be useful so they do not fall beyond the truncation point and hence out of a model.

3.5 Basis function selection utilizing stopping rules

Certain problems naturally arise when using orthonormal series expansions. Foremost among these problems is determining after how many terms a series expansion should be truncated. In an orthonormal basis function neural network, this determines the

complexity of the basis function layer. This section discusses stopping rules used in this dissertation.

Because the basis functions utilized are multidimensional, the approach considered here will be applied to complexity measures developed in Section 3.6.

3.5.1 Univariate stopping rules

One-dimensional series expansions require a single truncation point. In a trigonometric series expansion, the truncation point determines the degree of smoothing of the fit to the underlying data (Devroye, Györfi et al. 1996). Including an excessive number of terms will result in a model with an overfit, whereas including an insufficient number of terms will result in a model with an underfit.

Univariate stopping rules based on estimates of the MISE have been well studied in the statistical literature. Hart (1985) and Diggle and Hall (1986) both developed unbiased estimators of the MISE for Fourier series density estimates. All terms up to and including a cutoff term T are included in the estimator after the stopping rule is implemented. T is selected such that it minimizes an unbiased estimator of the MISE; in practice, this involves a line search of potential stopping points (Tarter and Lock 1993).

An MISE-based estimator for classifier systems, such as that introduced in this dissertation, can serve the same purpose for orthonormal basis function classifiers. For a one-dimensional problem, a cutoff point could be determined by choosing the set of terms, inclusive of all terms up to a given frequency, which minimizes the estimated MISE. This is well-defined for trigonometric series such as the discrete cosine transform

(DCT). It is problematic for other orthonormal series for which frequency is not as meaningful.

3.5.2 Multivariate stopping rules

Multivariate orthonormal series expansions require a more elaborate formulation of stopping rules. Efromovich's approach (1999) is typical. He selects a pair of stopping frequencies for a bivariate problem. This leads to a rectangular frequency window, which will contain low-frequency basis functions at one corner and high-frequency basis functions at the opposite corner. For a high-dimensional problem, selecting an ordered *n*-tuplet of stopping frequencies specifies a hyperrectangular frequency window that consists almost entirely of high-frequency basis functions that are the tensor products of one-dimensional low-frequency basis functions. The complexity of such a model is evidence that this method is particularly susceptible to the curse of dimensionality. It is clear that other approaches to selecting multidimensional frequency cutoffs need to be considered.

This dissertation examines several possible methods of selecting basis functions in a multidimensional frequency space. These include a method developed for this dissertation that assigns tensor product basis functions to frequency classes according to the product of the indices of their component one-dimensional basis functions, resulting in a hyperbolic frequency cutoff with a single cutoff parameter. Other possibilities include spherical and linear additive frequency cutoffs.

3.6 Scalar indexing of multidimensional tensor product bases

Most authors to date have studied orthonormal series density estimation and classification using a rectangular window. A stopping rule is used to determine either a maximum index value M_d for each dimension d or a single maximum index value M for all dimensions. While these approaches may work from problems of low dimensionality, they are not appropriate when the dimensionality can cause an exponential explosion in the number of coefficients. Moreover, some of the tensor product basis functions included in a rectangular window are enormously complex, involving many interacting terms in different dimensions. Comparatively simple basis functions of index $m_d = \mathrm{M}_d + 1$ or $m_d = \mathrm{M} + 1$ are excluded.

This section suggests ways in which multidimensional tensor product basis function indices can be mapped to a unidimensional index j. The aim is to construct a mapping that enables the application of unidimensional stopping rules and includes tensor product basis functions in order of complexity. Such mappings are dependent on the definition of complexity used. Each of the following primary ordering criteria implements a plausible measure of complexity. When mapping tensor product basis functions to a unidimensional index, if two functions have a different primary criterion value, the function with the higher value will always be assigned a higher unidimensional index j.

3.6.1 Linear complexity criterion

The linear ordering is applicable to both trigonometric and polynomial bases. It uses as its primary criterion r the sum of the multidimensional indices:

$$r = \sum_{d=1}^{D} m_d \tag{3.20}$$

For a polynomial basis, this implements a familiar complexity measure, the degree of a multivariate polynomial, plus a constant equal to the number of dimensions.

3.6.2 Spherical complexity criterion

The spherical ordering uses as the primary criterion:

$$r = \sum_{d=1}^{D} \left(m_d - 1 \right)^2 \tag{3.21}$$

This corresponds to the coefficients of the tensor product of trigonometric polynomials after application of the Laplacian.

3.6.3 Hyperbolic complexity criteria

3.6.3.1 Definition of zero-crossing order

Let the zero-crossing order (ZC order) of a univariate basis function with support on the interval be the number of subintervals separated by zero crossings, so that the ZC order is equal to one plus the number of zero crossings in the interval. For example, the constant basis function has ZC order one, and the half-cosine basis function has ZC order two. The Haar mother wavelet also has ZC order two.

3.6.3.2 Hyperbolic complexity criterion for trigonometric and polynomial bases

The hyperbolic ordering differs from the linear ordering in the primary criterion used. Instead of the sum of indices, the primary criterion is the product of indices:

$$r = \prod_{d=1}^{D} m_d \, . \tag{3.22}$$

Provided that the ZC order for unidimensional basis functions is equal to their index values, this is identically the product of the ZC orders of the component functions:

$$r = \prod_{d=1}^{D} Z_d \, , \tag{3.23}$$

where Z_d is the ZC order of the tensor product component in the d th dimension. The primary criterion in Equation (3.23) is equal to the number of regions of alternating sign separated by zero crossings in the tensor product basis function.

3.6.3.3 Hyperbolic complexity criterion for wavelet bases

For wavelet bases, the hyperbolic complexity criterion needs to reflect that the contraction operation on a wavelet effectively doubles its complexity by halving its support. Two daughter wavelets placed side-by-side would, when combined, have the same support as their mother wavelet but would have twice as many zero crossings. A multiplicative complexity measure should therefore be inversely related to the support of a basis function:

$$r = \prod_{d=1}^{D} \frac{Z_d}{S_d} \, , \tag{3.24}$$

where S_d is the length of the interval of support for the tensor product basis function component in the d th dimension. Equation (3.24) is also applicable to the special case

of (3.23) in which all basis functions have support over the entire unit interval. Treatment of the ZC order of a univariate wavelet function is thus analogous to that defined for trigonometric and polynomial bases above.

Some wavelets, such as the Daubechies D4 wavelet, have a large number of zero crossings, while effectively partitioning the interval into a small number of subintervals of alternating sign. The Daubechies D4 wavelet, for example, has four clearly identifiable subintervals of alternating sign, and none of the remaining subintervals approach these four in length or amplitude. Assigning the Daubechies D4 mother wavelet a ZC order of four more accurately reflects its capacity to separate classes than counting the actual number of zero crossings. The Daubechies D4 wavelet was therefore treated in this dissertation as if it had a ZC order of four. The practical impact of this is to determine the relative complexity of the linear and mother wavelet terms.

3.6.4 Ordering of functions with the same primary complexity criterion

Within a class of functions with the same primary criterion value, a secondary criterion is used to determine the ordering. If each variate were equally likely to be informative, it might be arbitrary to impose an ordering on basis functions of the same complexity. Assuming that the first variate is the most useful for classification and the last variate is the least useful, a condition that using canonical variates as in Section 4.2.3 aims to meet, it may be preferable first to include tensor product basis functions that represent the greatest complexity within the first variate. Within a class of tensor product basis functions with the same primary criterion, therefore, reverse lexicographic ordering is used. This assigns the lowest unidimensional indices to tensor product basis functions

with the highest first dimension indices m_1, representing the greatest complexity along that dimension.

3.7 An MISE-based measure for Devroye's discriminant method

The method of orthonormal series expansions has been employed extensively for density estimation. As reviewed by Devroye, Györfi et al. (1996), the method was initially developed by Cencov (1962) and further developed by numerous authors. Tarter and Lock (1993) and Efromovich (1999) have written books that form a comprehensive resource for the practitioner who desires to implement orthonormal series expansions for density estimation. They provide limited insights into how similar methods may be applied to multivariate problems and classification problems. For instance, Efromovich shows how separate density estimators for each subpopulation corresponding to a particular class may be compared to make a classification decision.

3.7.1 Model

It is assumed throughout that $\{\varphi_j(\mathbf{x}), j = 1, 2, \ldots\}$ form a basis in L2. The random variable or vector X and the random variable Y are assumed to be independent and identically distributed (i.i.d.).

3.7.2 Devroye's discriminant function

Devroye et al. (1996) present an elegant discriminant function that, for a two-class problem, can be estimated by a single function of all of the available data values. Devroye's discriminant function is given by

$$\alpha(\mathbf{x}) = p(\mathbf{x})\left[2P(Y=1|\, X=\mathbf{x})-1\right], \tag{3.25}$$

where X is a random variable or vector and Y is the random variable for the class associated with the data X, and $p(\mathbf{x})$ is the probability density of X at $\mathbf{x}$. Note that, since

$$p(\mathbf{x}) \geq 0 \quad (\forall \mathbf{x}),$$

the discriminant function takes on positive values only where

$$P(Y=1|\, X=\mathbf{x}) > P(Y \neq 1|\, X=\mathbf{x}). \tag{3.26}$$

This leads to the discrimination rule:

$$y^*(\mathbf{x}) = \begin{cases} 1 & \text{if } \hat{\alpha}(\mathbf{x}) > 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.27}$$

where $\hat{\alpha}(\mathbf{x})$ is an estimate of Devroye's discriminant function $\alpha(\mathbf{x})$.

Devroye et al. prove certain statistical properties of an estimator of $\alpha(\mathbf{x})$, but as their work on this subject is limited to discrimination of two classes, they do not present a multivariate classification system that uses this discriminant function and estimator as its foundation. Developing this estimator and its associated statistical properties into a usable method for fitting and selecting orthonormal basis function neural network models is the scope of this section.

3.7.3 Mean of an orthonormal series-based classifier

The coefficients for the orthonormal expansion of Devroye's discriminant function $\alpha(\mathbf{x})$ using the basis $\{\varphi_j(\mathbf{x}),\, j=1,2,\ldots\}$ are given by

$$a_j = \int_{\Xi} \varphi_j(\mathbf{x})\alpha(\mathbf{x})\,d\mathbf{x}. \tag{3.28}$$

This has the estimator (Devroye, Györfi et al. 1996)

$$\hat{a}_j = \frac{1}{n}\sum_{i=1}^{n}[2y_i - 1]\varphi_j(\mathbf{x}_i), \tag{3.29}$$

where

$$y_i = \begin{cases} 1 & \text{if } i \in \text{Class 1} \\ 0 & \text{otherwise} \end{cases} \tag{3.30}$$

and N is the number of samples.

The estimated expansion coefficients $\hat{a}_j$ are unbiased estimators of the true expansion coefficients a_j:

$$\begin{aligned} E[\hat{a}_j] &= E\left[\frac{1}{n}\sum_{i=1}^{n}[2y_i - 1]\varphi_j(\mathbf{x}_i)\right] \\ &= \int_{\Xi} E[2Y - 1 \mid X = \mathbf{x}]\varphi_j(\mathbf{x})\,p(\mathbf{x})\,d\mathbf{x} \\ &= \int_{\Xi} [2P(Y = 1 \mid X = \mathbf{x}) - 1]\,p(\mathbf{x})\varphi_j(\mathbf{x})\,d\mathbf{x} \\ &= \int_{\Xi} \alpha(\mathbf{x})\varphi_j(\mathbf{x})\,d\mathbf{x} \\ E[\hat{a}_j] &= a_j \end{aligned} \tag{3.31}$$

3.7.4 Variance of $\hat{a}_j$

The variance of $\hat{a}_j$ can be estimated from the sample variance.

Let

$$u_{ji} = 2(y_i - 1)\varphi_j(\mathbf{x}_i) \tag{3.32}$$

Then

$$Var(\hat{a}_j) = \frac{1}{n}\sum_{i=1}^{n} Var(u_{ji}) \tag{3.33}$$

$Var(u_{ji})$ has the unbiased estimator derived from the sample variance (Mendenhall, Wackerly et al. 1990):

$$\begin{aligned} \widehat{Var}(u_{ji}) &= \frac{1}{n-1}\sum_{i=1}^{n}(u_{ji} - \overline{u_{ji}})^2 \\ \widehat{Var}(u_{ji}) &= \frac{1}{n-1}\sum_{i=1}^{n}(u_{ji} - \hat{a}_j)^2 \end{aligned} \tag{3.34}$$

Thus,

$$\widehat{Var}(\hat{a}_j) = \frac{1}{n}\frac{1}{n-1}\sum_{i=1}^{n}(u_{ji} - \hat{a}_j)^2 \tag{3.35}$$

is an unbiased estimator for $Var(a_j)$.

3.7.5 The expected value of $\hat{a}_j^2$

$$\begin{aligned} E(\hat{a}_j{}^2) &= E\left[\left(\frac{1}{n}\sum_{i=1}^{n} u_{ji}\right)\left(\frac{1}{n}\sum_{l=1}^{n} u_{jl}\right)\right] \\ &= E\left[\left(\frac{1}{n^2}\sum_{i=1}^{n}\sum_{\substack{l=1\\ l=i}}^{n} u_{ji}{}^2\right) + \left(\frac{1}{n^2}\sum_{i=1}^{n}\sum_{\substack{l=1\\ l\neq i}}^{n} u_{ji}u_{jl}\right)\right] \\ &= E\left[\frac{1}{n^2}\sum_{i=1}^{n} u_{ji}{}^2\right] + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{\substack{l=1\\ l\neq i}}^{n} E[u_{ji}]E[u_{jl}] \\ &= E\left[\frac{1}{n^2}\sum_{i=1}^{n}(2y_i - 1)^2\varphi_j(x_i)^2\right] + \frac{(n)(n-1)}{n^2}a_j{}^2 \end{aligned}$$

since the u_{ji} are assumed to be i.i.d.

$$E(\hat{a}_j{}^2) = E\left[\frac{1}{n^2}\sum_{i=1}^{n}(2y_i - 1)^2\varphi_j(\mathbf{x}_i)^2\right] + \frac{n-1}{n}a_j{}^2 \tag{3.36}$$

Since $(2y_i - 1)^2 = 1 \quad (\forall y_i)$,

$$E(\hat{a}_j{}^2) = E\left[\frac{1}{n^2}\sum_{i=1}^{n}\varphi_j(\mathbf{x}_i)^2\right] + \frac{n-1}{n}a_j{}^2 \tag{3.37}$$

$$E(\hat{a}_j{}^2) = \frac{1}{n}\int_{\Xi}\varphi_j(\mathbf{x})^2\, p(\mathbf{x})d\mathbf{x} + \frac{n-1}{n}a_j{}^2 \tag{3.38}$$

By construction, this has the unbiased estimator:

$$\hat{E}(\hat{a}_j{}^2) = \frac{1}{n^2}\sum_{i=1}^{n}\varphi_j(\mathbf{x}_i)^2 + \frac{n-1}{n}a_j{}^2 . \tag{3.39}$$

This may not be used to estimate $E(\hat{a}_j{}^2)$ if a_j is unknown, but it proves useful in estimating $a_j{}^2$.

3.7.6 An estimator for $a_j{}^2$

Efromovich (1999) shows a useful technique for unbiased estimation of the squares of coefficients. This technique is readily adapted for orthonormal series based classification. $a_j{}^2$ can be stated in terms of $Var(\hat{a}_j)$ and $E(\hat{a}_j{}^2)$. From the unbiased estimators for both of these terms follows an unbiased estimator for $a_j{}^2$.

$$\begin{aligned} Var(\hat{a}_j) &= E(\hat{a}_j{}^2) - E(\hat{a}_j)^2 \\ &= \left[\frac{1}{n}\int_{\Xi}\varphi_j(\mathbf{x})^2\, p(\mathbf{x})d\mathbf{x} + \frac{n-1}{n}a_j{}^2\right] - a_j{}^2 \\ Var(\hat{a}_j) &= \frac{1}{n}\int_{\Xi}\varphi_j(\mathbf{x})^2\, p(\mathbf{x})d\mathbf{x} - \frac{1}{n}a_j{}^2 \end{aligned}$$

$$a_j{}^2 = \int_{\Xi}\varphi_j(\mathbf{x})^2\, p(\mathbf{x})d\mathbf{x} - n\,Var(\hat{a}_j) \tag{3.40}$$

Since unbiased estimators for both right-hand terms of Equation (3.40) are known, it is possible to construct an unbiased estimator for a_j^2 by substituting the estimators for these terms:

$$\widehat{a_j^2} = \tfrac{1}{n}\sum_{i=1}^{n}\varphi_j(\mathbf{x}_i)^2 - n\widehat{Var}(\hat{a}_j) \tag{3.41}$$

3.7.7 Mean integrated squared error (MISE) of an orthonormal series-based classifier

Using the uniform weighting function as in Tarter and Lock (1993), the mean integrated squared error (MISE) of the estimator $\hat{\alpha}(\mathbf{x})$ is defined as:

$$MISE\left[\hat{\alpha}(\mathbf{x})\right] = E\int_{\Xi}\left[\alpha(\mathbf{x}) - \hat{\alpha}(\mathbf{x})\right]^2 d\mathbf{x}\,. \tag{3.42}$$

By Parseval's Identity (Papoulis 1987), this can be written in terms of the expansion coefficients:

$$MISE\left[\hat{\alpha}(\mathbf{x})\right] = \sum_{j=1}^{\infty}\left(a_j - \hat{a}_j\right)^2 \tag{3.43}$$

3.7.8 Variance and squared bias terms of $MISE_W$

Let $W = \{w_1, w_2, \ldots\}$ be a set of zero-one weights that determine whether corresponding terms of the expansion $\{\hat{a}_1, \hat{a}_2 \ldots\}$ contribute to the estimate

$$\hat{\alpha}_W(\mathbf{x}) = \sum_{j=1}^{\infty} w_j \hat{a}_j \varphi_j(\mathbf{x}) \tag{3.44}$$

after a truncation of terms, where $w_j = 1$ indicates that the *j*th term of the expansion is included in the estimate, and $w_j = 0$ indicates that the *j*th term is excluded. For example,

W could result from truncating a Fourier or related series at a particular frequency, setting the w_j corresponding to higher frequencies to zero. Define

$$MISE_W = MISE\left[\hat{\alpha}_W(\mathbf{x})\right]. \tag{3.45}$$

Then

$$MISE_W = \sum_{j=1}^{\infty} Var\left(w_j \hat{a}_j\right) + \sum_{j=1}^{\infty} \left(1 - w_j\right) a_j^2 . \tag{3.46}$$

Determining the MISE would require the estimation of infinitely many terms in Equation (3.46). Although it is impossible to compute a full infinite series estimate of the MISE, a relative MISE measure can be computed quite readily for finite sets J. This relative measure can be used as a tool for model selection. If the relative MISE is defined as

$$RMISE_W = MISE_W - MISE_{\{w_j = 0,\ j = 1,2,\ldots\}}, \tag{3.47}$$

then

$$\begin{aligned} RMISE_W &= \sum_{j=1}^{\infty} w_j Var\left(\hat{a}_j\right) + \sum_{j=1}^{\infty} \left(1 - w_j\right) a_j^2 - \sum_{j=1}^{\infty} a_j^2 \\ RMISE_W &= \sum_{j=1}^{\infty} w_j \left[Var\left(\hat{a}_j\right) - a_j^2\right] \end{aligned} . \tag{3.48}$$

$RMISE_W$ thus has the unbiased estimator

$$\widehat{RMISE_W} = \sum_{j=1}^{\infty} w_j \left[\widehat{Var}\left(\hat{a}_j\right) - \widehat{a_j^2}\right]. \tag{3.49}$$

This is the MISE-based measure that is employed throughout the remainder of this dissertation for basis function set selection using a stopping rule and model simplification using a single-term exclusion rule.

3.7.9 Application of RMISE to model selection: series truncation (stopping)

The cumulative MISE of a model with ordered terms can be minimized by selecting a stopping term that minimizes the RMISE, an equivalent measure up to a constant:

$$\Gamma = \arg\min_{\gamma} \left(\widehat{RMISE_{W_\gamma}} \right), \tag{3.50}$$

where W_γ is the set of basis functions that includes the γ ordered terms, i.e.

$$w_j = \begin{cases} 1, & j \le \gamma \\ 0, & j > \gamma \end{cases} \quad \gamma = 0,1,2,\ldots \tag{3.51}$$

3.7.10 Application of RMISE to model selection: single term exclusion

Tarter, Holcomb and Kronmal (1967) initially suggested a single-term inclusion rule for term selection. They subsequently discovered (Tarter and Kronmal 1976; Tarter and Lock 1993) that deciding whether to include terms individually leads to inclusion of spurious high-order terms. However, a similar procedure inspired by the single-term stopping rule is a promising method of achieving significantly higher levels of compression in an orthonormal basis function model.

Regardless of the method used for stopping, including certain terms in the model may increase both the expected MISE and the complexity of the model. The RMISE is the sum of the individual terms in Equation (3.48), each of which corresponds to one term in the orthonormal series expansion. If the estimator corresponding to the *j*th term is positive:

$$w_j \left[\widehat{Var}\left(\hat{a}_j \right) - \widehat{a_j^2} \right] > 0 , \tag{3.52}$$

then the expected contribution of the *j*th term is an increase in the MISE of the model. Omitting the term decreases both the complexity of the model and the expected MISE error rate.

3.8 Evaluating the performance of classification methods

It is important that statistically valid experiments be performed to evaluate the performance of any pattern classification algorithm. Fortunately, there are several sources for standard benchmarking problems and techniques. These sources include the University of California Irvine (UCI) machine learning repository (Hettich, Blake et al. 1998), the University of Toronto's DELVE suite for statistical evaluation of machine learning algorithms (Rasmussen, Neal et al. 1996), and certain problems that have been formalized and popularized by authors such as Ripley (1996).

The strengths and weaknesses of a given algorithm can be probed by comparing it to alternative methodologies. In particular, this dissertation compares orthonormal basis function neural networks to the *k*-nearest-neighbors (KNN) algorithm, backpropagation neural networks, and two support vector machine (SVM) algorithms. KNN is a simple and robust classification algorithm which is frequently used as a point of comparison. Backpropagation neural networks (Werbos 1974; Rumelhart, Hinton et al. 1986) are multilayer perceptrons that use nonorthogonal basis functions of a particular functional form, typically a sigmoid function. Backpropagation is a popular mean squared error minimization method that is computationally intensive. The backpropagation algorithm utilizes an iterative nonlinear optimization to adaptively minimize the mean squared error of a model. SVMs employ support vector regularization, a nonlinear regularization

method that is applicable to basis expansions (Vapnik 1995; Schölkopf, Burges et al. 1999). As is the case for the linear regularization used for orthonormal basis function neural networks, support vector regularization minimizes an objective function in the space of a basis function expansion. The techniques differ in many respects, however, including the types of objective functions used, the methods of representing basis function expansions, and the applicability to particular basis function expansions.

A problem with these benchmark comparisons is that many of the standardized benchmark problems are small databases, containing at most a few thousand exemplars. Such databases do not demonstrate the applicability of algorithms to large, real-world problems. The problem of identifying land use changes in the Nile River delta from a sequence of ten satellite images (Lenney, Woodcock et al. 1996), described in further detail in Chapter 2 of this dissertation, is a useful platform to test orthonormal basis function networks on a larger scale. It is a real-world database consisting of eight different classes and sixty-five-dimensional data vectors. Orthonormal basis function networks were used to classify land use changes on the same database of images employed for ARTMAP neural network classification of land use change. This database contains approximately 25,000 pixels for which land use classifications are known. Millions of additional pixels must be classified to generate a map of land use changes in the study area. This demands fast testing performance.

3.8.1 Benchmarking results

One problem that illustrates certain properties of classifiers is Ripley's two-dimensional synthetic database (1994). This database is popular due to the ease of

visualization. Since the database is synthetic, the Bayes error rate of 8.0% is known. The data consist of 250 exemplars drawn from four bivariate Gaussian distributions. Two of these Gaussians correspond to each of the two classes.

The orthonormal basis function neural network methodology outlined above results in a discrete cosine basis model with 21 basis function units. Fitting this model is a speedy process: all of the computations involved take a fraction of a second on a typical Linux workstation. The error rate on an independent test set is 10.4%, which is comparable to the results using various other pattern classifiers reported by Ripley. As seen in Figure 3.5, the contours of the orthonormal basis function neural network model (solid lines) closely follow the optimal decision boundary (broken lines) in high-density regions of the decision space. Note that there are large regions in the decision space where this model yields the incorrect class; however, the likelihood of data appearing in these regions is expected to be very low. This is a data sampling problem: with a hundred or a thousand times more data points, the fact that this is a consistent estimator may cause the problem to disappear entirely. With sparse data requiring a tradeoff between being accurate in much of the decision space and being accurate where data are most likely to appear, this methodology will select basis functions that optimize the model where there is a higher density of data points.

If less complex decision boundaries were desired, one approach not investigated in depth in this dissertation would be to employ an appropriate regularization term to penalize models with complex boundaries.

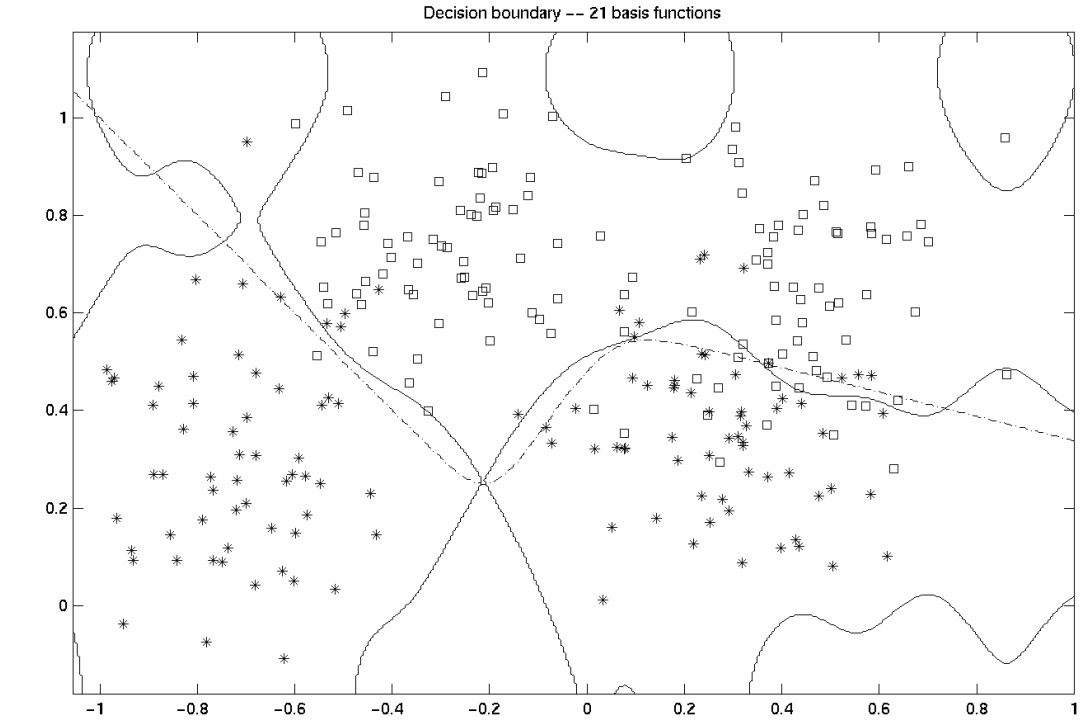

Figure 3.5: Decision boundaries of an orthonormal basis function neural network classifier applied to Ripley's synthetic dataset. The error rate on the test set is 10.4%. The Bayes decision criterion (dashed lines) gives an error rate of 8.0%

Chapter 4

Linear Preprocessing and Postprocessing to Improve Orthonormal Basis Function Neural Network Models

4.1 Introduction

Several important problems arise when the orthonormal basis function classifiers of Chapter 3 are applied to classification tasks. This chapter considers rotation of data, dimension reduction, and extension of a two-class algorithm to multiple classification.

Rotation of data prior to fitting with an orthonormal basis function needs to be considered because the functions in the system are fixed. The model is dependent on the orientation of the data with respect to the basis functions. Because of this, it is possible that changes in the orientation will have a significant impact on the goodness of a model as measured by classification rate. Common approaches to data rotation and dimension reduction include Principal Components Analysis (PCA) and Canonical Variate Analysis (CVA) (Mardia, Kent et al. 1979). This chapter considers both of these methods and introduces Extended CVA, which combines CVA with PCA when the number of dimensions equals or exceeds the number of classes.

The number of dimensions used as input to an orthonormal basis function network can also be an important factor in the goodness of a model. If too few dimensions are used, the discarded dimensions may contain important information. Using too many dimensions can also degrade the performance.

As reviewed by Bentler and Yuan (1996), one approach to dimension reduction is to test the hypothesis that the smallest q eigenvalues of the data covariance matrix $\mathbf{\Sigma}$ are equal for $q = 2,3,\ldots,D$ using Bartlett's test (Bartlett 1954). However, in practice it is often the case that few dimensions have statistically identical eigenvalues and can be eliminated in this way. Bentler and Yuan observe that instead, eigenvalues of real data, when plotted, tend to fall sharply from the largest values, then trend linearly toward the smallest value, and that Cattell's scree test (1966) makes use of this to select principal components that are most likely to bear useful information based on their eigenvalues. The portion of the eigenvalue plot that trends linearly after the sharp drop is designated as "scree", excess dimensionality that can be eliminated from a model. A failing of this test is that it is a subjective visual test not easily implemented as part of an algorithm. Several automated methods for performing a scree test have been proposed (Bentler and Yuan 1996). This chapter proposes another such test, designed to be less stringent in its criterion for an eigenvalue to be considered scree.

A third problem investigated in this chapter is that of extending a two-class method to multiple classes. Devroye's discriminant estimator yields appropriate decision boundaries for two-class problems, but such a simple decision rule is insufficient in a multiclass context. *Masking* (Hastie, Tibshirani et al. 2001) can occur, leading to poor performance when bivariate decision rules are extended to three or more classes. One solution is to employ linear discriminant methods appropriate for multiclass problems on bivariate discriminant estimators. This chapter shows that linear discriminant analysis (LDA) postprocessing is equivalent to optimal scoring criteria that minimize the average

squared residual (ASR) of a model when the underlying model is fixed and only the postprocessing parameters are permitted to change.

4.2 Linear preprocessing for data orientation and dimension reduction

4.2.1 Principal components analysis (PCA)

4.2.1.1 Computation of principal components

Principal components analysis (PCA) computes an ordered set of orthonormal vectors onto which a data matrix $\mathbf{X}$ can be projected such that the variance of the projection onto each successive vector in the set is maximal (Ripley 1996).

Ripley (1996) gives the following simple explanation for how these may be obtained for an $n \times p$ data matrix $\mathbf{X}$ consisting of row vectors $\mathbf{x}_i$:

> This is done by taking the *singular value decomposition* of the data matrix $\mathbf{X}$ (Golub and Van Loan 1989) $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{\Lambda}$ is a diagonal matrix with decreasing non-negative entries (λ_i), U is an $n \times p$ matrix with orthonormal columns, and V is a $p \times p$ orthogonal matrix. Then the principal components are the columns of $\mathbf{XV}$.

The first q columns of $\mathbf{XV}$ contain the first q principal components of $\mathbf{X}$, which maximize the variance of the projection of $\mathbf{X}$ onto q dimensions.

4.2.1.2 Use of principal components for pattern recognition

Because the principal components of $\mathbf{X}$ are not invariant to linear scaling of $\mathbf{X}$, it is important for the dimensions of $\mathbf{X}$ to be in comparable units. Where the units of $\mathbf{X}$

are not directly comparable, it is customary to rescale the columns of $\mathbf{X}$ to have unit variance and zero mean.

Let $\mathbf{V}_q$ be the matrix consisting of the first $q \le p$ columns of $\mathbf{V}$:

$$\mathbf{V}_q = \mathbf{V}\begin{bmatrix}\mathbf{I}_q \\ \mathbf{0}\end{bmatrix}, \tag{4.1}$$

where $\mathbf{I}_q$ is the $q \times q$ identity matrix. For any such q, $\mathbf{X}_q^* = \mathbf{X}\mathbf{V}_q$, the first q unscaled principal components of $\mathbf{X}$, can be used as instead of $\mathbf{X}$ itself as the input data for neural networks and machine learning algorithms. For the remainder of this dissertation, PCAq, where q is given as an integer value, will denote the transformed input data matrix $\mathbf{X}_q^*$ obtained in this manner.

Results of using PCAq as the input to orthonormal basis function networks are reported later in this chapter, where this method is also compared to using the untransformed data $\mathbf{X}$ and other linear transformations of $\mathbf{X}$.

Note that PCA does not take into account the class labels y_i associated with the rows of $\mathbf{X}$. This is a shortcoming of the use of PCA for classification problems. The directions which explain the maximal variance in the input vectors are not necessarily the directions that are most useful for classification. For classification problems, *canonical variates* (Mardia, Kent et al. 1979), to be discussed in Section 4.2.3, may be a more appropriate linear transformation of the data matrix.

4.2.2 An automated scree test for principal component dimensionality

A problem that often occurs with PCA is that the principal components with the largest eigenvalues are helpful for building a model while the principal components with the smallest eigenvalues act as distractors. As reviewed by Bentler and Yuan (1996), Cattell (1966) developed a *scree plot* method for determining the number of principal components to keep in a dimension reduction procedure. The method is so named because the plot of eigenvalues often resembles a steep mountain slope with scree or rubble collected at the bottom. The larger eigenvalues that comprise the steep slope are interpreted in Cattell's method as "important", and the corresponding principal components are kept. Cattell observed that the remaining eigenvalues formed a shallow slope that was approximately linear. His test is a visual test that requires the user to identify the elbow of the scree plot, the point at which the steep slope ends and the scree begins.

Bentler and Yuan (1996) developed a test for the linearity of the smallest eigenvalues. While useful, this test may be sensitive to small nonlinearities in the scree. In this section, an alternative test for the approximate linearity of eigenvalues is proposed. This test determines whether an individual eigenvalue is consistent with a linear trend in eigenvalues by measuring its *influence* (Weisberg 1985) on a linear fit to the sequence of eigenvalues. This is a formalization of Cattell's scree test that replaces the subjective analysis of plots with a simple objective test for the similarity of an eigenvalue to the trend in successive eigenvalues.

A standard measure of influence is Cook's distance (Cook 1977),

$$D_i = \frac{\left(\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}}\right)^T \left(\mathbf{V}^T\mathbf{V}\right)\left(\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}}\right)}{p'\hat{\sigma}^2}, \tag{4.2}$$

where $p' = 2$ is the number of parameters estimated in linear regression on a single scalar variable, $\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix}$ contains the estimated slope ($\hat{\beta}_1$) and intercept ($\hat{\beta}_0$) parameters, and $\hat{\boldsymbol{\beta}}_{(i)}$ is the estimated parameter vector when the ith data point is omitted from the regression on the data matrix $\mathbf{V}$.

Let $\{\lambda_i\}$ be the set of eigenvalues of a data covariance matrix in descending order of magnitude. Let Δ_i be the influence of

$$\mathbf{v}_i = \begin{bmatrix} 1 & \lambda_i \end{bmatrix}, \tag{4.3}$$

corresponding to the ith eigenvalue λ_i, on the linear regression model

$$y_j = \mathbf{v}_j^T\boldsymbol{\beta} + e_j, \quad j \geq i \tag{4.4}$$

of the eigenvalues with index greater than or equal to i. If Δ_i is large, this is indicative of an eigenvalue that is much larger or smaller than the trend of successive eigenvalues. Since eigenvalues that are much larger than typical eigenvalues will have large values of Δ_i, it may be possible to use this as a measure of the importance of an eigenvalue in a way that corresponds to visual interpretation of a scree plot. Let i_{elbow} be the index of the first eigenvalue that does not exceed a predefined influence threshold:

$$i_{elbow} = \arg\min_i 1\left(\Delta_i < \Gamma_{elbow}\right) \tag{4.5}$$

It is hoped that appropriate selection of Γ_{elbow} will result in automated determination of scree plot truncation points similar to those obtained by visual inspection.

Bentler and Yuan (1996) demonstrate their linear trend (LT) method for PCA model selection on two published psychological databases. The Lord (1956) database, as republished by Bentler and Yuan, concerns student performance on a variety of psychological tests yielding 15 variables. The eigenvalues of the covariance matrix are plotted in the upper half of Figure 4.1. The lower half of the figure plots the corresponding values of Δ_i. Bentler and Yuan conclude that "the last 13 eigenvalues do not show a linear trend as assessed by the LT-χ^2_{q-2} test statistic, while the last 12, and certainly the last 11, eigenvalues do exhibit a linear trend." Using the influence-based method of this section, a wide range of values for Γ_{elbow} leads to the conclusion that the last 11 eigenvalues are scree. This moreover agrees with a visual inspection of the scree plot, in which either the fourth or fifth eigenvalue would be determined to be the elbow location.

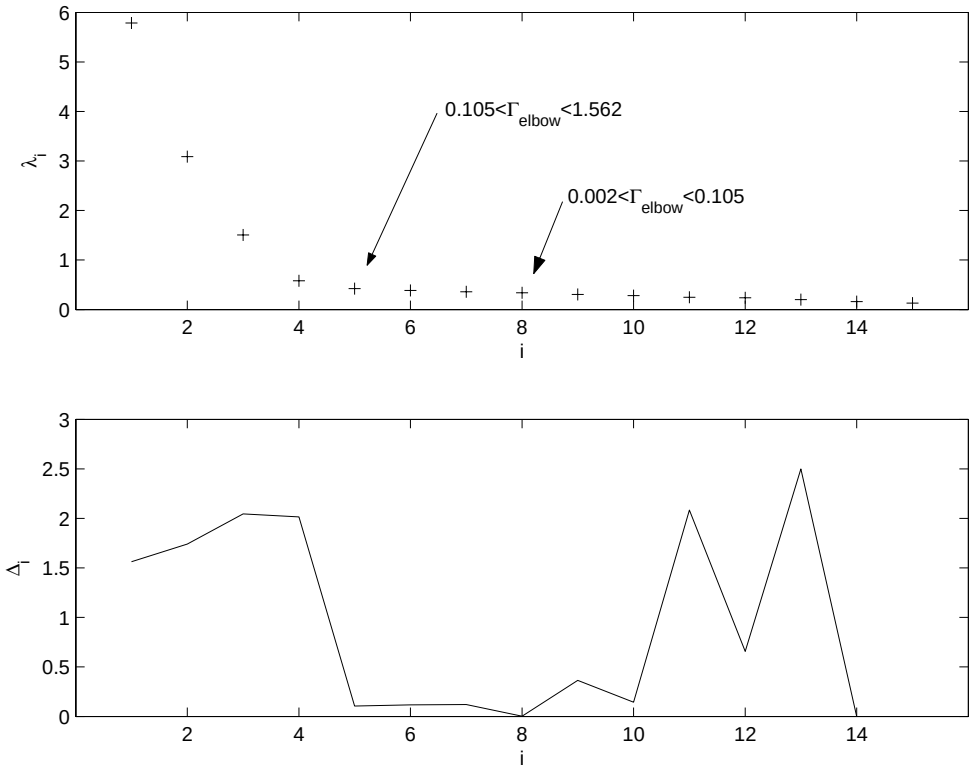

Figure 4.1: Automated scree plot analysis for the psychological test data of Lord (1956). A wide range of thresholds Γ_{elbow} lead to the determination that $i_{elbow} = 5$. This agrees with both a visual inspection of the scree plot and the linear trend (LT) eigenvalue analysis method of Bentler and Yuan (1996).

The 24-dimensional psychological test data of Holzinger and Swineford (1939), also as published in Bentler and Yuan (1996), are problematic for both a visual scree test and the automated method based on the influence measure Δ_i. The location of the elbow in the sequence of eigenvalues is ambiguous, as seen in Figure 4.2. The threshold range for Δ_i that best corresponds to visual selection of the elbow may be $0.398 < \Gamma_{elbow} < 0.839$, which places the elbow at the ninth eigenvalue. The LT method (Bentler and Yuan 1996) finds a linear trend commencing with the twelfth eigenvalue.

From these preliminary studies, it appears that an automated scree test method based on the influence of eigenvalues corresponds well with the visual scree plot method. A threshold value such as $\Gamma_{elbow} = 0.5$ may lead to the selection of components that are reasonably close to those selected by visual inspection and the existing LT method. Further studies might both determine the statistical properties of such a threshold and investigate whether a single value for Γ_{elbow} is applicable to a wide variety of datasets (Section 7.2.3). Where an automated scree test is used on the Adult dataset later in this dissertation, the threshold $\Gamma_{elbow} = 0.5$ is employed.

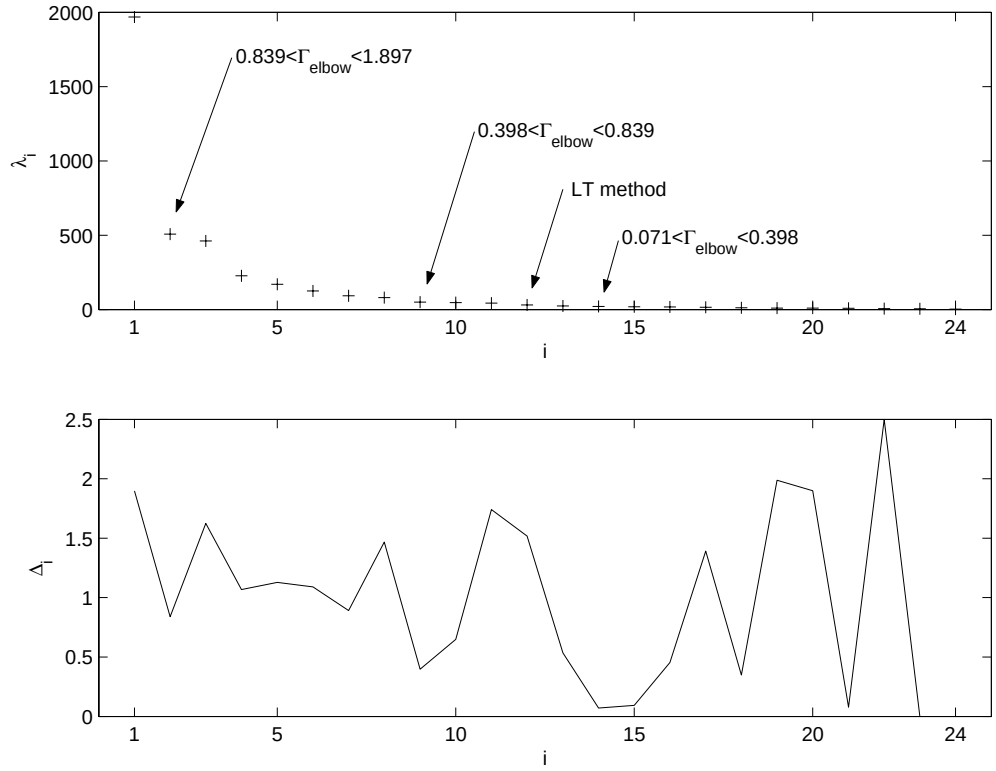

Figure 4.2: Automated scree plot analysis for the psychological test data of Holzinger and Swineford (1939). A threshold value $\Gamma_{elbow} > 0.839$ is aggressive, discarding components that a visual inspection of the scree plot would likely include. More reasonable elbow points are given by the threshold ranges $0.398 < \Gamma_{elbow} < 0.839$ and $0.071 < \Gamma_{elbow} < 0.398$. Neither is far from that given by the linear trend (LT) eigenvalue analysis method of Bentler and Yuan (1996). The middle range of threshold values $0.398 < \Gamma_{elbow} < 0.839$ may agree best with a visual inspection of the scree plot.

4.2.3 Canonical variates

Canonical variates, like principal components, are a linear transformation of the input matrix $\mathbf{X}$. Whereas principal components are selected to maximize the variance of the input data with respect to the components, canonical variates are selected to maximize the correlation between the input data and the output classes. The input data are

represented as rows of an input matrix $\mathbf{X}$, and the output classes are represented as a matrix of dummy zero-one class indicator variables $\mathbf{W}$, where

$$w_{i,j} = \begin{cases} 1 & \text{if the } i\text{th exemplar} \in \text{class } j \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

(Mardia, Kent et al. 1979).

Canonical variates can be obtained by performing a canonical correlation analysis on the input matrix $\mathbf{X}$ and the class indicator matrix $\mathbf{W}$ given in (4.6). The first canonical correlation vectors $\mathbf{a}_1$ and $\mathbf{b}_1$ maximize the correlation between $\mathbf{X}\mathbf{a}_1$ and $\mathbf{W}\mathbf{b}_1$. Subsequent canonical correlation vectors are chosen to maximize the correlation between $\mathbf{X}\mathbf{a}_q$ and $\mathbf{W}\mathbf{b}_q$ subject to the condition that $\mathbf{X}\mathbf{a}_q$ is uncorrelated with the previous canonical correlation variables $\mathbf{X}\mathbf{a}_1 .. \mathbf{X}\mathbf{a}_{q-1}$ (Mardia, Kent et al. 1979). Let

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_D \end{bmatrix}. \tag{4.7}$$

Then the column vectors of $\mathbf{XA}$ are the canonical variates.

Because of the relationship between canonical variates and linear discriminant analysis (discussed further in Section 4.3.1), it is standard to scale the canonical correlation vectors $\mathbf{a}_q$ such that each canonical variate $\mathbf{X}\mathbf{a}_q$ has unit within-class variance (Ripley 1996).

4.2.4 Extended canonical variates

A key limitation of canonical variates as inputs to an orthonormal basis function neural network is that the number of canonical variates is limited to $C-1$, where C is the number of classes in the dataset. In some cases, it is advantageous to present inputs

that have higher dimensionality than canonical variate analysis can provide. This may be the case when the number of classes is small relative to the number of input dimensions.

Extended canonical variates are the set of canonical variates augmented by the principal components of the orthogonal complement of the canonical vectors. These principal components represent the directions in which the data vary the most, yet the weighted means of the classes are equal. Although these components are not useful for linear classification in the original problem space, they can potentially improve the ability of other methods, including orthonormal basis function networks, to distinguish between classes.

Extended canonical variates are not invariant with respect to the scale of the unrotated data, although the first $C-1$ components are. The remaining components, derived using PCA, will vary if the dataset is rescaled prior to computing the extended canonical variates. Unless the relationship between the scales of dimensions in the original data is known and meaningful, it is appropriate to scale the original data to have mean $\mu = 0$ and standard deviation $\sigma = 1$.

4.2.4.1 Obtaining extended canonical variates

Let $\mathbf{A}$ be the matrix of canonical variate projection column vectors. The extended canonical variates are the canonical variates $\mathbf{XA}$ augmented by the principal components of the canonical variate transform residual matrix $\mathbf{XA}^{\perp}$, where the vectors in $\mathbf{A}^{\perp}$ are orthonormal to preserve the scale of $\mathbf{X}$.

The principal components of $\mathbf{XA}^{\perp}$ may be obtained by direct computation of $\mathbf{A}^{\perp}$. It is equivalent to find the principal components of

$$\mathbf{X}' = \mathbf{X}\left(\mathbf{I} - \mathbf{A}\mathbf{A}^{+}\right), \tag{4.8}$$

where $\mathbf{A}^{+} = \left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathbf{T}}$ is the standard Moore-Penrose matrix pseudoinverse , since $\mathbf{I} - \mathbf{A}\mathbf{A}^{+}$ spans the null space of $\mathbf{A}$.

The number of extended canonical variates is identical to the rank of $\mathbf{X}$. If fewer variates than this number are desired, those with the smallest PCA coefficients are discarded.

For the remainder of this dissertation, CVA*q*, where *q* is given as an integer value, will denote the first *q* extended canonical variates of $\mathbf{X}$:

$$\text{CVA}q = \mathbf{X}_q^{*} = [\mathbf{X}\mathbf{A} \quad \mathbf{X}'\mathbf{V}]\begin{bmatrix}\mathbf{I}_q \\ \mathbf{0}\end{bmatrix}, \tag{4.9}$$

where $\mathbf{I}_q$ is the $q \times q$ identity matrix. CVA*q* contains the first *q* columns of $\mathbf{X}^{*} = [\mathbf{X}\mathbf{A} \quad \mathbf{X}'\mathbf{V}]$.

4.3 Postprocessing to improve multiclass models

Any classification algorithm that assigns class scores based on a vector of input variables can be formulated as a regression technique if the class scores are taken to be the result of regressing class indicator variables on the input data. Hastie, Tibshirani et al. (1994) show that the problem of *optimal scoring*, selecting the linear transformation of class scores that minimizes the least squares error from the correct classification, is equivalent to performing linear discriminant analysis (LDA) on the class scores.

Hastie et al. make an argument for implementing optimal scoring for classification problems involving three or more classes whenever an algorithm is capable

of providing class scores. They show that other methods that can yield similar discrimination boundaries for two-class problems, such as Softmax on linear regression models (Bridle 1990), fail to find optimal linear decision boundaries for multiclass data.

4.3.1 Linear Discriminant Analysis (LDA)

Let $\mathbf{XA}$ be the canonical variates of the input matrix $\mathbf{X}$, where $\mathbf{A}$ is the linear transformation matrix of (4.7) in which each $\mathbf{Xa}_q$ has unit variance.

In this canonical variate space, Mahalanobis distance is identical to Euclidean distance to the class means (Ripley 1996). If the class prior probabilities are equal, linear discriminant analysis (LDA) assigns to a vector $\mathbf{x}_i$ the estimated class $\hat{y}_i$ whose linearly transformed class mean $\boldsymbol{\mu}_y^T\mathbf{A}$ is nearest $\mathbf{x}_i^T\mathbf{A}$. If the class prior probabilities π_y are not equal, the linear discriminant applies a correction factor adding $-2\log\pi_y$ to the distances to the class means.

This method of determining the linear discriminant shows the connection between LDA and canonical variates analysis (CVA), which in turn can be thought to be an application of canonical correlation analysis (CCA). The LDA model that this finds is the standard LDA model consisting of a multivariate Gaussian for each class with a single covariance matrix common to each Gaussian. The Gaussians differ in their means and amplitudes.

4.3.2 Optimal scoring for an orthonormal basis function model with fixed coefficients

Once the coefficients of an orthonormal basis function network are fixed, it may be possible to improve the decision boundaries by *optimal scoring* (Hastie, Tibshirani et

al. 1994). Let θ_k be a function that assigns scores to the class labels, and consider the model

$$\theta_k(z) \sim \mathbf{x}^T \boldsymbol{\beta}_k , \tag{4.10}$$

where the class label z is transformed by the optimal scoring function θ_k and the variates $\mathbf{x}$ are multiplied by the weight vector $\boldsymbol{\beta}_k$ to minimize the *average squared residual (ASR).* Hastie, Tibshirani et al. give this as:

$$ASR = \frac{1}{N} \sum_{k=1}^{C} \sum_{i=1}^{N} (\theta_k(z_i) - \mathbf{x}_i^T \boldsymbol{\beta}_k)^2 \tag{4.11}$$

Mardia, Kent et al. (1979) show the equivalence between this optimal scoring problem and linear discriminant analysis (LDA) (Section 4.3.1).

Flexible discriminant analysis (Hastie, Tibshirani et al. 1994) makes it easy to extend the model of (4.10) and (4.11) to a partial basis expansion $\boldsymbol{\varphi}(\mathbf{x}_i)$:

$$\theta_k(z) \sim \boldsymbol{\varphi}(\mathbf{x})^T \boldsymbol{\beta}_k \tag{4.12}$$

$$ASR = \frac{1}{N} \sum_{k=1}^{C} \sum_{i=1}^{N} (\theta_k(z_i) - \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\beta}_k)^2 \tag{4.13}$$

Minimizing the average squared residual in this case requires a full multivariate regression on the expanded basis terms $\boldsymbol{\varphi}(\mathbf{x}_i)$. Depending on the number of terms under consideration, this can be computationally intensive.

Instead of performing a full regression on the expanded basis terms, it is possible to estimate the coefficients of these terms using Devroye's discriminant estimator or an equivalent class density estimator and fix the coefficients. Let

$$A = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,C} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{M,1} & \alpha_{M,2} & \cdots & \alpha_{M,C} \end{bmatrix} \tag{4.14}$$

be the matrix of coefficients for Devroye's discriminant estimator. If A is fixed, the model of (4.12) can be modified to require a regression only on the outputs of the orthonormal basis function classifier developed in the previous chapter:

$$\begin{aligned} &\theta_k(z) \sim [\mathrm{A}^T \boldsymbol{\varphi}(\mathbf{x})]^T \boldsymbol{\beta}_k \text{ , or equivalently} \\ &\theta_k(z) \sim \hat{\mathbf{y}}^T \boldsymbol{\beta}_k \end{aligned} \tag{4.15}$$

This model has average squared residual:

$$ASR = \frac{1}{N} \sum_{k=1}^{C} \sum_{i=1}^{N} (\theta_k(z_i) - \hat{\mathbf{y}}_i{}^T \boldsymbol{\beta}_k)^2 \tag{4.16}$$

This clearly can be minimized in exactly the same way as (4.11), substituting $\hat{\mathbf{y}}_i^T$ for $\mathbf{x}_i^T$. The results of Mardia, Kent et al. and Hastie, Tibshirani et al. therefore show that this particular optimal scoring problem is solved by performing a procedure equivalent to LDA on the estimated class indicators $\hat{\mathbf{y}}_i$ and the associated actual class indicators $\mathbf{y}_i$.

In fact, it is clear that $\hat{\mathbf{y}}_i$ may always be substituted for $\mathbf{x}_i$ in this manner whenever a classification algorithm estimates a class indicator vector. Performing LDA on the estimated class indicator vectors minimizes the average squared residual *subject to the constraint that the coefficients of the underlying model are fixed by some alternate methodology*. Note that in most cases it would be possible to achieve a lower average squared residual without this constraint by performing a full regression to find the

coefficients of the underlying model, but the computational expense of doing so might be prohibitive if the number of terms under consideration is large.

4.4 ANOVA models of preprocessing and postprocessing performance

The method used for preprocessing, the dimensionality of the preprocessed data, and the method used for postprocessing all potentially impact the performance of an orthonormal basis function network. The combination of methods to use for benchmarking was selected by estimating the effect that each choice had on the classification error rate.

4.4.1 Procedure

Analysis of Variance (ANOVA) (Winer, Brown et al. 1991) is a standard statistical tool for linear modeling of the effects of known factors in an experimental setting. In this experiment, the classification error rate of an orthonormal basis function network was modeled as a linear function of four factors and their interactions. ANOVA attributes a portion of the total variance in the performance to each variable and interaction term, with the remainder accounted for as *residual*.

The factors under consideration included the preprocessing method, preprocessing dimensionality, postprocessing method, and basis used. The levels of these factors are summarized in Table 4.1. These experiments were constructed using a four-way full factorial design, meaning that every possible combination of factor levels was represented. ANOVA models based on this design were used to evaluate the performance of orthonormal basis function networks on benchmarks from the DELVE

suite (Rasmussen, Neal et al. 1996; Hettich, Blake et al. 1998) under the variety of conditions represented by the factors.

Factor	**Levels**
Preprocessing method	CVA, PCA
Preprocessing dimensionality	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Postprocessing method	LDA, Maximum (no postprocessing)
Basis	Cosine (discrete cosine basis with zero-crossing cutoff criterion), Daubechies (second-order Daubechies wavelets with scale product cutoff criterion), Legendre (polynomial basis with polynomial-order cutoff criterion)

Table 4.1: Factors and treatment levels for four-way ANOVA modeling of orthonormal basis function neural network performance on DELVE development benchmarks. In the *Maximum* treatment level, the class selected is that with the maximum value of the corresponding one-vs.-many discriminant at a particular $\mathbf{x}_i$.

The ad hoc selection of cutoffs reflects that exploratory data analysis on these databases did not yield significant differences between cutoffs (hyperbolic, linear, and spherical) within each basis, likely because the data were inadequate for this purpose. Therefore, a single representative cutoff was selected for each basis. It is possible that the cutoff selected could be a significant factor in classification performance on other databases. The hyperbolic cutoff, which uses the ZC order of tensor product basis functions, was selected for the cosine and Daubechies bases. The linear cutoff, which

corresponds to multivariate polynomial order, was selected for the Legendre basis. It should be noted that results in this chapter and following chapters showing comparative performance of bases are in fact comparing these particular ad hoc combinations of basis and cutoff, as summarized in Table 4.1.

The DELVE benchmark databases consist of between four and eight disjoint training sets. These were considered to be replicates for the sake of this analysis. Although the training sets are disjoint, they do not meet the independence requirement for ANOVA analysis since the same training sets are used for each and every combination of factor levels. This approach is thus deficient because the replicates are correlated across all treatments. While a repeated-measures ANOVA analysis might be more appropriate for these data, the results are very similar to those using standard ANOVA. Repeated-measures ANOVA is moreover known to be sensitive to violations of its assumption of *sphericity* (Keppel 1991; Winer, Brown et al. 1991), while standard ANOVA is relatively robust. Sphericity is the property that the group covariance matrix $\mathbf{\Sigma} = \begin{bmatrix} s_1^2 & s_{12} & \cdots & s_{1N} \\ s_{21} & s_2^2 & \cdots & s_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N1} & s_{N2} & \cdots & s_N^2 \end{bmatrix}$ has when $s_j^2 + s_k^2 - 2s_{jk}$, the variance in the difference between the groups, is identical for all group pairings $\{j,k \mid j \neq k\}$.

4.4.2 Results

4.4.2.1 Letter recognition database

The letter recognition task is classification of a sixteen-dimensional vector of letter attributes as one of twenty-six capital letters. Frey and Slate (1991) generated this

database by distorting twenty fonts and taking various metrics. DELVE splits the database into six disjoint training sets and six disjoint test sets. It is meant to be run three times, with each run doubling the number of training exemplars available. In the first run, there are 390 training exemplars per replicate; in the second, 780 exemplars; and in the third, 1,560 exemplars. Figure 4.3, Figure 4.4, and Figure 4.5 show the ANOVA models resulting from these three runs. The Shapiro-Wilk test as implemented by the R Development Core Team (2003), a standard normality test that has been extended to be applicable to a wide range of sample sizes, was used to inspect the residuals. With 780 or 1,560 exemplars per replicate, the four-way ANOVA models without interaction have residuals statistically indistinguishable from the normal distribution. The best models including interaction terms for all three runs have this same property. ANOVA therefore appears to be appropriate for characterizing these data. The ANOVA models show that each of the four factors under consideration is statistically significant at $p = .05$.

CVA preprocessing results in a lower mean classification error rate than does PCA on the letter recognition database. It is also clear that the number of dimensions is significant. However, on the 390-exemplar run, there was no significant difference between the best observed performance (for twelve dimensions) and anything else in the range of ten to sixteen dimensions. On the 780-exemplar and 1,560-exemplar runs, there was no significant difference between the best observed performance (for sixteen and fifteen dimensions, respectively) and anything else in the range of twelve to sixteen dimensions. These data are insufficient to determine whether any reduction in dimensionality is beneficial for application of orthonormal series classifiers to the letter

recognition benchmark. It is clear that reducing the dimensionality to nine or less is detrimental in this case.

For postprocessing, LDA consistently results in better performance than simply taking the class with the highest score.

The basis used (in combination with the ad hoc selected cutoff) also proved to be a significant factor. For the letter recognition problem, the cosine basis resulted in the lowest mean error rate, followed closely by the Legendre basis.

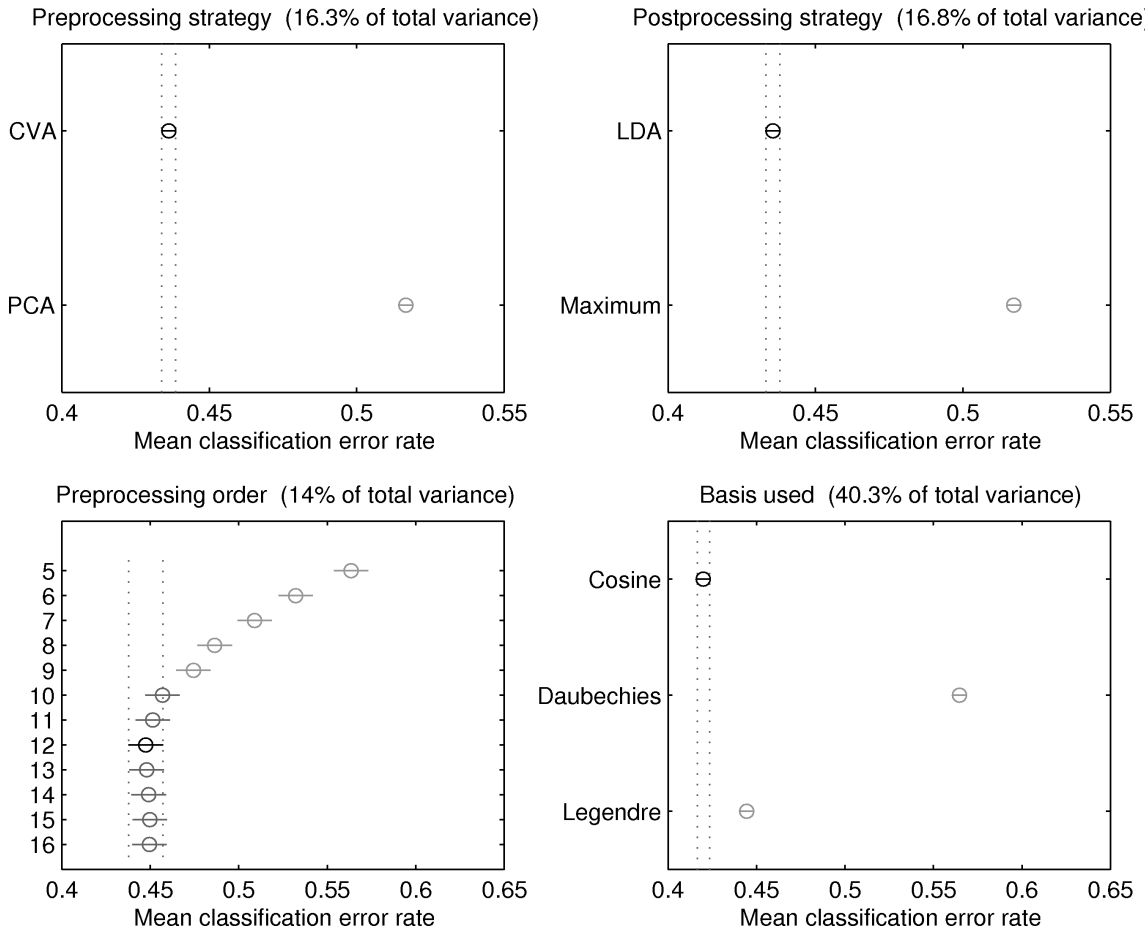

Figure 4.3: Multiple comparison of four-way ANOVA model factors for the DELVE letter recognition task with 390 training exemplars. The classification error rate has global mean $\mu = 0.476$, standard deviation $\sigma = 0.100$, and $\frac{\sigma}{\mu} = 0.209$. Residuals account for 12.6% of the total variance.

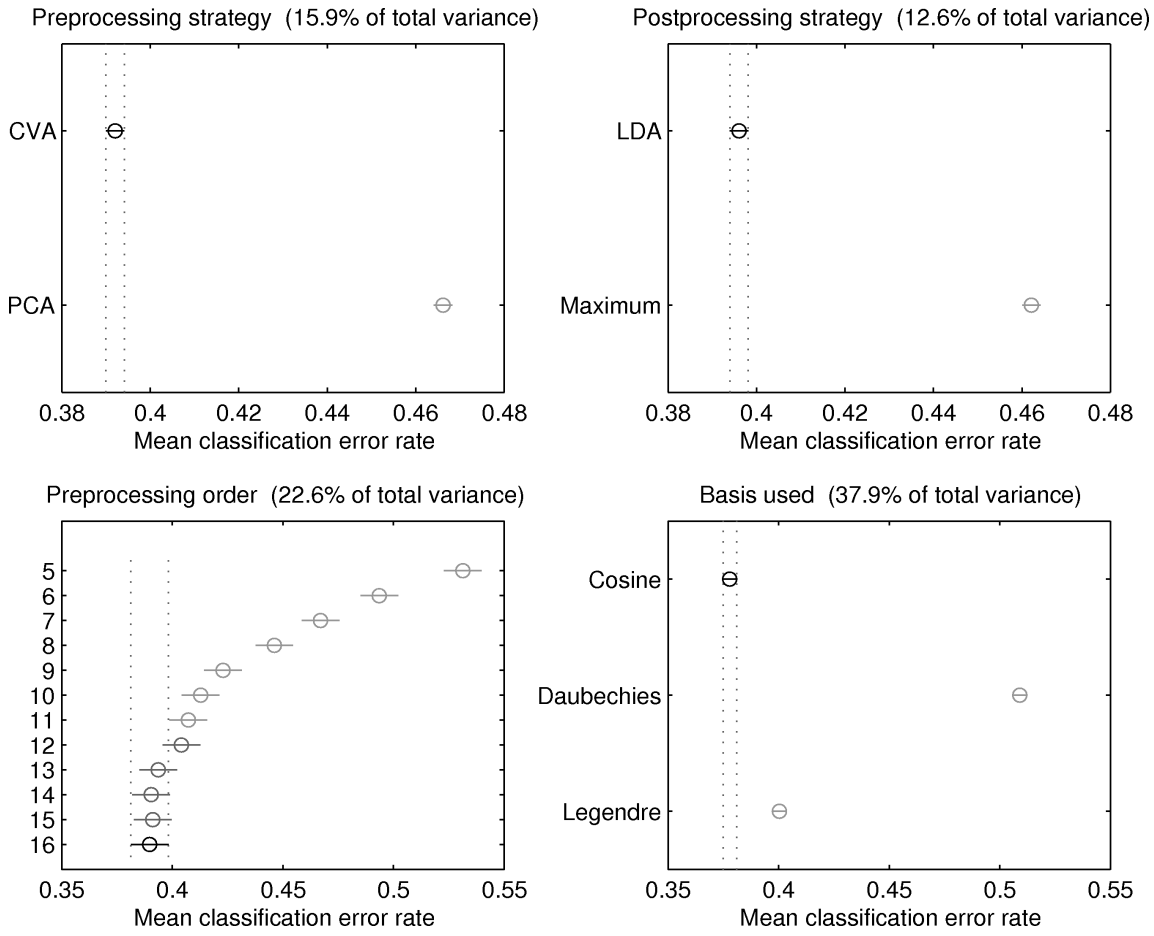

Figure 4.4: Multiple comparison of four-way ANOVA model factors for the DELVE letter recognition task with 780 training exemplars. The classification error rate has global mean $\mu = 0.429$, standard deviation $\sigma = 0.093$, and $\frac{\sigma}{\mu} = 0.217$. Residuals account for 11.1% of the total variance.

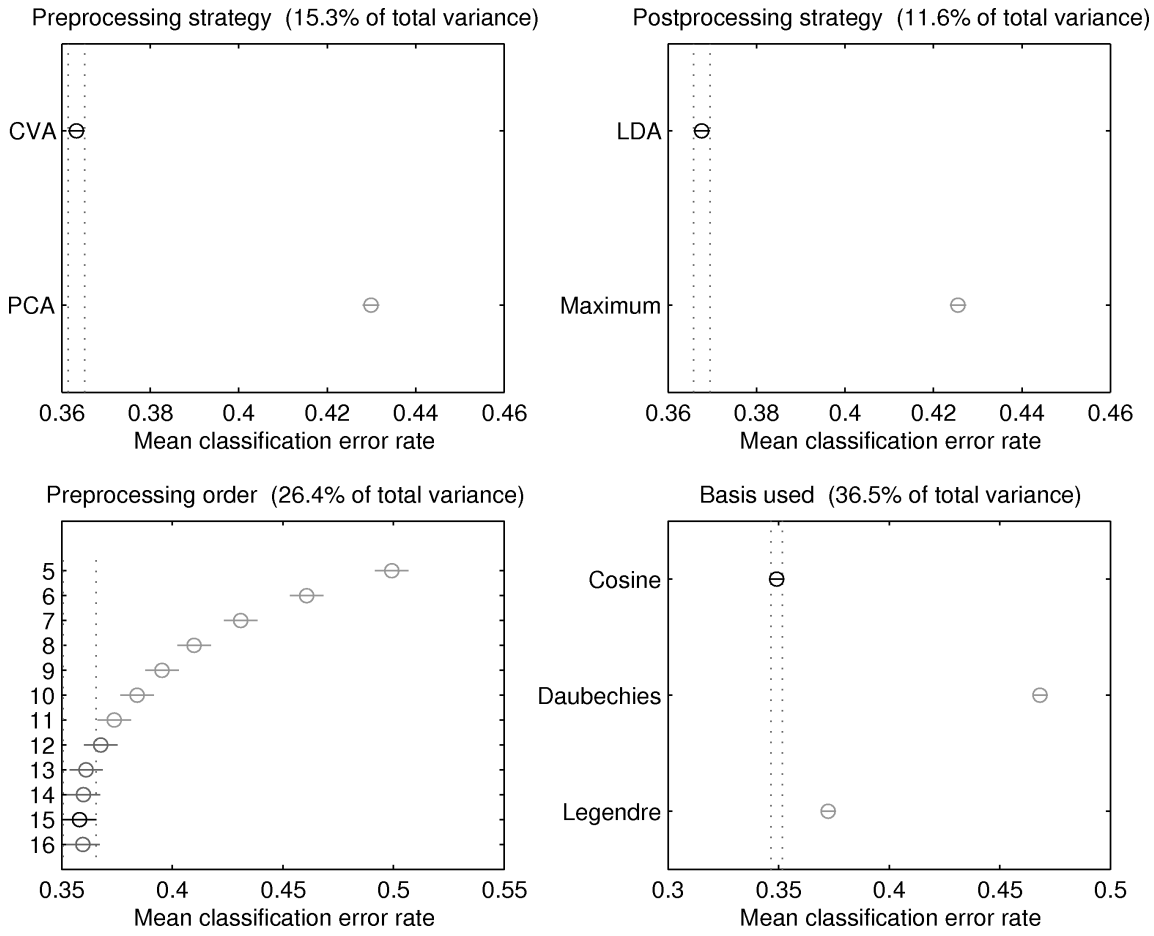

Figure 4.5: Multiple comparison of four-way ANOVA model factors for the DELVE letter recognition task with 1,560 training exemplars. The classification error rate has global mean $\mu = 0.397$, standard deviation $\sigma = 0.085$, and $\frac{\sigma}{\mu} = 0.215$. Residuals account for 10.3% of the total variance.

4.4.2.2 Image segmentation database

The image segmentation task is classification of a sixteen-dimensional vector of attributes as one of seven classes: *brickface, sky, foliage, cement, window, path,* or *grass*. The attributes are various measures taken from a 9-pixel region of a color image. The database was created by the University of Massachusetts Vision Group. DELVE splits

the database into eight disjoint training sets and a common test set of 1,190 exemplars. This database is meant to be run three times, with each run doubling the number of training exemplars available. In the first run, there are 70 training exemplars in eight replicates; in the second, 140 exemplars in eight replicates; and in the third, 280 exemplars in four replicates.

Figure 4.6, Figure 4.7, and Figure 4.8 show the ANOVA models resulting from these three runs. The residuals of these models are quite high, so much of the variance in performance is not explained. A Shapiro-Wilk test of the residuals rejects the hypothesis of normality for ANOVA models with or without interaction effects, indicating that they may not be appropriate for the data corresponding to 70 training exemplars ($p < 10^{-7}$) or 280 training exemplars ($p < 2 \times 10^{-3}$).

From the 140-exemplar run (Figure 4.7), it is possible to conclude that the preprocessing method and dimensionality both have a significant effect on performance. As with the letter recognition database, CVA results in a lower mean error rate than PCA. The best error rate, achieved with thirteen dimensions, is statistically indistinguishable from anything between ten and sixteen dimensions, inclusive. It is not possible to conclude whether a reduction in dimensionality could be beneficial.

The image segmentation database is inconclusive about postprocessing methods. The differences observed were not significant except in the 70-exemplar model, the validity of which is questionable.

The cosine and Legendre bases showed an advantage over the Daubechies basis on the 140-exemplar data, but they showed no significant advantage over each other. These observations are contradicted by the 70-exemplar model.

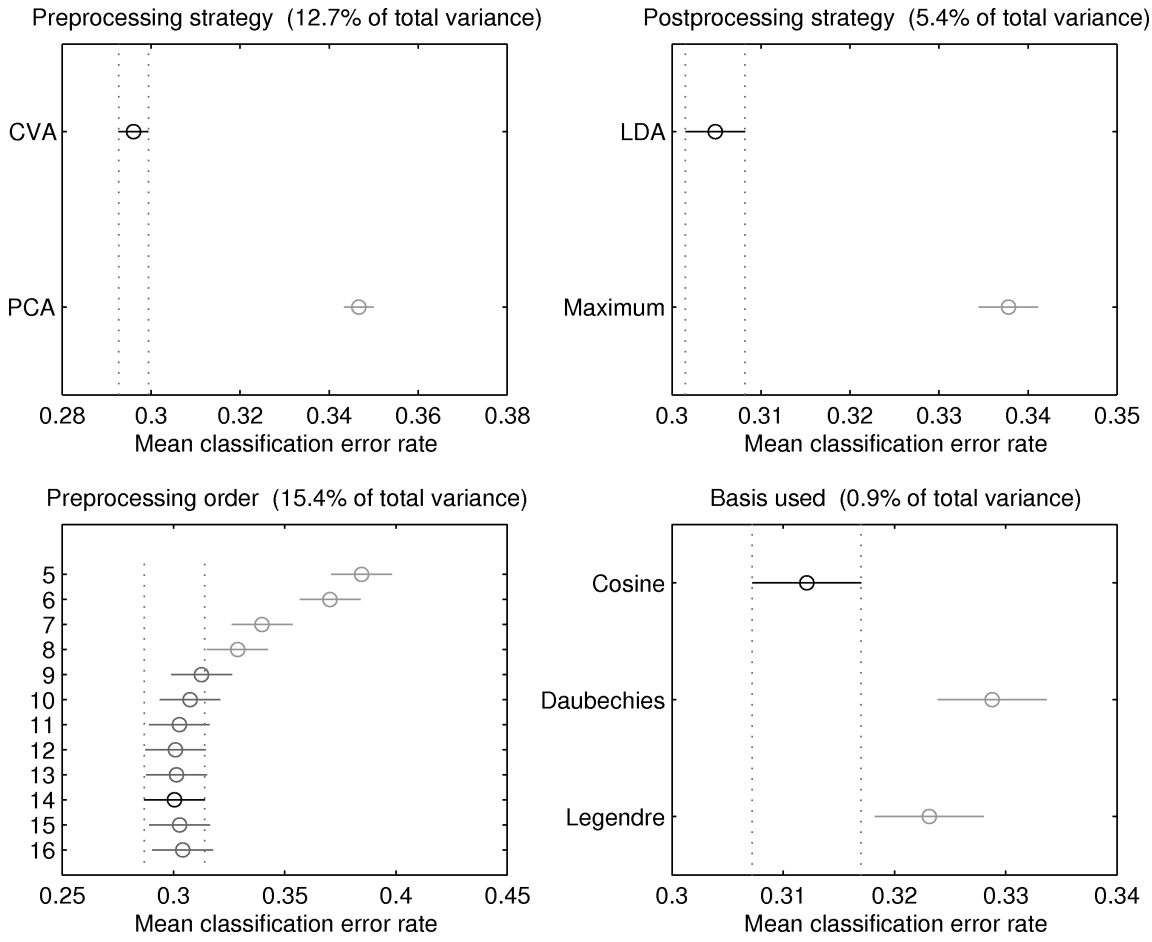

Figure 4.6: Multiple comparison of four-way ANOVA model factors for the DELVE image segmentation task with 70 training exemplars. The classification error rate has global mean $\mu = 0.321$, standard deviation $\sigma = 0.071$, and $\frac{\sigma}{\mu} = 0.221$. Residuals account for 65.6% of the total variance.

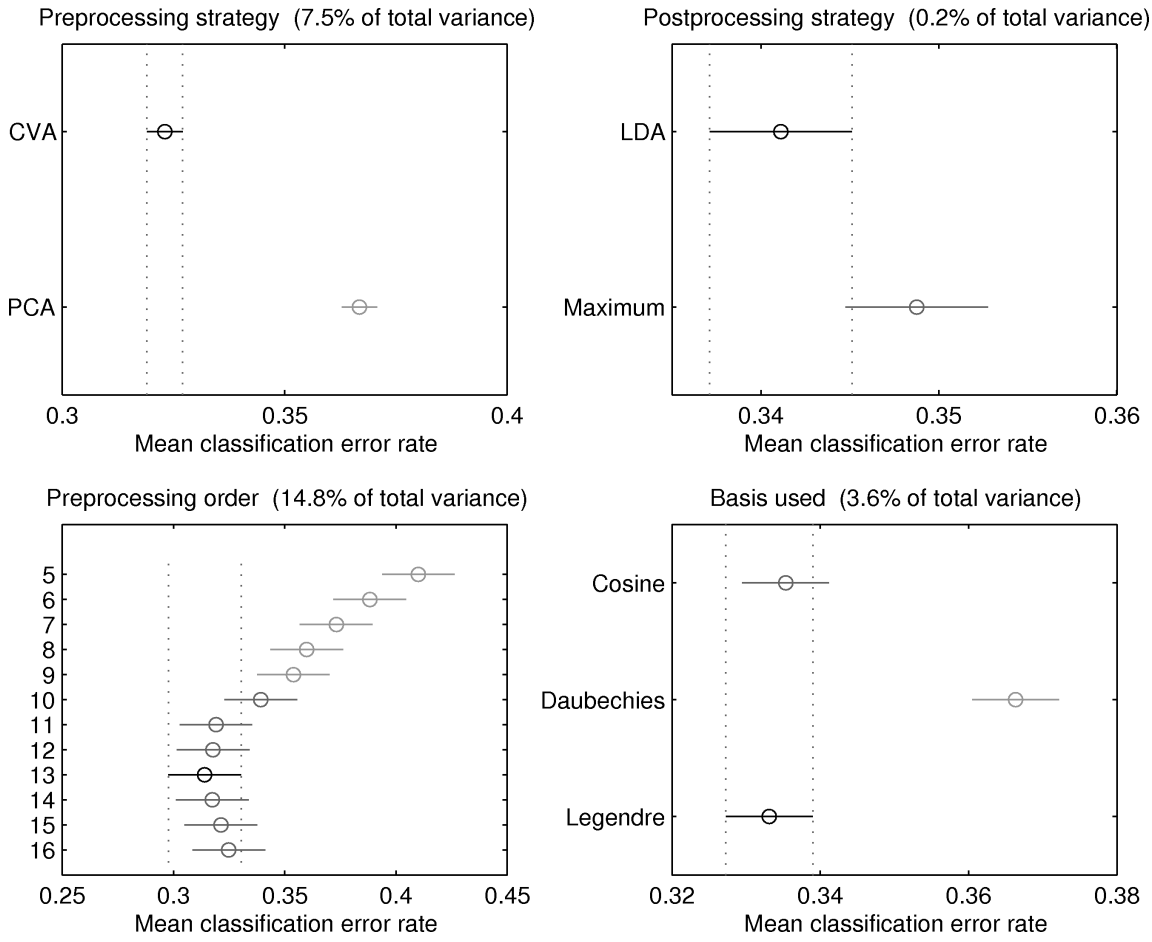

Figure 4.7: Multiple comparison of four-way ANOVA model factors for the DELVE image segmentation task with 140 training exemplars. The classification error rate has global mean $\mu = 0.345$, standard deviation $\sigma = 0.080$, and $\frac{\sigma}{\mu} = 0.232$. Residuals account for 73.9% of the total variance.

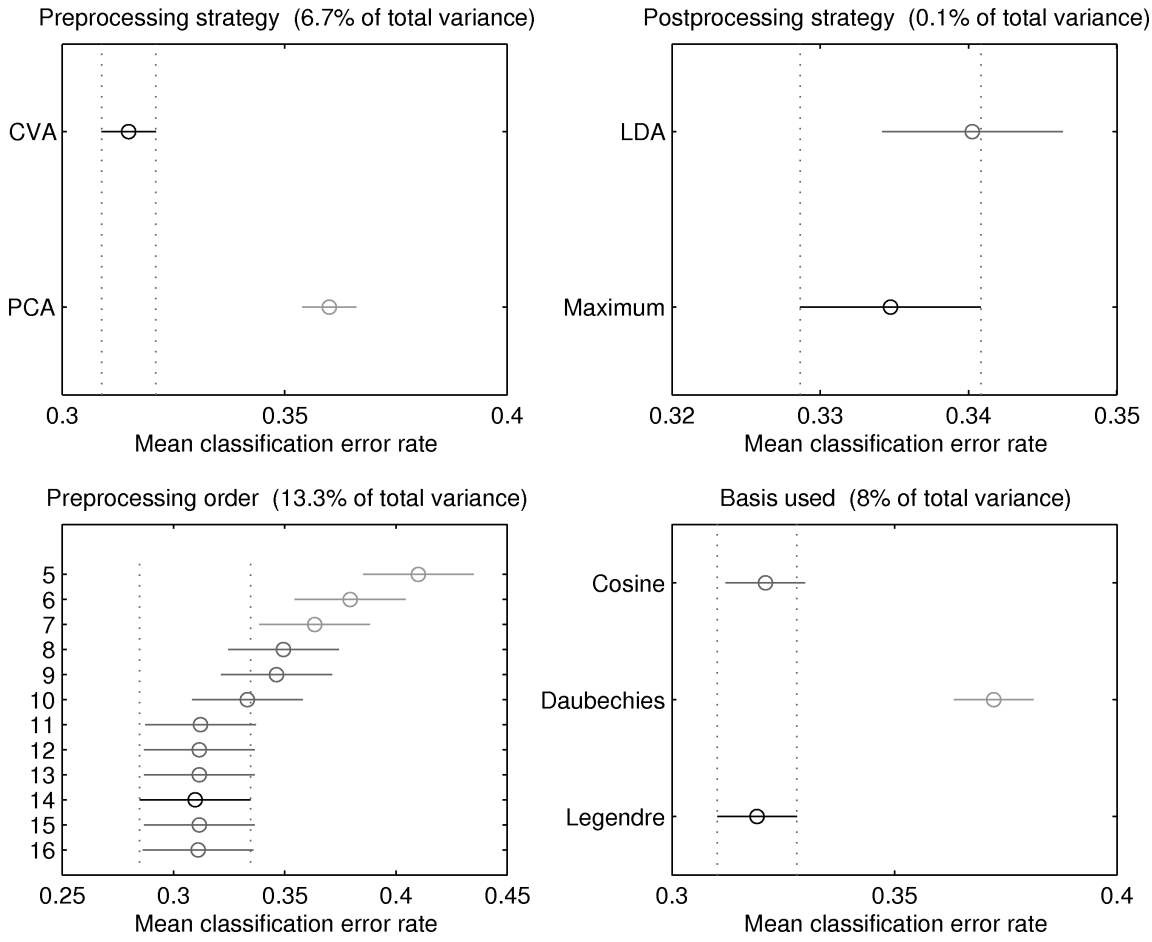

Figure 4.8: Multiple comparison of four-way ANOVA model factors for the DELVE image segmentation task with 280 training exemplars. The classification error rate has global mean $\mu = 0.338$, standard deviation $\sigma = 0.087$, and $\frac{\sigma}{\mu} = 0.258$. Residuals account for 71.8% of the total variance.

4.4.3 Conclusions

This study indicates that it may be advantageous to use CVA, rather than PCA, as the preprocessing component of an orthonormal basis function classification system. Such a limited selection of databases does not give a good estimate of the number dimensions of preprocessed data to keep. In these particular databases, there was no

significant disadvantage to utilizing all of the dimensions available from preprocessing with CVA or PCA.

LDA postprocessing for optimal scoring resulted in a significant reduction in the error rate for the letter recognition task. At worst, LDA postprocessing with adequate data should result in approximately the same error rate as a winner-takes-all scoring system. Since there is little cost or risk to using LDA as a postprocessing method, it is probably advantageous to do so as a matter of course for multiclass problems.

It is clear from this analysis that the choice of basis for an orthonormal basis function classifier can have a significant effect on the results. For the data under consideration, the Daubechies basis was consistently outperformed by the other two bases. However, it is possible to construct data for which the Daubechies basis is optimal. The best basis for a particular problem is a characteristic of that problem, so it is difficult to generalize this result.

The following chapter of this dissertation will reflect these empirical results unless otherwise noted, although for some problems, these may not be the best choices. All preprocessing will be done with CVA, and the maximum number of dimensions will be retained when feasible. LDA will be applied uniformly to discriminant estimates for multiclass problems. These options define a reference system for a particular basis. The only factor that will be varied among orthonormal basis function networks is the basis itself. There are inadequate data to probe the cutoff method, and as a matter of convenience, it is not varied. Each basis uses the cutoff assigned ad hoc in Table 4.1.

Chapter 5

A Comparative Study of Classification Performance

5.1 Introduction

This chapter is an in-depth analysis of DELVE benchmarking results for a variety of classification methods including linear models (LDA), *K*-nearest neighbors (KNN), classification trees (CART), neural networks using backpropagation of error (backprop), and support vector machine kernel methods (SVM) as well as orthonormal basis function neural networks. DELVE (Rasmussen, Neal et al. 1996) compares such classifiers using statistically valid methodologies applied to established databases.

5.1.1 Orthonormal basis function neural networks

This section summarizes the specific steps involved in orthonormal basis function classification. Flowcharts of the orthonormal basis function neural network training and testing procedures described here are shown in Figure 5.1 and Figure 5.2.

The data are prepared by computing all extended canonical variates. For some databases this may yield too many variates, for instance more than twenty. In that case an automated scree test may be applied to the extended variates (those obtained through PCA of the residual) for dimension reduction.

An initial set of orthonormal basis functions is selected by using the ordering criteria in Section 3.6 to determine the first N basis function, where N is the number of training data points.

For each class, the corresponding coefficients of the orthonormal series expansion of Devroye's discriminant function (3.25) are estimated (3.29), as are the associated variance (3.35) and squared bias terms (3.41).

The orthonormal series is truncated to minimize its mean integrated squared error (MISE) as in Equation (3.50), and individual terms that increase the expected MISE of the model are also eliminated (3.52).

For multiclass problems, LDA is employed to determine decision boundaries between the estimated discriminant functions for the various classes (Section 4.3).

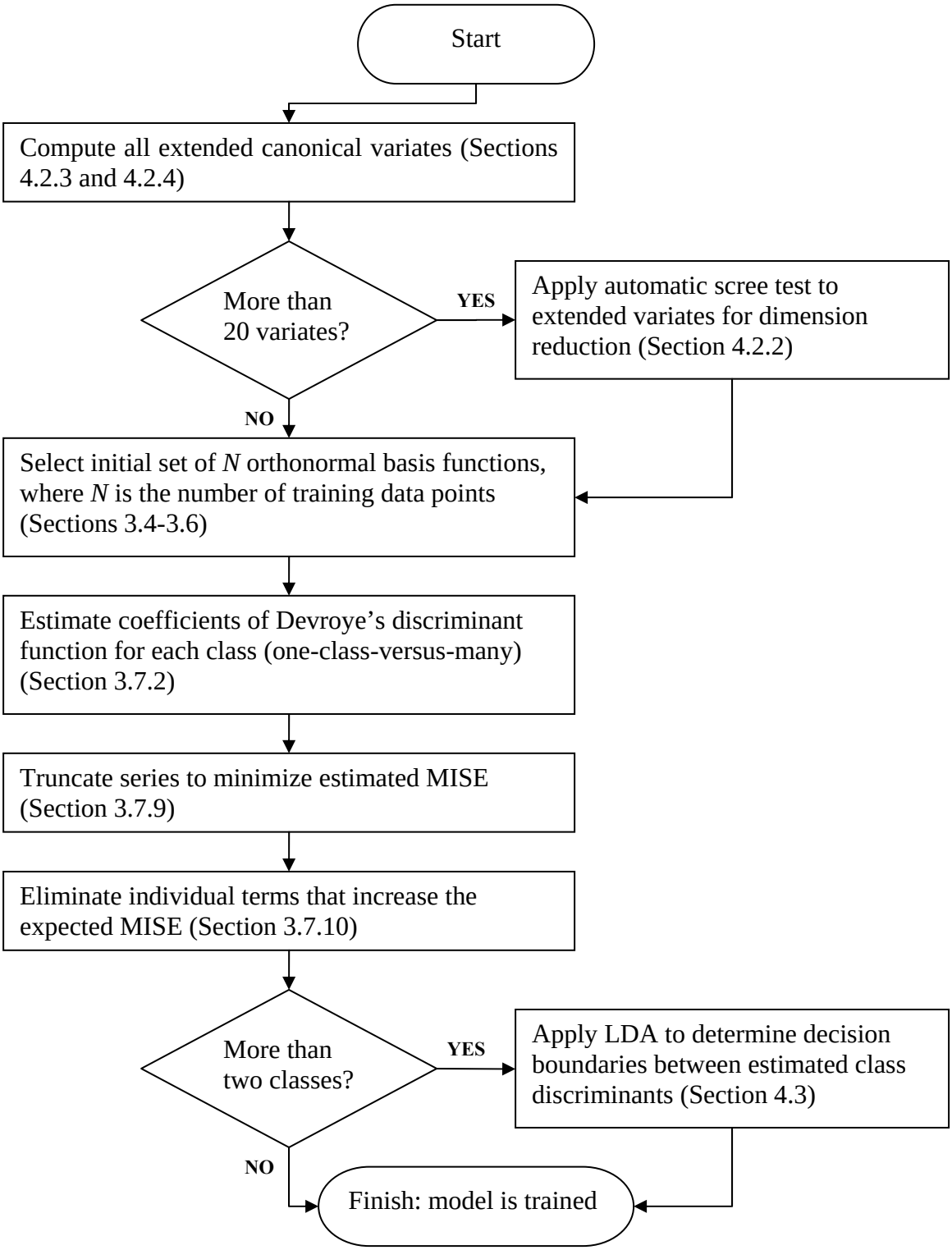

Figure 5.1: Flowchart of orthonormal basis function network training procedure.

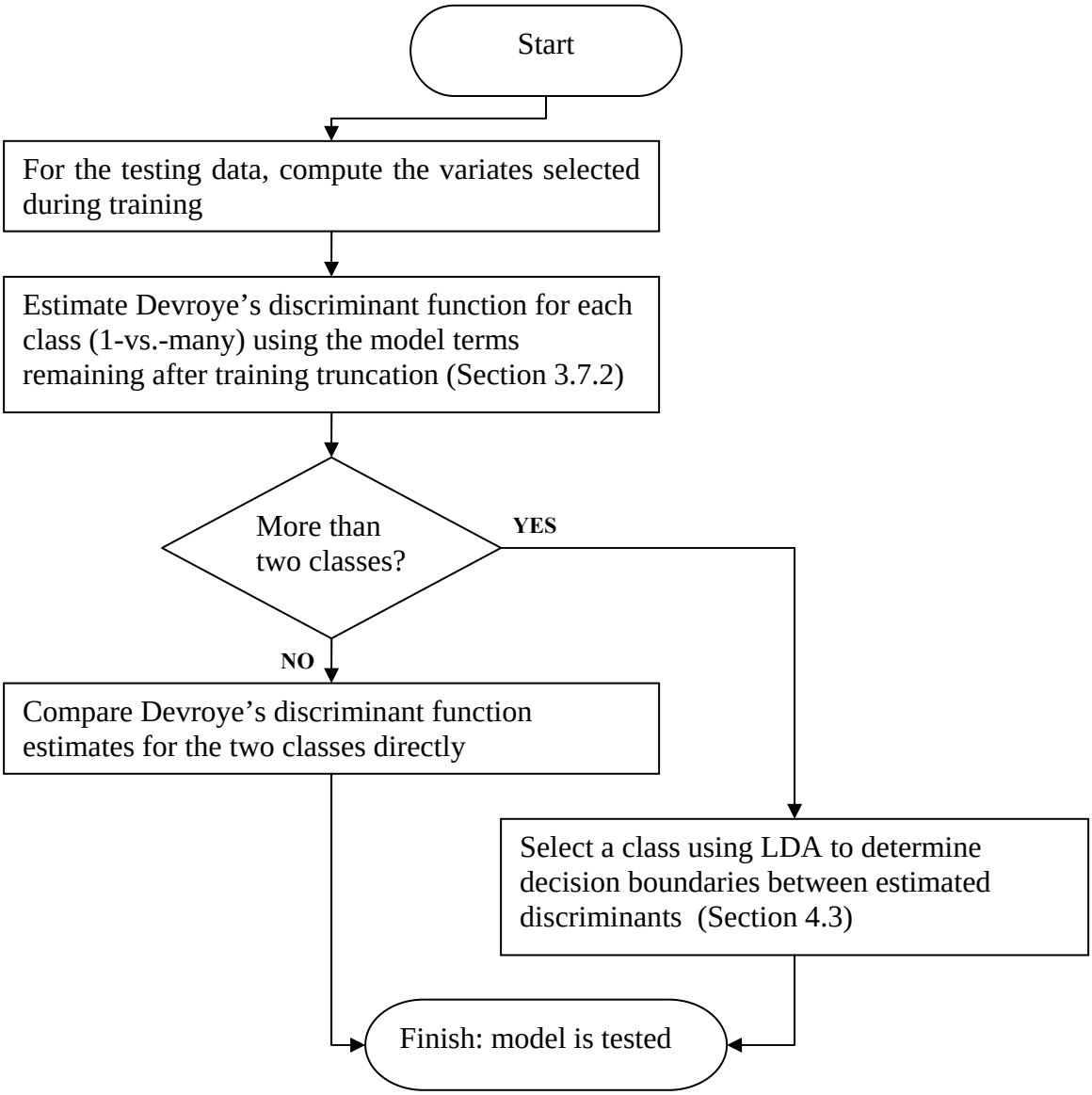

Figure 5.2: Flowchart of orthonormal basis function network testing procedure.

5.1.2 Backpropagation neural networks

Multilayer perceptron networks using backpropagation of error (backprop) for training are a system initially developed by Werbos (1974) and implemented in this

dissertation with the Netlab toolbox (Nabney and Bishop 2001). A three-layer perceptron utilizes hyperbolic tangent hidden layer nodes and linear output layer nodes. The number of output layer nodes is equal to the number of classes for a particular classification task.

The number of hidden layer nodes J_h is determined by a divide-and-conquer search to minimize the generalized cross-validation (GCV) criterion. For each potential value j of J_h, a backprop network containing j hidden layer nodes is trained for 1,000 iterations on the training data set. J_h is selected among these candidates to minimize the GCV (Craven and Wahba 1979; Friedman 1991; Stone, Hansen et al. 1997):

$$J_h = \arg\min_j \frac{MSE_j}{\left[1 - \frac{a(j-1)}{N}\right]^2}, \tag{5.1}$$

where MSE_j is the mean squared error

$$MSE_j = \frac{1}{N}\sum_{i=1}^{N}[\hat{f}(\mathbf{x}_i) - y_i]^2 \tag{5.2}$$

of the backprop network over the training set $\{\mathbf{x}_i\}$ with corresponding output class indicator function values $\{y_i\}$. $a = 2.5$ is a typical value for the GCV hyperparameter (Stone, Hansen et al. 1997) and is used for model selection for this backprop implementation.

The selected backprop network with J_h hidden layer nodes is trained to a total of 5,000 iterations before being applied to the DELVE test data.

5.1.3 Classification and regression trees (CART)

Classification and Regression Trees (CART) are a family of decision tree methods developed by Breiman, Friedman et al. (1984). A tree is constructed by progressively splitting the data into disjoint clusters by minimizing a split criterion at each step. The CART method used in this comparative study uses one-dimensional splitting to minimize the tree's total *Gini impurity*, defined at each node as:

$$i(N) = 1 - \sum_{c=1}^{C} \left(\hat{P}\left(y = c \mid \mathbf{x} \in N \right) \right), \tag{5.3}$$

where $\hat{P}\left(y = c \mid \mathbf{x} \in N \right)$ is the proportion of $\left(\mathbf{x}, y \right)$ pairs at node N that are associated with class c (Duda, Hart et al. 2000).

The full classification tree is computed for each of ten leave-out-ten-percent cross-validation samples to estimate the optimal pruning level. The final model is obtained by applying this pruning level to a full classification tree computed using all of data points.

In this comparative analysis, dimension reduction was not used prior to fitting the classification trees.

5.1.4 K-nearest neighbors (KNN)

As reviewed by Agrawala (1977), the *K*-nearest neighbors (KNN) method was first described by Fix and Hodges (1951). It remains a popular method for pattern classification due to its simplicity and statistical consistency. That many statistical properties of KNN are known also makes KNN useful for benchmark comparisons.

KNN requires that a distance metric $d(x_1, x_2)$ be defined over the domain of x. This is often taken to be the Euclidean distance metric. In KNN, the parameter k determines the number of elements in the set

$$\arg\min_{\{x_1^*, x_2^*, \ldots x_k^*\}} \sum_{i=1}^{k} d(x, x_i^*), \tag{5.4}$$

where $\{x_1^*, x_2^*, \ldots x_k^*\}$ are unique elements taken from the set $\{x_1, x_2, \ldots, x_n\}$ of known exemplars. These k elements are the nearest in distance to the point of interest x, and from the classes associated with these elements, it is possible to construct the class probability estimator for class C at x. Let y be the class associated with x and let y_i^* be the class associated with x_i^*. Then

$$\hat{P}(y = C) = \frac{1}{k} \sum_{i=1}^{k} 1(y_i^* = C) \tag{5.5}$$

(Bishop 1995).

In this comparative analysis, KNN models were applied without dimension reduction to data normalized to have zero mean and unit variance in each dimension. The optimal k was estimated by leave-out-one cross-validation.

5.1.5 Linear discriminant analysis (LDA)

Linear discriminant analysis (Mardia, Kent et al. 1979; Ripley 1996), as described in Section 4.3.1, was applied directly to the original variates of benchmark databases. In some cases, data were linearly dependent, requiring dimension reduction to ensure that input data had rank equal to its dimensionality.

5.1.6 Support vector machines (SVM)

This work used the support vector machine toolbox created by Cawley (2000). The toolbox employs Vapnik's (1995) support vector machine for classification by means of the sequential minimal optimization (SMO) algorithm established by Platt (1999). The algorithm represented utilizes a Gaussian radial basis kernel. Multiple classification was performed using the DAGSVM directed acyclic graph method (Platt, Cristianini et al. 2000). The DAGSVM method constructs a separate SVM classifier for each possible class pairing. These binary classifiers are combined according to the connections in the directed acyclic graph, which determines the order in which classes undergo elimination.

$\xi\alpha$ cross-validation (Joachims 2000) was used to select the kernel radius r and regularization parameter C. These were determined by performing a simplex minimization of the error estimate $\mathrm{Err}_{\xi\alpha}$ over the space of (r, C) for each binary classifier in a DAGSVM system.

5.2 Methods and metrics

This section summarizes methodologies incorporated in the DELVE software (Rasmussen, Neal et al. 1996), as described in detail in the DELVE Manual (Rasmussen, Neal et al. 1996), along with related methodologies used in this dissertation to extend the paired comparisons of algorithms in DELVE to multiple comparisons and their interpretation. DELVE employs one of two types of ANOVA model depending on whether disjoint test datasets are specified to correspond to the training replicates (the "hierarchical" model) or a single common test dataset is used for all training replicates.

5.2.1 DELVE bivariate comparisons with disjoint test data

When test datasets are disjoint, DELVE represents a classifier's loss on training set i and test case j by the linear model

$$y_{ij} = \mu + a_i + \varepsilon_{ij}, \tag{5.6}$$

in which the variable a_i accounts for the variation in classification performance measured across training sets i (Rasmussen, Neal et al. 1996). It is assumed for the sake of ANOVA analysis that a_i and ε_{ij} are independent Gaussian random variables. The same model can be used to represent the difference in performance between two different classifiers when y_{ij} is instead taken to be the difference in classification performance. For paired comparisons, the significance of this difference can be determined using a *t*-test. The relevant *t* statistic is

$$t = \overline{y}\left(\frac{1}{I(I-1)}\sum_i (\overline{y}_i - \overline{y})^2\right)^{-1/2} \tag{5.7}$$

with $I-1$ degrees of freedom (Rasmussen, Neal et al. 1996).

5.2.2 DELVE bivariate comparisons with common test data

In some cases, insufficient data exist to support the use of disjoint test datasets for each replicate. In these cases, DELVE employs a single common test set . Under these circumstances, it is necessary to model the effects of the test cases on the loss (Rasmussen, Neal et al. 1996):

$$y_{ij} = \mu + a_i + b_j + \varepsilon_{ij} \tag{5.8}$$

As in (5.6), a_i accounts for training set variation. The variable b_j accounts for variation in loss due to test case j (Rasmussen, Neal et al. 1996). For paired tests, this model can be used to represent the difference in performance between two classifiers when y_{ij} is taken instead to be the difference in classification performance. Rasmussen, Neal et al. employ a quasi-*F* test to test the hypothesis that the mean difference between the performances of the classifiers is nonzero. If the *p*-value of this test is greater than 0.05, the difference is taken to be not significant.

5.2.3 Multiple comparisons using DELVE

The DELVE methodology allows bivariate comparisons between any two algorithms. However, it does not provide an explicit way to perform multiple comparisons.

A conservative approach to multiple comparisons of n candidates utilizes the Bonferroni correction,

$$\alpha' = \frac{\alpha}{n(n-1)/2}, \tag{5.9}$$

which constrains the experiment-wide probability of error in determining the direction of performance differences to be less than or equal to α. Although this is very conservative in determining significance and direction of results comparisons, it assigns large confidence intervals. These might lead one to believe that an individual algorithm could potentially perform much better than it actually does. Adjusting for multiple comparisons in this way makes it very important not to draw conclusions from the lack of significance.

The staircase plots presented in this dissertation reflect pairwise differences between algorithm performances. The Bonferroni correction was not used because it might mask significant statistical differences between any two individual algorithms of interest.

5.2.4 Staircase plotting of multiple comparison data

Basford and Tukey (2000) introduced a new plot format that presents results and their multiple comparisons in a single display. The staircase plot organizes multiple comparison data so they may be readily interpreted. Significant differences are represented by the relative positions of results on this plot. A result that is on a higher tier and to the right of a given result is significantly different unless otherwise indicated. A result that is on a lower tier and to the left is likewise significantly different. Comparisons must take both the tier and horizontal position into account. A result on the same tier is never significantly different; nor is a result to the right on a lower tier or to the left on a higher tier.

A disadvantage of the staircase plot is that it requires that the order of presentation of results be flexible. However, the amount of information that can be presented in this manner would otherwise require multiple displays, one to show the results with error estimates and another to represent all of the $\frac{(n)(n-1)}{2}$ bivariate comparisons between n results. By using horizontal position for information display, the staircase plot successfully consolidates results and their bivariate comparisons. For this reason, staircase plots are used throughout this chapter.

5.3 DELVE letter recognition benchmark

The letter recognition database (Frey and Slate 1991; Rasmussen, Neal et al. 1996; Hettich, Blake et al. 1998) contains 20,000 cases, each consisting of sixteen input characteristics and an associated output class, one of twenty-six uppercase letters. DELVE segments the database into six test sets of 1,773 cases each and six disjoint training sets with 390, 780, or 1,560 exemplars each.

When trained on only 390 exemplars (Figure 5.3), six algorithms performed almost uniformly well, with between 64% and 68% correct classification rates. These included the cosine and Legendre orthonormal basis function networks, backprop, KNN, LDA, and SVM. Among these systems in the top group, only two significant differences were observed at $\alpha = .05$. The cosine system had a significantly better classification rate (66.1%) than the Legendre (64.3%), and the SVM performance (67.3%) was significantly higher than that of KNN (64.4%). Three systems failed to make the top group. Significantly lower performance was observed using the Daubechies (54.8%), CART (52.3%), and Haar (20.9%) classifiers.

With 780 exemplars for training, the best-performing group of algorithms, between 73% and 78% correct classification rates, consisted of backprop, KNN, and SVM. Within this group, SVM performed significantly better than KNN. Next in order of classification rate was the cosine basis function network (70.1%). This outperformed a third group consisting of the Legendre basis function network and LDA. Apart from the Haar classifier, the Daubechies and CART systems were least suited to the 780-exemplar letter classification task.

When trained with 1,560 exemplars, a similar ordering was observed, with greater differentiation between the algorithms. Only backprop (86.7%) and SVM (85.7%) had classification rates in a top grouping. In order, they were followed by the KNN, cosine, and Legendre systems. A lower group consisted of the Daubechies, CART, and LDA classifiers. Within this group, LDA had a significantly higher correct classification rate than did the Daubechies classifier. Only the Haar classifier performed beneath this lower group.

On all of these tasks, orthonormal basis function neural networks employing cosine and Legendre bases were competitive with popular standard classifiers, equalling or exceeding the classification performance of LDA. They were most competitive when only 390 training exemplars were available. On this task, their performance was statistically indistinguishable from backprop, KNN, LDA, and SVM. Of the six top-grouped systems on the 390-exemplar task, it appears that backprop, KNN, and SVM benefitted most from the availability of additional training data, while LDA benefitted the least.

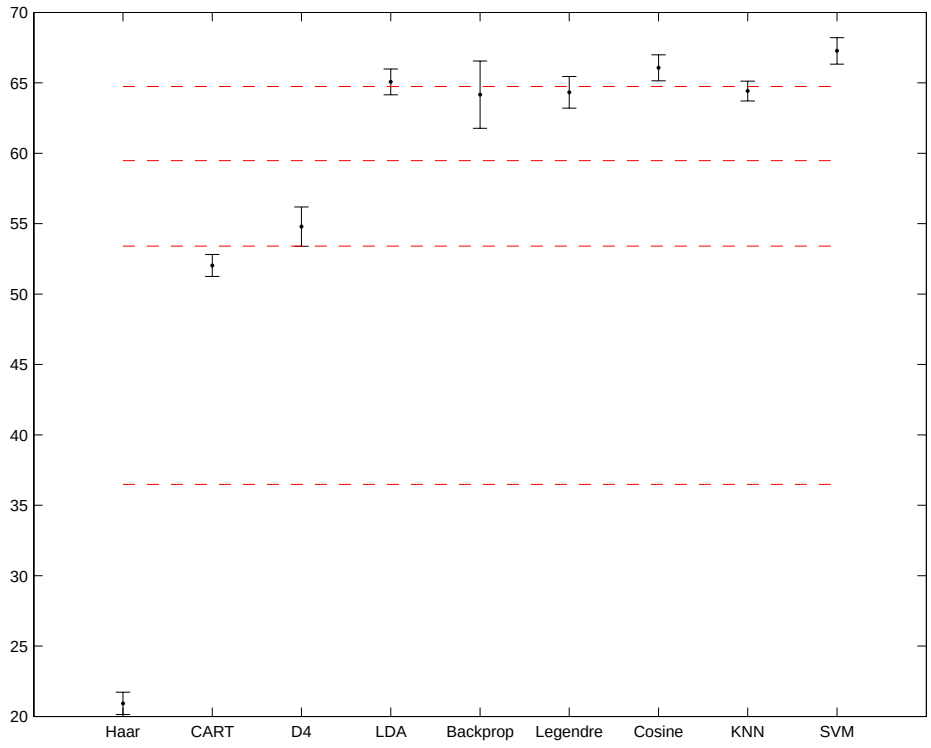

Figure 5.3: Staircase plot of DELVE benchmark performance for the letter recognition task with 390 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance. On this task, six algorithms had equivalent performance in the top tier, and all but one of these did significantly better than the two second-tier systems.

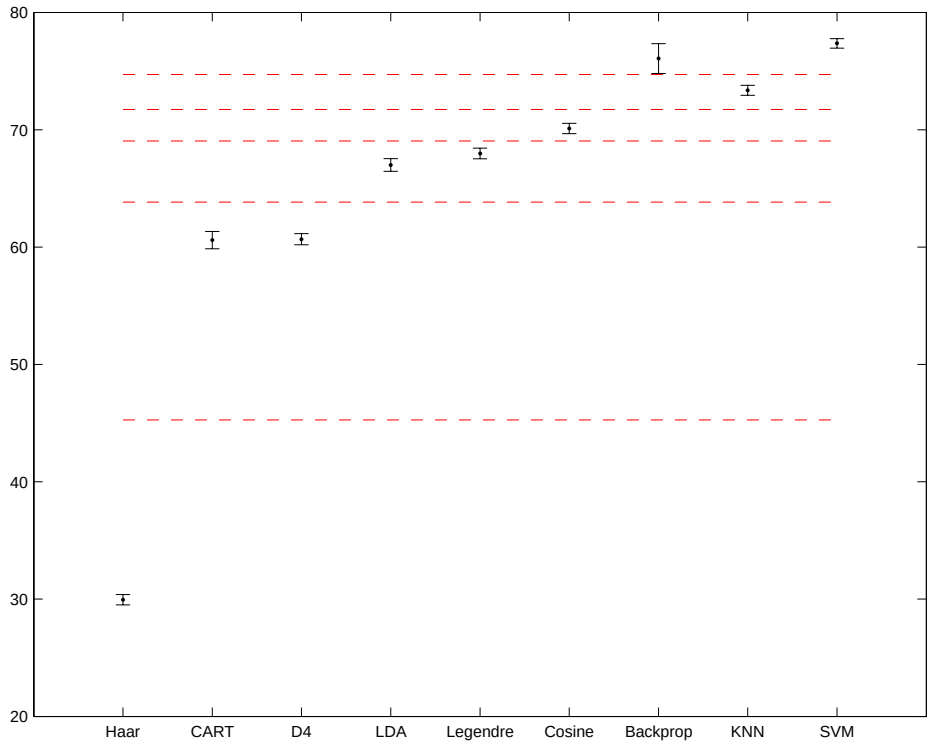

Figure 5.4: Staircase plot of DELVE benchmark performance for the letter recognition task with 780 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

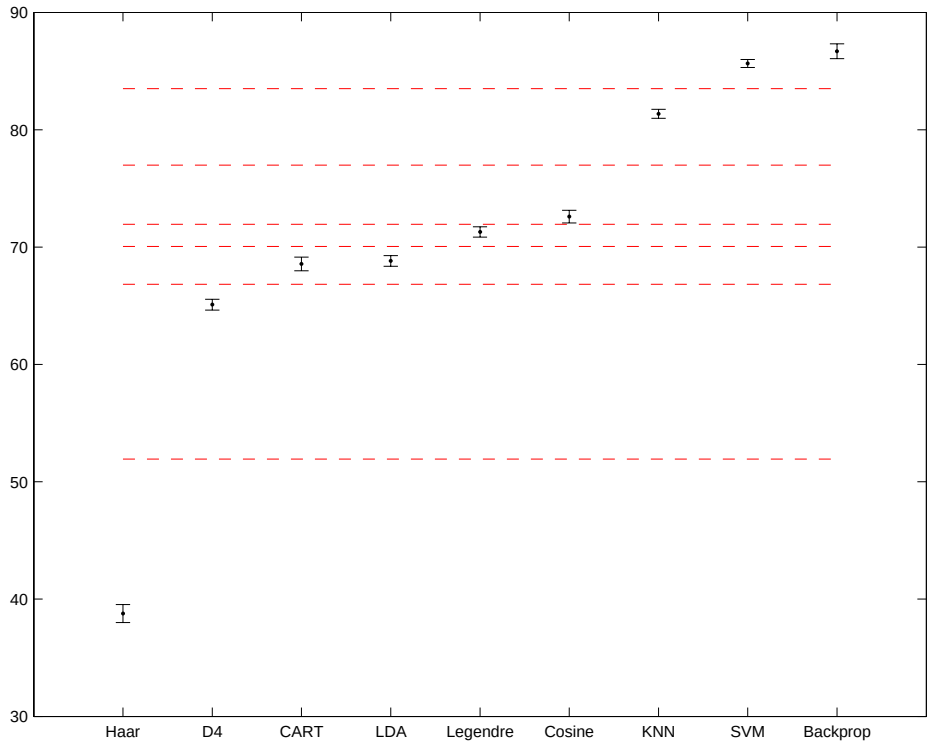

Figure 5.5: Staircase plot of DELVE benchmark performance for the letter recognition task with 1,560 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

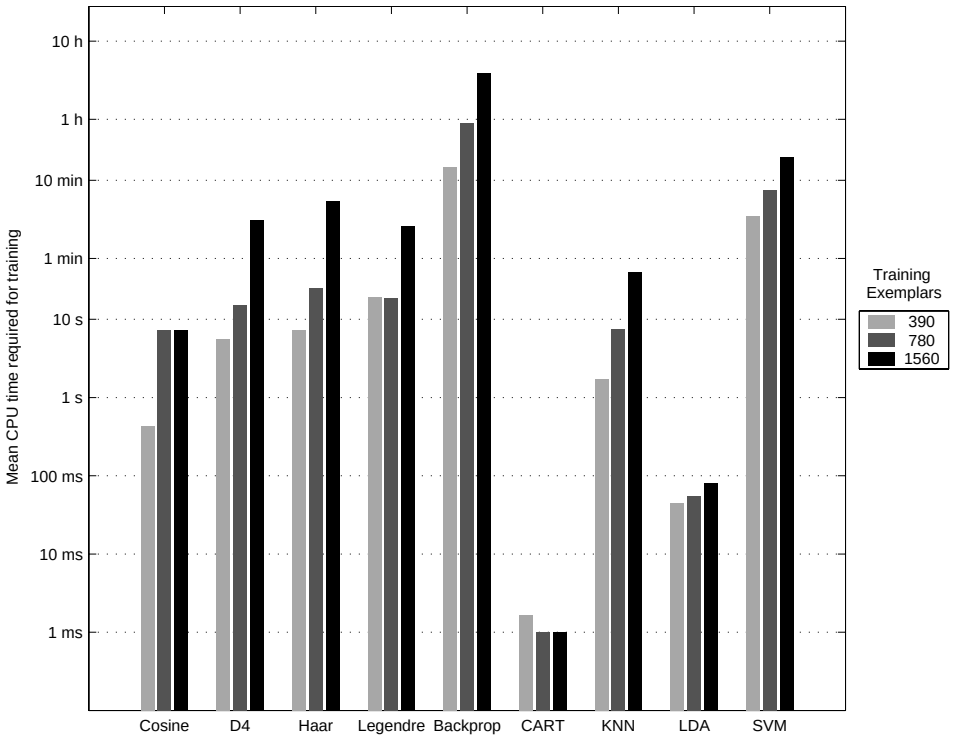

Figure 5.6: Mean CPU time required to train four orthonormal basis function networks (left) and five other classifiers (right) on the DELVE letter recognition database. CPU times are shown on a logarithmic scale. Values less than 1 ms are rounded up to 1 ms.

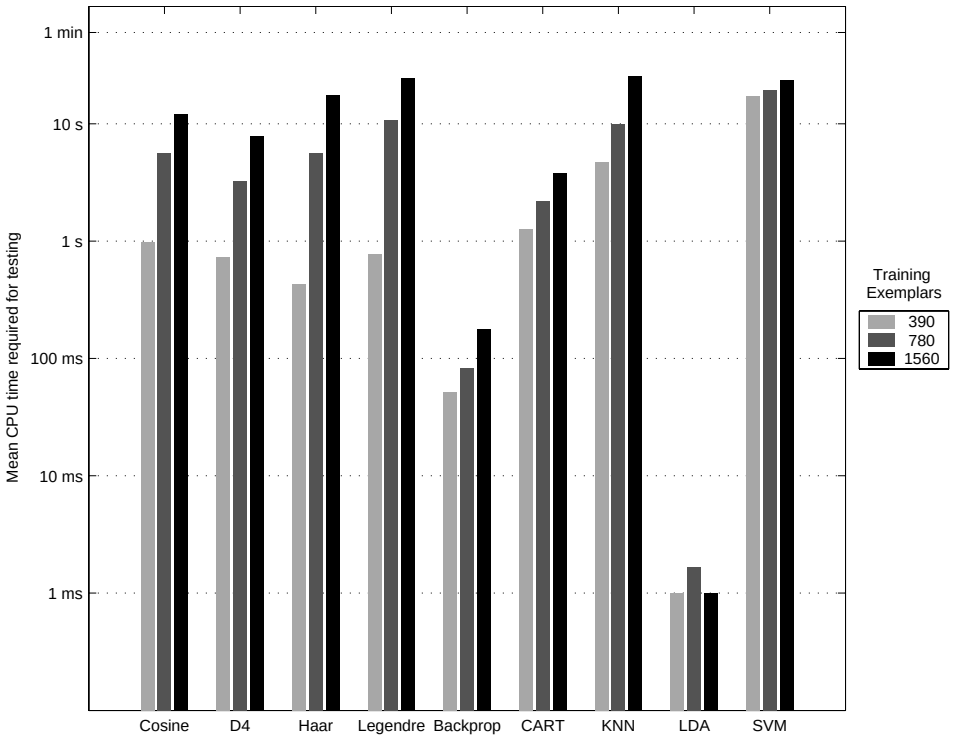

Figure 5.7: Mean CPU time required to test four orthonormal basis function networks (left) and five other classifiers (right) on 1,773 exemplars from the DELVE letter recognition database. CPU times are shown on a logarithmic scale. Values less than 1 ms are rounded up to 1 ms.

5.4 DELVE image segmentation benchmark

The University of Massachusetts Vision Group's image segmentation database (Rasmussen, Neal et al. 1996; Hettich, Blake et al. 1998) contains 2,310 cases, each consisting of sixteen local image attributes and an associated output class, one of seven textures. The DELVE specification for this database utilizes the data three times, with the number of training exemplars doubling in each run. DELVE segments the database into four or eight disjoint training replicates, depending on the number of training

exemplars (70, 140, or 280) used for a given run. A common test set of 1,290 cases is set aside.

Regardless of the number of training exemplars provided, top performing systems included cosine, Daubechies and Legendre orthonormal basis function networks and LDA. Only these algorithms were in the top group when trained on 70 exemplars, all falling between 72% and 75% correct classification. The performance of CART (67.7%) was not significantly worse than the LDA system on this task, nor was it significantly better than backprop or KNN, or SVM. These three algorithms and CART formed a second group for which the classification rates, between 63% and 68%, were not significantly different. The Haar orthonormal basis function network did significantly worse than any other system benchmarked on the image segmentation database regardless of the number of training exemplars provided.

When 140 training exemplars were provided, the top performing group of algorithms became more inclusive. In addition to the cosine, Daubechies and Legendre orthonormal basis function networks and LDA, top performers included CART and KNN. Among these algorithms, the only significant difference observed at $p = .05$ was between the cosine network (69.6% correct) and KNN (65.6% correct). All of these algorithms but CART and KNN performed significantly better than backprop (60.4% correct). All of the top group algorithms but CART performed significantly better than SVM (61.8% correct), which with backprop formed a second performance grouping.

With only four replicates, the 280-exemplar benchmark had few significant differences. There were no significant differences between the classification rates of the

cosine, Daubechies, and Legendre orthonormal basis function networks, CART, KNN, LDA, and SVM. Backprop had significantly lower classification performance than the Daubechies, Legendre (72.3% correct), and CART classifiers.

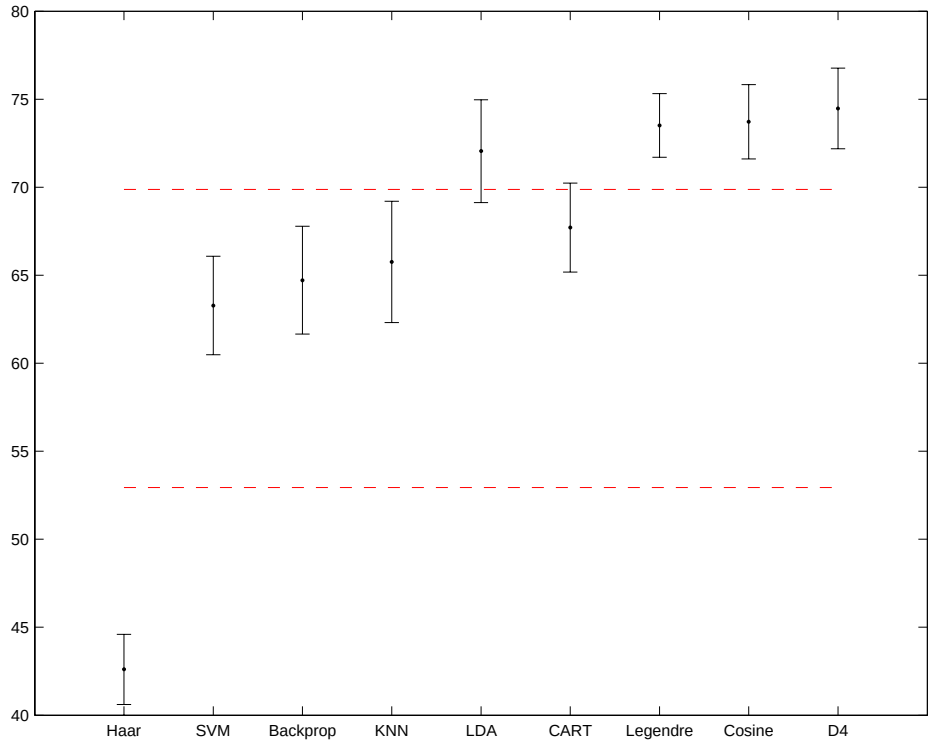

Figure 5.8: Staircase plot of DELVE benchmark performance for the image segmentation task with 70 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

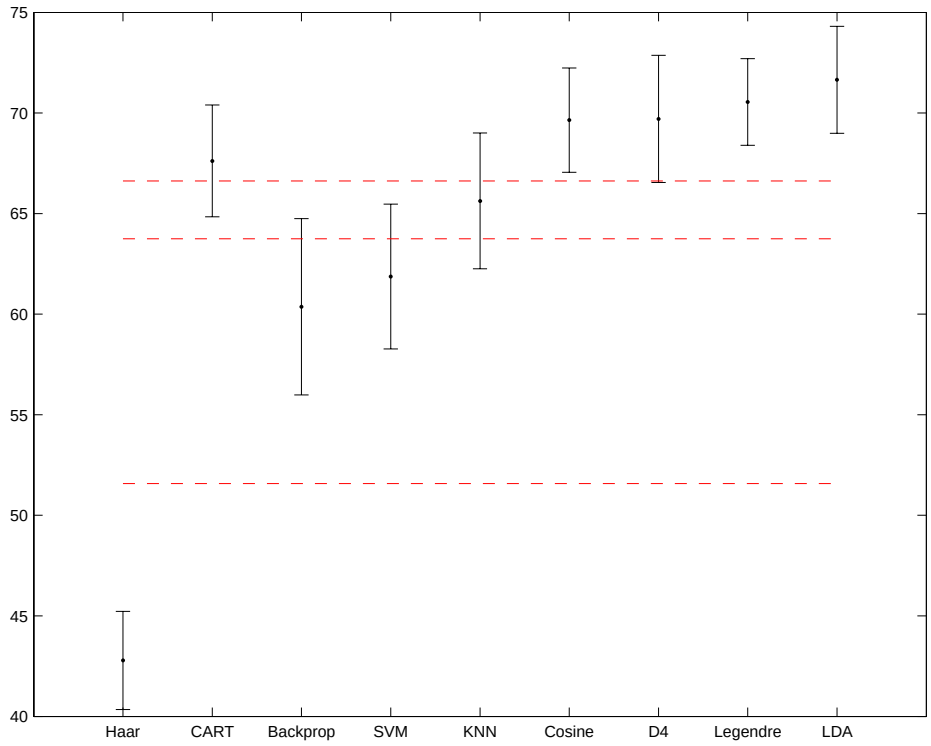

Figure 5.9: Staircase plot of DELVE benchmark performance for the image segmentation task with 140 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

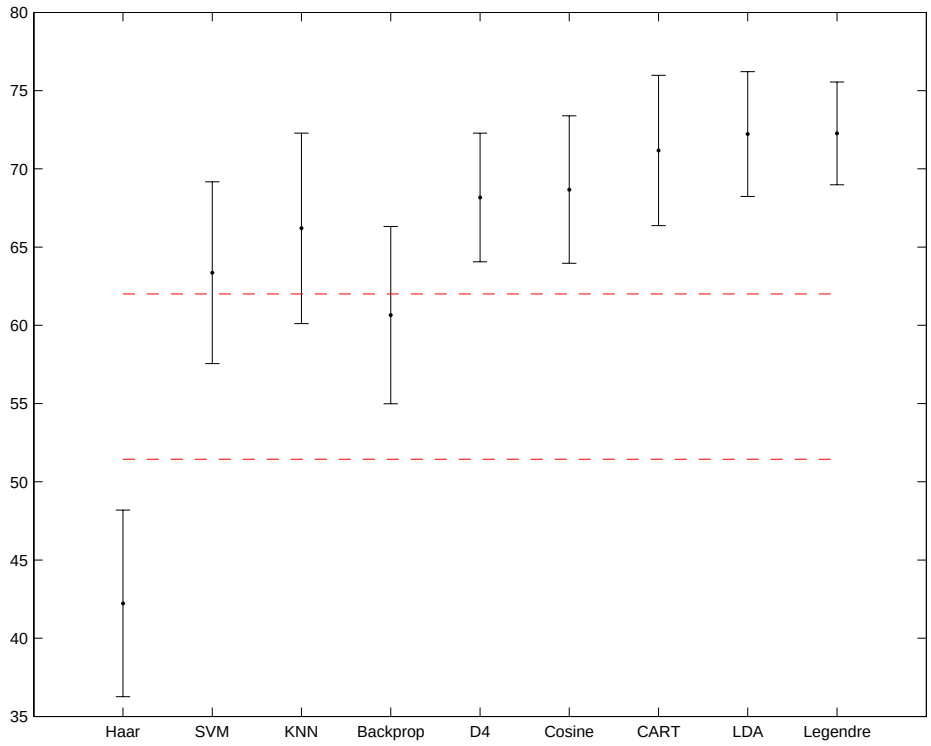

Figure 5.10: Staircase plot of DELVE benchmark performance for the image segmentation task with 280 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

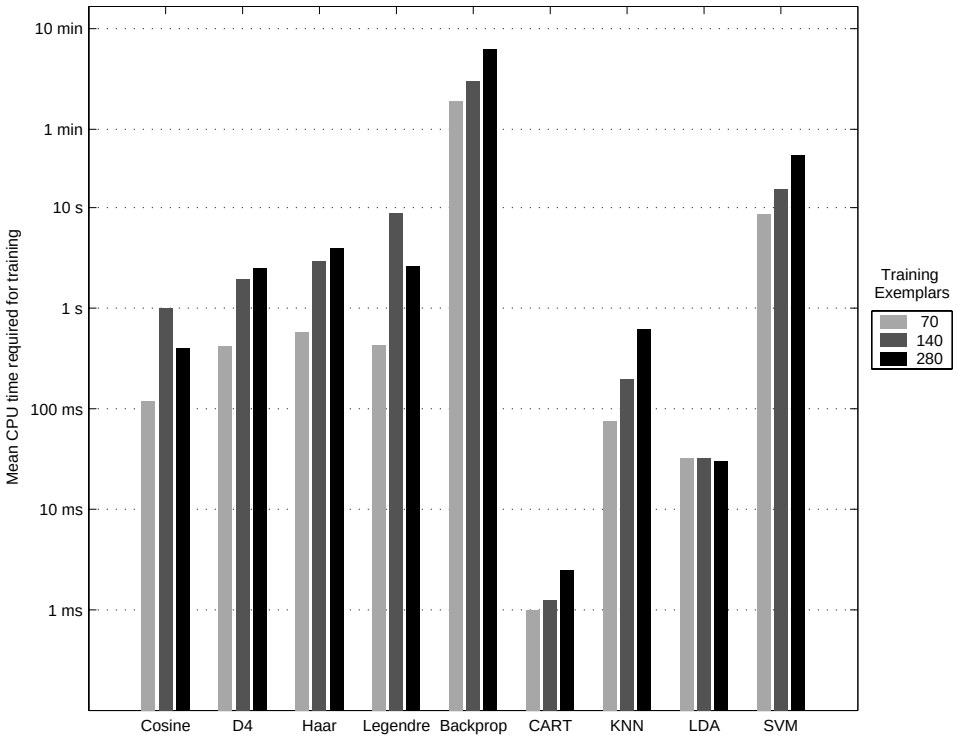

Figure 5.11: Mean CPU time required to train four orthonormal basis function networks (left) and five other classifiers (right) on the DELVE image segmentation database. CPU times are shown on a logarithmic scale. Values less than 1 ms are rounded up to 1 ms.

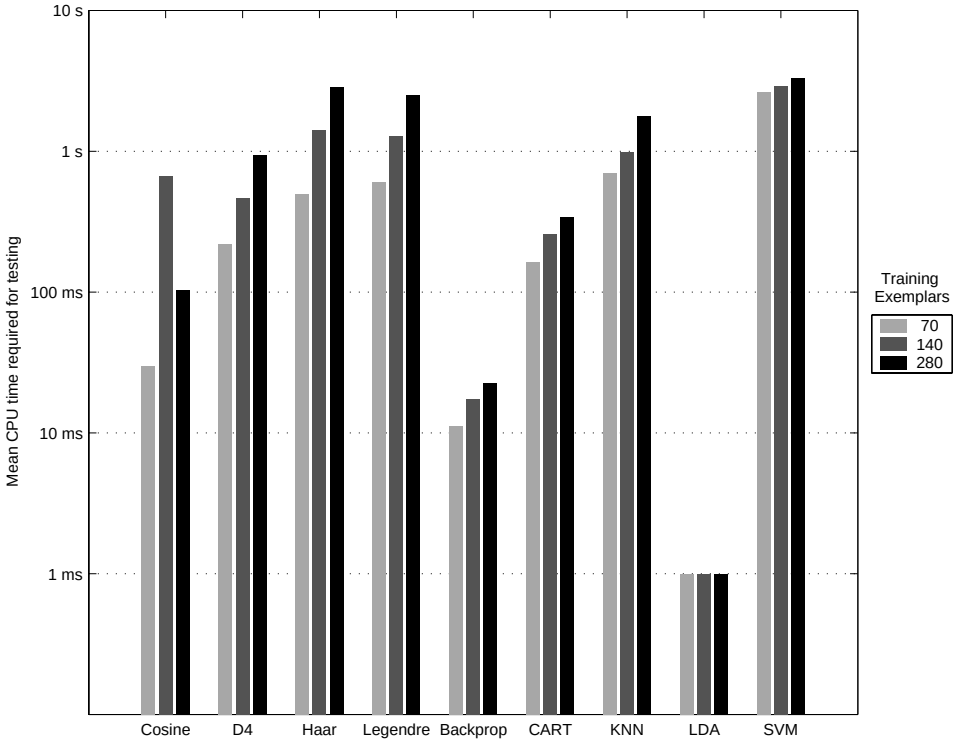

Figure 5.12: Mean CPU time required to test four orthonormal basis function networks (left) and five other classifiers (right) on 1,190 exemplars from the DELVE image segmentation database. CPU times are shown on a logarithmic scale. Values less than 1 ms are rounded up to 1 ms.

5.5 DELVE Titanic survival prediction benchmark

The Titanic database (Rasmussen, Neal et al. 1996) contains 2,201 exemplars. Each represents a passenger on the Titanic. The task is to classify an individual as a survivor or victim based on three variables: class of passage, sex, and whether a child or an adult. DELVE utilizes the training data to form sets of eight replicates with 20, 40, 80, and 160 exemplars.

Provided with 20 training exemplars, all algorithms tested, with the exception of SVM and CART, had equivalent classification performance. SVM performed significantly worse than all other algorithms, achieving 64.8% correct while other systems all achieved at least 69.3% correct. This value corresponded to the performance of the CART classifier, which was significantly worse than that of both LDA (72.5%) and backprop (72.3%). The remaining algorithms (all orthonormal basis function classifiers and KNN) had performance that was not significantly worse than backprop or LDA, nor significantly better than CART.

With 40 training exemplars, there were two tiers of classifiers based on performance. No significant difference was observed between LDA (74.7% correct), backprop, CART, and the Legendre-based classifier. All other algorithms could be ruled out from having the best performance on this task, and these formed a second tier. The following significant differences were observed between systems in the first tier and second tier: LDA had a significantly higher correct classification rate than all systems in the second tier, backprop performed better than all systems in the second tier except KNN, and CART outperformed the SVM classifier. The Legendre-based system could not be differentiated statistically from any other system.

With 80 training exemplars, first-tier systems included backprop (76.5% correct), LDA, KNN, and CART. The second tier contained the Legendre, cosine, and Haar orthonormal basis function classifiers. Backprop performed significantly better than all second-tier systems, and the Daubechies classifier performed significantly worse than the backprop, LDA, KNN, and Legendre classifiers. Other than these, the only distinction

that could be made was between SVM and all other systems: SVM had a significantly lower classification rate.

With 160 training exemplars, backprop (78.0% correct), CART, and LDA had top classification scores. A second group with no significant differences within the group consisted of KNN, the Legendre, cosine, and Haar basis function classifiers, and SVM. Of these systems, all but KNN performed significantly worse than LDA. The Daubechies orthonormal basis function classifier performed significantly worse than backprop, CART, and KNN, but not LDA. No system could be distinguished statistically from SVM due to high variance in the performance of SVM on this task.

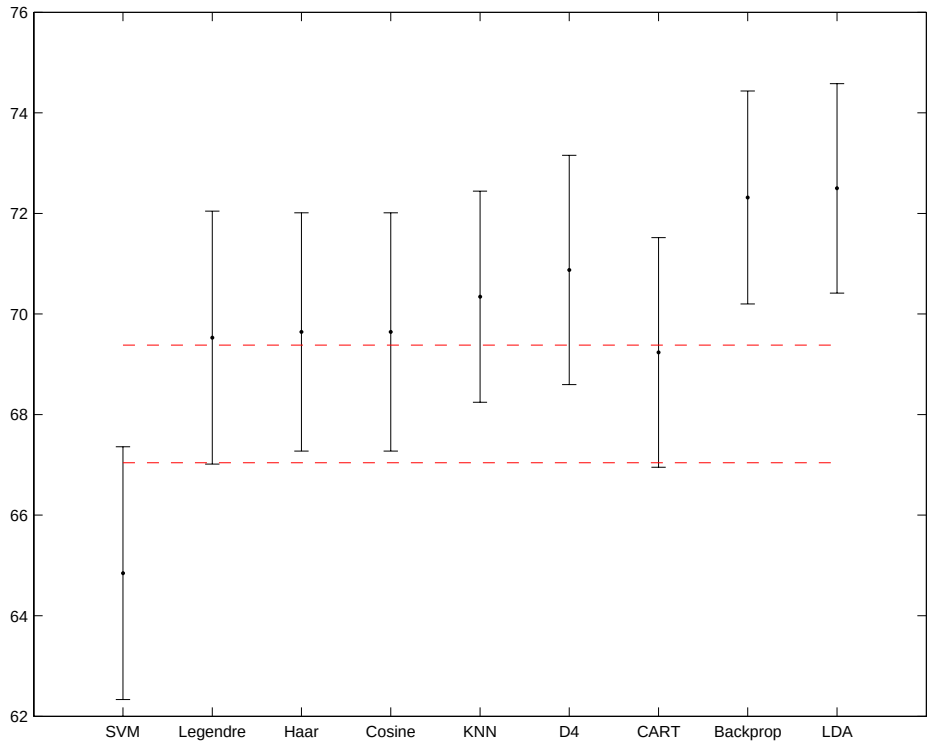

Figure 5.13: Staircase plot of DELVE benchmark performance for the Titanic survival prediction task with 20 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

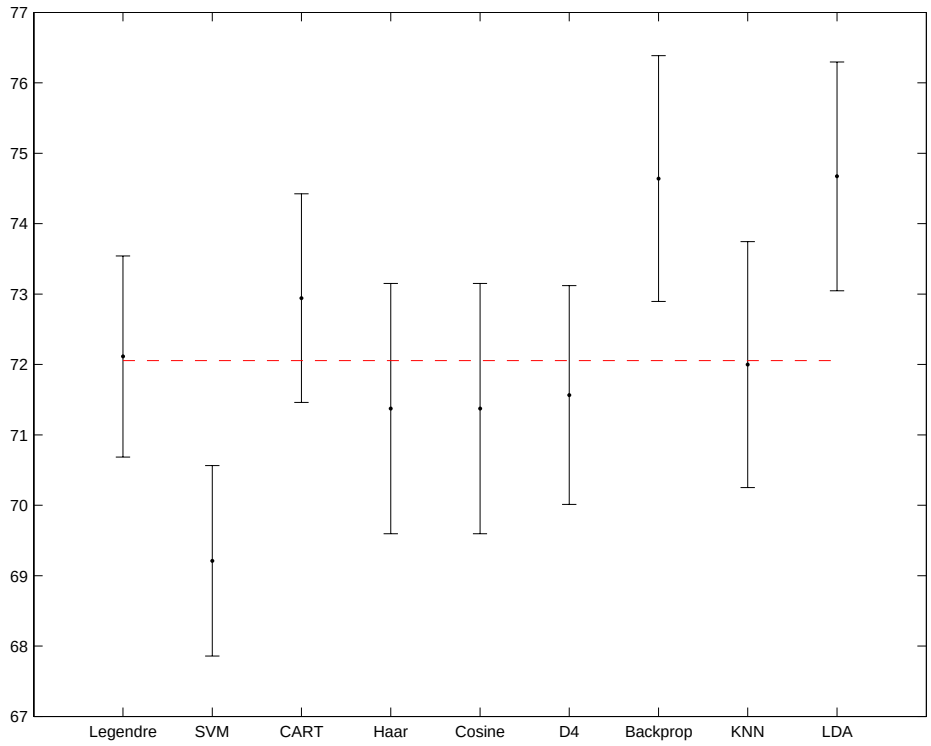

Figure 5.14: Staircase plot of DELVE benchmark performance for the Titanic survival prediction task with 40 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

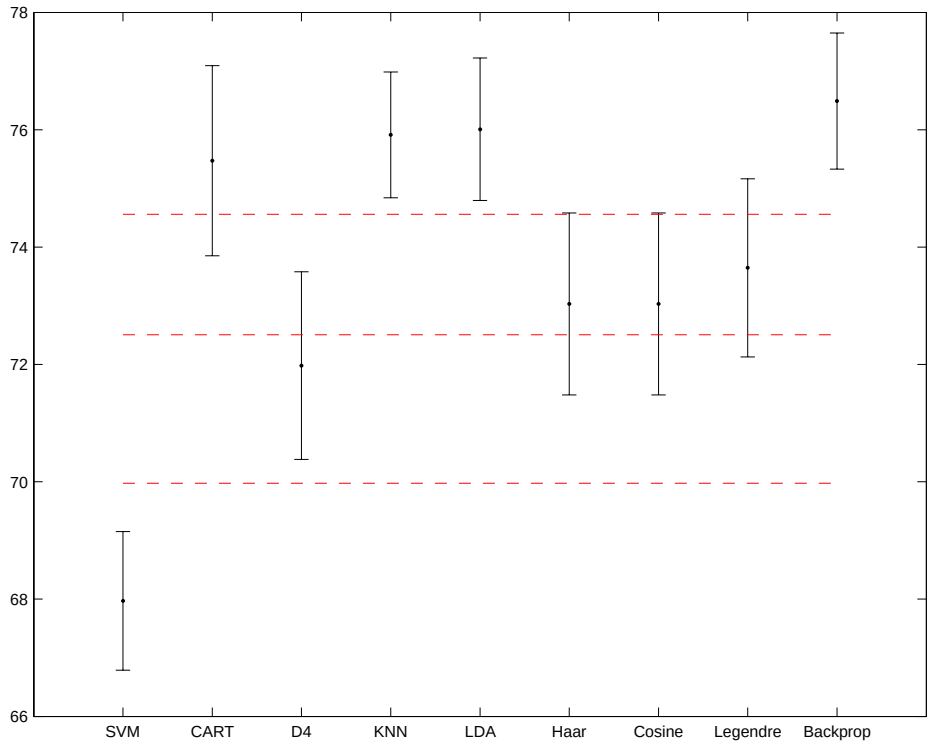

Figure 5.15: Staircase plot of DELVE benchmark performance for the Titanic survival prediction task with 80 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

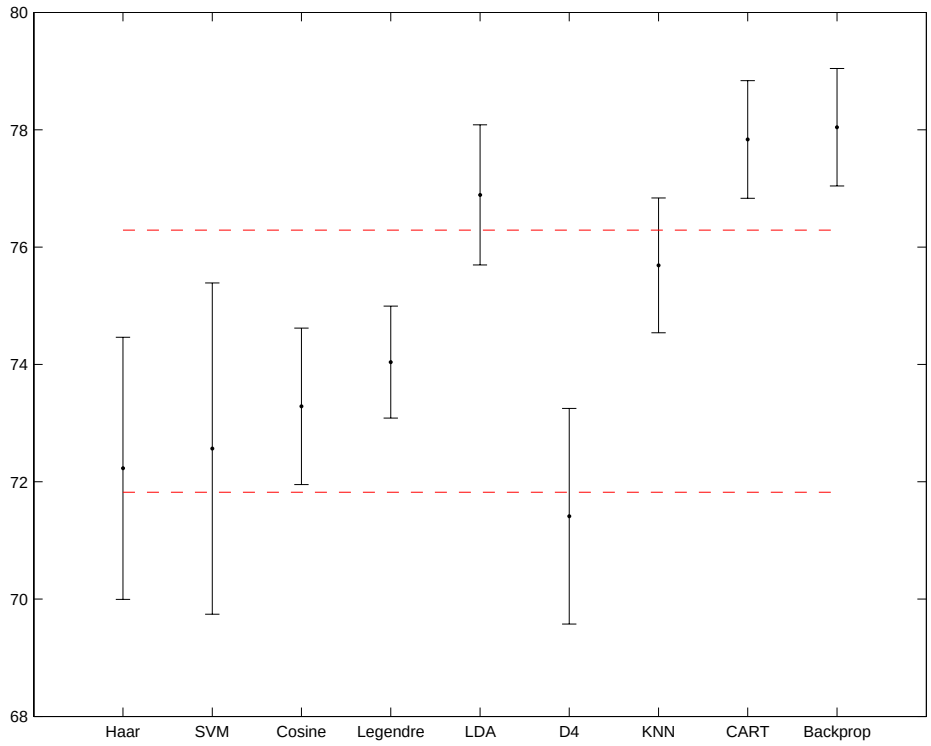

Figure 5.16: Staircase plot of DELVE benchmark performance for the Titanic survival prediction task with 160 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

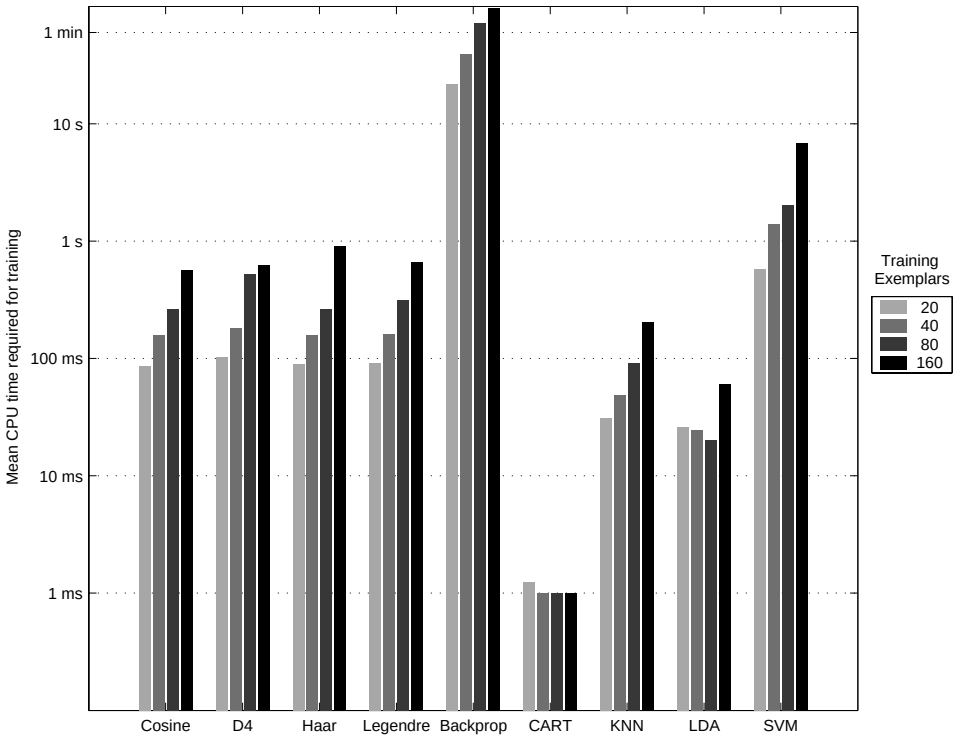

Figure 5.17: Mean CPU time required to train four orthonormal basis function networks (left) and five other classifiers (right) on the DELVE Titanic survival database. CPU times are shown on a logarithmic scale. Values less than 1 ms are rounded up to 1 ms.

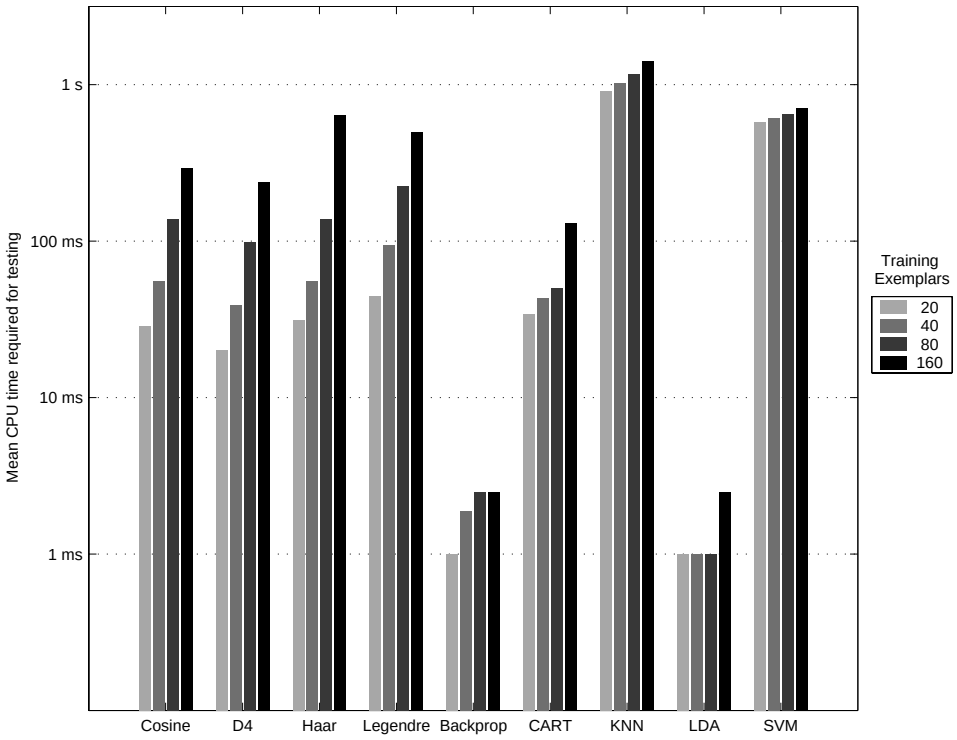

Figure 5.18: Mean CPU time required to test four orthonormal basis function networks (left) and five other classifiers (right) on 1,561 exemplars from the DELVE Titanic survival database. CPU times are shown on a logarithmic scale. Values less than 1 ms are rounded up to 1 ms.

5.6 DELVE Adult benchmark

The Adult database (Rasmussen, Neal et al. 1996) consists of six continuous attributes and seven categorical attributes from the 1994 U.S. Census database. Classifiers are to ascertain whether a particular exemplar (a census respondent) had a salary of greater than $50,000.

Although there are only 13 attributes, using category indicator variables to represent the categorical attributes requires a total of 62 dimensions. Extended canonical

variates were computed for the full dimensionality of the datasets. The automatic scree test method of Section 4.2.2 was used to reduce the dimensionality of the extended variates acquired via PCA.

With 256 training exemplars, top-tier systems included KNN, CART, and LDA. The second tier contained the Legendre, Haar, and cosine orthonormal basis function classifiers. Of these, the Legendre system did not have significantly lower classification performance than LDA, and was also the only system with significantly higher performance than the third-tier Daubechies orthonormal basis function classifier. SVM, also in the third tier, did not perform significantly worse than the Daubechies classifier.

Backprop was significantly worse than all other systems on all the tasks for the Adult benchmark. Regardless of the number of training exemplars, backprop with GCV model selection performed worse than chance, which appears to be due to the use of a generalized cross-validation criterion in place of actual cross-validation. This generalized cross-validation criterion was minimized when the number of hidden layer nodes was between 15 and 50. However, the performance of a simple linear discriminant on this task indicates that one hidden layer unit should be sufficient to provide good performance. The failure to find such a simple model is a shortcoming of the GCV methodology used herein.

When provided with 512 training exemplars, CART and LDA had significantly better classification performance than all other algorithms. KNN followed, with performance significantly worse than each of these algorithms and significantly better than all other classifiers. The cosine, Haar, and Legendre networks formed a tier of

equivalent performance with no distinctions among the algorithms. This tier was followed by the Daubechies, SVM, and backprop systems, respectively. Each had significantly different performance.

CART was again the best performing algorithm when 1,024 exemplars were provided. This was followed by LDA and KNN, each significantly different from all other algorithms tested. The cosine, Haar, Legendre, and SVM classifier systems followed. There were no significant differences between these systems. All the aforementioned systems except SVM had significantly better classification rates than the Daubechies orthonormal basis function system.

With 2,048 training exemplars, CART had a significantly higher classification rate than any other algorithm. This was followed by LDA, which also performed significantly better than the remaining algorithms, and KNN, which likewise was in a performance tier of its own. SVM and the cosine, Haar, and Legendre orthonormal basis function networks formed a tier of equivalent performance. The Daubechies system had significantly lower classification performance than all of these systems. Of the tested systems, only backprop performed worse.

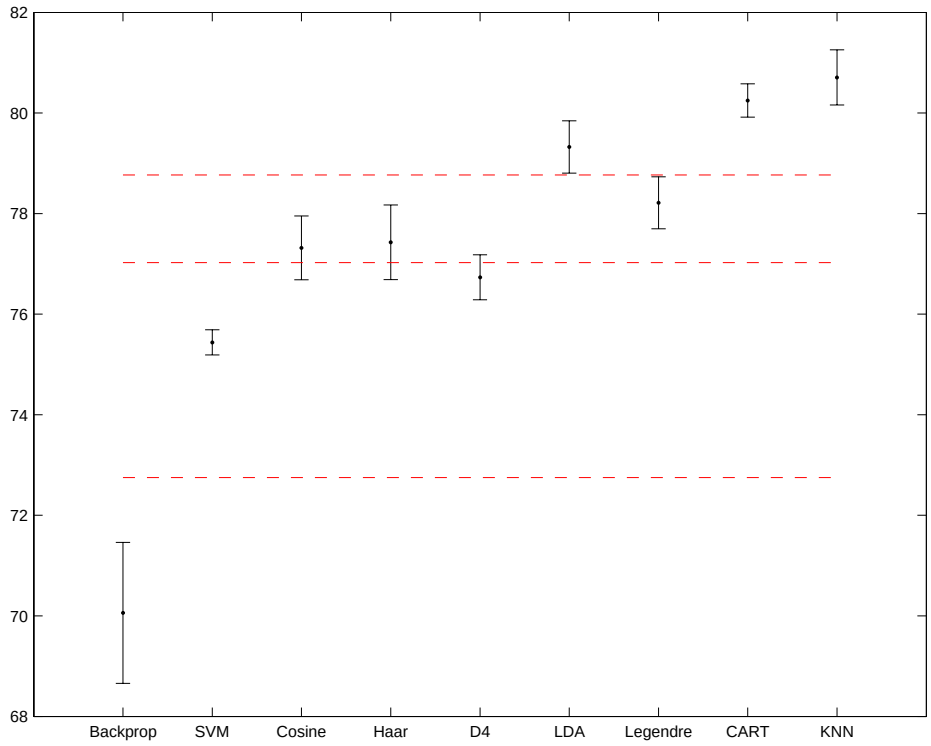

Figure 5.19: Staircase plot of DELVE benchmark performance for the Adult task with 256 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

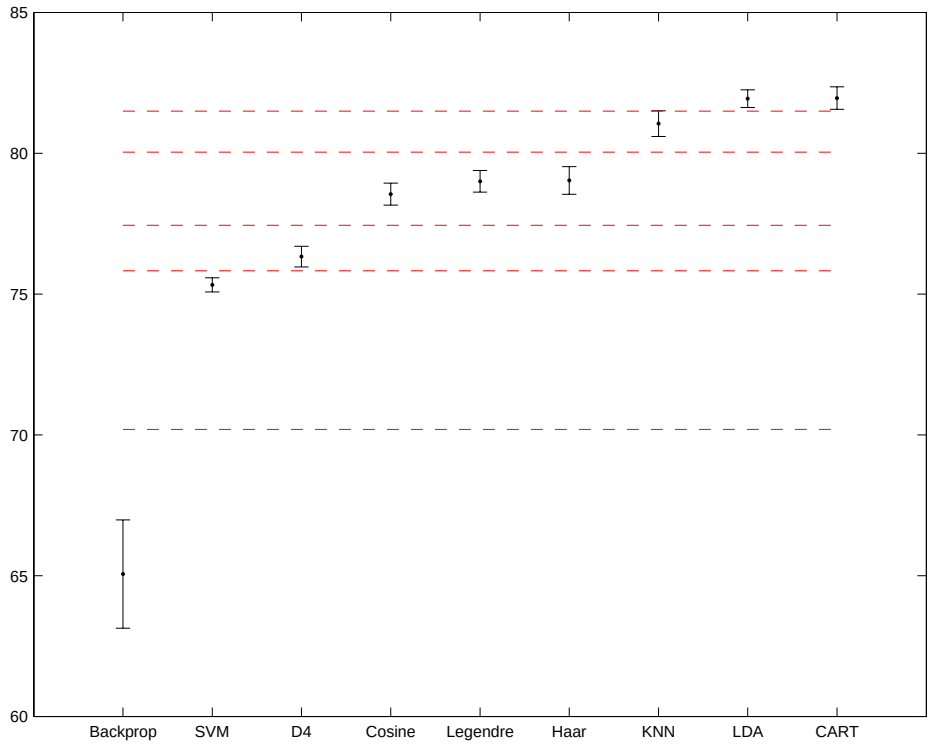

Figure 5.20: Staircase plot of DELVE benchmark performance for the Adult task with 512 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

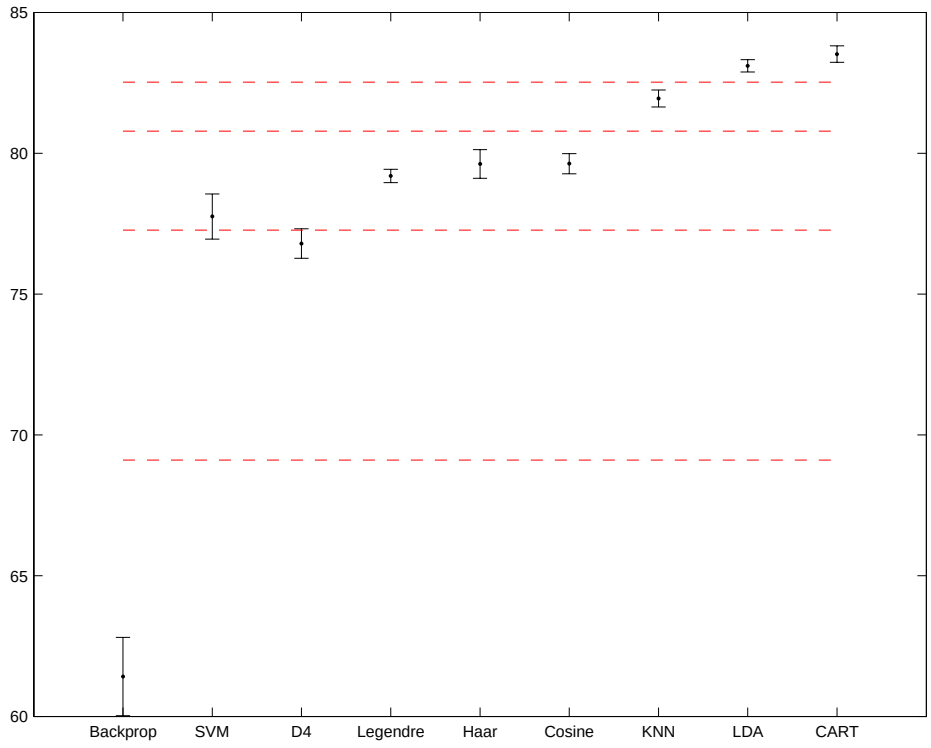

Figure 5.21: Staircase plot of DELVE benchmark performance for the Adult task with 1,024 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

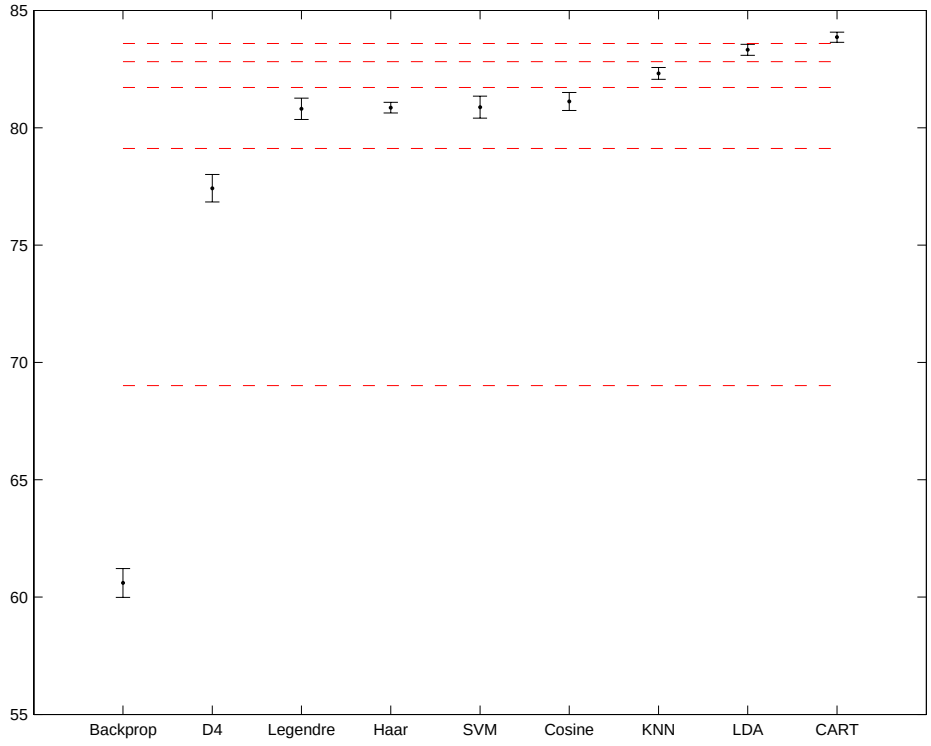

Figure 5.22: Staircase plot of DELVE benchmark performance for the Adult task with 2,048 training exemplars. Performance tiers are separated by dashed lines. Algorithms that are to the right and in a higher tier with respect to a selected algorithm have significantly better classification performance.

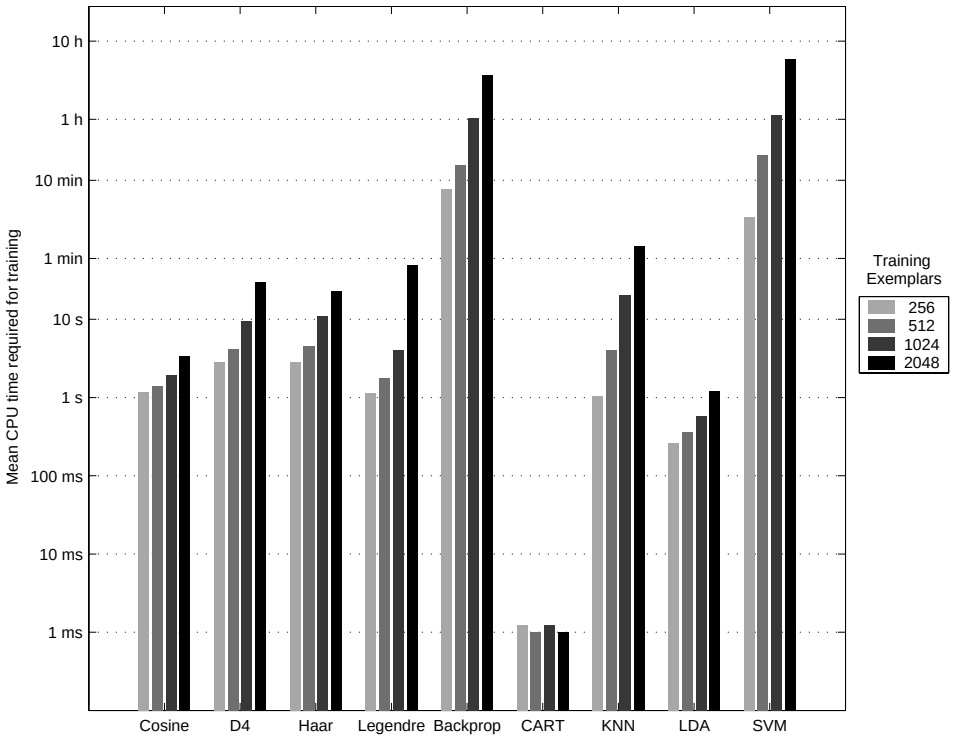

Figure 5.23: Mean CPU time required to train four orthonormal basis function networks (left) and five other classifiers (right) on the DELVE Adult database. CPU times are shown on a logarithmic scale. Values less than 1 ms are rounded up to 1 ms.

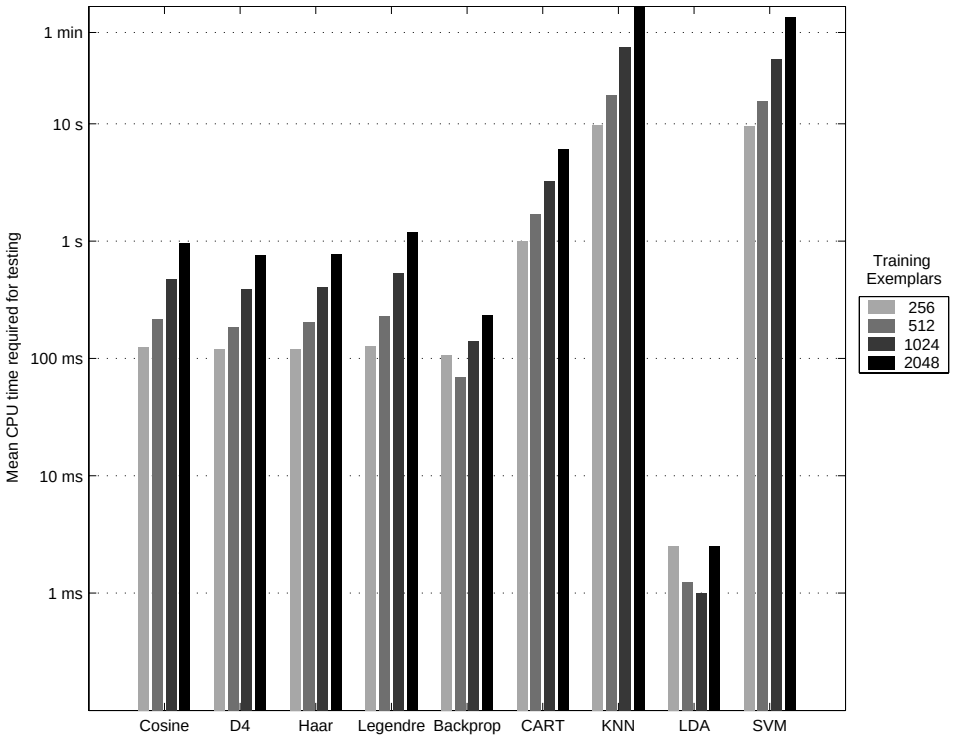

Figure 5.24: Mean CPU time required to test four orthonormal basis function networks (left) and five other classifiers (right) on 3,706 exemplars from the DELVE Adult database. CPU times are shown on a logarithmic scale. Values less than 1 ms are rounded up to 1 ms.

5.7 Discussion

Of the orthonormal basis function networks tested, systems that use two of the bases, Daubechies and Legendre, are able to exactly represent a linear classification function and linear decision boundary. Using functions at the fundamental frequency, the cosine basis can also represent monotonically increasing classification functions, although the boundaries and representations are nonlinear. Of the bases tested, the Haar basis is unique in its inability to represent any continuous monotonic function other than

the trivial constant function. It appears from these results that this may be disadvantageous for certain problems.

Besides classification performance, a salient difference between classifiers is the amount of computation required for training (Figure 5.6, Figure 5.11, Figure 5.17, Figure 5.23). CART is consistently the fastest system to be trained on all the benchmark tasks considered in this chapter, not requiring more than five milliseconds to fully fit a model. LDA is also extremely fast in this respect, with training times between 20 milliseconds and two seconds. At the other extreme are backprop and SVM, the methods that perform nonlinear optimizations. SVM scales poorly in the number of training exemplars, requiring only seconds for the smallest databases considered and several hours for the largest. With the exception of the Adult database, for which backprop and SVM training times are similarly long, backprop is consistently the most time-consuming classifier to train. While not as fast as either CART or LDA, orthonormal basis function and KNN classifiers require at most a few minutes for training on these databases.

Disparities in CPU time for testing (Figure 5.7, Figure 5.12, Figure 5.18, Figure 5.24) are not as extreme as for training. Where fast evaluation of test data is a concern, LDA, which requires a simple linear computation, is an excellent choice, taking around one microsecond of CPU time to classify a single exemplar, or only milliseconds to classify an entire database. From the empirical results obtained here, backprop appears to be the clear second choice for testing speed. KNN and SVM are consistently the two slowest systems for testing, typically between one and three orders of magnitude slower than backprop. In the worst case, however, both algorithms take just over a minute to test

all exemplars. Other algorithms, including CART and all orthonormal basis function classifiers, consistently test more rapidly than these systems but more slowly than backprop.

5.8 Conclusions

These DELVE classification benchmark results illustrate the need to test a variety of classification approaches on any particular dataset. It is difficult to ascertain *a priori* which classifier will yield the best classification performance on a given task. All of the classifiers tested had performance among the best results on one task or another in the series of benchmarks.

The comparisons performed in this chapter confirm the suitability of orthonormal basis function classifiers to multidimensional classification tasks of similar nature to those in the DELVE benchmark suite. It appears from these limited tests that orthonormal basis function neural network performance relative to other approaches may be best when the number of exemplars available for training is small.

Chapter 6

An Application of Orthonormal Basis Function Neural Networks to Land Use Change Classification

6.1 Introduction

The results of the previous chapter suggest that orthonormal basis function classifiers may be useful for processing remotely sensed data. This chapter revisits the Nile River delta land use change database of Chapter 2 to evaluate the performance of orthonormal basis function neural networks on the task. It is desirable to investigate a variety of systems for such a problem since it is unknown *a priori* which systems will perform well. This database differs from many remote sensing databases in that it requires the identification of changes in land use over time from a sequence of satellite images.

For remote sensing applications, there are several performance measures of interest. The *user's accuracy* assesses the classification rate on the subset of pixels or sites for which class labels are known. The *producer's accuracy* estimates the classification rate on all pixels in a map. The training and testing speeds are also important for many remote sensing applications due to the volume of data to be processed.

6.2 Methods

6.2.1 Data

The dataset, consisting of ten satellite images, had been previously been geometrically registered and radiometrically normalized (Lenney, Woodcock et al. 1996). For each pixel of the registered image set, a multi-date vector was prepared. Data consisted of six bands of 30m Landsat TM data from ten dates between 1984 and 1993, of which one band for one date was missing. The line and sample (vertical and horizontal) coordinates of each 30m pixel were also available. In addition, each pixel was associated with a geographic region (*delta, desert, coast,* or *wetlands*). The geographic region was represented by a 1-of-4 coding scheme, in which one of four vector elements was set to a value of one and the remaining three elements were set to a value of zero. The 59 bands of Landsat TM data, 2 coordinates, and 4 region indicator variables were concatenated to form a 65-dimensional feature vector for every pixel in the study area. Using extended canonical variate analysis (Section 4.2.4) on the labeled pixels, the dimensionality of the feature vectors was reduced to ten, including seven canonical variates and the three most prominent principal components of the residual (Figure 6.1). These canonical variates accounted for 1.08% of the total variance in the feature matrix $\mathbf{X}$. The first three principal components accounted for 87.24% of the total variance in $\mathbf{X}$. In all, the first ten extended canonical variates accounted for the full linear classification model and 88.33% of the variance in the independent variables.

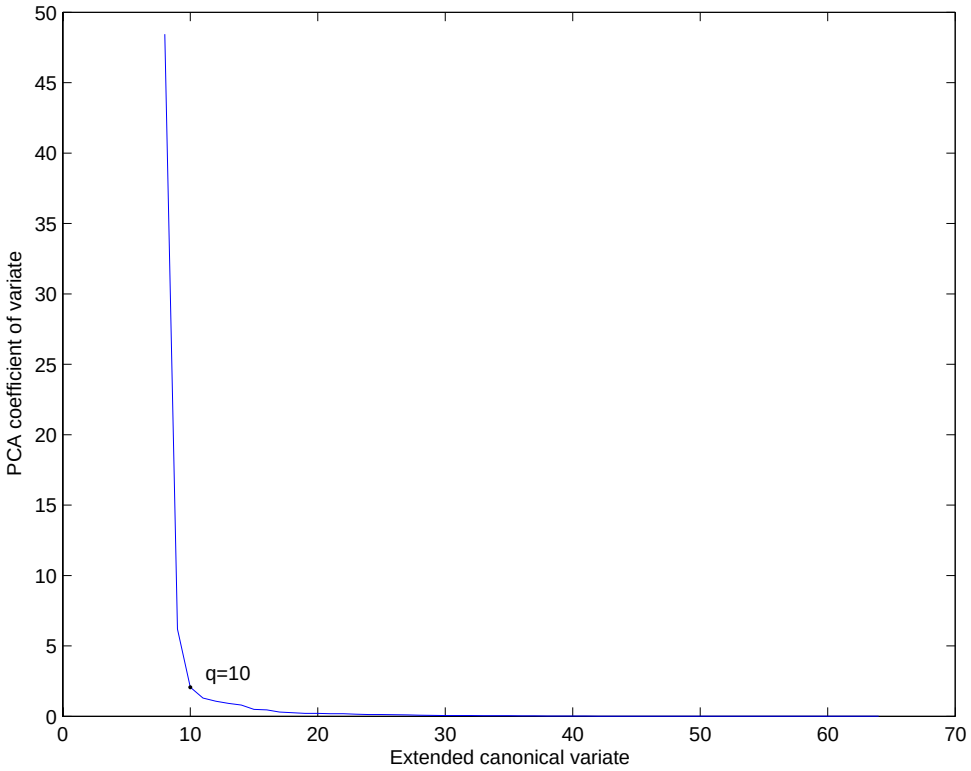

Figure 6.1: Scree plot of the principal components of the land use change database after removal of the canonical variates. The elbow at $q = 10$ (seven canonical variates and three additional components) was selected by visual inspection.

Values in these ten dimensions were normalized by linear transformation to be in the range $[0,1]$ for every pixel in the study area. This was necessary for the ARTMAP and orthonormal basis function classifiers. CART and LDA classifiers are not sensitive to rescaling, so both were unaffected by this transformation. The KNN algorithm used in this work rescales all variables to have zero mean and unit variance, and was likewise unaffected by the normalization of variates to the unit interval.

6.2.2 Site-based leave-out-one cross-validation

The Egypt land use change ground truth dataset consists of 358 sites, each of which contains four pixels. These pixels may be highly correlated within a site, so a typical leave-out-one cross-validation methodology would result in the presentation during training of pixels almost identical to the pixel left out.

To eliminate this anticipated source of bias, cross-validation was performed by leaving out all pixels within a site. The algorithms were trained on the remainder of the database, then tested on each of the four validation pixels.

It is also possible to determine a site classification rate by combining pixel classifications within a site. However, the classification rates obtained in this experiment refer to the number of pixels correctly classified.

A limitation of leave-out-one cross-validation is that cross-validation was not used to select the number of dimensions for training via the extended canonical variates procedure. It is possible that a bias could have been introduced in the dimension reduction process.

6.2.3 Algorithms tested

Algorithms considered in this experiment included LDA, CART, KNN, and orthonormal basis function networks employing the discrete cosine, Daubechies D4, and Legendre polynomial bases. The experiment also included the ARTMAP system of Chapter 2 with the following representative parameters: baseline vigilance $\bar{\rho} = 0$, choice parameter $\alpha = .001$, and $V = 3$ voters. Sites, each containing four pixels, were presented an average of 108 times.

Support vector machines and backpropagation neural networks were excluded due to their large training time requirements. On DELVE databases with a similar number of training exemplars, the versions of these algorithms employed in this dissertation required upwards of 20 minutes of CPU time for training, and typically an hour or more (Figure 5.6, Figure 5.23). Extrapolating from these numbers, in a leave-out-one-site cross-validation scheme with 358 sites, the expected CPU time necessary for the evaluation performed in this chapter on other algorithms is anywhere from 120 hours to 360 hours or more. This long running time is attributable to the cross-validated model selection integral to the algorithms presented in 5.1, which for backprop and SVM algorithms increases the running time by an order of magnitude or more over ad hoc parameter selection.

6.3 Results and discussion

Performance of the LDA classifier (93.5%) was significantly better than that of all other classifiers tested. The ARTMAP classifier (84.1%), followed by the Daubechies

orthonormal basis function classifier, (87.1 %) performed significantly worse than all other algorithms tested. Remaining classifiers (KNN, CART, and cosine and Legendre orthonormal basis function networks) did not have significantly different performance from one another; all had a classification rate between 90.1% and 91.0%.

An important observation is that the orthonormal basis function networks do not perform a full linear discriminant. Rather, LDA is used as a postprocessing tool to better draw decision boundaries. These boundaries are not drawn in the full space of the orthonormal basis transformation.

Canonical variate analysis on the 65-dimensional feature vectors allows the database to be visualized in up to seven dimensions. The seven-dimensional view shows why LDA is an effective approach for this classification task. The first two canonical variates, in Figure 6.3, show that most of the classes are in clusters that are nearly separable by linear boundaries. These include classes 5 through 8 (*agriculture in desert/coast*, *reclamation*, *wetland reclamation*, and *other*), each of which appears as a single cluster, and class 4 (*agriculture in delta*), which appears to be bimodal with two distinct clusters. Using the first two canonical variates alone, it is difficult to distinguish between classes 1, 2, and 3 (*urban*, *urbanization*, and *reduced productivity*).

Although it appears from the first two variates to be a difficult task to separate classes 1 through 3, additional variates provide a greater degree of separation between the classes. This can particularly be seen in Figure 6.9. Class 2 (*urbanization*) is the only class that does not form one or more visually distinguishable clusters in the canonical variate plots, although the seventh canonical variate in Figure 6.6 appears to separate

some members of this class from the remainder of the database. This visual analysis suggests that a linear discriminant that can determine appropriate separation boundaries between the clusters should classify most data with few errors.

Figure 6.2 shows a comparison of six classifiers on the Egypt land use change database. Significance of differences was evaluated using McNemar's test (Fleiss 1981; Ripley 1996). With the exception of the ARTMAP classifier developed in Chapter 2, all algorithms were applied to normalized 65-dimensional feature vectors containing the data listed in Table 6.1.

Figure 6.11 gives a measure of the importance of each of the variates in determining the classification. This chart highlights the relatively large import of geographical region in determining land use and land use change classification. Moreover, all of the most important image spectral bands are taken from the first three images of the dataset and the final image of the dataset, suggesting that land use change was occurring throughout the entire measured period.

Input vector element	Description
1-6	Normalized spectral bands from the June 7, 1984 image
7-12	Normalized spectral bands from the September 11, 1984 image
13-18	Normalized spectral bands from the June 10, 1985 image
19-24	Normalized spectral bands from the December 22, 1986 image
25-29	Normalized spectral bands from the May 15, 1990 image
30-35	Normalized spectral bands from the August 21, 1988 image
36-41	Normalized spectral bands from the August 3, 1990 image
42-47	Normalized spectral bands from the February 19, 1991 image
48-53	Normalized spectral bands from the June 13, 1992 image
54-59	Normalized spectral bands from the from April 29, 1993 image
60	Pixel *x*-coordinate
61	Pixel *y*-coordinate
62	Geographic region indicator for *delta* (Boolean)
63	Geographic region indicator for *desert* (Boolean)
64	Geographic region indicator for *coast* (Boolean)
65	Geographic region indicator for *wetlands* (Boolean)

Table 6.1: Classifier inputs for the Nile River delta land use change task.

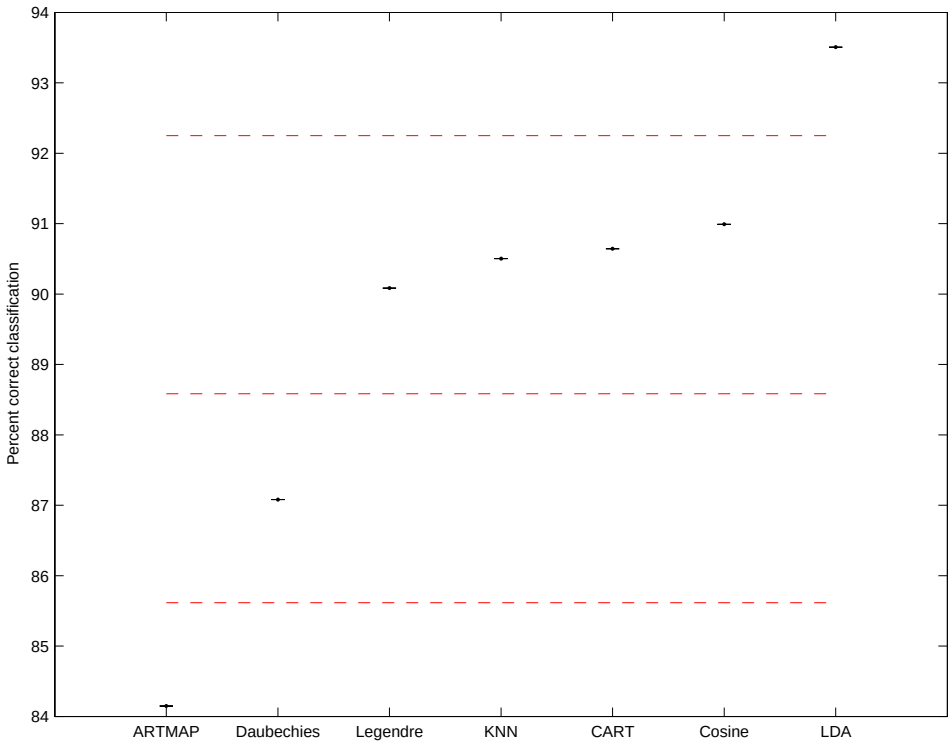

Figure 6.2: Staircase plot of Egypt land use change dataset results for six classifiers. Performance was evaluated using leave-out-one cross-validation in which each site was omitted in turn and the classifiers were trained on the remaining sites. Significance levels were determined with McNemar's test (Fleiss 1981; Ripley 1996). Error bars are not available due to the testing methodology. These results show three performance tiers with no significant difference between the Cosine and Legendre orthonormal basis function networks, CART, and *K*-nearest neighbors.

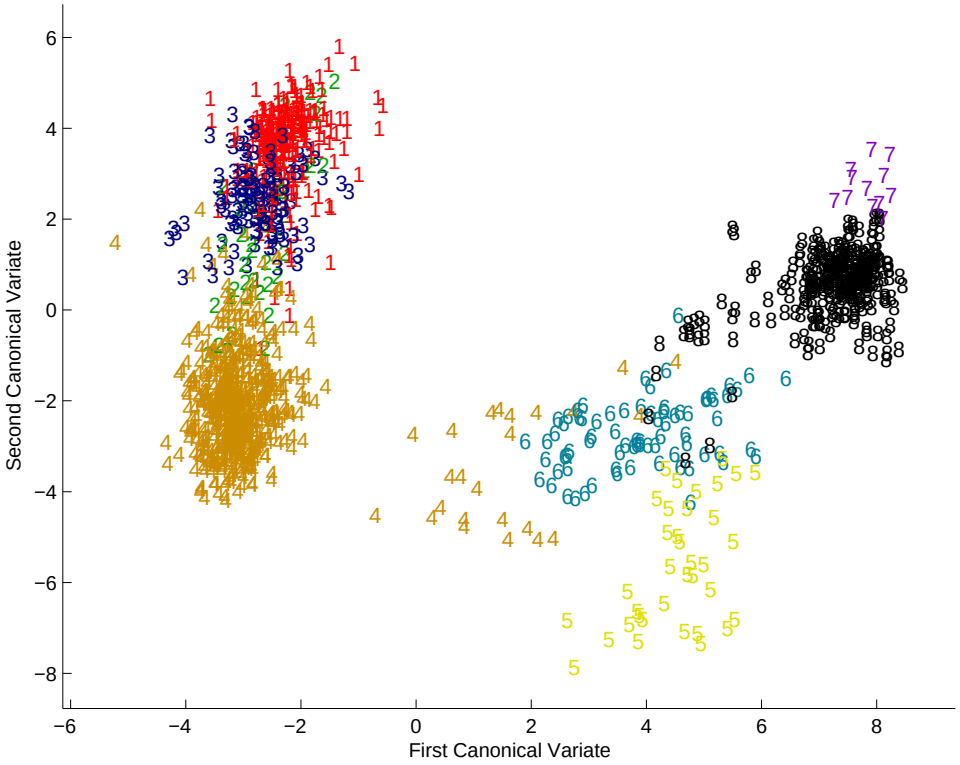

Figure 6.3: First and second canonical variates of the Egypt land use change database. Classes are indicated by digits one through eight: 1 = *urban*, 2 = *urbanization*, 3 = *reduced productivity*, 4 = *agriculture*, 5 = *agriculture in desert/coast*, 6 = *reclamation*, 7 = *wetland reclamation*, 8 = *other*.

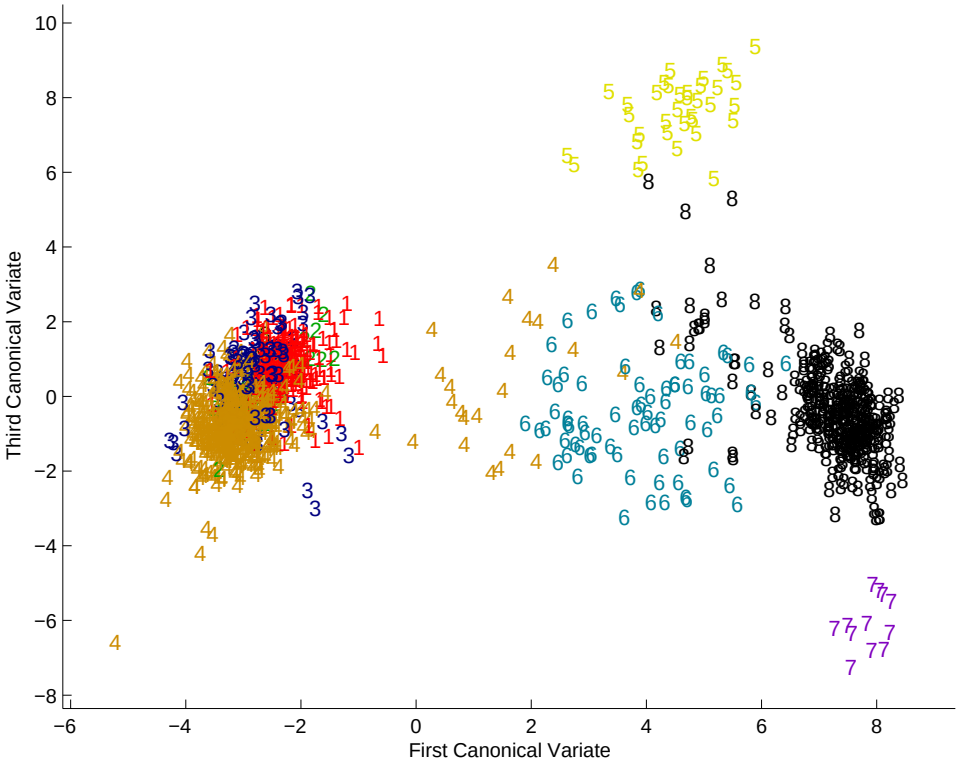

Figure 6.4: First and third canonical variates of the Egypt land use change database. Classes are indicated by digits one through eight: 1 = *urban*, 2 = *urbanization*, 3 = *reduced productivity*, 4 = *agriculture*, 5 = *agriculture in desert/coast*, 6 = *reclamation*, 7 = *wetland reclamation*, 8 = *other*.

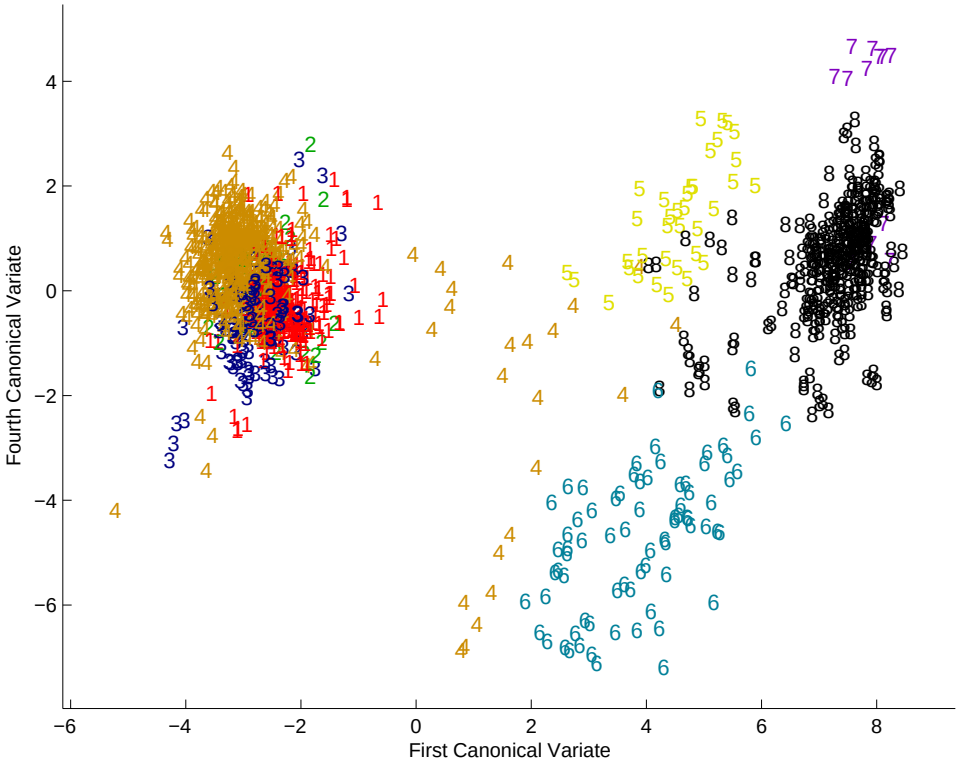

Figure 6.5: First and fourth canonical variates of the Egypt land use change database. Classes are indicated by digits one through eight: 1 = *urban*, 2 = *urbanization*, 3 = *reduced productivity*, 4 = *agriculture*, 5 = *agriculture in desert/coast*, 6 = *reclamation*, 7 = *wetland reclamation*, 8 = *other*.

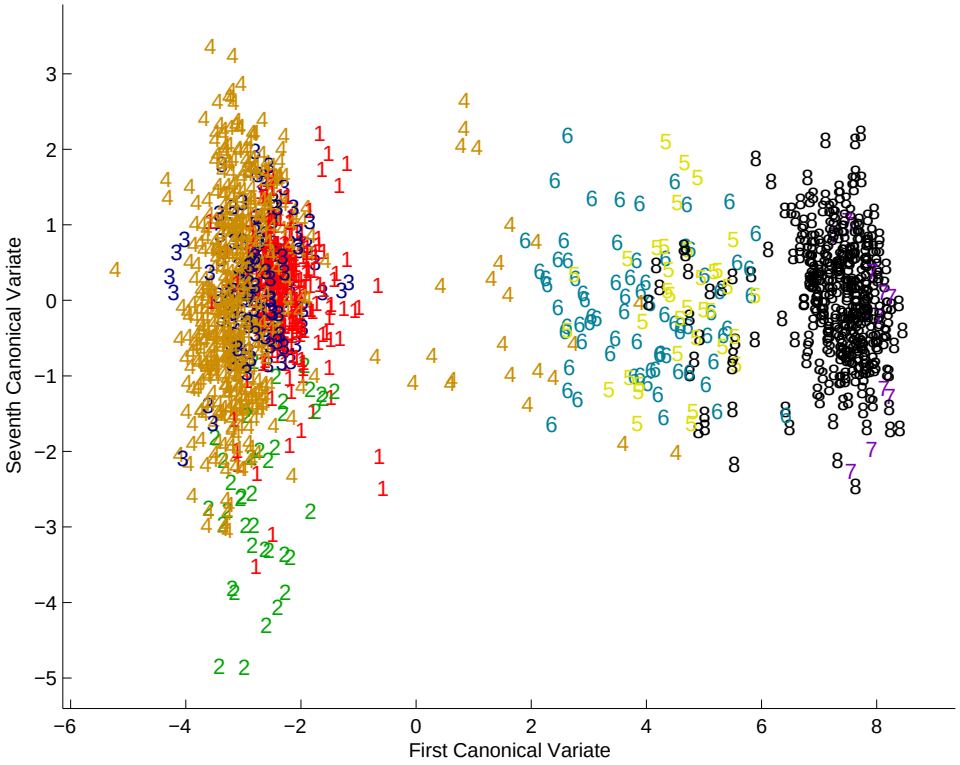

Figure 6.6: First and seventh canonical variates of the Egypt land use change database. Classes are indicated by digits one through eight: 1 = *urban*, 2 = *urbanization*, 3 = *reduced productivity*, 4 = *agriculture*, 5 = *agriculture in desert/coast*, 6 = *reclamation*, 7 = *wetland reclamation*, 8 = *other*.

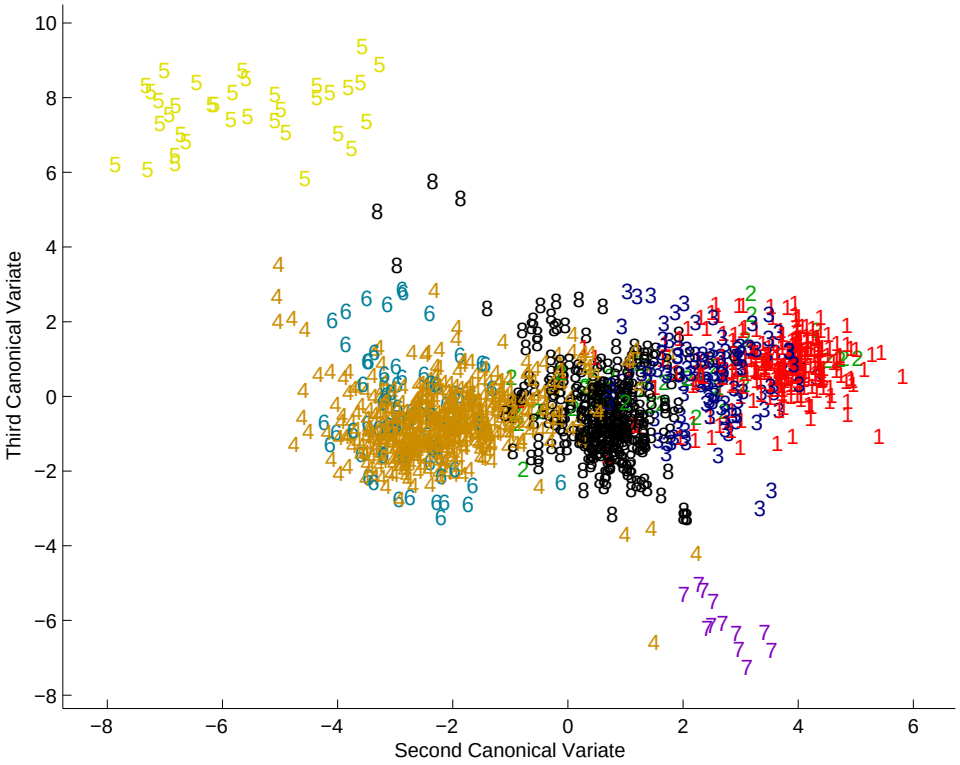

Figure 6.7: Second and third canonical variates of the Egypt land use change database. Classes are indicated by digits one through eight: 1 = *urban*, 2 = *urbanization*, 3 = *reduced productivity*, 4 = *agriculture*, 5 = *agriculture in desert/coast*, 6 = *reclamation*, 7 = *wetland reclamation*, 8 = *other*.

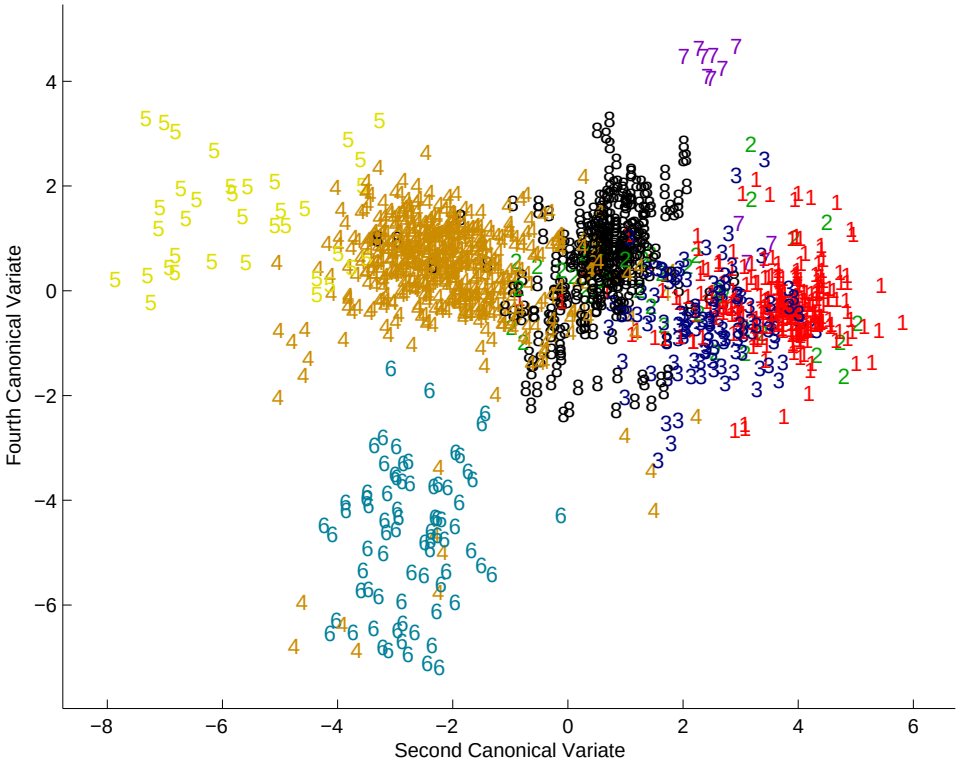

Figure 6.8: Second and fourth canonical variates of the Egypt land use change database. Classes are indicated by digits one through eight: 1 = *urban*, 2 = *urbanization*, 3 = *reduced productivity*, 4 = *agriculture*, 5 = *agriculture in desert/coast*, 6 = *reclamation*, 7 = *wetland reclamation*, 8 = *other*.

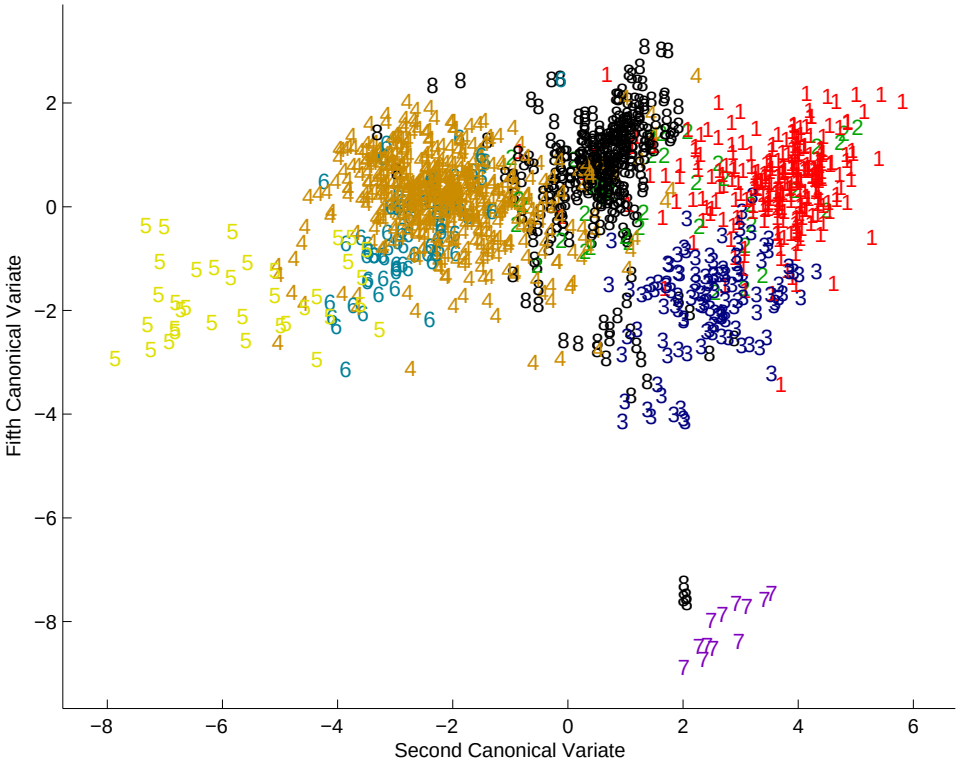

Figure 6.9: Second and fifth canonical variates of the Egypt land use change database. Classes are indicated by digits one through eight: 1 = *urban*, 2 = *urbanization*, 3 = *reduced productivity*, 4 = *agriculture*, 5 = *agriculture in desert/coast*, 6 = *reclamation*, 7 = *wetland reclamation*, 8 = *other*.

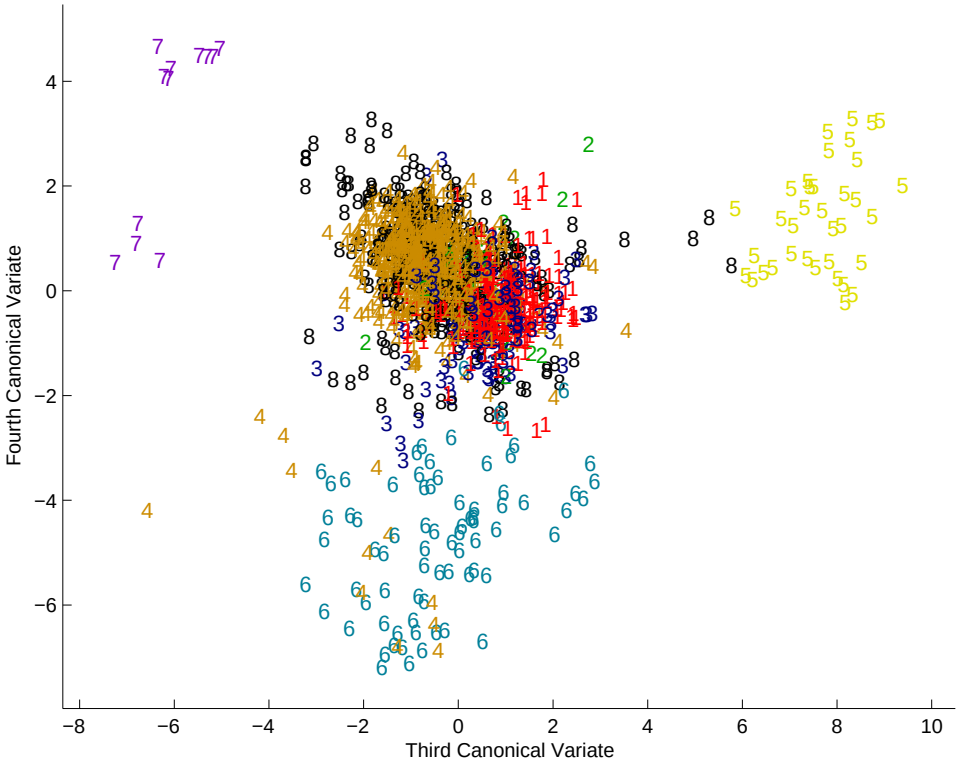

Figure 6.10: Third and fourth canonical variates of the Egypt land use change database. Classes are indicated by digits one through eight: 1 = *urban*, 2 = *urbanization*, 3 = *reduced productivity*, 4 = *agriculture*, 5 = *agriculture in desert/coast*, 6 = *reclamation*, 7 = *wetland reclamation*, 8 = *other*.

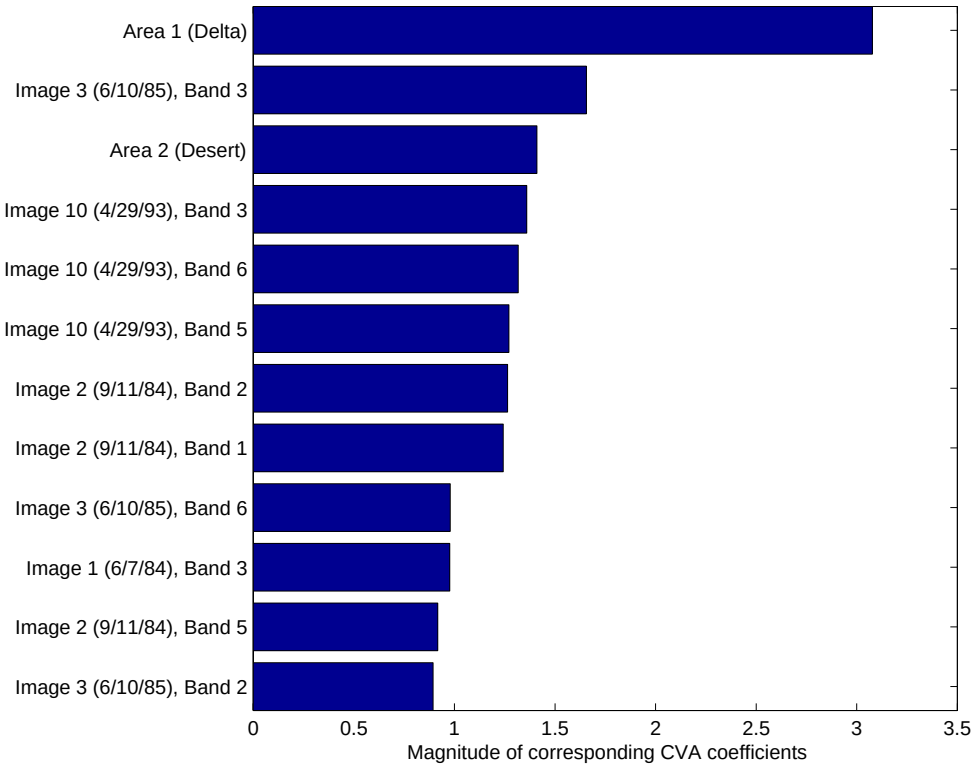

Figure 6.11: The twelve components of the Egypt land use change database with the largest root sum-of-squares λ weights in the canonical variate analysis (CVA). The indicator variables for the areas *Delta* and *Desert* are strongly related to the class labels. Other major class predictors include three bands each from the images taken on September 11, 1984; June 10, 1985; and April 29, 1993; and one band from the image taken on June 7, 1984. These represent the first year and the last year of the study period.

Land use classifications	Pixels	Urban	Urbanization	Reduced productivity	Agriculture in delta	Agriculture in desert/coast	Reclamation	Wetlands reclaimed	Other	User's accuracy
Urban	290	**263**	7	13	7					*90.7%*
Urbanization	33	3	**24**	1	5					*72.7%*
Reduced productivity	127	6	4	**110**	3				4	*86.6%*
Agriculture in delta	540	4	5		**531**					*98.3%*
Agriculture in desert/coast	41				2	**36**			3	*87.8%*
Reclamation	88				14		**72**		2	*81.8%*
Wetlands reclaimed	16							**12**	4	*75.0%*
Other	297				2		4		**291**	*98.0%*
Total	1432	276	40	124	564	36	76	12	304	***Overall 93.5%***

Table 6.2: User's accuracy assessment of the LDA classifier on 1,432 pixels cross-validated by leaving out one four-pixel site. Errors include difficulties distinguishing *urbanization* from other land uses, confusion between *urban* and *reduced productivity*, and errors of commission identifying *reclamation*, *agriculture in desert/coast*, and *wetlands reclaimed* sites. Overall accuracy on the ground truth dataset was 93.5%.

	Field assessments								
Land use classifications	Urban	Urbanization	Reduced productivity	Agriculture in delta	Agriculture in desert/coast	Reclamation	Wetlands reclaimed	Other	Map proportions
Urban	**1.609 %**	0.043 %	0.080 %	0.043 %					*1.774%*
Urbanization	0.038 %	**0.300 %**	0.013 %	0.063 %					*0.413%*
Reduced productivity	0.094 %	0.063 %	**1.732 %**	0.047 %				0.063 %	*2.000%*
Agriculture in delta	0.397 %	0.496 %		**52.656 %**					*53.549%*
Agriculture in desert/coast				0.044 %	**0.799 %**			0.067 %	*0.910%*
Reclamation				0.850 %		**4.370 %**		0.121 %	*5.341%*
Wetlands reclaimed							**0.557 %**	0.186 %	*0.743%*
Other				0.238 %		0.475 %		**34.559 %**	*35.271%*
Estimated true proportions	*2.137 %*	*0.902 %*	*1.824 %*	*53.940 %*	*0.799 %*	*4.845 %*	*0.557 %*	*34.996 %*	
Producer's accuracy	75.3 %	33.3 %	95.0 %	97.6 %	100.0 %	90.2 %	100.0 %	98.8 %	***Overall 96.6%***

Table 6.3: Producer's accuracy assessment of the LDA classifier on 1,432 pixels cross-validated by leaving out one four-pixel site. Producer's accuracy estimates the percentage of pixels of each ground truth class correctly identified. The overall producer's accuracy of 96.6% is an estimate of the percentage of pixels in the entire map that are correctly identified.

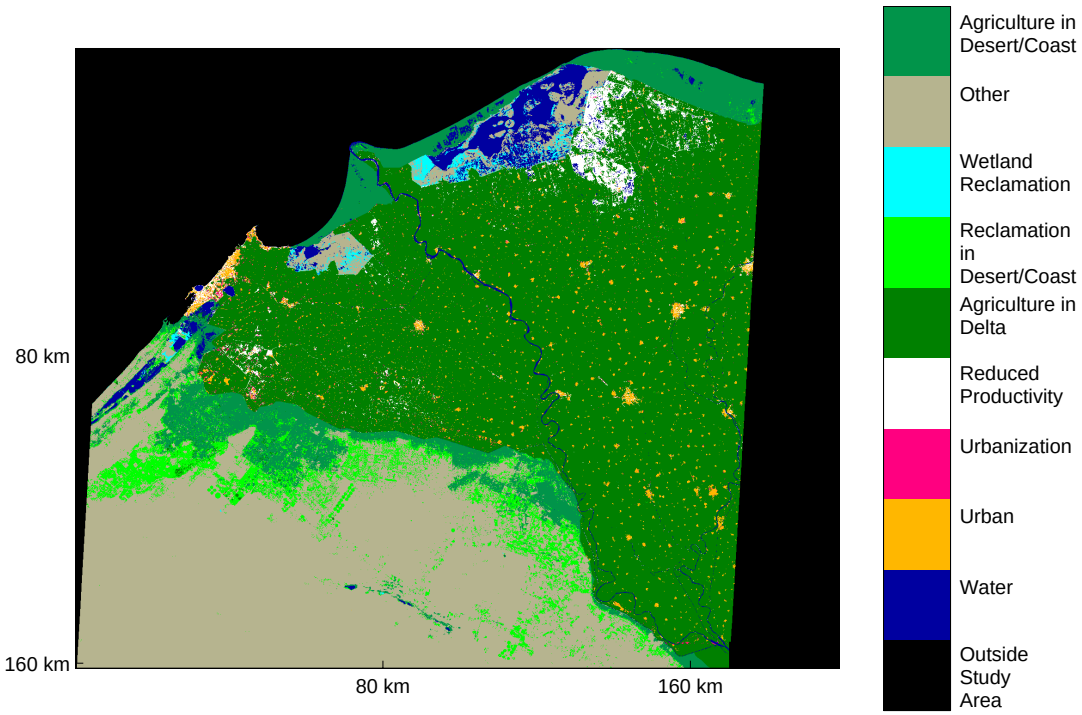

Figure 6.12: Map of labels assigned by the LDA classifier in the Nile River Delta study area. *Water* and *outside study area* were labeled by application of appropriate masks to the map. The most noticeable error is that coastal regions are labeled almost entirely as *agriculture in desert/coast* regardless of their land use.

6.4 Conclusions

Orthonormal basis function classifiers were found to be suitable for processing remotely sensed land use change data from the Nile River delta. The cosine and Legendre orthonormal basis function systems did not differ significantly from the CART and KNN classifiers. LDA had the best classification rate of all tested systems, suggesting that the data are close to linearly separable.

Visual inspection of the canonical variates appears to bear this out. It may not be necessary to perform a nonlinear transformation to get the data into a space in which the classes are for the most part linearly separable. For such a database on which LDA performs exceptionally well, it may not be advantageous to transform the data as a step in classification. It may instead be best to use a simple linear discriminant if the data already appear to be clustered by class.

Chapter 7

Future Work

7.1 Introduction

The orthonormal basis function neural network classifiers introduced in this dissertation provide a viable platform for multidimensional pattern classification. Many opportunities exist to explore variations on these classifiers, a few of which have been identified during the course of this study.

The multitemporal LDA classifier used for Nile River delta land use change classification in Chapter 6 also could serve as a starting point for interesting variations. These might improve the classification accuracy of a linear discriminant as applied to such remote sensing data.

7.2 Future work in orthonormal basis function pattern classification

7.2.1 Stepwise regression for selection of model terms

Adaptive spline fitting methods such as MARS (Friedman 1991) and POLYMARS (Stone, Hansen et al. 1997) incorporate algorithmic approaches to select the terms that will be incorporated in a model. These use a forward and backward stepwise regression methodology (Friedman and Silverman 1989) to build a sequence of models, each differing by the inclusion or exclusion of a single model term. Such an approach might prove useful for selecting orthonormal basis function models, in particular if minimization of an objective function other than the MISE is desired.

Stepwise regression would need to be coupled with a cross-validation or generalized cross-validation methodology to determine the goodness of fit. A potential advantage of stepwise regression is that basis functions could be selected from a very large pool, since it might not be necessary to restrict the size of the pool to prevent overfitting.

7.2.2 Shrinkage for selection and fitting of model terms

An alternative to truncation methods (stopping and single-term exclusion) for model refinement is *shrinkage* (Tibshirani 1996; Hastie, Tibshirani et al. 2001). Shrinkage methods define an objective measure with a penalty on the size of the coefficients, such as a penalized residual sum of squares, and iteratively attempt to minimize this objective measure by shrinking the coefficients of a fit toward zero. Terms are eliminated from a model if their coefficients are shrunk to zero. This results in a biased model that may have significantly less error than the original fit. A potential drawback is that the iterative optimization steps of shrinkage methods can be computationally expensive and time-consuming. This might negate a major advantage of using orthonormal basis function expansions, the speed of fitting a model.

7.2.3 Objective thresholds for automated scree tests based on eigenvalue influence

In the automated scree test for principal component dimensionality introduced in 4.2.2, the threshold Γ_{elbow} was based on an ad hoc evaluation of two well studied psychological databases. These databases give some indication of the suitability of this method and of appropriate values for the threshold Γ_{elbow}. Additional study is required to determine whether this thresholding method is appropriate for a wider variety of

problems. If so, a key problem is to determine a technique for selecting appropriate thresholds in an objective manner.

7.3 Future work in land use change classification

7.3.1 Improving error rates through discriminant analysis on low-confidence data

Although the multitemporal LDA classifier achieved 93.5% user's accuracy on the Nile River land use change database, it may be possible to identify systematic sources of classification error and improve these results further. A potential approach is to identify the small percentage of data points misclassified, and then run further discriminant analysis on these data. If a linear discriminant is used as this second classifier, however, it is unclear how to combine it with the main LDA model, since an additive model of linear discriminants would yield a linear model, only with parameters different from the main, optimal LDA model.

Appendix A

Analysis of Variance Tables

A.1 Introduction

The following are tables summarizing results for the four-way ANOVA multiple comparison of Section 4.4. For each task with a particular number of exemplars, two tables are given. The first is for a four-way model without interactions, while the second is for the best four-way model with interactions, selected to minimize the Akaike information criterion (AIC) as implemented by the R Development Core Team (2003). The estimated AIC for each model is given, as is The Shapiro-Wilk test statistic for normality of the residuals. (R Development Core Team 2003).

The response variable is the zero-one classification error. The four factors under consideration in the ANOVA models, and their corresponding abbreviations in the results of the following sections, are as follows:

Abbreviation	**Factor**	**Levels**
pre	Preprocessing method	CVA, PCA
dim	Preprocessing dimensionality	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
lda	Postprocessing method	LDA, Maximum (no postprocessing)
bas	Basis	Cosine (discrete cosine basis with zero-crossing cutoff criterion), Daubechies (second-order Daubechies wavelets with scale product cutoff criterion), Legendre (polynomial basis with polynomial-order cutoff criterion)

A.2 ANOVA tables for the DELVE image segmentation task

A.2.1 Image segmentation, 70 exemplars

```
Analysis of Variance Table for the four-way model without
interactions

Response: Classification error
            Df Sum Sq Mean Sq  F value    Pr(>F)
pre          1 0.7379  0.7379 220.0544 < 2.2e-16 ***
lda          1 0.3126  0.3126  93.2172 < 2.2e-16 ***
dim         11 0.8931  0.0812  24.2107 < 2.2e-16 ***
bas          2 0.0551  0.0276   8.2191 0.0002858 ***
Residuals 1136 3.8094  0.0034
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.05790834
Estimated effects may be unbalanced

AIC:  -3276.729

Shapiro-Wilk normality test
W = 0.9874, p-value = 2.066e-08
```

Analysis of Variance Table for the best four-way model with interactions

```
Response: Classification error
              Df Sum Sq Mean Sq  F value    Pr(>F)
pre            1 0.7379  0.7379 240.3786 < 2.2e-16 ***
lda            1 0.3126  0.3126 101.8267 < 2.2e-16 ***
dim           11 0.8931  0.0812  26.4468 < 2.2e-16 ***
bas            2 0.0551  0.0276   8.9782 0.0001355 ***
pre:dim       11 0.2182  0.0198   6.4618 1.754e-10 ***
lda:dim       11 0.1203  0.0109   3.5626 6.031e-05 ***
lda:bas        2 0.0445  0.0223   7.2548 0.0007409 ***
pre:lda:bas    3 0.0219  0.0073   2.3806 0.0681024 .
Residuals   1109 3.4045  0.0031
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.05540619
Estimated effects may be unbalanced

AIC:  -3352.208

Shapiro-Wilk normality test
W = 0.9784, p-value = 4.266e-12
```

A.2.2 Image segmentation, 140 exemplars

```
Analysis of Variance Table for the four-way model without
interactions

Response: Classification error
            Df Sum Sq Mean Sq  F value    Pr(>F)
pre          1 0.5510  0.5510 114.9709 < 2.2e-16 ***
lda          1 0.0168  0.0168   3.5011   0.06159 .
dim         11 1.0870  0.0988  20.6175 < 2.2e-16 ***
bas          2 0.2647  0.1324  27.6157 1.946e-12 ***
Residuals 1136 5.4446  0.0048
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.06922985
Estimated effects may be unbalanced

AIC:  2865.302

Shapiro-Wilk normality test
W = 0.9978, p-value = 0.1294
```

Analysis of Variance Table for the best four-way model with interactions

```
Response: Classification error
            Df Sum Sq Mean Sq  F value    Pr(>F)
pre          1 0.5510  0.5510 130.2908 < 2.2e-16 ***
lda          1 0.0168  0.0168   3.9676   0.04663 *
dim         11 1.0870  0.0988  23.3647 < 2.2e-16 ***
bas          2 0.2647  0.1324  31.2955 5.988e-14 ***
pre:dim     11 0.3855  0.0350   8.2858 4.194e-14 ***
lda:dim     11 0.2525  0.0230   5.4278 1.842e-08 ***
lda:bas      2 0.1037  0.0519  12.2612 5.406e-06 ***
Residuals 1112 4.7029  0.0042
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.06503251
Estimated effects may be unbalanced

AIC:  -2986.004

Shapiro-Wilk normality test
W = 0.9974, p-value = 0.05682
```

A.2.3 Image segmentation, 280 exemplars

```
Analysis of Variance Table for the four-way model without
interactions

Response: Classification error
           Df  Sum Sq Mean Sq F value    Pr(>F)
pre         1 0.29272 0.29272 52.5511 1.399e-12 ***
lda         1 0.00436 0.00436  0.7829    0.3766
dim        11 0.57833 0.05258  9.4387 1.103e-15 ***
bas         2 0.34944 0.17472 31.3670 1.224e-13 ***
Residuals 560 3.11932 0.00557
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.07463383
Estimated effects may be unbalanced

AIC: -1337.235

Shapiro-Wilk normality test
W = 0.9885, p-value = 0.0001773
```

Analysis of Variance Table for the best four-way model with interactions

```
Response: Classification error
           Df  Sum Sq Mean Sq F value    Pr(>F)
pre         1 0.29272 0.29272 55.0445 4.537e-13 ***
lda         1 0.00436 0.00436  0.8200  0.365572
dim        11 0.57833 0.05258  9.8865 < 2.2e-16 ***
bas         2 0.34944 0.17472 32.8553 3.361e-14 ***
pre:lda     1 0.05366 0.05366 10.0910  0.001574 **
lda:dim    11 0.12843 0.01168  2.1955  0.013492 *
lda:bas     2 0.03366 0.01683  3.1647  0.043005 *
Residuals 546 2.90357 0.00532
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.07292386
Estimated effects may be unbalanced

AIC:  -1350.519

Shapiro-Wilk normality test
W = 0.9901, p-value = 0.0006168
```

A.3 ANOVA tables for the DELVE letter recognition task

A.3.4 Letter recognition, 390 exemplars

```
Analysis of Variance Table for the four-way model without
interactions

Response: Classification error
           Df Sum Sq Mean Sq  F value    Pr(>F)
pre         1 1.3955  1.3955 1099.751 < 2.2e-16 ***
lda         1 1.4404  1.4404 1135.114 < 2.2e-16 ***
dim        11 1.2007  0.1092   86.018 < 2.2e-16 ***
bas         2 3.4561  1.7281 1361.818 < 2.2e-16 ***
Residuals 848 1.0761  0.0013
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.03562223
Estimated effects may be unbalanced

AIC:  -3292.734

Shapiro-Wilk normality test
W = 0.9944, p-value = 0.002576
```

Analysis of Variance Table for the best four-way model with interactions

```
Response: Classification error
             Df Sum Sq Mean Sq   F value    Pr(>F)
pre           1 1.3955  1.3955 2487.8702 < 2.2e-16 ***
lda           1 1.4404  1.4404 2567.8693 < 2.2e-16 ***
dim          11 1.2007  0.1092  194.5915 < 2.2e-16 ***
bas           2 3.4561  1.7281 3080.7195 < 2.2e-16 ***
pre:lda       1 0.0379  0.0379   67.5294 8.694e-16 ***
pre:dim      11 0.3646  0.0331   59.0876 < 2.2e-16 ***
lda:dim      11 0.0597  0.0054    9.6694 < 2.2e-16 ***
pre:bas       2 0.0220  0.0110   19.6272 4.840e-09 ***
lda:bas       2 0.0215  0.0107   19.1416 7.686e-09 ***
dim:bas      22 0.1055  0.0048    8.5469 < 2.2e-16 ***
pre:lda:bas   2 0.0066  0.0033    5.8951  0.002878 **
pre:dim:bas  22 0.0236  0.0011    1.9155  0.007064 **
Residuals   775 0.4347  0.0006
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.02368397
Estimated effects may be unbalanced

AIC:  -3929.829

Shapiro-Wilk normality test
W = 0.9967, p-value = 0.07378
```

A.3.5 Letter recognition, 780 exemplars

```
Analysis of Variance Table for the four-way model without
interactions

Response: Classification error
           Df  Sum Sq Mean Sq F value    Pr(>F)
pre         1 1.18686 1.18686 1214.01 < 2.2e-16 ***
lda         1 0.94247 0.94247  964.03 < 2.2e-16 ***
dim        11 1.69071 0.15370  157.22 < 2.2e-16 ***
bas         2 2.83378 1.41689 1449.31 < 2.2e-16 ***
Residuals 848 0.82903 0.00098
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.03126713
Estimated effects may be unbalanced

AIC:  -3518.069

Shapiro-Wilk normality test
W = 0.9972, p-value = 0.1392
```

Analysis of Variance Table for the best four-way model with interactions

Response: Classification error

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
pre	1	1.18686	1.18686	3654.0327	< 2.2e-16	***
lda	1	0.94247	0.94247	2901.6223	< 2.2e-16	***
dim	11	1.69071	0.15370	473.2059	< 2.2e-16	***
bas	2	2.83378	1.41689	4362.2540	< 2.2e-16	***
pre:lda	1	0.04572	0.04572	140.7748	< 2.2e-16	***
pre:dim	11	0.25904	0.02355	72.5005	< 2.2e-16	***
lda:dim	11	0.04562	0.00415	12.7680	< 2.2e-16	***
lda:bas	2	0.03213	0.01606	49.4552	< 2.2e-16	***
dim:bas	22	0.15739	0.00715	22.0263	< 2.2e-16	***
pre:lda:dim	11	0.00996	0.00091	2.7879	0.0014564	**
pre:lda:bas	2	0.00299	0.00149	4.5986	0.0103537	*
pre:dim:bas	22	0.01667	0.00076	2.3324	0.0005372	***
lda:dim:bas	22	0.01786	0.00081	2.4996	0.0001778	***
Residuals	744	0.24166	0.00032			

Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.01802241
Estimated effects may be unbalanced

AIC: -4375.158

Shapiro-Wilk normality test
W = 0.9981, p-value = 0.4374

A.3.6 Letter recognition, 1,560 exemplars

```
Analysis of Variance Table for the four-way model without
interactions

Response: Classification error
           Df  Sum Sq Mean Sq F value    Pr(>F)
pre         1 0.95754 0.95754 1258.03 < 2.2e-16 ***
lda         1 0.72469 0.72469  952.10 < 2.2e-16 ***
dim        11 1.65515 0.15047  197.69 < 2.2e-16 ***
bas         2 2.29102 1.14551 1504.99 < 2.2e-16 ***
Residuals 848 0.64545 0.00076
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` '
1

Residual standard error: 0.02758881
Estimated effects may be unbalanced

AIC:  -3734.34

Shapiro-Wilk normality test
W = 0.9973, p-value = 0.1730
```

Analysis of Variance Table for the best four-way model with interactions

Response: Classification error

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
pre	1	0.95754	0.95754	3945.3743	< 2.2e-16	***
lda	1	0.72469	0.72469	2985.9321	< 2.2e-16	***
dim	11	1.65515	0.15047	619.9764	< 2.2e-16	***
bas	2	2.29102	1.14551	4719.8632	< 2.2e-16	***
pre:lda	1	0.04969	0.04969	204.7326	< 2.2e-16	***
pre:dim	11	0.14906	0.01355	55.8338	< 2.2e-16	***
lda:dim	11	0.07244	0.00659	27.1334	< 2.2e-16	***
pre:bas	2	0.00388	0.00194	8.0031	0.000362	***
lda:bas	2	0.03961	0.01980	81.5946	< 2.2e-16	***
dim:bas	22	0.12894	0.00586	24.1486	< 2.2e-16	***
pre:lda:bas	2	0.00840	0.00420	17.3073	4.387e-08	***
Residuals	797	0.19343	0.00024			

Signif. codes: 0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.01557883
Estimated effects may be unbalanced

AIC: -4673.478

Shapiro-Wilk normality test
W = 0.9969, p-value = 0.08897

Appendix B

Software for Orthonormal Basis Function Neural Network Classifiers

The software for orthonormal basis function classification are published in a public repository, the home page of which can be found at http://vera.bu.edu/orth_basis/.

This web server is managed by Dr. Michael A. Cohen:

Dr. Michael A. Cohen
Associate Professor of Cognitive and Neural Systems and Computer Science
Department of Cognitive and Neural Systems
677 Beacon St
Boston, MA 02215
(617) 353-9484
mike@cns.bu.edu

References

Abuelgasim, A. A., W. D. Ross, et al. (1999). "Change detection using adaptive fuzzy neural networks: Environmental damage assessment after the Gulf War." Remote Sensing of Environment **70**(2): 208-223.

Agrawala, A., Ed. (1977). Machine Recognition of Patterns. New York, IEEE Press.

Barron, A. R. (1993). "Universal Approximation Bounds for Superpositions of a Sigmoidal Function." IEEE Transactions on Information Theory **39**(3): 930-945.

Barron, A. R. (1994). "Approximation and Estimation Bounds for Artificial Neural Networks." Machine Learning **14**(1): 115-133.

Bartlett, M. S. (1954). "A note on multiplying factors for various χ^2 approximations." Journal of the Royal Statistical Society **16**: 296-298.

Basford, K. E. and J. W. Tukey (2000). Graphical Analysis of Multiresponse Data. Boca Raton, FL, Chapman and Hall.

Bentler, P. M. and K. H. Yuan (1996). "Test of linear trend in eigenvalues of a covariance matrix with application to data analysis." British Journal of Mathematical & Statistical Psychology **49**: 299-312.

Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford, Clarendon Press.

Breiman, L., J. H. Friedman, et al. (1984). Classification and Regression Trees. Belmont, CA, Wadsworth.

Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. Neurocomputing: Algorithms, Architectures, and Applications. J. Hérault. New York, Springer-Verlag**:** 227-236.

Card, D. H. (1982). "Using known map category marginal frequencies to improve estimates of thematic map accuracy." Photogrammetric Engineering and Remote Sensing **48**(3): 431-439.

Carpenter, G. A., M. N. Gjaja, et al. (1997). "ART neural networks for remote sensing: Vegetation classification from Landsat TM and terrain data." IEEE Transactions on Geoscience and Remote Sensing **35**(2): 308-325.

Carpenter, G. A., S. Gopal, et al. (1999). "A neural network method for mixture estimation for vegetation mapping." Remote Sensing of Environment **70**(2): 138-152.

Carpenter, G. A., S. Gopal, et al. (1999). "A neural network method for efficient vegetation mapping." Remote Sensing of Environment **70**(3): 326-338.

Carpenter, G. A., S. Gopal, et al. (2001). A neural network method for land use change classification with application to the Nile River delta. Boston, MA, Boston University.

Cattell, R. B. (1966). "The scree test for the number of factors." Multivariate Behavioral Research **1**: 245-276.

Caudell, T. P., S. D. G. Smith, et al. (1994). "NIRS - Large-scale ART-1 neural architectures for engineering design retrieval." Neural Networks **7**(9): 1339-1350.

Cawley, G. C. (2000). MATLAB Support Vector Machine Toolbox (v0.54β). Norwich, U.K., University of East Anglia School of Information Systems.

Cencov, N. (1962). "Evaluation of an unknown distribution density from observations." Soviet Math. Doklady **3**(1559-1562).

Cook, R. D. (1977). "Detection of influential observations in linear regression." Technometrics **19**: 15-18.

Craven, P. and G. Wahba (1979). "Smoothing noisy data with spline functions." Numerische Mathematik **31**: 377-403.

Daubechies, I. (1992). Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics.

Devroye, L., L. Györfi, et al. (1996). A probabilistic theory of pattern recognition. New York, Springer.

Diggle, P. J. and P. Hall (1986). "The selection of terms in an orthogonal series density estimator." Journal of the American Statistical Association **81**: 230-233.

Duda, R. O., P. E. Hart, et al. (2000). Pattern classification. New York, Wiley.

Efromovich, S. (1999). Nonparametric curve estimation : methods, theory and applications. New York, Springer.

Fix, E. and J. Hodges (1951). Discriminatory analysis--nonparametric discrimination: consistency properties. Random Field, Texas, US Air Force School of Aviation Medicine.

Fleiss, J. L. (1981). Statistical Methods for Rates and Proportions. New York, Wiley.

Frey, P. W. and D. J. Slate (1991). "Letter Recognition Using Holland-Style Adaptive Classifiers." Machine Learning **6**(2): 161-182.

Friedman, J. H. (1991). "Multivariate adaptive regression splines." Annals of Statistics **19**(1): 1-67.

Friedman, J. H. and B. W. Silverman (1989). "Flexible parsimonious smoothing and additive modeling." Technometrics **31**(1): 3-21.

Golub, G. H. and C. F. Van Loan (1989). Matrix Computations. Baltimore, Johns Hopkins University Press.

Gopal, S., C. E. Woodcock, et al. (1999). "Fuzzy neural network classification of global land cover from a 1 degrees AVHRR data set." Remote Sensing of Environment **67**(2): 230-243.

Gradshteyn, I. S., I. M. Rhyzik, et al. (1994). Table of Integrals, Series, and Products. Boston, Academic Press.

Greblicki, W. (1978). "Asymptotically optimal pattern recognition procedures with density estimates." IEEE Transactions on Information Theory **IT-24**: 250-251.

Greblicki, W. (1981). "Asymptotic efficiency of classifying procedures using the Hermite series estimate of multivariate probability densities." IEEE Transactions on Information Theory **27**: 364-366.

Greblicki, W. and M. Pawlak (1981). "Classification using the Fourier series estimate of multivariate density functions." IEEE Transactions on Systems, Man, and Cybernetics **11**: 726-730.

Greblicki, W. and M. Pawlak (1982). "A classification procedure using the multiple Fourier series." Information Sciences **26**: 115-126.

Greblicki, W. and M. Pawlak (1983). "Almost sure convergence of classification procedures using Hermite series density estimates." Pattern Recognition Letters **2**: 13-17.

Hall, P. (1981). "On trigonometric series estimates of densities." Annals of Statistics **32**: 351-362.

Hart, J. D. (1985). "On the choice of a truncation point in Fourier series density estimation." Journal of Statistical Computation and Simulation **21**: 95-116.

Hastie, T., R. Tibshirani, et al. (1994). "Flexible discriminant analysis by optimal scoring." Journal of the American Statistical Association **89**(428): 1255-1270.

Hastie, T., R. Tibshirani, et al. (2001). The Elements of Statistical Learning.

Haykin, S. (1994). Neural networks: a comprehensive foundation. New York, Macmillan.

Hettich, S., C. L. Blake, et al. (1998). UCI Repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Holzinger, K. J. and F. Swineford (1939). A Study in Factor Analysis: The Stability of a Bi-Factor Solution. Chicago, University of Chicago, Department of Education.

Joachims, T. (2000). The maximum-margin approach to learning text classifiers: method, theory and algorithms. Department of Computer Science. Dortmund, University of Dortmund.

Keppel, G. (1991). Design and Analysis: A Researcher's Handbook. Englewood Cliffs, NJ, Prentice Hall.

Kooperberg, C. and C. J. Stone (1999). "Stochastic optimization methods to fit Polyclass and feed-forward neural network models." Journal of Computational and Graphical Statistics **8**: 169-189.

Kronmal, R. A. and M. E. Tarter (1968). "The estimation of probability densities and cumulatives by Fourier series methods." Journal of the American Statistical Association **63**: 925-952.

Lenney, M. P., C. E. Woodcock, et al. (1996). "The status of agricultural lands in Egypt: The use of multitemporal NDVI features derived from Landsat TM." Remote Sensing of Environment **56**(1): 8-20.

Lord, F. M. (1956). "A study of speed factor in tests and academic grades." Psychometrika **21**: 31-50.

Mardia, K. V., J. T. Kent, et al. (1979). Multivariate Analysis. London, Academic Press.

McIver, D. K. and M. A. Friedl (2001). "Estimating pixel-scale land cover classification confidence using nonparametric machine learning methods." IEEE Transactions on Geoscience and Remote Sensing **39**(9): 1959-1968.

Mendenhall, W., D. D. Wackerly, et al. (1990). Mathematical statistics with applications. Boston, PWS-KENT.

Muchoney, D. and J. Williamson (2001). "A Gaussian adaptive resonance theory neural network classification algorithm applied to supervised land cover mapping using multitemporal vegetation index data." IEEE Transactions on Geoscience and Remote Sensing **39**(9): 1969-1977.

Nabney, I. and C. M. Bishop (2001). Netlab. Birmingham, Aston University.

Papoulis, A. (1987). The Fourier Integral and Its Applications. New York, McGraw-Hill.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods : support vector learning. A. J. Smola. Cambridge, Mass., MIT Press**:** 185-208.

Platt, J. C., N. Cristianini, et al. (2000). Large margin DAGs for multiclass classification. Advances in Neural Information Processing Systems 12, MIT Press.

R Development Core Team (2003). R: A language and environment for statistical computing. Vienna, Austria, R Foundation for Statistical Computing.

Rasmussen, C. E., R. M. Neal, et al. (1996). DELVE. Toronto, University of Toronto.

Rasmussen, C. E., R. M. Neal, et al. (1996). The DELVE Manual. Toronto, University of Toronto.

Ripley, B. D. (1994). "Neural networks and related methods for classification." Journal of the Royal Statistical Society **56**(3): 409-456.

Ripley, B. D. (1996). Pattern recognition and neural networks. Cambridge ; New York, Cambridge University Press.

Rubin, M. A. (1995). "Application of Fuzzy ARTMAP and ART-EMAP to automatic target recognition using radar range profiles." Neural Networks **8**(7-8): 1109-1116.

Rumelhart, D. E., G. E. Hinton, et al. (1986). Learning internal representations by error propagation. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. J. L. McClelland. Cambridge, MA, MIT Press. **1:** 318-362.

Schölkopf, B., C. J. C. Burges, et al. (1999). Advances in kernel methods : support vector learning. Cambridge, Mass., MIT Press.

Shock, B. M., G. A. Carpenter, et al. (2002). ARTMAP neural network classification of land use change. World Congress on Computers in Agriculture and Natural Resources, Iguaçu Falls, Brazil.

Singh, A. (1989). "Digital change detection techniques using remotely-sensed data." International Journal of Remote Sensing **10**(6): 989-1003.

Specht, D. (1971). "Series estimation of a probability density function." Technometrics **13**: 409-424.

Stone, C. J., M. H. Hansen, et al. (1997). "1994 Wald Memorial Lecture: Polynomial splines and their tensor products in extended linear modeling." Annals of Statistics **25**(4): 1371-1470.

Strang, G. (1993). "Wavelet transforms versus Fourier transforms." Bulletin of the American Mathematical Society **28**(2): 288-305.

Strang, G. and T. Nguyen (1997). Wavelets and Filter Banks. Wellesley, MA, Wellesley-Cambridge Press.

Tarter, M. E., R. Holcomb, et al. (1967). "A description of new computer methods for estimating the population density." Proceedings of the ACM **22**: 511-519.

Tarter, M. E. and R. A. Kronmal (1976). "An introduction to the implementation and theory of nonparametric density estimation." American Statistician **30**: 105-112.

Tarter, M. E. and M. D. Lock (1993). Model-free curve estimation. New York, Chapman & Hall.

Tibshirani, R. (1996). "Regression shrinkage and selection via the Lasso." Journal of the Royal Statistical Society Series B-Methodological **58**(1): 267-288.

Vapnik, V. N. (1995). The nature of statistical learning theory. New York, Springer-Verlag.

Weinstein, E. W. (1999). Haar function. MathWorld--A Wolfram Web Resource, Wolfram Research. http://mathworld.wolfram.com/HaarFunction.html.

Weinstein, E. W. (1999). Legendre polynomial. MathWorld--A Wolfram Web Resource, Wolfram Research. http://mathworld.wolfram.com/LegendrePolynomial.html.

Weisberg, S. (1985). Applied Linear Regression. New York, John Wiley & Sons.

Werbos, P. (1974). Beyond Regression: New Tool for Prediction and Analysis in the Behavioral Sciences, Harvard University.

Winer, B. J., D. R. Brown, et al. (1991). Statistical Principles in Experimental Design. New York, McGraw-Hill.

Curriculum Vitae

Byron Mitchell Shock

(831) 428-9006
broogle@gmail.com
bshock@alum.albertson.edu

Education

1997-2005 Ph.D., Cognitive and Neural Systems
Boston University, Boston, Massachusetts

Dissertation: *ARTMAP and Orthonormal Basis Function Neural Networks for Pattern Classification*

1990-1994 B.S., *summa cum laude*, Computer Science/Mathematics
Albertson College of Idaho, Caldwell, Idaho

Honors Paper: "Analytic Bach: computer-based mathematical style analysis of harmonic progressions"

Research Interests

Pattern recognition, neural networks, nonparametric statistics, artificial intelligence

Awards

1997-2001 National Science Foundation Graduate Research Fellow

1997-2001 Boston University Presidential University Graduate Fellow

1993 William Lowell Putnam Mathematical Competition
Nationally Ranked

1992-1994 Barry M. Goldwater Scholar

Professional Experience

1997-2005 Boston University, Boston, Massachusetts
Graduate Research Fellow (1997-2001) and Ph.D. Candidate
Graduate Student – In conjunction with advisors, developed machine learning algorithms based on neural networks and statistical

pattern recognizers. Responsible for research, technical implementation, analysis, and presentation of results.

1994-1997 Shock Consulting, Ontario, Oregon
Owner and Consultant – Provided a wide array of computer-related services to clients in rural communities. Services included training, relational database design and implementation, network client and server management, system troubleshooting, installation and configuration. Under consulting contract, primary point of contact for internal customers in the Research and Development departments at Ore-Ida Foods, Inc. Supported multi-user technical software packages. Designed and implemented network asset tracking database.
Instructor – As an instructor, helped launch the Industrial Training Center, now the Workforce Training & Education Department of Treasure Valley Community College, as one of the premiere providers of computer education in the region.

Summer 1993 Oregon State University Malheur Experiment Station, Ontario, Oregon
Research Intern – Designed, implemented and maintained software and hardware solutions to monitoring soil water infiltration and runoff, soil erosion, and soil moisture.

Summer 1992 Camp Morrison, Boy Scouts of America, McCall, Idaho
Ecology/Conservation Area Director – Obtained National Camping School certification in planning, developing, and implementing Ecology/Conservation programs. Taught courses in Ecology/Conservation. Trained and supervised the camp Ecology/Conservation instruction staff.

Publications

Shock, Byron M., Gail A. Carpenter, Sucharita Gopal, and Curtis E. Woodcock. (2001). “ARTMAP neural network classification of land use change”. Proceedings of the World Congress on Computers in Agriculture and Natural Resources, Iguassú Falls, Brazil, March, 2002. Technical Report CAS/CNS-TR-2001-009, Boston, MA: Boston University.

Carpenter, Gail A., Sucharita Gopal, Byron M. Shock, and Curtis E. Woodcock. (2001). "A neural network method for land use change classification, with application to the Nile River delta". Technical Report CAS/CNS-TR-2001-010, Boston, MA: Boston University.

Shock, Clinton C., Lynn B. Jensen, Joe H. Hobson, Majid Seddigh, Byron M. Shock, Lamont D. Saunders, and Timothy D. Stieber. (1999). “Improving onion yield and market grade by mechanical straw application to irrigation furrows”. HortTechnology 9:251-253.

Shock, Clinton C., Majid Seddigh, Joe H. Hobson, Ian J. Tinsley, Lucia R. Durand, and Byron M. Shock. (1998). "Reducing DCPA losses in furrow irrigation by herbicide banding and straw mulching". Agronomy Journal 90: 399-404.

Shock, Clinton C. and Byron M. Shock. (1998). "Comparative Effectiveness of Polyacrylamide and Straw Mulch to Control Erosion and Enhance Water Infiltration". In Arthur Wallace and Richard E. Terry, eds., Handbook of Soil Conditioners. New York: Marcel Dekker, pp. 429-444.

Shock, Clinton C., Joe H. Hobson, Majid Seddigh, Byron M. Shock, Timothy D. Stieber, and Lamont D. Saunders. (1997). "Mechanical straw mulching of irrigation furrows: soil erosion and nutrient losses". Agronomy Journal 89 (6): 887-893.

Shock, Clinton C., Lamont D. Saunders, Mary J. English, Robert W. Mittelstadt, and Byron M. Shock. (1995). “Water savings through surge irrigation, 1994”. OSU, Malheur Experiment Station Special Report 947:189-193.

Shock, Clinton C., Lamont D. Saunders, Byron M. Shock, Joe H. Hobson, Mary J. English, and Robert W. Mittelstadt. (1994). “Improved irrigation efficiency and reduction in sediment loss by mechanical furrow mulching wheat”. OSU, Malheur Experiment Station Special Report 936:187-190.

Shock, Clinton C., Lamont D. Saunders, Mary J. English, Robert W. Mittelstadt, and Byron M. Shock. (1993). “Surge Irrigation of Wheat, to Increase Irrigation Efficiency and Reduce Sediment Loss, 1993”. Oregon State University Agricultural Experiment Station Special Report 936: 157-161.

Computer Skills

Scientific and technical computing: Technical and database programming. Programming languages: MATLAB, SQL, S-PLUS and R, Prolog, Perl, Visual Basic, Borland Pascal, x86 Assembler. Knowledge of C, C++, FORTRAN, and JavaScript.

Workstation management: Installing, upgrading, maintaining, troubleshooting, and repairing network workstations. Workstation platforms: Linux, Windows (95, 98, ME, NT, 2000), Mac OS. Familiarity with SunOS, Solaris, and other Unix variants. Installing, upgrading, and supporting scientific application software. Key applications: MATLAB, R, SPSS, NCSS.

Network management: Installing and configuring network hardware and software. Standards and protocols: Ethernet, TCP/IP, IPX, SMB, HTTP, FTP, ssh. Installing, upgrading, and maintaining network file and print servers. Network operating systems: Linux (Red Hat), Netware.

Security: Implementing fundamental data security measures, including data backups, virus protection, user and group permissions, file encryption, and secure communications.

Webmaster

Languages

Spanish	Recent fluency College minor
Portuguese	Intermediate

References

Dr. Michael A. Cohen
Associate Professor of Cognitive and Neural Systems and Computer Science
Department of Cognitive and Neural Systems
677 Beacon St
Boston, MA 02215
(617) 353-9484
mike@cns.bu.edu

Dr. Eric L. Schwartz
Professor of Cognitive and Neural Systems, Neurobiology, and Electrical Engineering and Computer Systems
Department of Cognitive and Neural Systems
677 Beacon St
Boston, MA 02215
(617) 353-6179
eric@cns.bu.edu