Simultaneous Adaptive Fractional Discriminant Analysis: Applications to the Face Recognition Problem

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

John Daniel Draper, B.S., M.S.

Graduate Program in Statistics

The Ohio State University

2012

Dissertation Committee:

Dr. Prem Goel, Advisor

Dr. Radu Herbei

Dr. Yoonkyung Lee

Abstract

Linear Discriminant Analysis (LDA) has served as a standard technique in classification for many years. Many improvements have been proposed to boost the performance of LDA yet maintain its simplicity, intuitive appeal, and robust nature. Lotlikar and Kothari proposed fractional LDA (F-LDA) as an improved version of LDA. However, for a large number of classes, F-LDA is not feasible to implement in real time due to huge computational effort in the sequential search process for each dimension to be reduced. In addition, F-LDA is a directed projection pursuit technique, which takes several iterations to reduce just one dimension. Our research is focused on modifying these methods to be applicable to the face recognition problem (high-dimensional image data, large number of classes). Simultaneous Adaptive Fractional Discriminant Analysis (SAFDA) is a procedure developed specifically to learn a specified or fixed low-dimensional subspace in which classes are well separated by sequentially downweighting *all* dimensions to be removed *simultaneously*. Via analysis of a weighted between class scatter matrix of whitened data, $\tilde{S}_B$, the best projected space is learned through a directed projection pursuit method that focuses on class separation in the reduced space (rather than the full space like LDA). An adaptive kernel (Gaussian) was found to be the most suitable to avoid extra time considerations inherent in cross-validation by allowing the data to determine optimal bandwidth choice. While the SAFDA algorithm showed a marked improvement over standard LDA techniques in terms

of classification, the additional computational time, compared to LDA, was minimal in situations involving a small number of classes (MNIST) as well as a large number of classes (AR face database). SAFDA also provides a procedure that matches (or in some cases, outperforms) the F-LDA benchmark in terms of classification, yet is much more feasible in computational time and effort.

This work is dedicated to my family for their constant support throughout my life–my grandfather, Alfred C. Draper, for showing me the wonders of numbers and mathematics; my sister, Allison G. Draper, for always keeping me grounded; my mother, Karen C. Zawada, for her ceaseless love and support; and my father, Daniel V. Draper, for his constant thoughts, prayers, advice, and late night conversations that kept me going whenever things seemed overwhelming. Thank you all for everything!

Acknowledgments

This work would never have been possible without the support of countless individuals and organizations. My deepest thanks go out to:

- My advisor, Dr. Prem Goel, for sticking with me through the years and providing countless hours of advice and statistical knowledge.
- My committee, Dr. Yoonkyung Lee and Dr. Radu Herbei, for their patience and insight.
- The OSU Department of Statistics, for providing countless teaching opportunities and allowing me to find my niche.
- Dr. Jon R. Woods and The Ohio State University Marching Band (TBDBITL), for providing me with an organization that has made my college years a joy. I only hope that I can say that I left TBDBITL a little better than I found it.
- And finally, and most importantly, to my family, for their countless thoughts, prayers, and support throughout the years.

I could have never reached this milestone without the immense support and advice I have received from everyone. I can never repay the debt of gratitude that I owe to these individuals, but I will continue to strive for the highest and, as Woody said, 'Pay it Forward'.

Vita

August 23, 1981	Born - Dayton, OH
2003	B.S. Mathematics–Florida State University
2003	B.S. Statistics–Florida State University
2005	M.S. Statistics–The Ohio State University
2004-present	Graduate Teaching Associate/Lecturer, The Ohio State Univeristy.
2011	Interdisciplinary Specialization in Comprehensive Engineering and Science of Biomedical Imaging–The Ohio State University

Fields of Study

Major Field: Statistics

Table of Contents

List of Tables

Table **Page**

List of Figures

Figure **Page**

Chapter 1: Introduction and Literature Review

A common problem in science and engineering applications involves classifying objects using features of the data. With the explosion of sensors and computers, data is simpler to collect than ever. In fact, while the primary issue facing statisticians used to be a lack of sufficient data to solve a problem, the modern challenges stem from the sifting through the myriad data that is much easier to collect and store. Now that the data collection/storage process is quite simple, researchers tend to collect more data than may be truly useful to classification. As the number of dimensions increases, both signal and/or noise may increase. Therefore, additional information can be harmful as well as helpful (if the added noise is greater than the signal). A classic strategy in statistics is to select features that help to separate the classes. These techniques have been studied in great depth for separating the noise from the signal. This chapter explores the techniques that seek to find features/dimensions that maximize separation between classes, thereby maximizing classification accuracy. Section 1.1 examines steps of the general classification process from full data-to-optimal reduced dimension-to-classification paradigms. Section 1.2 provides an in-depth literature review of the face recognition problem and how current techniques have been used to solve the very complex problem.

1.1 Dimension Reduction, Feature Selection, and Classification

1.1.1 Why not use all the data?

Statistical science remains an ever growing field due in part to its great inroads in extracting relevant information from large quantities of data. An explosion of these applications has occurred recently as database size dwarfed computational ability and the technology grew to make computations feasible in real time or even tractable. Once the computational power caught up with the theory, the field increased in scope greatly. One of the main components of the discipline is the classification of new datapoints based either on a labelled set of training points or simply on the clustering of the test points themselves in some space. To meet this goal of classification, there are often dimensions (or features) that are not useful to the actual process of discrimination, so the researcher must first remove the 'noise' dimensions and make sure that the data used for classification is informative. Determining optimal subspaces in which to perform classification procedures is one of the main hurdles in tackling the classification problem. Using statistical reasoning, the data from separate classes are assumed to come from some class specific distribution (whose shape is either known or potentially unknown to the researcher) and methods of feature selection or projection are used to mine the data for the most useful dimensions.

Due to a huge increase in computing power as of late, more research has been devoted to the study of data mining - searching huge datasets for interesting/useful features that may be used for classifying data. Many of these techniques were developed before the technology existed to carry out calculations in real time, but now the focus is tackling even larger problems that, even with today's computing power, are somewhat intractable. Much research has been focused on not only finding new dimension reduction/classification techniques, but also on implementing existing techniques in a more efficient manner. This is

more than a matter of expediency as the exponential growth of databases has again outpaced computing power on large datasets.

The first thought that arises post-data collection is to organize the data and use all of it to understand the underlying dataspace; however, this is often not the most advantageous approach when seeking to maximize classification accuracy. Often, data contains few real features/dimensions useful for classifying new test points. While it may be counterintuitive to seek to use only some of the collected data, it is imperative to first identify which features of the data actually elicit separation of the classes and which features correspond to background noise or model a feature that is identical in all classes. A simple metaphor in face recognition would state that there are two eyes in the image, but this information would not lead to identifying the individual. Of course, noting aspects of those eyes (shape, color, location, etc.) might lead to understanding the space in which individuals are separated, but the fact of having two eyes will not be very important in discrimination (except, of course, finding the one-eyed man). Many statistical techniques have found that a classification procedure's performance can, in fact, suffer if the entire set of collected features is used indiscriminately. The addition of noisy features/dimensions that yield no appreciable information regarding class separation can lead to a degraded understanding of the lower-dimensional space. A statistical idea that explains this concept is the bias-variance trade-off. The researcher needs to find the delicate balance of using the training data to garner the best understanding of the class structure while still maintaining the ability to generalize to unseen test points. A solid dimension reduction technique is crucial in identifying those dimensions that are useful for classification and those that are extraneous noise.

One of the areas of interest today is image analysis/classification. Images are in essence large, possibly multi-layered matrices that, when stored in a database, are too large to be

analyzed in great depth in real time. Simple techniques such as matrix inversion become intractable and impossible with respect to storage capacity and computing time when applied to these small n (training sample size) large p (dimension) datasets, which have now become the norm in many practical problems.

The following notations will be used throughout this dissertation. Each observation in the training or test set is denoted by an ordered pair $(\mathbf{x_i}, \mathbf{y_i})$, where $\mathbf{x_i}$ is the p-variate vector of predictors associated with the i^{th} observation and $\mathbf{y_i} \in \{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{C}\}$ is the respective class label. The mean of the training set observations in class j is denoted

$$\bar{x}_{[j]} = \frac{1}{n_j} \sum_{i:y_i=j} x_i, \tag{1.1}$$

where n_j is the number of observations in class j.

The grand mean of all points in the training set is denoted by

$$\bar{\bar{x}} = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{1.2}$$

where $N =$ number of observations.

The between and within class scatter matrices, the cornerstones of linear discriminant analysis methods, are defined below.

Between class scatter matrix

$$S_B = \sum_{j=1}^{C} n_j (\bar{x}_{[j]} - \bar{\bar{x}})(\bar{x}_{[j]} - \bar{\bar{x}})^T = \sum_{k=1}^{C-1} \sum_{j=k}^{C} \frac{n_i n_j}{N} (\bar{x}_{[j]} - \bar{x}_{[k]})(\bar{x}_{[j]} - \bar{x}_{[k]})^T. \tag{1.3}$$

Pooled estimate of within class scatter matrix

$$S_W = \sum_{j=1}^{C} S_{W[j]}, \tag{1.4}$$

where

$$S_{W[j]} = \sum_{i:y_i=j} (x_i - \bar{x}_{[j]})(x_i - \bar{x}_{[j]})^T. \tag{1.5}$$

Section 1.1.2 discusses how determining proper features is essential to good classification. Section 1.1.3 examines various dimension reduction techniques and their associated assumptions, as well as situations in which a specific technique is useful. Section 1.1.4 concludes Section 1.1 with a comprehensive look at the classification schemes in the literature.

1.1.2 Dimensions vs. Features: Is there a difference?

Dimension in a statistical context can be used as a pseudonym for *feature*. In other words, the dimension of the data refers to the number of features or attributes available to describe one data point in the data set. It is important to note that not all dimensions lead to valuable information concerning the classification of a specific data point. Some information is often redundant or simply not informative from a classification standpoint. Herein lies the impetus for dimension reduction in a regression or classification problem.

The rank of the data matrix tells the number of *linear* combinations of features that are not redundant (sometimes nonlinear combinations could be useful as in the kernel framework and not be directly discoverable via the rank). In many cases with large p small n (i.e., face recognition), the rank is *much* smaller than the dimension of the data. Some may term the actual dimension of the data more in line with the rank (number of non-redundant linear combinations) rather than the number of pure features. If the most informative pieces of information can be extracted with fewer features, the signal is more easily extractable from the noisy dataset, which leads to a clearer separation of the classes. This is because the non-informative "dimensions" are predominantly detrimental noise, which can cause the algorithm to focus on irrelevant features in the data and thus degrade classification results. Dimension reduction is an extremely useful technique used to remove pieces of the data

that are not informative for classification purposes and focus on the underlying features that describe the class structures. Dimension reduction techniques attempt to find optimal *linearly independent* directions that correspond to combinations of the original dimensions. Not only do these techniques "weed" out the uninformative directions by removing the directions associated with noise, but also combine the "informative" directions in such a way as to impose maximum separability in the orthogonal projections.

In a regression framework, dimension reduction can be likened to feature/variable selection. When performing a regression, there are often a large number of predictors from which one can select a few to explain the response variable. Often, multiple variables explain the variability in a similar subspace which means there is a redundancy that is unnecessary in the model. Perhaps the variables can be combined to form one direction or simply drop one or all of the variables that do a poor job in describing the response. When variables in the predictor set for regression have a very low multiple correlation with the response (after accounting for those predictors already included in the model), their additional contribution to understanding of the response is negligible and the variable is often dropped.

Because the dimension reduction procedure is used to find which dimensions or features are useful in classification, it must be performed prior to the implementation of one of the classification techniques listed in Section 1.1.4. Linear Discriminant Analysis (LDA) can be seen as a dimension reduction technique with the goal of minimizing classification error because it involves retaining dimensions that are most significant in class separation (i.e., maximizing separation between class centroids while minimizing within class variation). Other common dimension reduction procedures include principal component

analysis (PCA), independent component analysis (ICA), Projection Pursuit, various combinations of the aforementioned, etc.

Dimension reduction is often used as a "top-down" approach in which one starts with the full dataset, and pares it down to its most important components, whereas feature selection is used to refer to a "bottom-up" approach in which the procedure starts with nothing and selects the most useful dimensions (or features) given those already chosen. While these two "flavors" of selection are not equivalent in the context of the forward stepwise vs. backward stepwise regression procedures, the ideas listed here refer more to the standard usage of the words. Both refer to the selection of optimal attributes and could be used interchangeably with no confusion.

Sometimes the semantic usage of the word *feature* in image/face recognition refers to something tangible about the item to be classified or regressed. Examples of common features in a face recognition framework are distance between the eyes, skin color, face shape, hair, presence/absence of glasses, etc. All of these are physical features that can be individually *explained* with substantive contextual knowledge, as to why these features are important. With the term dimension, the tendency is to believe the dimensions are simply an abstract concept some of which are tangible and explainable features but others that are deeply buried in the data and not easily individually explained. In fact, factor analysis is based on combining data 'dimensions' into several informative and explainable 'features'. Obviously, the label for a given feature in factor analysis is assigned by the expert who combines background knowledge to choose an appropriate label for the conglomeration of dimensions.

The relationship between these synonymous ideas seem to stem from differing disciplines. Mathematicians and mathematically inclined literature tends to focus on the idea of

dimension reduction and using these techniques to disregard dimensions of no assistance in classification with complete comfort in the abstract nature of dimensions/directions in high-dimensional space. Feature selection tends to be used more in the image classification, engineering, and psychological fields as the concrete tends to override the abstract in the "real-world" focus of most applied sciences. It is quite an interesting philosophical question in which to delve to root out the intricacies of the nomenclatures, but when looking at the core definition that one tends to use for *dimension*, the commonalities emerge. A dimension can truly be defined as a feature or attribute of a datapoint, which exemplifies the dual nature of this concept.

1.1.3 Dimension Reduction Techniques

There are two basic types of statistical learning frameworks known as supervised and unsupervised. The supervised learning paradigm focuses on finding criterion that maximize class separability with respect to some metric (centroid distance, classification rate, etc.) using the points in the training data set that are labelled *a priori*. Unsupervised learning techniques focus on optimizing a criterion that does not involve any labelled data. For example, principal component analysis seeks to find directions corresponding to the greatest variability in the data (no class consideration is included). Other unsupervised learning methods are generally focused on clustering the data into classes that are unknown *a priori* (i.e., k-means, k-medoids, etc.). The supervised learning techniques are the focus when dealing with a general classification problem as the class labels are known for the training set, and the goal is to find projections that maximally separate the classes. The criteria of dimension reduction for classification stem from improving class separability. However,

there are some unsupervised learning methods that are prevalent in the classification literature for their generalizability such as principal component analysis (PCA) and independent component analysis (ICA).

Principal Component Analysis: PCA

Principal component analysis (also called Karhunen-Loève transform in engineering literature) is a technique of feature selection that seeks orthogonal projection directions of the data that maximize the total variability of the dataset (see, e.g., Abdi and Williams (2010)). To discover directions with high variability, the eigenvectors of the covariance matrix of the data associated with large eigenvalues are used. The standard method of solving for these eigenvectors is to perform a spectral decomposition of the covariance matrix. The eigenvectors associated with the largest eigenvalues are retained and form the columns of the projection matrix, while the others are simply dropped as these directions explain little variability in the data. As PCA is an unsupervised methodology (does not use data labels for its separability criteria), it is not specifically geared toward the goal of the classification problem, but has been used due to the simplicity and speed of the algorithm, as well as good classification results in some situations.

An implicit assumption of any classification problem is that the variability between classes is inherently larger than the variability within a class. If this were not the case, it would be impossible to determine whether the test image was the target person or simply a new (not in the database) image of another. PCA focuses this search for directions with high variability on the total scatter (or covariance) matrix. Directions associated with larger eigenvalues explain more variability in the data (in that reduced space), and, as between class variability is assumed to be much larger than within class variability, these directions may be most closely associated with class separability. While the algorithm does not use

the labels in determining the optimal projection direction, the labelled data enter post-projection to determine class centroid/covariance structure in the projected space. A test point is then projected into the derived space and the label of the nearest class (using some distance metric) is assigned to the test point.

Questions arise as to the usefulness of creating a projection space that does not consider the class labels when class separation is the final goal. However, when dealing with a very high dimensional space (often the case in image recognition), the between scatter indeed dominates the within, which leads to positive classification results shown by using PCA methods. Of course, this need not necessarily be the case in general. Turk and Pentland (1991) propagated the idea of PCA into face recognition and showed the separation created in the principal component space was indeed pronounced in most cases and led to good classification results, as the projection was able to remove some of the noise and select features that led to the separation of individuals. Sometimes classes are separated more easily in dimensions of low variability, but PCA is still pervasive in the literature.

Independent Component Analysis: ICA

Another commonly used unsupervised learning technique for the classification problem is independent component analysis (ICA). Similar to PCA, ICA focuses on maximizing a criterion not related to the labels of the data. Rather than maximizing the variability, ICA seeks to find projections that minimize the statistical dependence between an observation's components. To find directions that yield independent components, the observed data, $\mathbf{z}$, is assumed to be a transformed version of another vector $\mathbf{s}$ which has p independent components. This is an inverse problem in which the goal is to solve for the underlying independent signal $\mathbf{s}$ given the noisy, observed $\mathbf{z}$.

ICA is often referred to as Blind Source Separation (BSS) or "the cocktail party problem" (many people talking at once create a loud din, yet individuals can often pick out several independent conversations simultaneously). Comprehensive overviews of the concept of ICA are available in Comon et al. (1994), Hyvarinen and Oja (2000), and Leino (2004). The motivation behind ICA is the fact that the distributions of independent components are as far from normal as possible. The kurtosis (fourth central moment) of a distribution is often used as a measure of non-normality because the kurtosis of a normal distribution is zero. Other criteria that measure "non-normality" include negentropy (difference between the entropy of a Gaussian variable and the entropy of the measured variable) or the negative of mutual information (mutual information is large for a normal distribution).

As the ICA seeks to maximize some function of the kurtosis, it is much more complex and requires an iterative procedure. For situations in which the data is indeed a mixture of many independent features, the procedure can yield a better projection for classification than PCA, but PCA is much more common due to comparable results in most cases and a much simpler framework that uses only the first two moments of the data.

Locality Preserving Projections–Laplacian

Spectral graph theory (see, e.g., Chung (1997)) explores techniques, called locality preserving projections (LPP), of projecting data into a lower dimensional space to preserve the local structure of the data is preserved. To accomplish this goal, points that lie within an ε-neighborhood (or k- nearest neighbors) of each point are connected to form an adjacency map. Let $v_i = A^T x_i$ be a projection of the points into a lower dimensional space where $A = ||a_{ij}||$ is a connectivity matrix. The objective function to minimize is

$$g(\mathbf{v}) = \sum_{\mathbf{ij}} (\mathbf{v_i} - \mathbf{v_j})^{\mathbf{2}} \mathbf{W_{ij}}, \tag{1.6}$$

where

$$a_{ij} = \begin{cases} 1 & : v_i \text{ and } v_j \text{ are connected in the adjacency graph} \\ 0 & : \text{otherwise.} \end{cases}$$

$$W_{ij} = \begin{cases} e^{-\frac{||v_i - v_j||^2}{h}} & : a_{ij} = 1 \\ 0 & : \text{otherwise.} \end{cases} \tag{1.7}$$

The methodology of finding the best projection that maintains the local neighborhood (Niyogi, 2004) shows that minimizing this criterion can be learned via decomposition of the Laplacian matrix, $D - W$, where W is the weight matrix defined above and D is a diagonal matrix in which each element is the sum of the corresponding row (column) of W. Eigenvalue decomposition of this matrix leads to the best projection matrix under this criteria. Mathematical details of this procedure for dimension reduction can be found in Belkin and Niyogi (2003). This idea has been expanded on with the methodology of local linear embedding (Saul and Roweis, 2000; Pan et al., 2009; De Ridder and Duin, 2002).

Additional references on unsupervised learning methods, specifically clustering, are found in Jain et al. (1999). The remainder of this section will center on supervised learning techniques.

Fisher Linear Discriminant Analysis Motivated Dimension Reduction

Supervised learning dimension reduction techniques are intricately intertwined with the idea of classification. Duda et al. (2001) state the duality in the following manner: "The conceptual boundary between feature extraction and classification proper is somewhat arbitrary: An ideal feature extractor would yield a representation that makes the job of classifier trivial; conversely, an omnipotent classifier would not need the help of a sophisticated feature extractor." This summary brings to light the dual nature of the dimension reduction

(feature selection) algorithms and the classifier. Any method of supervised dimension reduction is developed to improve upon classification in some way.

To begin the description of the variety of the dimension reduction techniques, the most sensible beginning is the seeming gold standard in most discriminant literature: Fisher linear discriminant analysis (LDA). R. A. Fisher, often referred to as the father of modern statistics, first discussed the idea of finding an optimal linear boundary between two classes with common covariance as well as extending this to a multi-class framework (Fisher, 1936). In the paper, he shows the optimal projection direction for classification (orthogonal to the boundary) to be that corresponding to the direction of least within class variability, while maximizing the distance between class centroids (between class variability).

To find the best direction of projection for classification, Fisher examined the problem under some assumptions:

1. Multivariate Normality of within class distribution for all classes,

2. Common within class variance/covariance structure among all classes.

While these appear to be quite restrictive assumptions, the Fisher LDA technique has been shown to be quite robust in practice to deviations from the assumptions. The assumptions are required to prove *optimality* of the solution in the $(C-1)$ dimensional space, although good performance is not contingent solely upon these assumptions.

One of the primary research goals has been extending the optimal LDA solution in the two-class framework to a multi-class (C class) paradigm when seeking optimal solutions. It is important to note that while the solution to the two-class problem yields the optimal *single* dimension for classification, the same does not hold true (finding an optimal single dimension) in the C $(C > 2)$ class case. LDA optimality rules can, however, find the optimal

projection in $(C-1)$ dimensional space (Fisher, 1936; Johnson and Wichern, 1988; Rao, 2002). With that being said, many researchers desire to reduce the space to lower than $(C-1)$ dimensions as the classes may be quite separated in this reduced space. The fewer features used, the lesser the chance that noise can confuse the classification. However, the strict optimality results do not hold when the space is less than $(C-1)$ dimensional. Many of the techniques discussed below search for methods that can improve upon the low dimensional projections garnered via LDA that can often outperform the optimal $(C-1)$ dimensional space (and those of smaller dimension yielded by Fisher's approach).

Schervish (1984) showed that an optimal single dimension existed for a three-class normal problem and found an implicit form of such a solution. As this solution was certainly nontrivial and required a numerical method to find it, there was a gap of time before the problem of finding optimal subspaces of size less than $(C-1)$ was solved. In examining the nature of the performance of LDA in dimensions lower than the optimal $(C-1)$, Hamsici and Martinez (2008) have shown that there is indeed a Bayes optimal one-dimensional subspace that can be learned from a C-class normal linear discriminant problem. First, they proved that the region associated with the boundaries of the Bayes error function in one dimension was convex and then used convex optimization techniques to find the optimal projection yielding a one-dimensional subspace that minimizes Bayes error. From the optimal single dimension, more dimensions for separability could then be added sequentially using the same methodology, with the constraint that each added dimension was orthogonal to the previously determined dimensions. This solution is computationally quite intensive, as the minimization of Bayes error in the convex region involves an extensive search, but the results prove that LDA does not provide an optimal one-dimensional solution in the C-class framework.

Fisher's criterion to find the best single direction of separation was to maximize:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}, \tag{1.8}$$

with respect to the vector, $\mathbf{w}$, where the between and within scatter matrices are defined in Equations (1.3) and (1.4) .

For the two-class problem, the solution is simply

$$\mathbf{w} = S_W^{-1}(\bar{x}_{[1]} - \bar{x}_{[2]}), \tag{1.9}$$

where S_W is the pooled within class covariance matrix.

To find the best one-dimensional projection, Fisher's solution to maximize $J(w)$ for the C-class problem was to find the eigenvector of the matrix $S_W^{-\frac{1}{2}} S_B S_W^{-\frac{1}{2}}$ (or singular vector of $S_W^{-1} S_B$) associated with the largest eigenvalue (singular value). This is a special case of the problem of finding the best d-dimensional projections described in Equation (1.10) below. In most cases, S_W is not full rank (more dimensions than observations) so the Moore-Penrose generalized inverse is used, and the solution to the optimization problem is the generalized eigenvector of S_B with respect to S_W associated with the largest eigenvalue. This corresponds to maximizing the numerator of J, subject to the constraint $w^T S_W w = 1$ with respect to w. It is often implemented by first whitening the data and searching in the space of length one vectors (assuming S_W is full rank). After whitening the data, the optimization function for LDA is based *only* on the between scatter matrix of the whitened means. Problems may arise if the assumptions listed above (multivariate normality and more importantly equal class covariances) are violated, as the estimate of Σ_W (S_W) that is used to whiten the data may be a poor estimate for the true within class covariance matrix.

When extending this Fisher criterion to finding the best d-dimensional subspace, the trade-off between maximizing the between and minimizing the within scatter matrices simultaneously took the form of maximizing the following objective function:

$$J(\mathbf{W}) = \frac{|\mathbf{W}^T S_B \mathbf{W}|}{|\mathbf{W}^T S_W \mathbf{W}|}, \tag{1.10}$$

where the between and within scatter matrices are defined in Equations (1.3) and (1.4). Duda et al. (2001) states that this is only truly optimized in $(C-1)$ dimensions as the S_B matrix lies, at most, in a $(C-1)$-dimensional space. The columns of the optimal $\mathbf{W}$ are the generalized eigenvectors satisfying

$$S_B w_i = \lambda_i S_W w_i, \tag{1.11}$$

corresponding to the largest d roots of the polynomial $|S_B - \lambda S_W| = 0$ (see Duda et al. (2001) for details). The eigenvectors of $S_W^{-\frac{1}{2}} S_B S_W^{-\frac{1}{2}}$ make up the columns of the optimal projection matrix and can be found via spectral decomposition performed on this matrix, i.e.,

$$S_W^{-\frac{1}{2}} S_B S_W^{-\frac{1}{2}} = U \Lambda^{-1} U^T, \tag{1.12}$$

where Λ is a $diag(\lambda_1, \lambda_2, \ldots, \lambda_p)$. The λ's are the eigenvalues of $S_W^{-\frac{1}{2}} S_B S_W^{-\frac{1}{2}}$ arranged in decreasing order, and $U = [u_1, u_2, \ldots, u_p]$ in which u_i is the eigenvector associated with the i^{th} largest eigenvalue (λ_i).

Often, LDA is used as the standard in classification due to the supervised nature of the algorithm and its simplicity, but PCA is simpler algorithmically. Martinez and Kak (2001) provided an analysis of PCA techniques vs. LDA techniques. They compared these techniques and found that, in most cases, LDA outperformed PCA as expected. PCA and LDA were shown to be fairly competitive in very low-dimensional projections (1-6 roughly), whereas LDA became the more accurate classifier as the number of retained dimensions

increased. The primary cases in which PCA tended to provide superior performance were when there were few samples per class or when the samples were non-uniformly selected from the underlying distribution (poor explanation of the class covariance structure). This is to be expected as poor understanding of the within scatter matrix (not enough data spanning the space) can muddle the results of LDA.

The LDA algorithm has retained a position of prominence in the classification community because of its equivalence to a Bayesian optimal boundary between two normal classes. The assigned class can also be viewed as the closest class centroid with respect to Mahalanobis distance. As LDA supplies optimal separation in a decision theoretic framework as well as the between versus within framework, its use as a discriminant technique remains popular.

LDA Variants

Hastie et al. (1994) refer to flexible discriminant analysis (FDA) using optimal scoring in which LDA is likened to a multiple nonparametric regression paradigm using multiple dummy variables, one for each class, to denote class membership. Standard LDA is shown to be equivalent to canonical correlation analysis that associates directions of high correlation between the data and the response (class label) with the projection directions (Mardia et al., 1979). FDA modifies the standard linear regression technique to use more sophisticated nonlinear regression techniques such as MARS to find nonlinear boundaries. The purpose of FDA is to create or allow more flexibility to the rigid linear boundary constraints of standard LDA and allow for nonlinear separation between classes via nonparametric methods. Obviously, the computational cost is higher due to the more complex regression techniques, but if the problem at hand is more suited to a nonlinear boundary, FDA will outperform LDA.

To provide a smooth decision boundary, a penalty term is often added to the regression loss function in canonical correlation analysis which leads to penalized discriminant analysis (PDA). This is discussed in a companion paper by Hastie et al. (1995). PDA is used to combat the issue of overfitting in LDA when there are many correlated predictors (as there are in image space). This paper examines the LDA paradigm from the canonical correlation framework and seeks to find the directions associated with least squares in a regression setup; however, unlike the FDA methodology of using nonlinear regression techniques, a penalized least squares approach is used to find smoother boundaries. Penalized discriminant analysis is shown to be equivalent to penalized canonical correlation analysis and penalized optimal scoring methodologies.

Hastie and Tibshirani (1996) discuss evolving LDA methods to include Gaussian mixtures (mixture discriminant analysis – MDA). The essential technique stems from modelling the individual classes as mixtures of Gaussian distributions in which individual components of the mixture are discovered using the EM algorithm. These subclass components are then shrunken toward the class (component) centroid to induce more separation between classes. As proposed, MDA also incorporates FDA and PDA when finding optimal projections to form a smooth boundary between classes.

One detractor from the standard LDA framework is the concept of *outlier classes*. Outlier classes are described in the literature (Loog et al., 2001) as classes that are very easily separated from the others because the class mean is at a significant distance from the other class means. Outlier classes can inflate the S_B matrix by adding a class mean that significantly departs from the others. This inflation can cause the direction of outlier class separation to dominate the optimal projection direction at the cost of optimizing the separation between the more "difficult to classify" groups. Fortunately, an outlier class is very

simple to classify in most directions (as it is significantly different from the other classes with little to no overlap) so there is really no reason to use that direction as a focus for discrimination.

The weighted LDA methodology (wLDA) is a simple extension of the standard LDA algorithm that can ease the effects of outlier classes. The only difference is a weight term input into the calculation of the between scatter matrix. This weight term is inversely related to the distance between class centroids so it downweights the contribution of a class that is easily separated when calculating the between scatter matrix. This allows the algorithm to focus on finding discriminative directions that focus on difficult to classify classes as the outlier classes should be reasonably separated under almost any projection. The between scatter matrix in the weighted LDA framework is defined as

$$\tilde{S}_B = \sum_{k=1}^{C-1} \sum_{j=k}^{C} w(\delta_{jk})(\bar{x}_{[j]} - \bar{x}_{[k]})(\bar{x}_{[j]} - \bar{x}_{[k]})^T, \tag{1.13}$$

where $\delta_{jk} = ||\bar{x}_{[j]} - \bar{x}_{[k]}||$ and the weight function, $w(.)$, is a monotonically decreasing function of the distance that decreases faster than the inverse pairwise distance squared. Commonly used $w(\delta_{jk})$ are simply δ_{jk}^{-h}, where $h > 2$. Note the introduction of another parameter h in this choice of $w(.)$ that can possibly be optimized via cross validation. Everything else in this methodology proceeds in the same manner as in standard LDA.

This weighted LDA framework is another formulation of the Laplacian method of dimension reduction. The only difference is that the adjacency matrix is constructed on the space of means rather than all datapoints. Kouropteva et al. (2003) expands the Laplacian LPP usage by connecting points that are of the same class in the adjacency graph. Weighted LDA connects all class means and seeks to find a dimension that maintains the local properties of the class means.

Lotlikar and Kothari (2000) posited a unique framework of the LDA approach that utilized not only the aspects of weighted discriminant analysis, but also discussed down-weighting the least informative dimension in an iterative manner, thereby allowing the space to rotate as the dimension was downweighted. This introduced more flexibility into the LDA paradigm as the dimension was removed fractionally rather than as a whole. The fractional LDA framework is discussed more completely in upcoming sections. The results, while quite computationally taxing, provide excellent separation of class centroids.

Tang et al. (2005) discuss this issue similar to Loog et al. (2001), but the focus is on the effect of the outlier class on the *within* scatter matrix estimation. The rationale being that if a class is well separated to begin with, its within covariance matrix should not be as important in developing an estimate of the pooled covariance matrix. To accomplish this goal, Tang et.al. propose a relevance weighted within scatter matrix.

$$S_W = \sum_{j=1}^{C} r_i S_{W[j]}, \tag{1.14}$$

where $S_{W[j]}$ is defined in Equation (1.5) and the weight r_i is related to the distance of the i^{th} class to the others:

$$r_i = \sum_{j \neq i} \frac{1}{w(\delta_{ij})}. \tag{1.15}$$

The proposed modification utilizing both weighted between and weighted within scatter matrices produces results that are comparable or superior than the standard LDA framework. The weight function defined by Tang, et.al. is *directly* proportional to the distance between classes (opposite of the definition utilized in wLDA). Tang et.al. also discuss updating the weights using an adaptive strategy.

In the image classification problem, the dimensionality of the data is much larger than the number of observations which yields many zero eigenvalues in the within class scatter

matrix. There have been a variety of solutions to this problem, the most common of which is replacing the within scatter matrix with the total scatter matrix (Chen et al., 2000). The optimal direction found in this framework leads to the same solutions, as the total scatter matrix can be decomposed into the sum of the between and within matrices. Therefore, finding directions which maximize the criterion

$$J^*(\mathbf{W}) = \frac{|\mathbf{W}^T S_B \mathbf{W}|}{|\mathbf{W}^T S_T \mathbf{W}|} \tag{1.16}$$

achieves the same goal as the standard LDA setup described in Equation (1.10). This allows use of the total scatter matrix (S_T) which is more likely to be of higher rank than S_W ($rank(S_T) \leq rank(S_B) + rank(S_W)$). S_T still may not be of full rank when working with the extremely high-dimensional image space, because S_W is almost certainly not of full rank and it is doubtful that S_B would contain enough classes to make up the discrepancy (recall, $rank(S_B) \leq (C-1)$). Other solutions discussed include projecting the data into the null space of S_W before maximizing the quantity $a^T S_B a$. This diagonalizes the within scatter matrix before maximizing over linear projections of the between scatter.

Yu and Yang (2001) suggest an alternate formulation of the problem that seeks to first diagonalize the between scatter matrix (project the data into the column space of S_B) then minimize over all remaining projections of the within scatter matrix. They called their approach direct-LDA (D-LDA) that only required an optimization over a (relatively) small subspace (maximum dimension $C-1$) and claimed the solution was equivalent to LDA. Gao and Davis (2006) proved that the D-LDA solution did *not* necessarily match the LDA optimal solution as the projection into the column space of S_B ignores S_W and can have significant effects in the within space. Regardless, D-LDA has been used as a time saving method that yields fairly strong (although not necessarily optimal) classification rates with

a much smaller space to search. This methodology is often used to combat the small sample size (SSS) problem which is often the case in face recognition as small sample sizes severely limit the estimation of the within scatter matrix (recall, D-LDA only focuses on S_W after the S_B has been maximized).

Estimation of the class covariance structure for LDA is often an issue. Croux et al. (2008) provide an efficiency analysis for the LDA procedure. They discuss using other measures for class center and spread that are more robust in the same LDA framework and its effects on classification error. The effects of these estimates are found using influence functions. Outlier points' effects are estimated and level of significance in overall performance is calculated analytically in the two-class case, but calculations become intractable for more classes.

Friedman (1989) examines a regularized framework of the standard LDA problem (RDA). As estimation of the class covariances is often an issue due to small sample size, a more robust estimate of the class covariance matrix can be garnered by shrinking the individual classes toward a common covariance matrix. Friedman went a step further to "shrink the shrunken estimates" toward a multiple of the identity in which the trace of the matrix is held constant. This is a method of finding more robust estimates for the class covariance matrices which in turn lead to better estimation of the optimal projection directions for discrimination. As each class covariance matrix was estimated separately, the final classification was done using minimum Mahalanobis distance with different class covariance matrices leading to quadratic boundaries.

Bensmail and Celeux (1996) further examine the regularized framework when estimating the within class covariance matrix. Often, the number of samples per class is small when compared to the dimensionality of the space which leads to difficulty in estimation of

S_W. Their approach is a regularization approach (in a sense) in which the covariance matrix is fit into one of a list of constrained covariance forms with fewer parameters (i.e., diagonal, spherical, etc.). Another constraint commonly imposed in the literature is on the set of all class covariance matrices (common (LDA), same eigenvectors different eigenvalues, or free (QDA)). This technique, referred to as Eigenvalue Decomposition Discriminant Analysis (EDDA), was then compared to Friedman's RDA procedure. The results were generally comparable as the EDDA framework allowed the data covariance structure to determine the model selected. The computational cost was higher as many models are examined, but the cost of forming the models is not very high due to the parsimonious nature.

PLS and other techniques

Barker and Rayens (2003) discuss a fairly simple yet not as pervasive method of dimensionality reduction that bridges the gap between PCA and LDA known as partial least squares (PLS), which has been very popular in Chemometrics (see, e.g., Wold et al. (2001); Dayal and MacGregor (1997)). It focuses on an eigen-decomposition of the correlation structure between the predictors and the response (in this case, class label). The PLS directions seek to maximize correlation between predictor and response. The technique is very similar to LDA and requires labelled classes for training, but the standard between and within class scatter matrices are slightly modified in the PLS paradigm.

More complex methods of dimensionality reduction, such as nonlinear manifold learning (Law and Jain, 2006) and kernel PCA/LDA (Yang et al., 2004), have been proposed. Nonlinear techniques seek to find a lower dimensional manifold on which the data lie and project the data into that space. While this is a nice technique in concept, solving for the underlying manifold is quite complex. Kernel Fisher Discriminant (KFD) as discussed by Mika et al. (1999) for two classes and Bavaud (2010) for multiple classes involves finding

optimal separation in the projected high dimensions. KFD becomes a dimension reduction technique *after* the data have been implicitly projected into a higher dimension in which nonlinear boundaries become linear. These techniques convert nonlinear classifiers in the data dimension into linear classifiers in much higher dimensions.

Projection Pursuit

Friedman and Tukey (1974) described a technique in which 'interesting' directions in the data were sought to yield understanding of the data structure. This is a very loose definition since what is 'interesting' for one study may not be so for another. For their criteria, they developed a function that examined overall spread of the data as well as the *local density* that they termed the P-index. The algorithm was akin to many clustering algorithms, yet it focused on linear mappings that yield high local density clusters as well as high overall variability (measure of cluster separation) in the data. This trade-off was achieved by using the product of trimmed standard deviation (spread) and a local concentration measure (density) as the objective function. To maximize this objective function, a random search algorithm based on simulated annealing is often used in which the current projection is perturbed in a random amount to yield a new direction. This direction is maintained if the objective function has increased. The annealing is applied to start with large random perturbations and slowly 'cool' the process such that the perturbations are smaller as the new projections near the optima.

One benefit of the random search is that the algorithm can escape from local minima, but it is more computationally expensive than other gradient methods (which are not applicable here as the objective function is not smooth). Other benefits of the projection pursuit paradigm are that the directions garnered are linear mappings of the original data space and therefore simpler to interpret than many nonlinear functions. However, it is important to

note that the original projection pursuit (PP) algorithm was defined for unsupervised learning in which the interesting directions led to a clustering of data with the only goal being maximal data concentration and clustering (no labels or objective clusters sought). While this is useful for exploratory data analysis, many classification methods seek to maximize separability of predetermined classes (supervised) or using some predetermined information.

Posse (1992) introduced supervised methods in which projection pursuit density estimators were used for discriminant analysis for two classes. Polzehl (1995) extended this approach using a measure theoretic perspective. These results stray from the LDA paradigm but use kernel density estimators to yield classification rates. The end goal is construction of a classifier based on data driven densities. This leads to a nonparametric approach but is more of a black box outcome with little interpretability.

Torkkola (2001) brought some LDA techniques into document classification with extremely high-dimensional data defining a document (class). He used projection pursuit with random projection search to reduce the extremely high dimensionality of the problem to a feasible size on which LDA can be effectively performed. Gribonval (2005) merged some of the ideas of projection pursuit with CART to form what is termed adaptive discriminant analysis. In this technique, linear features are iteratively selected from a dictionary based on some information criterion (mutual information, Kullback-Leibler divergence, etc.).

Lee et al. (2005) describe a similar projection pursuit method using a supervised objective function similar to Fisher's LDA in which the interesting directions were determined to be those which maximized between class scatter and minimized within class variation. This technique also used a simulated annealing approach in updating the projection space

from the principal axes which prevents the need for eigenvalue decomposition but introduces a random search criterion that is not guaranteed to converge to the optimal direction in an efficient manner. The major contributions of this paper involve bridging the gap between supervised projection pursuit and LDA. By using LDA driven objective functions, the projection pursuit methodology is focused on finding directions that yield class separation under the Gaussianity assumption. The technique proposed maintains many of the pitfalls of LDA (i.e., Gaussianity, common class covariance matrices, centroid based, etc.), but it focuses the search for optimal directions in the final projected space rather than the whole space. A big issue with this method is the random search of directions which is computationally very expensive (especially when matrix inversions/decompositions are involved).

Fractional LDA

Fractional linear discriminant analysis (F-LDA) was first discussed in Lotlikar and Kothari (2000). It was suggested as a technique that would allow a little more flexibility to standard LDA. Rather than dropping dimensions completely that did not appear to be useful, the authors sought to slowly (*fractionally*) downweight the dimension so as to allow *any* useful information from that dimension/direction to remain in the classifier. It provides a nice compromise between the strict gold standard of Fisher LDA which selects directions concurrent with the eigenvectors associated with the larger eigenvalues of $S_W^{-\frac{1}{2}} S_B S_W^{-\frac{1}{2}}$ and a more flexible method. This new method can be likened to a slow "squeezing" or "contracting" along the dimension/direction of least importance while allowing said dimension to slow mold to the optimal compression direction as the dimension is collapsed. The dimensions which are downweighted are determined via the weighted between class scatter matrix to emphasize directions of class confusion similar to the Laplacian methodology.

In fact, the algorithm as explained in Lotlikar and Kothari (2000) is actually more likened to a projection pursuit method in which the optimization criterion is (essentially) modified to only focus on class separation *in the final projected dimension*. To optimize in a rotated and lower dimensional space, Lothlikar and Kothari update the projection direction that leads to discarding the lowest $p-d$ dimensions (with regards to eigenvalue size) by 'shrinking' the dimensions to be dropped by larger and larger amounts to give the appearance of the lower dimensional space. By shrinking the dimension, the researcher is able to 'learn' the class separation in the projected space rather than in the full space as the lower dimensional space may induce confusion not seen in the full space.

The algorithm for the F-LDA procedure (preprocessing and dimension reduction procedure) is given below in Algorithm 1:

The primary purpose of the F-LDA algorithm was to take into account the difficult to distinguish classes in a lower dimension when they are easily separable in a higher dimension. The LDA framework focuses on an optimal projection for C classes into $(C-1)$ dimensions. An outlier class can cause significantly increased eigenvalues of the between class covariance matrix which inflate the directions that separate the outlier class from the others. With this discrepancy, the optimal projection often focuses on the dominating factors in S_B without considering its effect on the within separability (whitening) or more importantly the resulting class separability in the reduced space. Weighted LDA has been used to combat the outlier class problem, but because it is a one-pass problem, it does not allow the least informative direction to change as it is fractionally downweighted.

F-LDA has the major advantage of slowly downweighting the least-informative dimension which has the effect of not allowing one class to dominate any aspect of the separability criterion. The *fractional* step-down approach allows an adaptive framework to adjust

Algorithm 1 Fractional LDA

Require: $(\mathbf{X}_i, \mathbf{Y}_j)$

Ensure: W FLDA projection matrix

$\tilde{X}_i \leftarrow X_i - \bar{\bar{X}}$ {Mean Centering}

$S_T \leftarrow \tilde{X}'\tilde{X}$ {Total Scatter Matrix}

Remove the Null Space of S_T from the data

$S_{W[j]} \leftarrow \sum_{i:y_i=j} (\tilde{x}_i - \bar{\tilde{x}}_{[j]})(\tilde{x}_i - \bar{\tilde{x}}_{[j]})^T$

$\tilde{S}_W \leftarrow \varepsilon I + \sum_j \frac{n_j}{n} S_{W[j]}$ {Regularized Within Class Covariance Matrix}

$\tilde{\tilde{X}}_i \leftarrow \tilde{S}_W^{-\frac{1}{2}} \tilde{\mathbf{X}}_i$ {Whiten the Data}

$M_{[j]} \leftarrow \frac{1}{n_j} \sum_{i:y_i=j} \tilde{\tilde{X}}_i$ {Whitened class means}

$\alpha = r^{\frac{1}{r-1}}$ where r is the number of fractional steps

for $q = p$ to d **do**

Initialization

$W \leftarrow I_q$

$S_{FLDA} = \begin{bmatrix} I_{q-1} & 0 \\ 0 & \alpha \end{bmatrix}$

for $k = 0$ to $r-1$ **do**

Rotate class means to current 'optimal' rotation and downweight

$M^{(k)} \leftarrow W^T M$ {This is an identity transformation when k=0}

$Z \leftarrow S_{FLDA}^k M^{(k)}$ {This scales down the 'least informative dimension'}

$\tilde{S}_B \leftarrow \sum_{j=1}^{C} \sum_{i \neq j} \frac{n_i n_j}{n} w(\delta^{(ij)})(z_i - z_j)(z_i - z_j)^T$ where $w(\delta^{ij}) = ||z_i - z_j||^2$

$\tilde{S}_B = \Phi \Lambda \Phi^T$ {Spectral Decomposition where the eigenvalues in Λ are in decreasing order}

$W \leftarrow W\Phi$ {Updating the 'optimal' direction in the 'fractionally lower dimension space'}

end for

$W \leftarrow$ First q columns of W {Discard the least informative dimension}

end for

The final W is a pxd dimensional matrix to project the p dimensional data into d dimensional space

the optimal projection dimension as the importance of the least-informative dimension is lessened. It provides the option of modifying the projection dimension as the dimension of least importance is incrementally reduced, which permits the projection space to rotate to a direction in which better class mean separation is achieved for the training data. The least informative direction in a high-dimensional space may correspond to the direction of an outlier class (recall, the modified formulation of S_B downweights the impact of the outlier class on the between scatter). If that dimension were simply dropped, a great deal of confusion could be inadvertently introduced as the dimension separating the aforementioned outlier class has been removed *completely*. F-LDA permits a slow contraction along this dimension until the confusion increases (i.e., the distance between the outlier class and the others becomes small enough to effect the between scatter matrix) and rotates the projection space to maximize separation to improve upon the wLDA paradigm.

Of course, this stepwise procedure introduces a great deal of computational complexity as each fractional step requires a new computation of the between scatter matrix, its spectral decomposition, and projection matrix. Also, each of these fractional steps occurs many times in a reduction of just one dimension! The computational cost of reducing the dimension of the data from image space, which at the absolute least is of dimension 256 (often much larger), to a feasible realm for discrimination (i.e., 10-50) takes an enormous amount of time as there are r fractional steps for each dimension reduced. This is a significant disadvantage as the computational feasibility of the algorithm in the context of image classification (or more generally large p small n) is quite low. Recall, that this computational cost is incurred in training only, so once the algorithm has been trained to find the optimal projection, the cost is minimal, but if there was any hope to update the training set online

or at any reasonable time interval, the standard F-LDA algorithm is not very practical in real time.

While the early views of F-LDA have shown difficulty in training in an efficient time frame, the classification results are often quite good (outperforming LDA and many other techniques in most cases). F-LDA not only maintains the benefits of weighted LDA in dealing with outlier classes, but also modifies the projection to allow for heretofore unseen confusion upon the removal of dimensions of lower importance. LDA finds the direction of least importance given a criterion and collapses over it with no regard to the class separation in the resulting space. F-LDA allows the algorithm to mold to the projected space and the separability within it rather than finding the least informative direction prior to projection as LDA does.

1.1.4 Classification Techniques

The classification problem has taken on a variety of facades over the years. While new techniques have arisen to attack the disadvantages encountered by certain methods, one constant has consistently emerged – optimal methods are situation dependent. Nevertheless, new and more sophisticated techniques are being developed currently to provide more options in lowering misclassification rate. In this section, an in-depth review of a multitude of classification methods is conducted. This list is by no means exhaustive but provides an examination of the most popular classification techniques.

In its simplest form, the problem revolves around developing a rule for assigning an unlabelled datapoint to one of possible C classes based on a set of measured and/or derived predictor variables. In the supervised learning case (which is the focus of the research), a set of labelled datapoints is provided to form a training dataset. From the data labelled

a priori, the decision rule is constructed to minimize a penalty functional (the definition of which separates many of the classification methods) in terms of misclassification. The possible set of classes to which a datapoint could be assigned is, therefore, defined before the decision rule is constructed.

Linear Discriminant Analysis

Fisher's linear discriminant analysis (LDA) is one of the most commonly used methods to solve the classification problem. This procedure optimally separates two multivariate classes with common covariance matrix by finding directions in which the variability within a class is minimized while maximizing the variability of the class centroids. Once this direction is determined, the data is projected into this direction and is assigned the class of the centroid closest to it (in the projected space).

One of the major benefits of LDA is the flexibility to choose the dimension size of the projection. While the optimal projection dimension is related to the problem and the specific data provided, there is often a significant dropoff in the relative sizes of the eigenvalues. This indeed is a question for dimensionality reduction rather than classification, but the dimension size that is chosen can affect classification results.

Under the primary assumptions that the individual classes follow a multivariate normal distribution and with a common covariance matrix Σ, this method was extended to C class problem. LDA has stood the test of time due to the simplicity of the algorithm and the ability to work reasonably well in cases in which the assumptions are not satisfied. Even if the data does not adhere to these underlying assumptions, LDA tends to produce reasonably good classification results.

Quadratic Discriminant Analysis

The Quadratic Discriminant Analysis (QDA) methodology (see, e.g., Hastie et al. (2001)) is quite similar to linear discriminant analysis except that the assumption of common covariance matrices (homoscedasicity) is removed. To estimate the class label of a test point, the class means and within covariance matrices for each class are estimated from the training data. The Mahalanobis distance is computed from the test point to each class mean using the respective class covariance matrix. The class yielding the smallest distance is chosen as the label for this datapoint.

When the assumption of equal class covariance matrices is removed, every class covariance matrix must be estimated individually, and the added cost is in estimating and storing a great deal more parameters. Rather than estimating and storing $\frac{p(p-1)}{2}$ parameters in the pooled within covariance matrix using all the data in the training set, QDA requires computing as many parameters *for each* of the C classes using only points in the training data set of a given class. More data is essential to approximately estimate the class covariance matrices. In addition to the greater storage requirement and the added variance in estimation, there is also more computational cost in classification as the distance from a test point must be computed to each class centroid under a different Mahalanobis metric.

Support Vector Machines

Support vector machines (SVM) focus on determining class boundaries that maximize the separation between classes (margins) (see, e.g., Hastie et al. (2001); Steinwart and Christmann (2008)). In the simplest case when the classes are linearly separable, the margin is determined by the points nearest a potential boundary (called the support vectors). The boundary is chosen to maximize the spacing between the support vectors (margin).

Similarly to LDA, this method finds *optimal* separating hyperplanes in $(p-1)$ dimensions, but the criterion is different. SVM is a margin based approach in which points at the boundaries of the respective classes determine the separating hyperplane, whereas LDA is a centroid-based method (i.e., centroids and class covariance *structure* determine class separation).

The separating hyperplane is defined as $w^T x - b = 0$ and the two parallel planes that determine the boundaries of the margin in the separable case are defined as $w^T x - b = 1$ and $w^T x - b = -1$. This implies all points in class '1' follow the rule $w^T x_i - b \geq 1$ and those in class '-1' follow $w^T x_i - b \leq -1$ as there are no points within the margin (in the separable case). To simplify matters, these constraints can be combined into one using $c_i \in \{1, -1\}$ as the class label. The new constraint function is then $c_i(w^T x_i - b) \geq 1$. The width of the margin is $\frac{2}{||w||}$ so minimizing $||w||$ is the objective under the individual constraints on the points lying on the correct side of the margin.

Using LaGrange multipliers, the objective function is

$$argmin_{w,b} \left\{ \frac{1}{2}||w||^2 + \sum_{i=0}^{n} \alpha_i c_i \left[(w^T x_i - b) - 1 \right] \right\}. \tag{1.17}$$

Using standard quadratic programming techniques, the solution is of the form $w = \sum_{i=0}^{n} \alpha_i x_i$. While the solution appears on the surface to be heavily dependent on the data as well as the coefficients (α_i where $i = 1, 2, \ldots, n$), many of the α_i's are zero yielding a sparse solution. In fact, only the points that lie on the boundary of the margin contribute to the construction of the separating hyperplane (nonzero α).

The formulation of the problem (known as the primal problem) given above is fairly difficult to solve in practice as the optimization is occurring over the vectors w and b. Another approach (known as the dual form) provides more constraints but optimizes over a smaller space. The primal problem seeks to optimize over a space corresponding to the

size of the dimension of x (which in many useful cases is extremely large), whereas the dual problem seeks to maximize over the number of α's (which is the sample size). This may seem to make the problem more difficult, but in many cases related to image analysis and face recognition, the number of training data points generally does not exceed the dimension of the data (large p small n).

The dual formulation seeks to solve the following optimization problem:

$$argmax_{\alpha}\left\{\sum_{i=1}^{n}\alpha_i+\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j c_i c_j x_i^T x_j\right\}, \tag{1.18}$$

with the constraints of $\alpha_i \geq 0$ and $\sum_{i=1}^{n}\alpha_i c_i = 0$ for $i = 1,2,\ldots,n$. Again, the problem can be solved using quadratic programming.

It is important to note that the above formulation is for two separable classes. The practice of *soft thresholding* has been used to allow for mislabelled data points (points that lie in the margin or on the incorrect side of the margin). To accomplish this goal, the constraints in the primal problem are replaced by $c_i(w^T x_i - b) \geq 1 - \xi_i$ and the new optimization problem is

$$argmin_{w}\left\{\frac{1}{2}||w||^2 + C\sum_{i=0}^{n}\xi_i\right\}. \tag{1.19}$$

The ξ_is are referred to as slack variables and measure the distance the misclassified data point is from the margin. As before, this function is minimized to find the optimal separating hyperplane because the goal is to create a smooth boundary that minimizes the number of misclassified data points as well as making sure misclassified points are near the boundary (small ξs) to aid generalization.

While the introduction of slack variables accounts for non-separable classes, the formulation (as it stands) only deals with the two-class problem. There have been new methods designed to accomplish multi-class SVM involving a vector class label. Lee et al. (2004)

developed multi-class SVM defining the class label vector **y** for each point in the training set as follows:

$$\mathbf{y}_i = \begin{pmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{iC} \end{pmatrix}$$

$$y_{ij} = \begin{cases} 1 & \text{if } i^{th} \text{ point is in class j} \\ -\frac{1}{k-1} & \text{otherwise.} \end{cases}$$

This formulation extends the SVM paradigm to a multi-class problem, but it involves the estimation of many more parameters (as each training point yields a multivariate response). Standard SVM requires the estimation of n parameters, whereas MSVM requires $n*C$ parameters as a separate function is estimated for each class. It is similar to the previously used one-vs-the rest approach but by estimating all class similarities simultaneously, contradictions can be avoided. The MSVM technique has shown great benefits in the classification realm, but the issues of small sample size and large number of classes in most face recognition databases are problematic in the MSVM implementation. Also, face images of different individuals have different feature based centroids whereas outlier images overlap among different individuals. Of course, overlapping images from different subjects increases classification error in LDA as well; however, the centroid-based approach of LDA techniques is conjectured to be more robust than the boundary based approach of SVM as the centroids are more stable than the edges of the point clouds.

k-Nearest Neighbor

Another classification method that is commonly used is the k-nearest neighbor (k-NN) method (see, e.g., Hastie et al. (2001)). This is a non-parametric method that requires no real assumptions. This framework takes in a set of labelled training data and an unlabelled test point. The distance of the test point to all training points is calculated and the closest k

training samples are selected. Within the set of k nearest neighbors, the class that is most represented is chosen for the class label of the test point; i.e., each training point votes in this neighborhood and a plurality wins.

This method is quite robust as it does not have any restrictive assumptions, however, a high level of computational effort is needed for computing the distance from the test point to every training point. It also has the disadvantage of tending to label the test point as the class with more observations in the training dataset because they are more likely to be a *nearest neighbor*. One solution to this problem is to weight the votes accounting for the distance between the test point and the neighbor as well as the proportion of the class labels in the training data. Often the weights are inversely proportional to the distance from the training point as well as the proportion of a given class in the training data. This increases the weights of points very close to the test point and downweights the importance of the training points of classes more prevalent in the training data to account for the higher likelihood of occurrence.

Large values of k tend to yield boundaries that discount noise but produce boundaries that are generally too smooth, whereas small values of k yield classification boundaries that overfit and model the noise more so than the signal. Cross-validation is commonly used to find appropriate values of k for a given dataset.

Kernel Methods

All of the above methods involved searching for optimal separation in the data dimension or lower. Kernel methods seek to project the data (indirectly) into higher dimensional spaces that may induce further class separation (see, i.e., Hastie et al. (2001)). Popular kernels that often perform very well in many general cases are the polynomial kernel and Gaussian kernel. The motivation of the kernel methods for classification is generally to

increase the dimension (feature space) of the data by including other potentially nonlinear functions of the predictors and then use linear projections. It is possible that some nonlinear features introduce better separation in the classes. Once the data has been *projected up* into a larger (sometimes infinite dimensional in the Gaussian kernel case) feature space, the methods described above (LDA, kNN, SVM, etc.) can be used to construct a separating boundary that is useful for classification or simply provide a black box for classifying test points.

Determining which nonlinear functions to use to project the data into an informative space can be a tedious process that yields questionable results as there is no good way to find features that produce interesting directions. Also, there are infinitely many transformations that can be applied, so ascertaining which transformations are likely to yield useful results is difficult. The "kernel trick" allows for distance measures to be computed in a higher dimensional space without actually specifying which transformations are used. It is well known that the distances between points in the higher dimension is all that is needed to construct the classifier as the distance provides a similarity measure from which to determine class membership. By using the kernel trick to garner some similarity measure, one can examine the relationships in a much higher dimension without finding interesting projections to be used to separate classes.

While kernel selection is another added layer of complexity to the classification process, the only assumption that is used is that the chosen kernel indeed induces a greater level of class separation than just the collected features. More importantly, the kernel induces *linear separability* of the classes in the vector space spanned by the kernel basis function whereas a nonlinear boundary fits the raw data. Linear separability of classes is not guaranteed in general for any kernel so the kernel must be chosen to fit the specific problem. In fact,

LDA corresponds to a kernel method using a linear kernel. Kernel methods, as referred to here, are more of a feature enhancing method rather than classification method, but using the extra features in the general classification problem can yield much better classification results.

Decision Trees and Random Forests

The decision tree approach (the simplest type is known as CART) to classification focuses on classifying observations by sorting based on individual feature values and determining a cut point that induces classification (Breiman, 1984). The approach is referred to as a tree as an observation moves down a tree diagram and at each node (feature of interest), the observation is split along a specific branch. These branches are then split again and again until unique classification is achieved at the terminal nodes of the tree. The features chosen for the nodes are selected as those that best divide the training data by minimizing some loss function such as misclassification error, Gini index, etc. The tree classifier is quite simple as each decision node corresponds to a simple split in one dimension, but the optimal dimensions to choose for decision nodes must be learned.

As trees tend to overfit the training data, an important step in fitting the tree classifier is "pruning" the tree. This involves collapsing a fully grown tree under internal nodes that do not improve classification via some cost function. The growing and pruning steps are generally terminated by applying a stopping criterion at which the gain in continuing is not significant. Pruning seeks to remove unnecessary nodes as a tree with fewer nodes that achieves comparable classification performance is desired. Tree growing and pruning is an art since different cost functions applied can lead to very different decision trees. Learning the decision rules in a tree can be a time comsuming process, but the simplistic nature of

the classifier can be approached algorithmically. Furthermore, since final classification is based on many simple classifiers, computational time is generally not an issue.

Breiman (2001) describes the concept of a random forest in which many decision trees are formed on different bootstrapped training sets chosen at random from the actual training dataset. Each tree is individually formed as normal on the given training set but not pruned (as that would introduce much more computational effort). A common technique in the general random forest procedure is to randomly select a set of predictors on which to perform the node split. This random feature selection seemed to produce the smallest generalization error. After the ensemble of trees is formed (random forest), the test point is individually classified by each tree. The class receiving the most 'votes' is chosen to be the predicted class. The random forest concept has been shown to be quite simple and yet very effective for many classification problems. The large dimensionality and spatial relationships in image data may make this technique a bit poor for image classification as the storage requirements quickly increase in large dimensional datasets (as they do for any bootstrapping method). Also, the lack of a large number of training points per class in the face recognition problem is a cause for concern from the generalizability view of this procedure.

Naive Bayes classifier

Naive Bayes classifiers (see, i.e., Murphy (2006)) are simplistic Bayesian networks that are based on the assumption of independence of the predictor variables. This is, of course, not of much use in a face recognition framework as the features are highly correlated, but the naive Bayesian framework is a well-known method in classification that deserves mention. The classifier essentially solves for a posterior probability of each class given the data. With the assumption of independence of predictors, the calculations are

quite simple (simple product of the class likelihoods for each predictor – sum of the log likelihoods). This assumption of independence is often violated in practice, so naive Bayes classifiers are often less accurate than more sophisticated techniques. However with certain datasets, even with strong dependencies among predictors, the naive Bayes classifier performs comparably or even more favorably in terms of classification rate. Due to the simplicity of the algorithm, the computational cost is small. Semi-naive Bayes classifiers relax the independence condition slightly by only requiring conditional independence of predictors if the points are in different classes.

Bickel and Levina (2004) showed that in some cases (especially situations in which the dimension p is much larger than the sample size n, naive Bayes techniques can outperform techniques focused on the estimating covariance matrices. This is because covariance estimation in a large dimensional framework is not very stable for a small number of datapoints. We will use a regularized estimate to overcome the small sample size problem, but Bickel and Levina showed the naive Bayes technique worked quite well in a two-class problem of large dimension.

Bickel and Levina (2008) examined other methods of estimating the covariance matrix via banding (either the matrix itself or its inverse). While regularization introduces some inherent bias into the problem by artificially shrinking the estimate to make the matrix invertible, banding only focuses on estimating covariances near the diagonal and assumes zero covariance outside the band. With fewer parameters to learn, the small sample size problem can be controlled although this technique imposes an additional assumption of no covariance between certain features. This can be seen as a more relaxed naive Bayes technique.

1.2 Face Recognition Problem

The face recognition problem as well as other biometric techniques have been some of the hottest topics of interest within classification literature. With the reemergence of fighting terrorism as one of the highest priorities of United States law enforcement agencies following the 9/11 attacks, face recognition has taken on a more prominent role in the classification research community. Also, with high-tech security and the importance of company secrets certainly taking center stage in the corporate world, biometrics (of which face recognition is a part) have also burst onto the scene. Within the law enforcement community, the emphasis is on fast, accurate identification with less than optimal pose, lighting etc., whereas the biometric focus is on near-perfect classification in a controlled environment. These two areas look to solve the problem of finding the optimal classification under different sets of objectives. Mitra et al. (2006), Jain et al. (2006), and Hong and Jain (1998) discuss general biometric techniques in fingerprint and face recognition. The remainder of this section will focus on methodologies specifically geared to the face recognition problem under differing conditions.

For the purpose of this research, the face recognition problem is broken down into four distinct steps (this breakdown is motivated from a similar flowchart in Queirolo et al. (2010)):

1. Image Acquisition (camera)

2. Localization and Alignment (find the face and align anchor points)

3. Dimension Reduction (eliminate information that detracts from classification)

4. Classification (Assign class labels to training set based on test set).

Perhaps the simplest source of variation in face recognition is the localization problem (Step 2). Any template-based face recognition approach relies on images being localized, aligned, and standardized. The localization of the face in an image is very important when using a template-based approach as the correlation structure in the pixels of the image is completely dependent on all faces lying in the same area of the image. Many face localization techniques are discussed in the literature (Lowe, 1999; Sivic et al., 2005; Sznitman and Jedynak, 2010; Tu and Lien, 2010; Pamplona et al., 2010; Huang et al., 2011). The task of localization itself can be separately studied in the face recognition realm as localization is more of an engineering preprocessing issue. The most common "normalization" of spatial variation is to detect various key points around the face (eyes, nose, chin, ears) and transform all faces using an affine transformation to align those feature points. The most common features involve fixing the height of the eyes, the distance between eyes, the chin and the outer dimensions of the face. Many simple algorithms exist to align the faces to a specified grid and will be assumed for the remainder of this section. Also, it is common to normalize intensity as well as to focus the discriminator on features of the faces rather than photographic imaging characteristics.

Huang et al. (2007) developed a database of "faces in the wild" to begin the more complex face recognition task of locating, aligning, and standardizing facial images as they are found in real-world applications. This is taking the science of face recognition to a new level of complexity through the introduction of more intrapersonal variation due to background, angle, clothing, etc. Edge detection, key point anchors (inside of the eyes, nose midline, ear height, philtrum location, etc.), and warping techniques are applied to the raw images to align them for classification.

Our research will assume localized, aligned images and is focused on the statistical considerations of discrimination rather than the engineering concerns of automatic detection of features and alignment. While the acquisition and alignment of images are difficult problems (steps 1 and 2), especially when dealing with the newest challenge of classifying 'faces in the wild' (Huang et al., 2007), this is primarily an engineering problem. The tasks of finding the best lower dimensional representation of the faces for classification is the goal of this research. Our research is completely focused on steps 3 and 4.

1.2.1 What Aspects are Central to Human Cognition of Faces?

Before delving into the face recognition literature, it is important to note the aspects of the problem that make it unique in the classification realm. Bruce and Young (1986) provide an excellent overview of the background of the face recognition problem and what separates it from other classification problems. They also discuss many of the elements of human facial recognition prevalent in other fields such as psychology and behaviorology. Human face recognition stems from a variety of pointers in the human brain that are analyzed in depth by Bruce and Young. The primary problem when attempting to automate the face recognition system in the brain is that the images of faces show much more than extrapersonal variation (between class). In fact, there are many cases in which the intrapersonal variation (within class) actually exceeds the extrapersonal. This is due to many exterior factors (such as background, lighting, angle/pose) as well as intrinsic factors (such as expression, accessories, occlusion, etc.). These create a variety of problems for the face recognition problem in any classification framework. This is precisely what makes the face recognition problem unoptimizable in the general case, because the many potential sources of variation are simply too many to account for in a single algorithm. The

biometric situation (using a face to confirm identity) is a much more reasonable goal as the image collection can occur in a controlled setting (fixed lighting, pose, expression, etc.). The following sections speak to methods that attempt to control some of the many possible sources of variation and find optimal directions for class separation under the most general conditions.

Brunelli and Poggio (1993) provide an excellent analysis of what they term *feature* vs. *template* matching. This seminal paper divides the face recognition methods into a dichotomous classification framework that speaks to the reasoning for the differing nomenclature of feature selection and dimension reduction. Feature matching involves discovering features in the face such as relative position of eyes, nose, mouth, etc. and matching those features to a training set. Closest matches within the set of features yield the label. Template matching is more akin to the "dimension reduction" phraseology as it attempts to match the image as a whole vector of data values to a corresponding training vector. Of course, dimension reduction is performed using eigenanalysis or LDA, but features are chosen in a less concrete manner. The results suggest that the template matching scheme is often the easier of the two as the user need not define the features a priori and the computations are generally simpler (automatically locating specific features in a face image is often quite difficult).

1.2.2 Dimension Reduction for Faces

Eigenfaces, Eigenphases, Laplacianfaces, and Fisherfaces

The first general approach to dimension reduction for faces in the literature comes from Sirovich and Kirby (1987), where they begin to scratch the surface on using eigen-projection techniques in both the spatial and frequency domain for use in face recognition.

Turk and Pentland (1991) discuss the most simple template matching scheme termed eigenfaces. The eigenfaces are the eigenvectors of the covariance matrix of the rasterized face images. These eigenfaces represent directions learned from the training set to which all test points can be projected and distance measures can be calculated to points in the training dataset. Eigenfaces follow the PCA paradigm discussed in Section 1.1.3 and are, as such, an unsupervised algorithm, yet produce good classification results in practice. Moghaddam et al. (1998) discuss an eigenspace method in which the final classification is determined by a probabilistic matching to the training points rather than a distance metric approach.

Savvides et al. (2004) discussed the comparisons of eigenfaces and *eigenphases*. In the eigenphase paradigm, the images are transformed into the frequency domain via a Fourier transform and the transformed data is spectrally decomposed in the same manner as the original images. The optimal projection is realized in wave space rather than point space. Results tended to be promising under variations in lighting and less so under occlusion but not particularly impressive under other sources of variation. Chien and Wu (2002) also examined the wavelet transform and how the transformed data effects various classification rules such as nearest neighbor, nearest feature line (Li and Lu, 1999), nearest feature space, and nearest feature plane. Etemad and Chellappa (1997) investigated the effect of LDA in the wavelet domain as well as the spatial domain and found promising results in each.

Similar to the eigenface approach, there are many other template matching schemes, the most prevalent of which is Fisherface. As eigenfaces parallel the PCA technique, Fisherfaces parallel the Fisher LDA approach discussed in Section 1.1.3. Belhumeur et al. (1997) performed a comprehensive comparison of the techniques under varying expression and lighting and find that Fisherfaces tend to outperform the eigenface techniques (not surprising due to the supervised nature of the Fisherface algorithm). Fisherfaces also

tend to perform better over variation in lighting and expression. They also discuss a common technique in the eigenface image analysis which involves dropping the three largest principal components as these directions are often believed to measure background and lighting variation without examining between-class variation. Often this reduced eigenface approach outperformed general eigenfaces, but still fell short of the Fisherface standard. Belhumeur et al. (1997) finished the analysis by finding the technique that had the highest classification rates when glasses were placed on the subjects for the test images (no glasses incorporated in the training set). These results clearly favor the Fisherface method, as the eigenface methodology showed an error rate of greater than 50%. While most published results suggest the superiority of the LDA approach to the PCA approach, Beveridge et al. (2001) exhibit an example for which PCA performs uniformly better than LDA. They do not discuss why this is the case, but comment on the surprising nature of these results.

Laplacianfaces were explored in He et al. (2005) and Cai et al. (2006) in which the Laplacian technique was used in face recognition to find the optimal projection direction in which the local structure of the data is preserved as in Section 1.1.3. LDA and PCA can also be couched into the Laplacian framework as specific cases as shown in He et al. (2005). The Laplacianface eigenmap showed improvement over the standard LDA and PCA methods and was more robust. Niu et al. (2008) used a two-dimensional Laplacianface approach to utilize the spatial dependence of the figure. This suggests that the Laplacian methods are very useful for the face recognition problem. In fact, our research is directly linked to a Laplacian approach that is more tailored to class separation in the projected dimension.

Fisherface Extensions

The Fisherface approach has been extended in many ways in the face recognition literature along the lines of LDA extensions. Cevikalp et al. (2005) present an interesting take

on the direct-LDA framework for face recognition known as common vectors. The essential component of the algorithm includes finding the null space of S_W and first projecting a single point from each class into it. The vectors spanning the null space of S_W are known as the common vectors. There is significant computational savings by only projecting a single point for each class as all points in a common class project to a singularity as the points lie in the null space of S_W (note: this of course assumes common within covariance matrices among classes and deviations from this assumption will significantly degrade performance). Once the classes have been identified in the null space of S_W by the common vectors, the class between scatter is maximized ($J(W_{opt}) = argmax_W |W^T S_{com} W|$). These vectors can be identified using eigenanalysis on the matrix of common vectors (S_{com}).

Zhou et al. (2006) discussed a method called Improved LDA (I-LDA). This method is similar to the weighted-LDA approach discussed in Section 1.1.3, but Zhou et. al. modified the approach to include local information as well as global information. The local nature of the algorithm includes features focusing on landmarks; e.g., eyes, nose, and mouth specifically and included this information in the holistic face image. This combination of global and local information is quite robust to moderate changes in illumination, pose, and facial expression. Rather than PCA, Zhou's approach for I-LDA uses a discrete cosine transform (DCT) for dimensionality reduction rather than PCA which allows for increased computational efficiency. Price and Gee (2005) provided a similar approach that combines the weighted penalty in the between scatter matrix with a direct-LDA variation that projects the data into the column space of S_B. They also included a modular subspace approach that combines three classifiers, one of the whole face, a second focused on the region containing the eyes and nose, and a third containing the eye region only. This provides an

approach that is generally more robust to different facial expressions, but localizing the modular views is more complex under differing poses.

Lu et al. (2003b) were the first to propose a combination of the direct-LDA and fractional-LDA approach for the face recognition problem and termed their technique DF-LDA. The approach is very sensible as the current F-LDA paradigm is not geared toward finding low-dimensional representation of very high-dimensional spaces due to the extreme computational complexity of the algorithm. DF-LDA was developed to combat the small sample size problem (projection into the column space of S_B which is of maximal dimension $(C-1)$) which allows for a lower dimensional framework in which to start to which the F-LDA framework can be applied. Even though the results were quite promising in the experiments performed, it suffers from the same issues of Direct LDA as proven by Gao and Davis (2006). This provides a method of applying fractional LDA techniques to high-dimensional data in real time, but there are no guarantees of reaching the best projected space for classification. The methods explored in our research seek to apply a variation of F-LDA methods to problems in the high-dimensional space without sacrificing information through the direct-LDA procedure.

Chen et al. (2004) developed a methodology that sought to make F-LDA applicable to the classification problem with one sample per class. Their approach was to partition the single training image into non-overlapping regions and use each of these regions as training datapoints. The testing images are partitioned similarly and classified using two different techniques. The first classifies the vector of subimage projections via nearest neighbor methods to the training data projected vector of subimages, while the second classifies each subimage separately and employs a voting scheme to determine the class label. Dai et al. (2007) examined extending the F-LDA framework into a kernel framework for face

recognition. This provides the flexibility of F-LDA as well as finding non-linear boundaries (linear boundaries in the kernel basis space may be nonlinear in the image space). Of course, computational cost is intense at high dimensions, especially when using the F-LDA framework so future methods need to be investigated to make this technique tractable in a large face database case. Our approach can also be applied to a kernel framework but simultaneous shrinkage of all dimensions leads to a tractable solution for high-dimensional problems.

Other Cognitive Recognition Techniques

Departing from the LDA realm, other facial recognition methods have been discussed in many different fields including psychological markers that are pertinent in human recognition (Mitra et al., 2007). One such measure is facial asymmetry. Liu et al. (2003) used facial asymmetry as measured by the difference of an image and its reflection across the vertical midline (D-face) and the measure which is formed by the cosine of the angle formed between edges in the original image and edges in the reflected image at a given point (S-face). Liu, et. al. found that such asymmetry is actually quite constant across varying poses for a given individual and that this biometric produced excellent results across various poses (both D-face, S-face, and a combination). Mitra et al. (2005) examined the facial asymmetry measures in a frequency domain.

Martínez (2002) discussed a novel approach to dealing with imprecisely localized, occluded, or expression variant faces in a small sample framework. The localization problem is attacked by attempting to model the subspace from which images of a given class come. The subspace is often a feature space in which the class localization error is modelled using Gaussian distributions. The primary mechanism employed to combat occlusion is breaking the image into a variety of subimages for use in classification similar to Chen et al. (2004).

If a region of the face is occluded, this region would clearly be useless for classification, yet the others could still be used. Rather than a voting scheme usually employed, a probabilistic scheme is developed that weights subimages invariant to pose more highly than those that are highly variable under differing poses.

Tensor methods have also arisen in the face recognition framework. Vasilescu and Terzopoulos (2002) approached the idea of "tensorfaces" by constructing high-dimensional tensors of faces that contain different types of variation along different dimensions – one each for illumination, expression, pose/angle, individual, etc. Once an enormous tensor of training images is compiled, N-mode SVD (a multilinear SVD) is applied in which the extremely large dimensional tensor is projected into a more manageable size that parsimoniously explains each of the constituent factors (illumination, expression, pose, etc.). Obviously this incurs an incredibly large cost in constructing a training data set (images of each person in each illumination, pose, expression variation). Perhaps there are extensions in the tensor methodology that can find useful projection directions under a tensor model that is not completely saturated (i.e., not every pose, illumination, etc. available for every individual).

Other approaches have been discussed involving deformation measures of the face (Leroy et al., 1996) using certain measures of the eyes, nose, and mouth on a standardized image and the effects various expressions have on certain individuals. These deformation measures often involve gradient calculations in the image space and seek to find individual specific characteristics in these identified locales. Similarly, more advanced approaches have been taken to match a 2.5D face image to a hypothetical 3D face model (Lu et al., 2006). Lu et.al. referred to an image (or more precisely a collection of images) as a 2.5D scan as using multiple angles of a common face to construct a 3D model. The 3D model is

then compared to training 3D models for classification. Clearly, much more computational complexity is involved in this methodology to construct the model of the face and store a variety of training models for classification purposes.

As the imaging technology continues to improve and become more widespread, more techniques are being developed to analyze 3D images. Queirolo et al. (2010) examined a subimage simulated annealing approach to classifying 3D images. While this technique is relatively new, the prevalence of more accessible cameras and computers for analysis has made this a reality. Queirolo focused on segmented matching of the whole face, circular nose region, long elliptical nose region, and upper head as the nose region is the least affected by changes in facial expression. A conglomeration of classifying the four regions separately yielded the most consistent results.

Another approach to the face recognition is examining it from locality preserving projections (Lu and Tan, 2010). Lu, et.al. attacked the small sample size (SSS) problem by stating that PCA prior to discriminant techniques can lead to a significant loss of local information that may be useful to classification. The parametric regularized locality preserving projection (PRLPP) is an extension of other LPP methods rampant in the literature of late (He et al., 2005; Yu et al., 2006; Zhu and Zhu, 2007). The LPP technique in general incorporates some affinity measures (via kernels) when solving the generalized eigenvalue problem. Most LPP techniques in the literature are inherently unsupervised methods that seek to find an underlying nonlinear manifold in which points close in the full space remain close in the reduced space.

1.2.3 Unifying Aspects of Face Recognition

The methods discussed in this chapter seem as if they have been applied in an isolated context. This could not be further from the truth since many of the aforementioned classification methods are often combined into hierarchical contexts. The most common of these hybrid/hierarchical classifiers is projecting the high-dimensional images to a more manageable size using PCA, then applying LDA to the already reduced data. Also, there are methods in which various classifiers are constructed separately and the conclusions are drawn from integrating the individual classifications into a single output (Lu et al., 2003c). Another common technique utilized to integrate many simple learners is boosting which was applied to the face recognition problem by Lu et al. (2003a). The boosting paradigm focuses on improving overall classification by weighting misclassified observations more heavily in future learners to provide a holistic integrated technique that can handle a wide range of test samples. Other hybrid techniques are discussed in the literature to combat specific areas of discrimination, such as gender classification (Xu et al., 2008).

Chapter 2: Simultaneous Adaptive Fractional Discriminant Analysis

2.1 Motivation and Notation

Many traditional dimension reduction techniques for classification focus on maximizing the separation of the classes in the large dimensional space. Unfortunately, using standard techniques neglect to examine how the projection process itself may actually degrade separation. It is known that Fisher's LDA is optimal only when the data is projected into $(C-1)$ dimensions and (potentially) suboptimal when the desired dimension is much less than $(C-1)$ (which is often the case in the face recognition paradigm when there are a multitude of potential individuals).

Fractional Linear Discriminant Analysis (F-LDA), Lotlikar and Kothari (2000), showed a great deal of promise in classification improvement over other linear techniques as it focuses on projections that maximize separation in the projected space rather than in the full space in that a dimension is 'fractionally' removed iteratively across a multitude of substeps. Therefore, the separation criterion is optimized in the lower dimensional space, which lead to better separation. Of course, this is a useful method only when the weighted between class scatter matrix is used, as the weighting function (kernel) yields changing 'optimal directions for separation' after downweighting the less informative dimensions. The fractional modification involves extra computational costs which are generally considered

to be more than the benefits in classification accuracy. Since F-LDA only downweights incrementally and finally removes the least informative dimension one at a time, it requires repeating the eigenvalue decomposition process many times for reduction of one dimension. This is clearly not feasible to use for image classification or other large-dimensional problems as the increased cost in computing time is further exacerbated by iterating this process for each dimension to be removed.

F-LDA has shown success in classification accuracy (Lotlikar and Kothari, 2000; Lu et al., 2003b), but the prohibitive computing time prevents exploration of its benefits in large-dimensional problems. Also, the sequential nature of the dimension removal loses interdependence between dimensions. In other words, removing k dimensions from a p dimensional space sequentially may be 'optimized' by some criterion (LDA) at each step but not optimized with regard to finding the best d dimensional space due to interrelatedness of the dimensions removed. This is similar in spirit to the step-down method in regression model selection.

The proposed algorithm seeks to merge the benefits of F-LDA (seeking optimal separation space in the projected space rather than the full space) with a computationally streamlined method. The new method, termed Simultaneous Adaptive Fractional Discriminant Analysis (SAFDA), requires multiple steps of $\tilde{S}_B$ (as defined in Equation (1.13)) eigenvalue decomposition to optimize class separation in the reduced space, but rather than shrinking (downweighting) only one dimension at a time (the least informative), it simultaneously shrinks all $(p-d)$ dimensions ultimately to be dropped by varying degrees.

Three main downweighting schemes listed below are examined in Sections 2.3.1 and 3.5:

1. Downweight all dropped dimensions by the same amount.

2. Downweight the dimensions by an amount based on the rank of the respective eigenvalue.

3. Downweight by an amount based on the relative size of its respective eigenvalue according to some function.

The goal is to perform the fractional technique across many dimensions *simultaneously* and remove a 'chunk' of dimensions at once in the last step. This allows the classification benefits of F-LDA with computational feasibility, while also introducing the removal of all dimensions at once which leads to a more global solution rather than one that is derived via sequential optimization.

2.1.1 Notation

M denotes the $p \times c$ matrix containing the *whitened* centroids of the data:

$$M_{p\times c} = (\mu_1, \mu_2, \ldots, \mu_c) \tag{2.1}$$

D denotes a $C \times \binom{C}{2}$ differencing matrix that, via matrix multiplication with M, creates a matrix containing all pairwise differences of means.

$$D_{C\times\binom{C}{2}} = \begin{bmatrix} 1 & 1 & \ldots & 0 \\ -1 & 0 & \ldots & 0 \\ 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \\ 0 & 0 & \ldots & -1 \end{bmatrix} \tag{2.2}$$

W denotes a diagonal matrix in which each diagonal element corresponds to the weight function applied to the given pairwise mean difference in the construction of the weighted between scatter matrix. This particular construction allows for any weighting function (kernel) to be used as long as it is a similarity measure.

$$W = \begin{bmatrix} w_{1,2} & 0 & \dots & 0 \\ 0 & w_{1,3} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_{(c-1),c} \end{bmatrix}, \tag{2.3}$$

where $w_{ij} = K(\mu_i, \mu_j)$

Using the notation above, the weighted between class scatter matrix as defined in Equation (1.13) can be rewritten as:

$$\tilde{S}_B = \sum\sum w_{ij}(\bar{x}_i - \bar{x}_j)(\bar{x}_i - \bar{x}_j)' = (MD)W(MD)'. \tag{2.4}$$

Since the downweighting process is repeated many times, the savings garnered in matrix computation over loops is amplified. Also, this matrix representation makes it much easier to understand both the F-LDA methodology and the SAFDA modification while streamlining the algorithm.

2.2 Why Weight?

The central motivation behind the SAFDA paradigm is the fact that the standard LDA paradigms are not made to be optimized in a sub-$(C-1)$ dimensional space. This suboptimality stems from the criterion used to elicit separation residing in the large space rather than the reduced space. While the criterion is indeed optimized in the *complete* space under LDA conditions, the process of projection to a lower-dimensional space can introduce confusion between classes that were simple to distinguish in the full $(C-1)$ space. Even under the weighted LDA framework, in which classes that are more difficult to distinguish drive the dimensions of importance, the focus is on separation in the full space. Classes that are very separated in the full space may yield little information to the weighted between

class scatter matrix used for feature selection. Therefore, they can become confused in the reduced space unless they are also separated in the dimensions than have greater confusion among other classes. It is also important to realize that these fractional techniques must take place in a weighted framework. If there is no weighting of the between scatter matrix based on class mean separation, the directions of optimal separations will never change as the dimensions of lesser importance will cause no reordering of the eigenvalues of the unweighted between scatter matrix. By adding a kernel-like weighting scheme, fractional removal of dimensions can yield rotations of the original LDA orientation derived in the full space that is more aligned to the weighted criterion in the reduced dimensional space. The weighting function and successive fractional removal of dimensions lead to separation tailored to the reduced space.

The selection of the optimal (or even appropriate) weighting function for a given problem is of paramount concern in practice. Since the best weighting function for a given problem is data dependent, there is no weighting function or kernel that is always the 'best'. Therefore, Section 3.3 analyzes the costs and benefits of three weighting functions, listed below, that are based on the Euclidean pairwise distance only:

1. Inverse Pairwise Weighting Function.

2. Bounded Inverse Pairwise Distance Weighting Function.

3. Gaussian Kernel.

While this is by no means a comprehensive list of weighting functions that may be used, these three exhibit a range of characteristics that can be extended to other weighting functions (unbounded, bounded, adaptive, etc.). Functions that are based on any criteria other than Euclidean distance are not used because the weighting function is not useful in the

construction of the weighted between class scatter matrix when other extraneous information is included (i.e., vector length). This extra information can give larger weight to the directions of lesser importance simply due to scaling.

The inverse pairwise distance weighting function was used in Lotlikar and Kothari (2000) and is shown here:

$$K(\bar{x}_{[i]},\bar{x}_{[j]}) = ||\bar{x}_{[i]} - \bar{x}_{[j]}||^{-h}, \tag{2.5}$$

where the parameter h is chosen via cross validation (so long as $h > 2$).

A bounded variant of the same weighting function is also examined to prevent one pair of classes from dominating the separation criterion.

$$K(\bar{x}_{[i]},\bar{x}_{[j]}) = \frac{1}{(1+||\bar{x}_{[i]} - \bar{x}_{[j]}||)^{h}}, \tag{2.6}$$

where h is again learned via cross validation.

The Gaussian kernel,

$$K(\bar{x}_{[i]},\bar{x}_{[j]}) = e^{-\frac{||\bar{x}_{[i]} - \bar{x}_{[j]}||^2}{h}}, \tag{2.7}$$

is, perhaps, the most attractive kernel as the bandwidth parameter h can be adapted to the down-weighted data at each step. When referring to simultaneous *adaptive* fractional discriminant analysis, this is the kernel of choice due to the ability of choosing the bandwidth online. This leads to a tremendous time savings by avoiding CV on an already computationally intensive procedure.

2.3 Description and Algorithm

The SAFDA method focuses on shrinking multiple dimensions (rather than only one dimension at a time in F-LDA) simultaneously and removing several of them at the end of the process. The benefits of this modification are twofold:

- Immense time savings as the computational cost to remove several dimensions is about the same order of magnitude as removing one dimension in F-LDA.
- Best dimensions in the reduced space are found while accounting for the interdependence that may exist between dimensions (no sequential/conditional optimization).

The SAFDA algorithm is described below using the previous notation.

Simultaneous Adaptive Fractional Discriminant Analysis

1. Initialization–Set parameters
   (a) Weighting function tuning parameters (may be determined via cross-validation or adaptive scheme)
   (b) Final dimension of projection (d)
   (c) Number of iterations before removal (r_{max}) and determine downweighting factor for the most important dimension to be dropped ($\alpha = r_{max}^{-\frac{1}{r_{max}-1}}$) which makes the geometric downweighting scheme mimic an arithmetic downweighting paradigm after r_{max} steps ($\alpha^{r_{max}-1} = \frac{1}{r_{max}}$)
   (d) Choose downweighting scheme
      - Common downweight
      - Downweight based on rank of eigenvalues
      - Downweight based on a function of magnitudes of eigenvalues
   (e) Scaling matrix, which has the following form:

$$S = \begin{bmatrix} I_d & 0 & 0 & \dots & 0 \\ 0 & f_{d+1}(\alpha) & 0 & \dots & 0 \\ 0 & 0 & f_{d+2}(\alpha) & \dots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & 0 & 0 & \dots & f_p(\alpha) \end{bmatrix}.$$

The functions within the scaling matrix are dependent on the downweighting scheme, but $f_{d+1}(\alpha) = \alpha$ for all scaling matrices.

2. Preprocessing Step; Algorithm 2 described below

Algorithm 2 Preprocessing

Require: $(\mathbf{X}_i, \mathbf{Y}_i)$ (where X is the data and Y is its class label)
Ensure: M whitened class means
$\tilde{X}_i \leftarrow X_i - \bar{X}_.$ {Mean Centering}
$S_T \leftarrow \tilde{X}'\tilde{X}$ {Total Scatter Matrix}
Remove the Null Space of S_T from the data
$\bar{x}_{[j]} = \frac{1}{n_j} \sum_{i:y_i=j} \tilde{x}_i$ {Compute Class means}
$S_{W[J]} \leftarrow \sum_{i:y_i=j} (\tilde{x}_i - \bar{\tilde{x}}_{[j]})(\tilde{x}_i - \bar{\tilde{x}}_{[j]})^T$
$\tilde{S}_W \leftarrow \varepsilon I + \sum_j \frac{n_j}{n} S_W[j]$ {Regularized Within Class Covariance Matrix}
$\tilde{\tilde{X}}_i \leftarrow \tilde{S}_W^{-\frac{1}{2}} \tilde{X}_i$ {Whiten the Data}
$M_j \leftarrow \sum_{i:y_i=j} \tilde{\tilde{X}}_{ij}$ {Whitened class means}
Remove the null space of M via eigendecomposition as it is not helpful in classification {M is at most $C \times (C-1)$}

---

3. Simultaneous Adaptive Fractional LDA; Algorithm 3 given below

4. Project whitened training/test points into reduced space using P

5. Classify using any classification algorithm (we use nearest mean in the projected space)

The key differences between SAFDA and F-LDA (Algorithm 1 in Chapter 1) involve the functions used to calculate distances in the weighted between class scatter matrix and the construction of the scaling matrix (S). In F-LDA, S has the much more simplified form

Algorithm 3 Simultaneous Adaptive Fractional LDA

Require: M whitened training means, r= max number of iterations
Ensure: P= SAFDA projection matrix
$U \leftarrow I_p$
$W_0 \leftarrow e_1$
$\alpha = r_{max}^{\frac{1}{r_{max}-1}}$
$B \leftarrow$ Stopping criterion boundary for Hellinger distance between successive normalized weights
for $r = 0$ to $r_{max} - 1$ **do**
$M_{(r)} = S^r U' M$
$h_r = min_{i,j}(||S^r U' M(i) - S^r U' M(r)||^2)$ for Gaussian kernel
$\hat{W}_{r+1}[i,j] \leftarrow K(S^r U' M[i], S^r U' M[j])$
$W_{r+1} =$ rasterized vector of $\hat{W}_{r+1}$
$\tilde{W}_{r+1}[i] \leftarrow \frac{W_{r+1}[i]}{||W_{r+1}||}$
$\tilde{S}_B = (S^r U' MD) diag(\tilde{W}_{r+1})(S^r U' MD)$ {Weighted Between Class Scatter Matrix}
$\tilde{S}_B = \Phi \Lambda \Phi'$ via eigendecomposition with λ_i's in decreasing order
$U \leftarrow U\Phi$
if $k \geq 1$ **then**
$H^{(r)} = \sqrt{2(1 - \sum_i \sqrt{\tilde{W}_{r-1}[i]\tilde{W}_r[i]})}$
if $H^{(r)} < B * H^{(1)}$ **then**
break
end if
end if
end for
return $P =$ first d columns of U

in which only the least informative dimension is downweighted:

$$S_{FLDA} = \begin{bmatrix} I_{p-1} & 0 \\ 0 & \alpha \end{bmatrix}.$$

By downweighting all dimensions to be removed *simultaneously*, the external loop, in Algorithm 1 that peels the dimensions off one-at-a-time, is not needed. Clearly, this modification is a *huge* computational time saver, as F-LDA has the same basic structure except the removal of one dimension at a time requires repeating step 3 in the above algorithm as many times as the number of dimensions to be removed $(p-d)$. Recall, each pass through the loop in step 3 above (loop that shrinks the least informative dimensions and finds the optimal rotation) has the same cost as a standard LDA algorithm (after whitening) with costs incurred through the weighted between matrix eigenvalue decomposition. SAFDA removes the need of peeling the dimensions off one at a time with expanded computational cost for each dimension. By construction, the SAFDA algorithm will take longer than standard LDA as the matrix decomposition is performed r_{max} times to optimize spacing in the projected space rather than the full space, but it prevents the expanded F-LDA cost of performing this time-intensive procedure repeatedly, once for each dimension to be reduced. The time savings alone is large when dealing with large C, large p problems (such as image classification) as matrix decompositions are very costly for large p. Since the size of the M matrix is dependent on the number of classes as seen in Algorithm 2, this modification is particularly useful in cases of large C.

Another big computational savings for large p problems from the original F-LDA algorithm is in the preprocessing step. We show that the whitened class means lie in at most a $(C-1)$ dimensional manifold. As there is no new information that can be garnered from

the null space of $\tilde{S}_B$ (as any element in the null space of $\tilde{S}_B$ projects all classes to a singularity and therefore no separation), there is no penalty for removing its entire null space of the *whitened* class means *before* the shrinking and optimization portion of the algorithm.

Proposition 1. *The removal of the null space of $\tilde{S}_B$ after whitening but before shrinking, incurs no loss in classification from running SAFDA on the full $\tilde{S}_B$ matrix.*

Proof. For the first iteration in the inner loop, l=0 so S=I.

$$\tilde{S}_B = M_0 D W D' M_0' = QQ' \text{ where } Q = M_0 D W_0^{\frac{1}{2}} \tag{2.8}$$

Using singular value decomposition of Q

$$Q = M_0 D W_0^{\frac{1}{2}} = U \Lambda^{\frac{1}{2}} V', \tag{2.9}$$

$$M_0 D = U \Lambda^{\frac{1}{2}} V' W_0^{-\frac{1}{2}} \tag{2.10}$$

For high-dimensional data, there are often zero eigenvalues (when the data space is not full rank). Therefore, the matrices take on a block structure as shown:

$$M_0 D = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1^{\frac{1}{2}} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}' W_0^{-\frac{1}{2}} \tag{2.11}$$

The projection matrix is U and the data (means) are projected into the U space before the shrinkage step of the inner loop in F-LDA. Let

$$M_1 = U' M_0$$

The differenced means have a simplified structure by substituting from equation (2.10).

$$
\begin{aligned}
M_1 D &= U' M_0 D \\
&= U' U \Lambda^{\frac{1}{2}} V' W_0^{\frac{1}{2}} \\
&= \Lambda^{\frac{1}{2}} V' W_0^{\frac{1}{2}} \\
M_1 D &= \begin{bmatrix} \Lambda_1^{\frac{1}{2}} V_1' \\ 0 \end{bmatrix} W_0^{\frac{1}{2}}
\end{aligned} \tag{2.12}
$$

When scaling is applied to the new data, there is no change as the scaling only affects the last (or in the SAFDA case, last several) eigenvalue(s) which are all zero. Clearly, scaling yields no effect until the dimensions in the column space of the data are impacted via scaling, as shown below.

$$
\begin{aligned}
SM_1 D &= SU' M_0 D \\
&= \begin{bmatrix} \mathbf{I}_{p-1} & 0 \\ 0 & \alpha^l \end{bmatrix} \begin{bmatrix} \Lambda_1^{\frac{1}{2}} V_1' \\ 0 \end{bmatrix} W_0^{\frac{1}{2}} \\
&= \begin{bmatrix} \Lambda_1^{\frac{1}{2}} V_1' \\ 0 \end{bmatrix} W_0^{\frac{1}{2}} \\
&= U' M_0 D \\
SM_1 D &= M_1 D
\end{aligned} \tag{2.13}
$$

Because this process can be repeated for all of the inner shrinkage steps in F-LDA for each dimension in the null space of $\tilde{S}_B$, this result holds until the scaling impacts a dimension with non-zero eigenvalue. The same holds true for SAFDA when directions of zero eigenvalue are shrunken.

□

Removing the (potentially very large) null space of $\tilde{S}_B$ *before* the iterative downweighting steps is a very large savings for large p problems as C is usually much smaller than p

for most problems, and the matrix decomposition is much faster. Facial recognition is an example of a problem with a large number of classes for which this modification of projecting the data into the space spanned by the class means is not enough, because the class means span the potentially large C-dimensional space, in which the computational requirements are still too large in practice. Since C is large and calculations in the large space ($(C-1)$ dimensions) are costly, SAFDA ability to remove all dimensions simultaneously rather than one-at-a-time like F-LDA is a significant boost to the speed of the process.

2.3.1 Downweighting Schemes

One of the primary elements in the algorithm is the downweighting scheme. We do not believe that the final direction of projection would be greatly affected by the scheme chosen, but perhaps convergence to the best space may be achieved more quickly using more information (size and/or magnitude of the eigenvalues) in determining how far to downweight the data. This section will examine three schemes:

- Scheme 1: Common downweight for all dropped dimensions.
- Scheme 2: Downweights based on the rank of the associated eigenvalue.
- Scheme 3: Downweights based on the relative size of the associated eigenvalue.

The SAFDA algorithm is designed to elicit optimal separability in the lower (d) dimensional space under any downweighting scheme, but it is hypothesized that using more adaptive informative downweighting schemes may achieve convergence more quickly. This is desirable in order to minimize the computational effort.

Scheme 1: Common α

The simplest scheme involves a common downweighting factor for all dimensions. The result of this scheme is to slowly reduce the importance of each dimension to be discarded equally to examine the separability in the pseudo-lower dimensional space. The scaling matrix for this scheme is a block diagonal matrix with unity for the retained dimensions and the same downweighting factor α for the dropped dimensions.

$$f_i(\alpha) = \alpha \qquad i = d+1, d+2, \ldots, p \tag{2.14}$$

Scheme 2: α based on eigenvalue rank

When using the order of the eigenvalues only, a recursive scheme was developed to define the shrinkage matrix as follows.

$$\begin{aligned} f_{d+1}(\alpha) &= \alpha \\ f_i(\alpha) &= \frac{\frac{1}{2} + f_{i-1}(\alpha)}{2}, \quad i = d+2, d+3, \ldots, p. \end{aligned} \tag{2.15}$$

This recursive scheme was chosen as a geometric shrinking toward $\frac{1}{2}$. The least amount of shrinkage was chosen to be the same α as in F-LDA to ensure the most informative discarded dimension is shrunken to $\frac{1}{r_{max}}$ after r_{max} iterations (prior to dropping). The others are downweighted more heavily as they are less important (relatively). This method uses a geometric progression of weights that begin at α as chosen by the number of shrinkage steps (*a priori*) using a factor of $\frac{1}{2}$. An asymptote of $\frac{1}{2}$ was chosen (arbitrarily) so that no dimension was removed too quickly in the shrinkage steps. This guarantees that a single dimension (no matter its import) cannot be shrunken by more than half in a single step. This asymptote was put in place to ensure that a dimension that has little import in the full space is not removed immediately as it may have some discriminative powers in the projected

space. Standard LDA techniques (LDA, wLDA, etc.) remove these dimensions without regard to their import in the projected space. This scheme ensures that this downweighting method improves separation in the reduced space over standard LDA, but it is hypothesized that scheme 2 may reach a more stable solution in fewer iterations than scheme 1.

Scheme 3: α based on relative magnitude of eigenvalues

Another downweighting scheme that was examined involves the magnitude of the eigenvalues. Ideally, this technique downweights dimensions with similar eigenvalues almost identically whereas the scheme 2 places greater importance on the order regardless of how close they are. This technique was expected to provide a better downweighting scheme (quicker convergence to a stable solution) as more information is used in determining how important a given dimension is for classification. This scheme involves weighting the dimensions based on the ratio of the associated eigenvalue to all those dropped. While the ratio drives the amount of shrinkage, a lower bound on the shrinkage is desired to prevent a dimension with little impact from being almost completely removed in one iteration.

The scheme we used in which the eigenvalue magnitude is used in the downweighting calculations is given below in Equation (2.16).

$$f_i(\alpha) = (c + (1-c)\frac{\lambda_i}{\sum_{k=d+1}^{p}\lambda_k})\alpha, \quad c \in (0,1) \quad i = d+1, d+2, \ldots, p \tag{2.16}$$

The above scheme allows for two important facets not available in the rank based downweighting scheme (Scheme 2) described by Equation (2.15).

1. The parameter c can be chosen by the experimenter as the asymptote to which the shrinkage factor decreases. In fact, the function is constrained to be between $c\alpha$ and α.

2. The scaling (shrinking) function is *updated* within each iteration.

The shrinkage factors used in schemes 1 and 2 are constant given the number of dimensions to remove, but this technique allows dimensions that have similar eigenvalues to be scaled by a like weight. However, in scheme 3, the downweighting factors are not directly related from iteration to iteration; i.e., the amount of shrinkage in this scheme is not necessarily monotonic. This can cause one dimension to be downweighted more in an earlier iteration as the scaling matrix is raised to a power based on the iteration. To allow for these changing scaling matrices, the previous scaling matrices will be retained and multiplied together to produce the next iteration's scaling matrix.

An issue to be addressed in schemes 2 and 3 is that the final iteration has the dropped dimensions weighted differently (even though they have all approached near zero after r_{max} iterations). As all $p-d$ dimensions will be dropped after r iterations, it is hoped that all dropped dimensions have been, eventually, reduced to more or less equally poor status in discrimination. One final modification to the shrinkage scheme would be to base the shrinkage factors on the *differences* of successive eigenvalues which may alleviate this effect of varying scales as we approach the terminal shrinkage step, but we are not examining this approach in this work.

2.4 Connections between SAFDA and Projection Pursuit

Simultaneous adaptive fractional discriminant analysis methodology is focused on finding low-dimensional spaces that yield close to optimal separation of the classes *in the reduced dimensional space*. This focus on separation in the projected space differs from most standard LDA practices in which informative directions are sought out in the full data space without regard for the effect of projection on class separation into said space. Fisher's

LDA is only known to be optimal in $(C-1)$ dimensional space (under homoscedasticity and Gaussian assumptions) but when the goal is to project to a lower dimensional subspace, the optimal solution need not match up with the most informative few directions in LDA. While there is indeed an ordering of importance among the directions discovered in LDA, the entire $(C-1)$ space is needed to ensure optimality.

Projection pursuit was initially proposed as a tool for clustering by seeking projections in which an objective function involving local clustering behavior and overall spread was optimized. The projection pursuit paradigm is essentially an optimization problem using a simulated annealing algorithm to find the directions that optimize the given function (see Section 1.1.3 for details). Laud et al. (1992) proposed a probing algorithm to improve standard simulated annealing. In this process, rather than generating one random perturbation in search of an improvement, many randomly perturbed points are selected under some prior distribution centering at the current point. Each of the randomly chosen points are evaluated and the step is taken in the most optimal direction of all the probe points. The procedure allows for fewer stationary steps as the optimal probe point from among many is more likely to provide improvement over the current point (unless the optimum has been reached). Using a probing procedure in this fashion would be computationally more expensive by generating and evaluating more test points, but the optimum would most likely be found in fewer steps.

Recently, Lee et al. (2005) developed more directed objective functions for supervised learning in which the objective function involves the class labels in some sense. While Lee uses a Fisher-type criterion for class separation, the search algorithm used to find optimal directions still relies on simulated annealing. This procedure has a (somewhat) undetermined stopping criterion as the search for optimal directions is random and refines itself

based on the updated directions as the optima is approached. Another caveat in using this LDA criteria is that the within class covariance matrix is often singular (large p small n cases) which causes intractability of Lee's optimization function. Lee and Cook (2010) addressed issues by introducing a regularized measure for calculating the within or total scatter matrices in the objective function. This is a simplistic fix to deal with singular matrices, but a small regularization is considered acceptable under the small sample size estimate of the covariance matrices.

The current literature about projection pursuit is focused on developing new objective functions that determine 'interesting' directions with little focus on finding more useful methods of optimizing the functions. SAFDA can be likened to a *directed* search under the weighted between scatter matrix criterion. The shrinking steps are akin to examining the data in a lower dimensional space but rather than examining the class separation in $d << p$ space and optimizing using standard simulated annealing techniques, the dimensions are reduced *fractionally* (i.e., via shrinking) and the optimal dimensions are *updated* based on the separation of the class means (under the functional criteria) in that space. A major gain in efficiency is that a random search is avoided (along with the multitude of steps yielding no improvement and somewhat nebulous stopping criterion). Viewing SAFDA in a directed projection pursuit framework allows for:

- A search similar to a gradient search in that a random probe need not be generated, but an updated rotation of the basis vectors improves upon the weighted class separation based on the shrunken space which is a pseudo-approximation of the final low-dimensional space (at every step).

- A predetermined stopping criterion in which at most a fixed number of steps leads to termination of the algorithm rather than forcing pure convergence.

This connection has not been examined as yet because simulated annealing techniques have become so streamlined that the objective function is generally the focus of improving projection pursuit techniques. The SAFDA paradigm provides the researcher more control concerning the time at which the optimum is achieved (the algorithm is halted after r_{max} iterations of shrinking and rotating). Of course, a larger r_{max} leads to a longer calculation time but also a more exhaustive search. The size of r_{max} can be likened to the inverse step size in the annealing process. While it is not exactly parallel in nature, the larger the r_{max}, the smaller the amount of difference in shrinkage which leads to more precise updates whereas a small r_{max} might 'jump' across the global optimum in favor of a local optimum.

The directed nature of the algorithm also removes the need for the accept/reject nature of the annealing algorithm in which the current point is updated to a random perturbation only if the objective function is larger when evaluated at the random probe point. SAFDA always updates the current space of interest via rotation to one that optimizes the separation criterion for the 'fractional' data space (dimensions to be dropped have been fractionally lessened in importance). Each probe direction in this case is optimizing the weighted LDA objective function on the shrunken data which mirrors the effect of removing those dimensions entirely.

2.4.1 Simple Example

A simple example of three normally distributed classes illustrating the power of SAFDA in three-dimensional space centered at $\mu_1 = (0, 1, -50.5)$, $\mu_2 = (0, -1, -49.5)$, and $\mu_3 = (0, 0, 100)$ and a common class covariance matrix equal to the $\mathbf{I}$ is examined to show the pitfalls of LDA methodology in the full space. Optimizing the standard LDA criterion with a one-dimensional projection would focus on the third dimension only due to the

outlier class 3. While the third dimension clearly separates class 3 from the other two, that single dimension is almost useless in separating classes 1 and 2. The SAFDA method employs a directed projection pursuit approach in which the classes are shrunken along the dimensions of lesser importance (by larger and larger degrees) and those dimensions of least importance are updated as the shrinking occurs via maximizing the wLDA criterion. The net outcome for this example would be that the third variable would remain the least important for many steps, until the shrinking causes classes 1 and 2 to reach near class 3, at which time the optimal space would be slightly rotated to allow for maximal separation.

This example was run under both the LDA and SAFDA paradigms. The training data set was generated with 500 independently normally distributed samples per class and the test set had 300 independently normally distributed samples per class. For a one-dimensional projection, LDA had a test set correct classification rate (CCR) of 79.89% whereas SAFDA improved the CCR (on the same training/test set) to 91.89%. The confusion matrices for the one-dimensional projections in optimal LDA and SAFDA are given in Tables 2.1 and 2.2, respectively.

	Predicted Class		
Truth	1	2	3
1	205	95	0
2	86	214	0
3	0	0	300

Table 2.1: LDA Confusion Matrix: Simulated Data

	Predicted Class		
Truth	1	2	3
1	260	40	0
2	33	267	0
3	0	0	300

Table 2.2: SAFDA Confusion Matrix: Simulated Data

We can see that class 3 is easily separated from the other classes as expected, because class 3 is an outlier class. Optimal LDA with one-dimensional projection focuses on the third dimension, which separates class 3 from classes 1 and 2, ignoring the second dimension which separates classes 1 and 2. This is evident in the LDA confusion matrix. The factor loadings for the optimal LDA are $[0.0063, -0.0016, 1.0000]$ which shows that LDA substantially downweights the first two dimensions and basically classifies on the third dimension. The slight loadings on the first and second dimensions and noise in the data led to classification rates better than guessing. For SAFDA, the loadings are $[0.0240, 0.9021, -0.4310]$. SAFDA actually puts most of the weight on the second dimension as class 3 is easily separated in the third dimension without heavy loadings. More discriminatory effort must be placed on the second dimension which serves to separate the most confused classes 1 and 2. Neither case places much effort on the first dimension as there is no difference between the classes in that dimension.

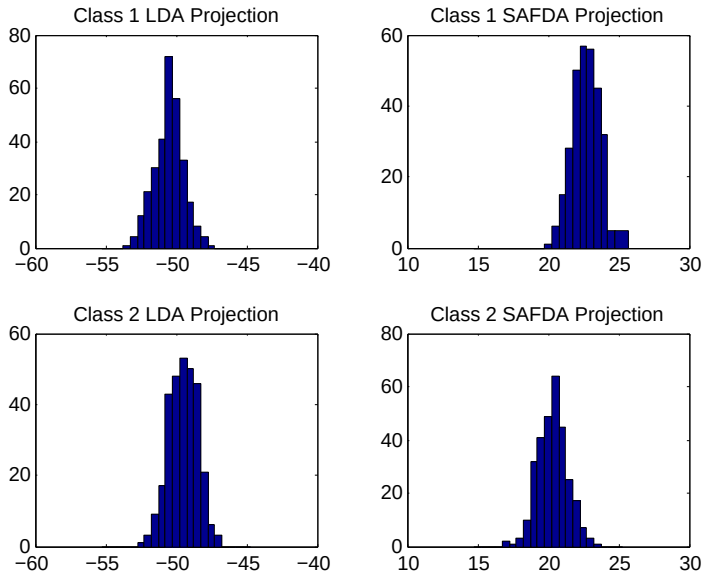

Figure 2.1: One-Dimensional Projection of Test Data in Classes 1 and 2 under LDA vs. SAFDA

Figure 2.1 shows a close-up view of the effect of the projection on classes 1 and 2 by showing the actual distribution of the test points projected in the optimal LDA and SAFDA directions. We see that while classes 1 and 2 still have some overlap under the SAFDA projection due to their proximity in the full space, there is much less overlap than under the LDA projection. Class 3 was easily separated under both paradigms and, therefore, not shown.

Chapter 3 will examine the SAFDA procedure in more depth with regards to performance, weighting function selection, computational time requirements, and potential algorithmic improvements.

Chapter 3: Case Studies, Performance, and Implementation Issues

3.1 Introduction

This chapter will explore the SAFDA algorithm's performance as well as the issues that arise in its implementation ranging from weighting function choice, scale parameter selection, downweighting scheme and stopping criterion. A thorough investigation of these issues allows for a better understanding of the separation of class means under the SAFDA and the standard LDA paradigms to improve the SAFDA technique to the utmost extent. The simultaneous shrinkage procedure of SAFDA attempts to maximize the correct classification rate while substantially reducing the computational time to make it feasible in practical situations.

Section 3.2 describes the databases used for the case studies. The MNIST handwritten database is used for a multi-class problem with a small number of classes and the AR face database is used as an example of standard face databases with a large number of classes.

Issues of weight function selection and parameter estimation are examined in Section 3.3. In terms of the Gaussian kernel weight function, the bandwidth parameter can be determined via a function of pairwise mean distances and, therefore, an acceptable function for adaptive parameter formulation needs to be investigated. An adaptive kernel is used for the Gaussian kernel (rather than using cross-validation) because the shrinking process

changes the effective dimension of the data and, therefore, shrinks the pairwise distances between means during dimension reduction steps. The shrinking distances imply that the bandwidth of the Gaussian kernel needs to change throughout the process to prevent over-compensation at the early stages or undercompensation at the later stages. An adaptive parameter search is suggested via the percentiles of the pairwise distances at a given stage.

Section 3.4 examines the potential of terminating the SAFDA procedure earlier than the fixed number of r_{max} shrinking iterations. The early stopping is used to prevent waste of time once the best lower dimensional space has been learned reasonably well. With the expectation that using downweighting schemes that were more geared to the data (eigenvalue order and/or magnitude) would lead to quicker convergence to a stable solution, Section 3.5 examines more informative downweighting schemes (rather than shrinking all dimensions simultaneously by a constant factor of α).

The overarching goal of this research was to make the F-LDA procedure feasible for large-dimensional data with a large number of classes (aka face recognition). While F-LDA provides improvements over many of the other LDA paradigms by integrating projection pursuit and Laplacian techniques into a linear discriminant analysis framework, its implementation becomes intractable in the large dimension framework due to computational complexity. SAFDA takes more time than the simplest, most pervasive standard LDA technique, but the computational burden is *significantly* less than that of F-LDA. Also, the positive results in higher correct classification rate achieved in F-LDA are also realized in SAFDA (and in some cases surpassed). The procedure focuses on searching for optimal projection directions while *simultaneously* reducing the import of all dimensions to be removed, which allows the *dependence* among the dimensions to be exploited. While F-LDA peels the dimensions off one at a time, SAFDA allows for a large number of dimensions

to be shrunken simultaneously and removed in a single step at the end (rather than sequential dimension removal one at a time). Section 3.6 examines the comparisons between the various algorithm's performance and time requirements.

3.2 Case Study Databases

We consider two case studies in this investigation: one for a smaller number of classes (MNIST) and another for a large number of classes (AR).

The MNIST database of handwritten digits (LeCun and Cortes, 1998) was developed by Lecun (NYU) and Cortes (Google Labs) to study a large-dimensional multi-class (10 classes) problem in the image analysis space. Handwriting contains a great deal of variation between individuals which makes this an ideal image discrimination problem. The database contains 60,000 training images and 10,000 test images that have been size-normalized and centered. To avoid cutting off part of the image in the localization process, all 20×20 images are centered in a 28×28 pixel bounding box using the center of mass. The grayscale images are then rasterized to create 784-dimensional feature vectors, which contain some 0-padded pixels introduced via the localization. This database is used to examine the properties of SAFDA on an image analysis problem that has few classes and is oversampled, thus allowing the within scatter matrix to be completely learned without need of regularization to make it invertible (although regularization is still employed to aid estimation). Figure 3.1 shows some example images of the MNIST data.

Clearly, the face recognition problem, whether in the biometric framework or faces in the wild, never has an oversampling of each individual to allow for a complete understanding of the within class (interpersonal) variation. To examine the SAFDA procedure on a face recognition problem with a large number of classes, the AR face database (Martinez,

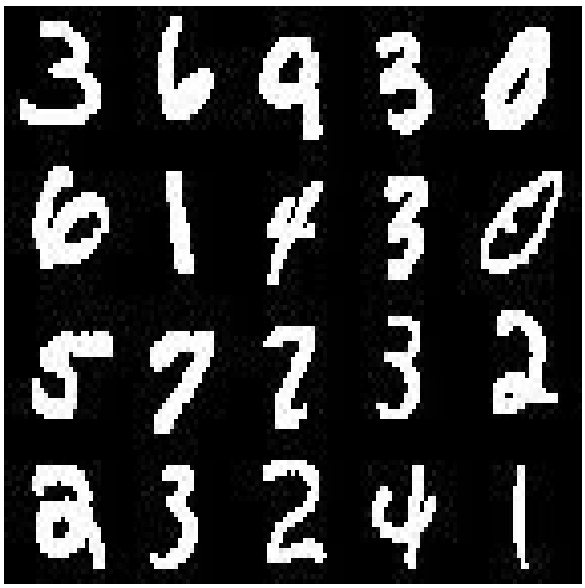

Figure 3.1: MNIST examples

1998) was used. The AR face database was developed in 1998 at the Purdue Computer Vision Center. Thirteen images per person of 116 people were taken under different characteristic changes (glasses, illumination, expressions, scarves, etc.) at two different times (2 weeks apart). While the images themselves were taken under strictly controlled conditions, no constraints were enforced on clothing, make-up, hair, etc. For purposes of this paper, the grayscale cropped face images of 100 people were used (the cropped data was preprocessed by the makers of the database). The cropped images were 160×120 leading to rasterized feature vectors of size 19,200. Training, validation (if needed), and testing sets were randomly generated from the complete set of images to simulate a realistic, random split of training and test data. The large number of classes (100) and undersampling that is inherent in the face recognition problem make this database a great tool for exploring the

Figure 3.2: AR Example Images

performance of the SAFDA procedure in a face recognition framework. Figure 3.2 shows some example face images (accessories, expressions, illumination, etc.).

3.3 Choice of Weighting Function and Parameter Selection

As stated in Chapter 2, there is no weight function (similarity measure) that is optimal for every problem. For the implementation of a simultaneous fractional methodology, we now investigate three weight functions (given in Section 2.2) in the construction of $\tilde{S}_B$.

One of the issues to be examined within both the F-LDA and SAFDA process is the separation of the sets of eigenvalues that are retained and those that are discarded. While this ratio of discarded to retained eigenvalue sums is not always linked to better classification, the progression of the individual sums can yield information about how the algorithm is affected by a single pair of classes that become arbitrarily close. The F-LDA process

shrinks the lowest eigenvalue on each step so the least informative dimension clearly approaches zero (at a rate determined by α) while the retained dimensions grow relatively. The shrinkage step only directly affects the least informative dimension, but its effect in the rotated space of retained eigenvalues could be noticeably detrimental to classification if the chosen shrinkage factor (α) was too large (thereby shrinking/removing information important to classification). This is not of much interest in the standard F-LDA methodology as shrinking a single eigenvalue has a limited effect on the retained data in the rotated space. Within the SAFDA framework, the effects of shrinking multiple eigenvalues (perhaps a large percentage of the trace of the weighted scatter matrix) can lead to a significant degradation of class separation in the retained space. This degradation can lead to washing out all discriminative information not only in the dimensions to remove, but also in those retained for classification. If the dimensions shrunken in the initial stages of the algorithm are important in the lower dimensional space, potentially important information can be lost in the early stages (since the eigenvalue decomposition only examines the separation criterion in the full space). This section is devoted to the understanding of the effects of various weighting functions, changing parameters of the weight function, and of the final dimension of projection on the progression of the sum of retained eigenvalues and the sum of the discarded eigenvalues as well as the ratio of those quantities.

We will examine the progression of the weights across the shrinkage steps to see how the SAFDA procedure 'molds' itself to the data and learns the appropriate space as well as seeing how the pairwise distances change in the rotating/shrinkage steps. The data will also be projected into the spaces learned during the 'intermediate' shrinkage steps to see if the algorithm provides a smooth, (mostly) monotone convergence to a lower dimensional space with regard to classification of the testing data. This progression of classification rate

across shrinkage steps can also yield insight into the potential for stopping the algorithm early.

As well as choosing which weight function to use, parameters for the function must be learned. In the standard machine learning problem, cross validation is traditionally used to select appropriate regularization parameters.

1. Solve for the projection matrix from a set of training data given an assortment of parameters.

2. Classify the projected data (using the various projection matrices) of a validation set and choose which parameter maximizes performance.

3. Project the disparate test data using the best parameter's projection matrix and classify.

Note that, cross-validation requires the entire procedure to be repeated many times (once for each attempted parameter value). The SFDA (non-adaptive) methodology improves the speed when compared to F-LDA, so cross validation (CV) can be performed (whereas the time commitment of F-LDA prohibits reasonable CV time) even though it is more costly than LDA.

The following sections will focus on the selection of the weighting function parameters for each of the measures discussed in Section 2.2. We hope that there is some directed path to parameter learning that may be discovered across a broad parameter grid, but given that the data drives optimal weight function formulation, there will be no clear methodology for choosing parameters other than a brute force search.

For each of these similarity measures in each case study, we will examine the following:

1. Effects of varying the parameter on correct classification rate (CCR),

2. Progression of ratio of eigenvalues retained to the whole trace across shrinkage steps,

3. Progression of pairwise weights across shrinkage steps,

4. Progression of CCR across shrinkage steps (to investigate if an effective optima is reached prior to the completion of r_{max} steps).

3.3.1 Inverse Pairwise Distance Similarity Measure

4

Lotlikar and Kothari (2000) used the weighting function given in Equation (2.5). One aspect that was never addressed in the original paper was the use of an unbounded weighting function when the parameter h was chosen via cross-validation. To make sure that the weights in successive shrinkage steps are comparable (since pairwise distances are a function of dimension), the weights are normalized to sum to 1. The shrinkage procedure can lead to intermediate steps in which two (or more) means are extremely close together and, therefore, *dominate* the criteria for determining the most informative dimensions. This causes a major problem in the weighted between scatter matrix criterion used for discrimination in that the impact of the dimension in which the means are incredibly close overrides all other conditions of separation, as was exhibited in the simple example at the end of Chapter 2.

A plot of interest examines the progression of the sum of the d retained eigenvalues (referred to as trace of the retained) and the trace of $\tilde{S}_B$ (i.e., Figures 3.4, 3.7, and 3.11). When utilizing the unbounded weighting function of inverse norm, the behavior of both these plots is quite erratic, especially when the data is projected into low-dimensional spaces (spaces in which two or more class means are far more likely to project very close to one another) and the choice of tuning parameter h is large. Notice that in low-dimensional spaces,

there are instances in which the retained eigenvalues 'spike' and lead to a convergence that is not monotonic. Smooth monotone behavior is desired to prevent overcorrecting to a single pair of classes.

Under this weighting function, the pairwise weights are relatively small for most pairs of classes and dominated by a single pair of classes. The progression of weights plotted with respect to r for the case studies that follow (i.e., Figures 3.5, 3.8, and 3.12) exhibit the impact a single pair of classes has on the overall weighting between class scatter matrix.

Case Study 1: MNIST Database

It is important to note that the random splits of validation and test set led to different plots; however, these plots had many characteristics in common. All exhibited a jagged pattern in eigenvalue and weight progression, indicative of overcorrection to easily confused classes. Also, the plots of CCR versus parameter choice showed much more local stability in CCR for values of h less than 10. Finally, the progression of correct classification always had the same increasing level of CCR across shrinkage steps. There is definitely a concern over the generalizability of the parameter in this small number of classes problem, especially due to the instability at larger choices of h.

First, to explore the effects of the parameter h on classification performance, we ran the SFDA algorithm many times using different values of h ranging from 2 to 25 by 0.1 increments. The lower bound of 2 was based on the fact that Lotlikar and Kothari (2000) stated that the weighting function must decrease faster than $\frac{1}{||\bar{x}_i - \bar{x}_j||^2}$. The upper bound of 25 was chosen arbitrarily to encompass a very broad range of parameters, yet be tractable in a reasonable amount of time. In traditional cross validation, this particular grid of parameters is more extensive than would be time/cost effective, however this section is focusing on

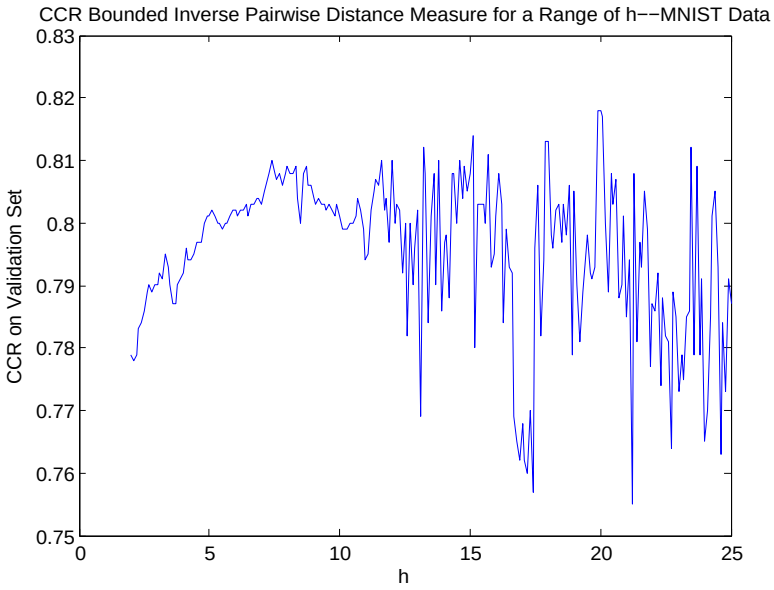

Figure 3.3: Effects of Parameter h on CCR Inverse Pairwise Distance Measure for MNIST d=4 (Training/Validation/Test Split)

discovering if any relationship between parameter and classification performance exists. If a pattern emerges, a more insightful search may be possible.

There is no discernible pattern immediately apparent from the plot (Figure 3.3) of the correct classification rate (CCR) over the grid of parameters (and this lack of pattern holds across many different training/validation/test sets). One particular facet of this analysis is that performance is much more erratic (higher variability) at the higher levels of h. This local stability in CCR for lower values of h and instability at larger values h was consistently seen across different training/validation/test splits. While the *'best' h* on the validation set occurs in the upper middle of this range of h (at 19.9), there is much more instability (generalization error) in performance for small changes in h for values of $h > 10$. This instability is intuitively explained by the space over-correcting to the training data (large values of h

give relatively more weight to directions in which classes are very close together). This suggests focusing our parameter search in the future on lower values of h (< 10). As a fine grid spanning a large range of h is not feasible in most cases and the results are much more stable for lower values of h, we will focus on a more coarse grid for $h < 15$. The stability of CCR for low h values suggests that a coarse h grid will find an approximate maxima.

For the plots that follow, we use the 'optimal' h parameter learned via our cross validation. For these random splits of MNIST data, the optimal h was 19.9. While this is not in the 'stable range' of h, it achieved the highest CCR on the validation set.

Next, to see how the SFDA procedure effects the relative sizes of the eigenvalues kept (as compared to the trace of the weighted between class scatter matrix), we'll examine the separation of the eigenvalues retained and the total trace. This visually demonstrates that, downweighting many dimensions at once does not remove all discriminative information in the dimension reduction process. The plot in Figure 3.4 shows (on an absolute scale) the progression of the traces of retained eigenvalues and trace of $\tilde{S}_B$ for the MNIST digit recognition data across the shrinkage steps for various final projection dimensions.

The most important aspect of these plots (Figures 3.4) is the erratic progression of the sum of the retained eigenvalues (the non-monotonic nature) as well as the total trace even for a problem with a relatively small number of classes. The zig-zag pattern exhibited by the sum of the retained eigenvalues is due to *overcorrection* that occurs in the early stages of the algorithm when a pair of classes gets very close together. As the shrinkage algorithm focuses on shrinking the least informative dimensions of the weighted between class scatter matrix, the choice of similarity measure (weighting function) can have a major impact on the rotation. The weighting function given by Equation (2.5) is not bounded and purely based on the inverse squared norm (which grows enormous for arbitrarily close

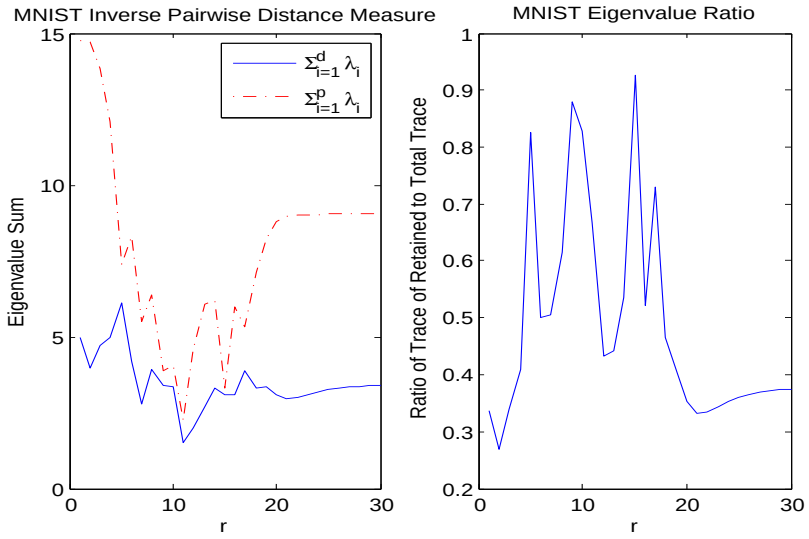

Figure 3.4: Eigenvalue Separation for MNIST data using Inverse Pairwise Distance Measure; h=19.9, d=4

classes). This concept leads a single direction (corresponding to the direction separating two arbitrarily close classes) dominating the weighted between class scatter matrix because the weight associated with that direction is incredibly large (demonstrated in Section 2.4.1). The next rotation is selected (via the eigenvalues) to separate those confused classes with little regard for directions that separate other classes. This, in effect, causes the class means to 'bounce' away from each other in the next rotation as the directed projection pursuit paradigm forces classes near each other to be easily separated in successive iterations. The righthand plot in Figure 3.4 shows the progression of the ratio of retained eigenvalues to the trace of the weighted between class scatter matrix across the shrinkage steps. This weight function eventually leads to a stable solution, but this is due more to the fact that the difference between the successive downweights gets smaller ($\alpha^{k-1} - \alpha^k < \alpha^{k-2} - \alpha^{k-1}$),

than to the choice of similarity measure. Eventually, there is little effect due to the shrinking as the values of the data in those dimensions have little variability.

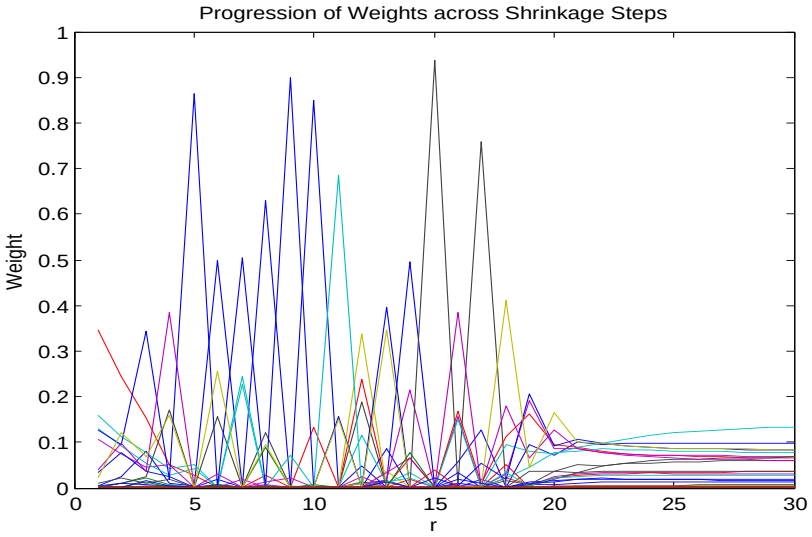

Figure 3.5: Progression of Weights using Inverse Pairwise Distance Measure for MNIST data; h=19.9, d=4

Figure 3.5 exhibits the progression of the individual weights associated with the pairwise distances to examine how the distribution of weights, spread amongst the pairs of classes, changes throughout the downweighting process. This plot also allows us to find the effect of a few pairs of classes coming very close together. While there are many distances to track (45 in the MNIST 10 class case), the important factor is the oscillatory nature of the weights. This shows conclusively that many of the shrinkage steps are driven by just one pair of classes that dominate the weighted between class scatter matrix at each intermediate step, even though this pair could be different at every step. The jagged rise and

fall in the weights assigned to different pairs of means exhibit that, at every iteration, a single pair of means dominates the eigenvalue determination (as those classes are very close together) and the correction leads to confusion among other classes. Stability is achieved only because the geometric progression of the downweighting shrinks the dimensions to be dropped to near zero ($1/r_{max}$).

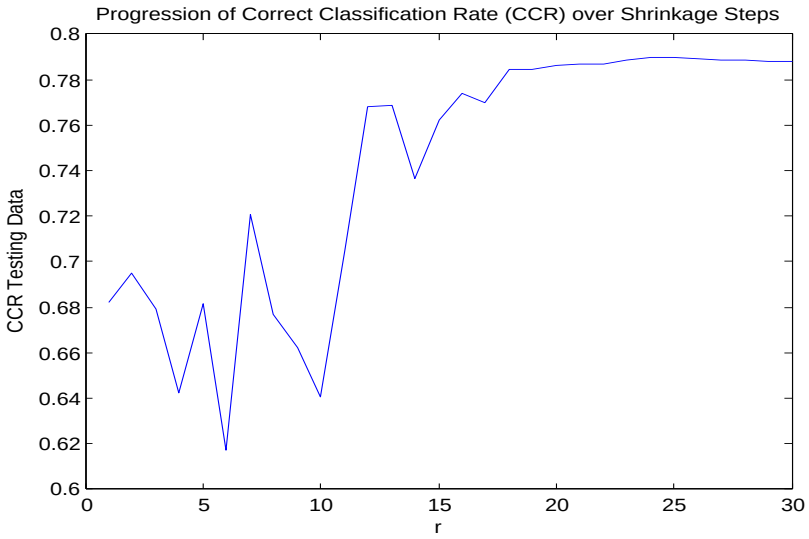

Figure 3.6: CCR Progression Inverse Pairwise Distance Measure for MNIST data; h=19.9, d=4

Figure 3.6 shows the progression of the CCR of the *testing* data as the shrinkage steps progress. The classification was performed using a nearest mean classifier on a testing dataset that was different from the training data set. While the improvement in classification is not monotonically increasing in r, the SFDA algorithm shows that using this inverse pairwise weight function leads to an improvement over standard LDA methodology, because the CCR of SFDA when $r = 1$ (first step of the procedure) is equivalent to weighted

LDA CCR (under a given choice of h). The monotonicity of the CCR is never guaranteed as the test and training sets are disparate. However, the erratic behavior that occurs in the early steps of the algorithm is primarily an artifact of the overcorrections that occur in the earlier stages, because larger downweighting (in the early stages) leads to pairs of class means getting closer together. This, in turn, causes the directions separating those means to have much more weight in $\tilde{S}_B$. The effect is forcing the largest eigenvalue to be related to the direction separating the two close means without considering the separation of all the means in the lower dimensional space, as exhibited in Section 2.4.1.

When only values of h from 2 to 15 were examined (the more stable range of h), the oscillatory nature was not as prevalent because lower values of the parameter do not exacerbate small distances as severely, causing less overcorrection to the validation set. We can see in Figures 3.7, 3.8, and 3.9 that the progressions of eigenvalues, weights, and CCR across the shrinkage steps are much smoother at $h = 7.1$ than at $h = 19.9$ because smaller values of h do not lead to as much overcorrection. There are still some remnants of the jagged pattern present in the early progression of weights in Figure 3.8, and a CV still needed to be run to find the best h for the dataset.

In this case, comparing Figures 3.6 and 3.9 shows that the lower value of h yields a better CCR on the *test* set (CCR=78.82% for $h = 19.9$ and CCR=79.1% for $h = 7.1$). This is because overcorrection in a situation with a small number of classes is detrimental to general classification on a disparate test set. Also, different randomly selected validation sets were used which also contributed to the discrepancy. In the MNIST case, all of the calculations are in 9-dimensional space and distances between class centroids often get very small during the downweighting process. If h is chosen relatively high, the performance on a given validation set can be optimized, but generalized results will suffer on a test set. We

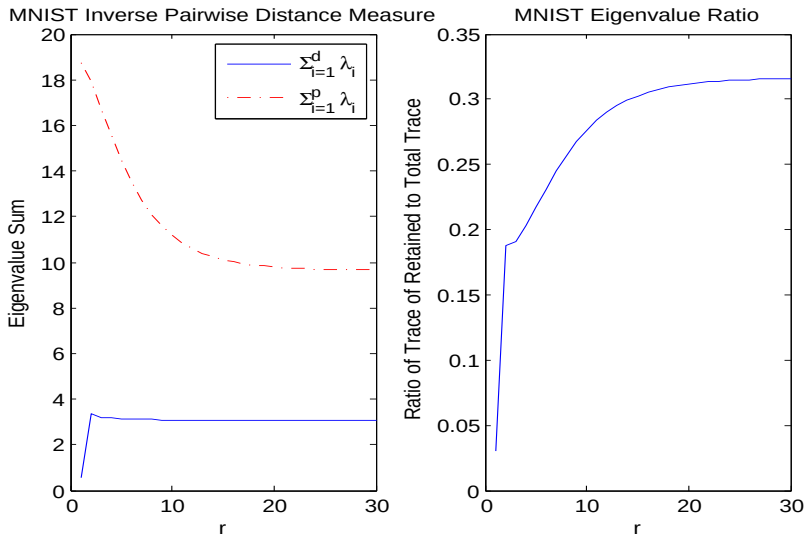

Figure 3.7: Eigenvalue Separation for MNIST data using Inverse Pairwise Distance Measure; h=7.1, d=4

show in Case Study 2, that this overcorrection phenomenon is not as costly in a problem with a large number of classes due to the curse of dimensionality. In either case, choosing an appropriate h via CV is computationally costly.

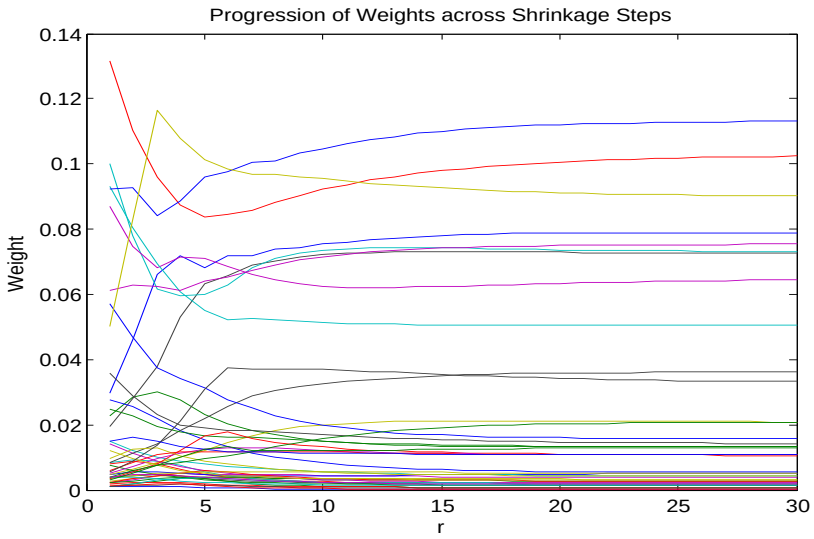

Figure 3.8: Progression of Weights using Inverse Pairwise Distance Measure for MNIST data; h=7.1, d=4

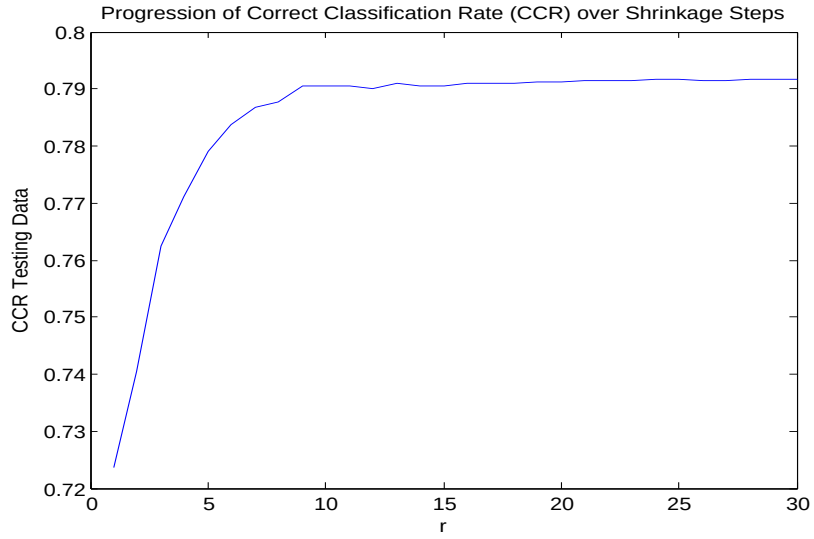

Figure 3.9: CCR Progression Inverse Pairwise Distance Measure for MNIST data; h=7.1, d=4

Case Study 2: AR Database

The same procedure was implemented in the AR face database (Martinez, 1998).

When choosing h via cross validation on the AR database, we see many of the same phenomena present in the MNIST. The variability in performance seems to increase as h increases leading to overcorrection which led us to focus on a more fine grid on the mid to lower levels of h ($7 \leq h \leq 15$). Figure 3.10 shows this phenomenon.

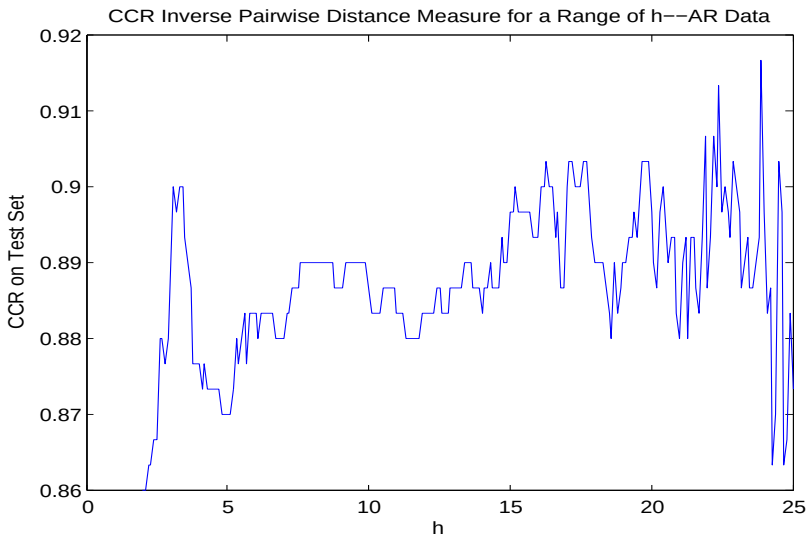

Figure 3.10: Effects of Parameter h on CCR Inverse Pairwise Distance Measure for AR Data d=10 (Single Training/Validation/Test Split)

While the middle values of h provide a more stable parameter for correct classification rate, the best rates examined over 10 different splits (training/validation/test) of the AR data yielded a wildly varying best h parameter (min=4.9, max=23.9, sd=6.9423). This is not a surprise given the weight measure's propensity for overcompensating when two

classes become very close. Because there was no consistency across various data splits, there is no reason to think the best parameter for this kernel can be achieved by any method other than brute force trial and error. CV is a very costly procedure as the algorithm has to be run many times under different parameters to find the best space. Weight functions that require cross validation to learn the tuning parameters (non-adaptive) will probably not be the best choice when compared to adaptive functions that only require one pass of the SAFDA algorithm. An adaptive kernel will be examined in Section 3.3.3.

We can see in Figures 3.11, 3.12, and 3.13, the results are more or less the same with a large number of classes (100 classes in AR database compared to 10 for MNIST). Clearly, the inverse pairwise distance similarity measure overcorrects in both large and small C problems thus leading to some questions of overfitting. This phenomenon is particularly noticeable at $r = 12$ in the weight progression plot (Figure 3.12) as one pair of classes overwhelmingly dominates the weighted between class structure. The inverse pairwise distance weight function's biggest issue in the large C problems (face recognition) is in excessive time and not in performance.

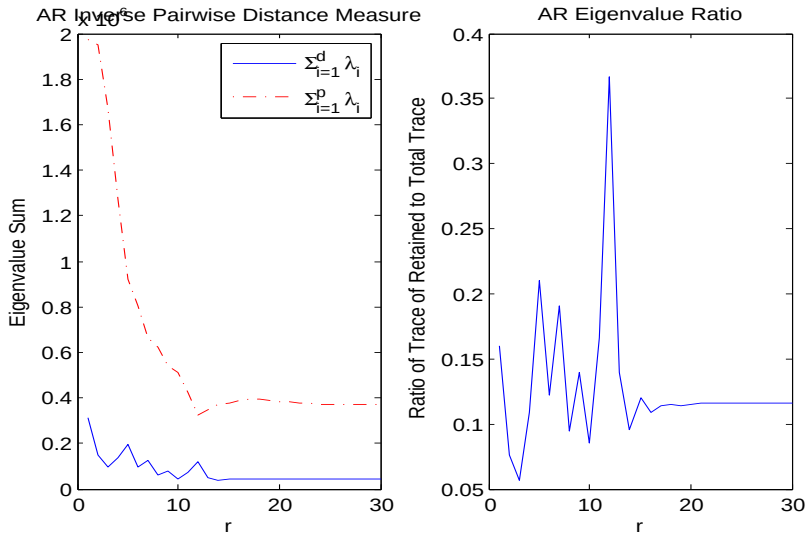

Figure 3.11: Eigenvalue Separation for the AR face database under Inverse Pairwise Distance Measure; h=23.9, d=10

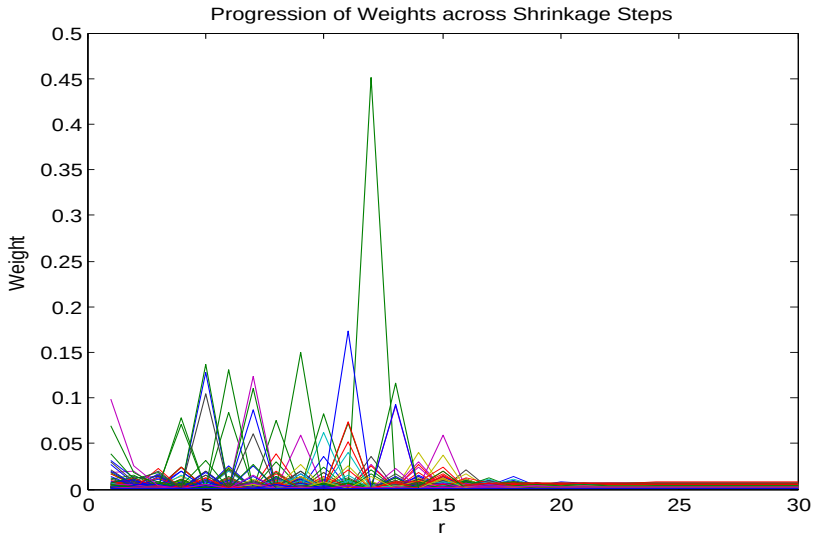

Figure 3.12: Progression of Weights using Inverse Pairwise Distance Measure for AR face database; h=23.9, d=4

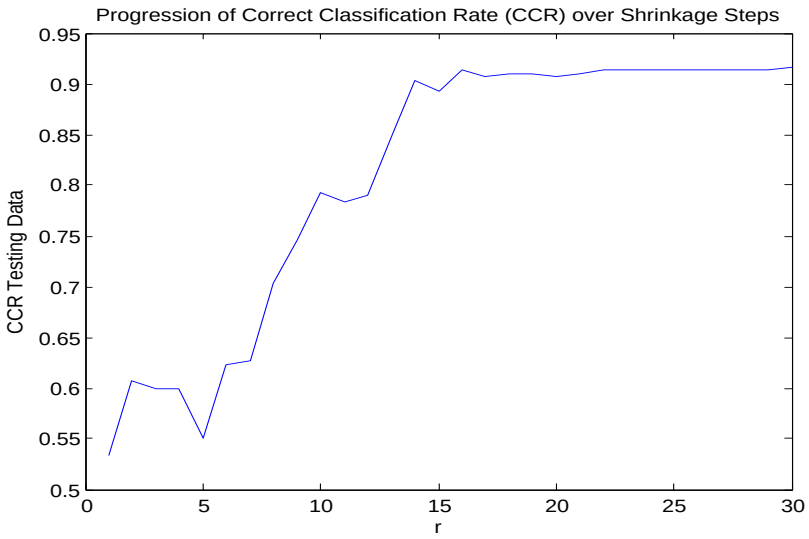

Figure 3.13: CCR Progression Inverse Pairwise Distance Measure for the AR face database; h=23.9, d=10

When the grid for examining h was restricted to lower range of values ($h < 15$), the plots were much smoother, once again, exhibiting that lower values of h may not lead to the 'best' performance on the validation set, they yield solutions that are more generalizable due to the stability. The additional cost of CV, even on this restricted range of h is still prohibitive. Figures 3.14, 3.15, and 3.16 show a smoother progression to a more stable solution using a value of h ($h = 3.1$) that is suboptimal on the validation set, yet in a more stable region and, therefore, more generalizable to a disparate test set.

Comparison of Figures 3.13 and 3.16 actually show that, in this particular case, the larger value of h performs slightly better (CCR= 91.67% for $h = 23.9$ versus CCR=90% for $h = 3.1$). This is most likely because the classes, being in a much higher dimensional space, are, therefore, spread farther apart. In the case of many 'outlier classes', overcorrection to those classes that are highly confused, may be beneficial. Classification results are quite good for both values of h, however, it is important to remember the additional cost inherent in the CV procedure. Section 3.3.3 will examine an adaptive procedure that can avoid the costly CV.

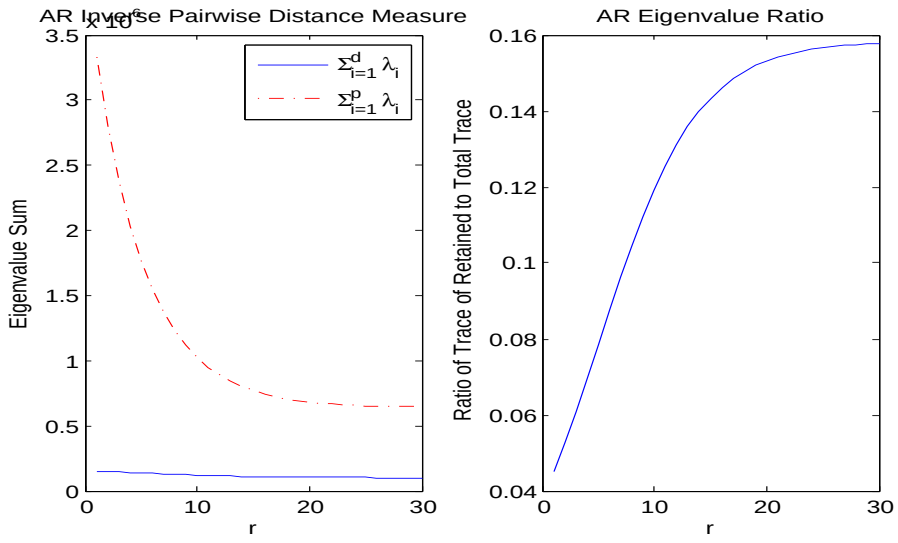

Figure 3.14: Eigenvalue Separation for the AR face database under Inverse Pairwise Distance Measure; h=3.1, d=10

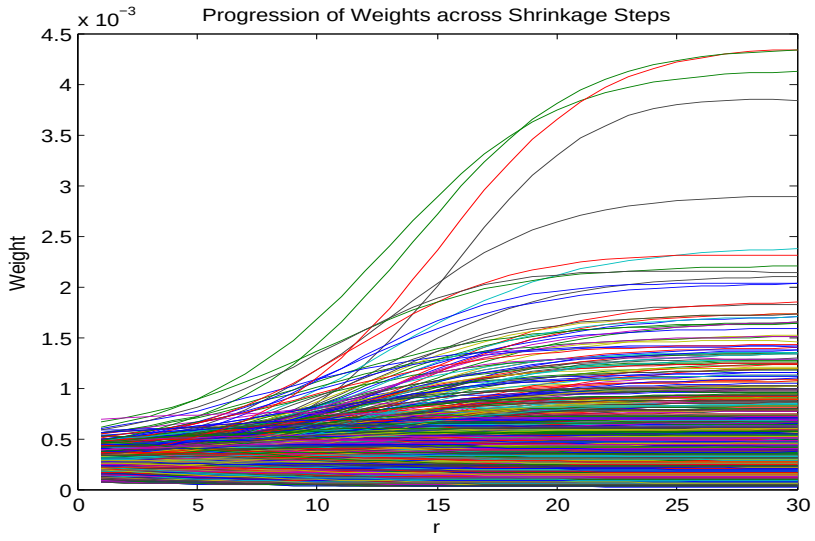

Figure 3.15: Progression of Weights using Inverse Pairwise Distance Measure for AR face database; h=3.1, d=4

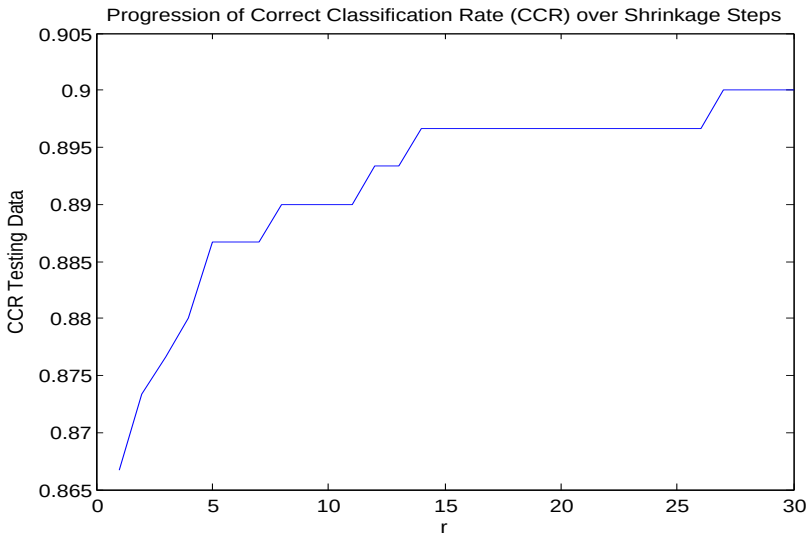

Figure 3.16: CCR Progression Inverse Pairwise Distance Measure for the AR face database; h=3.1, d=10

3.3.2 Bounded Inverse Pairwise Distance Similarity Measure

It was hypothesized that a bounded measure based on the same criterion may somewhat help the overcorrection problem as the values in the unnormalized weight matrix could not grow infinitely large. If a single pair of class means get too close together, the weight measure corresponding to the unbounded similarity measure can lead to machine infinity for the weight and thereby, completely dominate the weighted between class scatter matrix. For this analysis, the similarity measure described in Equation (2.6) is examined.

While this similarity measure is bounded between 0 and 1, the weights are still normalized to sum to one in order to ease comparison. The only difference between this measure and the inverse pairwise distance is that the quantity in the denominator is always greater than one. In the (unbounded) inverse pairwise distance measure, small distances are exacerbated (made even more important) by the parameter, h. Bounded measures don't overcorrect as severely as the unbounded version to classes that have a very small pairwise distance.

While the bounded measure prevents some of the overcorrection, it is still quite focused on classes that are relatively close together which still yields overcorrection in the intermediate steps. Quick examination of the plots showed that the bounded measure produces almost exactly the same results as the unbounded measure. In fact, the results are equivalent for most practical problems. The only cases in which results may vary between the two measures is when two (or more) classes are (or become via shrinkage) extremely close together ($||\bar{x}_i - \bar{x}_j|| \to 0$). In a large class framework (since all work occurs in $(C-1)$ dimensional space), the curse of dimensionality generally prohibits class means from being too close together in a large dimensional space. When the means are not close together, the effect is minimal as the weights are normalized.

Case Study 1: MNIST database

Compared to Section 3.3.1, the plots in Figures 3.17, 3.18, 3.19, and 3.20 are very similar. The differences are due more to the randomly generated validation set than the different choice of weight function.

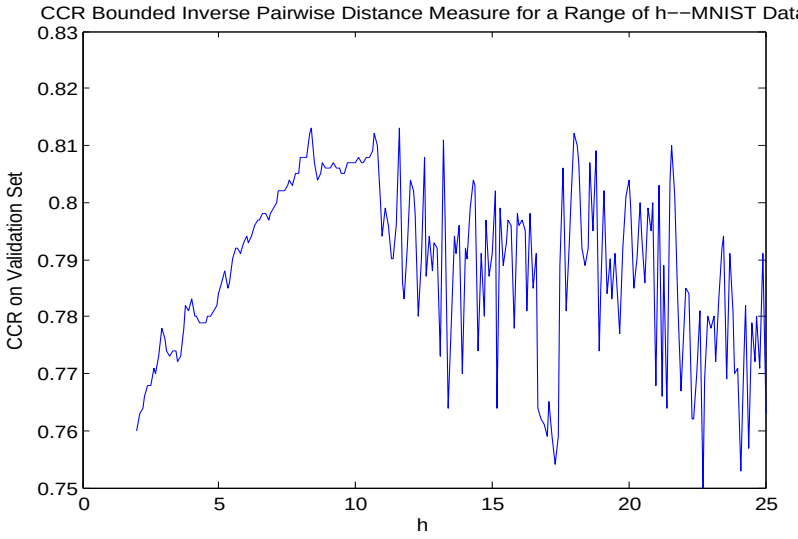

Figure 3.17: Effects of Parameter h on CCR Bounded Inverse Pairwise Distance Measure for MNIST database

It is interesting to note that the best h is lower than that of the unbounded weighting function. Similar local stability at the lower values of h is again observed until approximately $h = 10$. For this particular validation/test split, the best h occurred in the more stable region, but the performance was more akin to the large h's in the unbounded case in which the learned directions overcorrected to classes that were close (easily confused) in the projected directions.

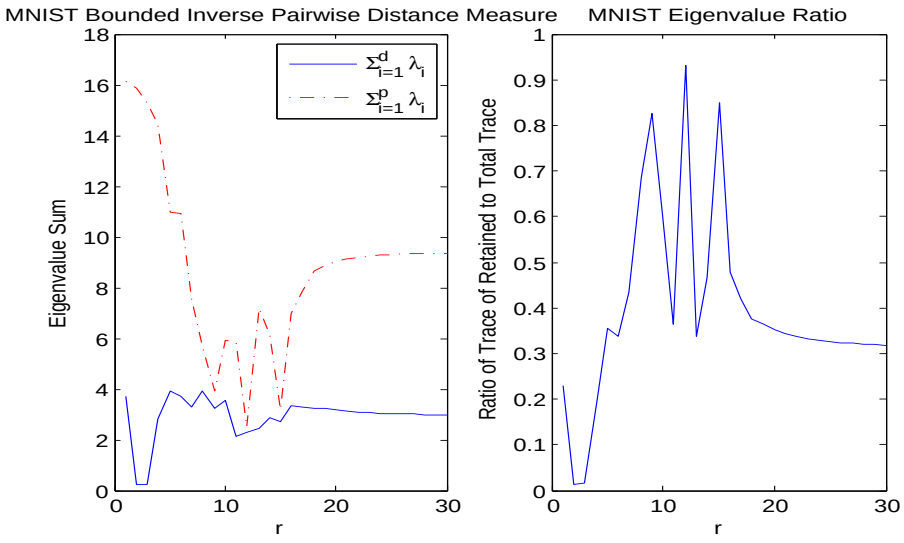

Figure 3.18: Eigenvalue Separation for the MNIST database using Bounded Inverse Pairwise Distance Measure; h=8.4, d=4

No appreciable difference was noted when comparing these plots to those of Section 3.3.1. Because the bounded weighting function is more suited to prevent very small distances from dominating the structure of $\tilde{S}_B$, the bounded weighting function is slightly preferable to the standard inverse distance weighting function of Lotlikar and Kothari (2000).

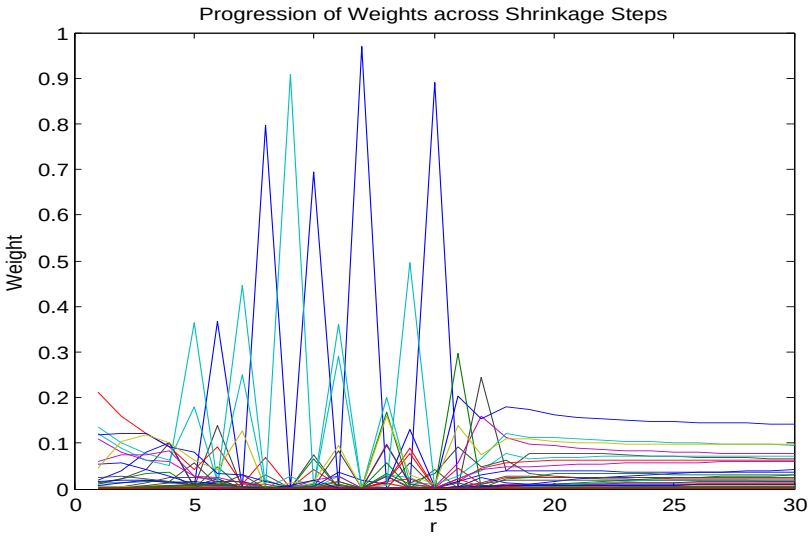

Figure 3.19: Progression of Weights using Bounded Inverse Pairwise Distance Measure for MNIST database; h=8.4, d=4

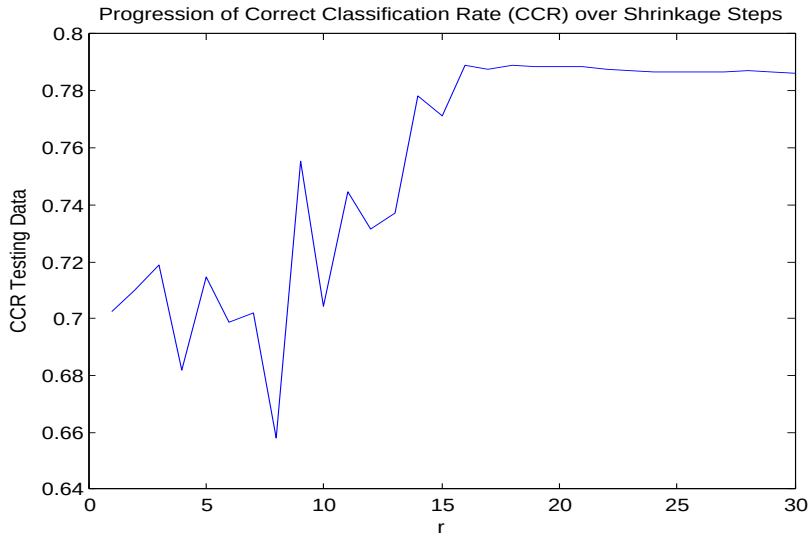

Figure 3.20: CCR Progression Bounded Inverse Pairwise Distance Measure for the MNIST database; h=8.4, d=4

Case Study 2: AR database

For the AR database, the plots below in Figures 3.21, 3.22, 3.23, and 3.24 are essentially *identical* to those in Section 3.3.1. While the weighting functions are indeed run on different training/validation/test set splits, the results are not visibly different due to the large-dimensional space. Recall, all work is done in $(C-1)$-dimensional space (99 in this case), so the likelihood of two classes getting close enough together for the bounding factor to make an appreciable difference is essentially zero.

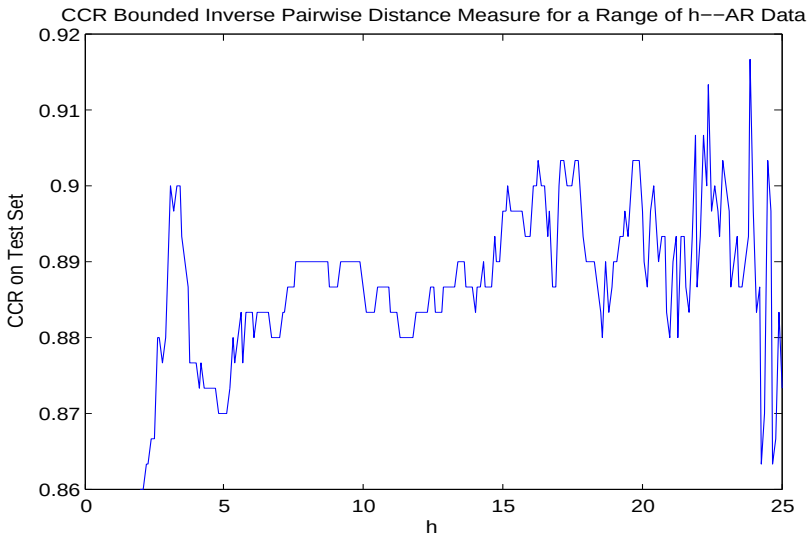

Figure 3.21: Effects of Parameter h on CCR Bounded Inverse Pairwise Distance Measure for AR face database; h=23.9, d=10

As shown in the next section, both these inverse pairwise distance measures (bounded and unbounded) suffer from a lack of stability, propensity to overcorrect to easily confused classes, as well as additional computational time for cross-validation. Therefore, we will

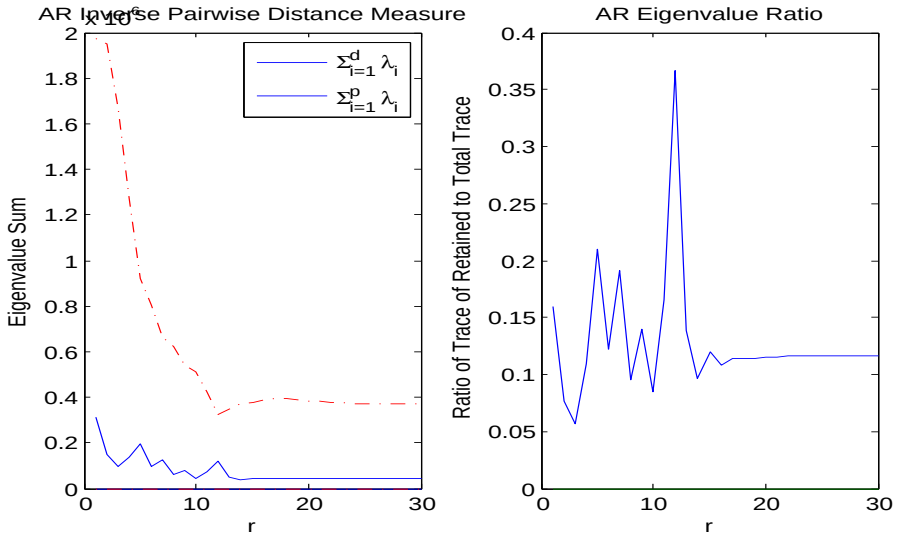

Figure 3.22: Eigenvalue Separation for the AR face database using Bounded Inverse Pairwise Distance Measure; h=23.9, d=10

focus on an adaptive weighting function. The Gaussian kernel provides a nice weighting function that allows us to avoid cross-validation through an adaptive framework easily learned via bandwidths.

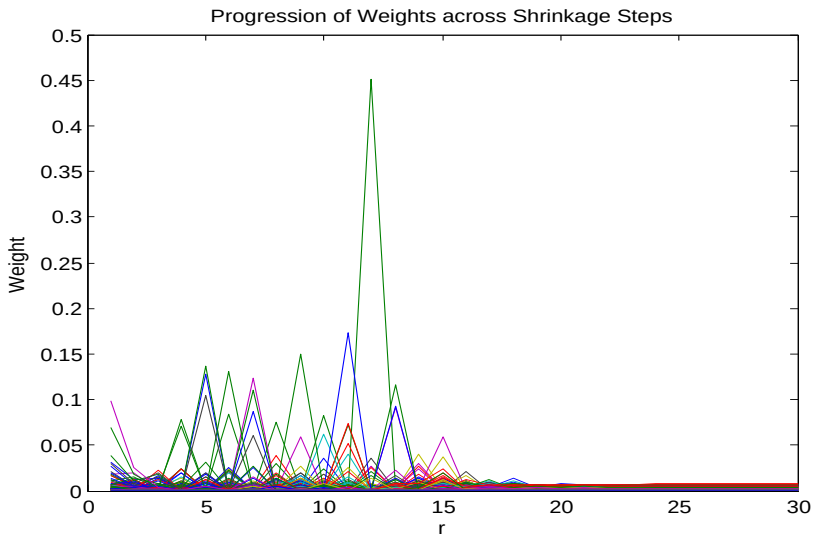

Figure 3.23: Progression of Weights using Bounded Inverse Pairwise Distance Measure for AR face database; h=23.9, d=10

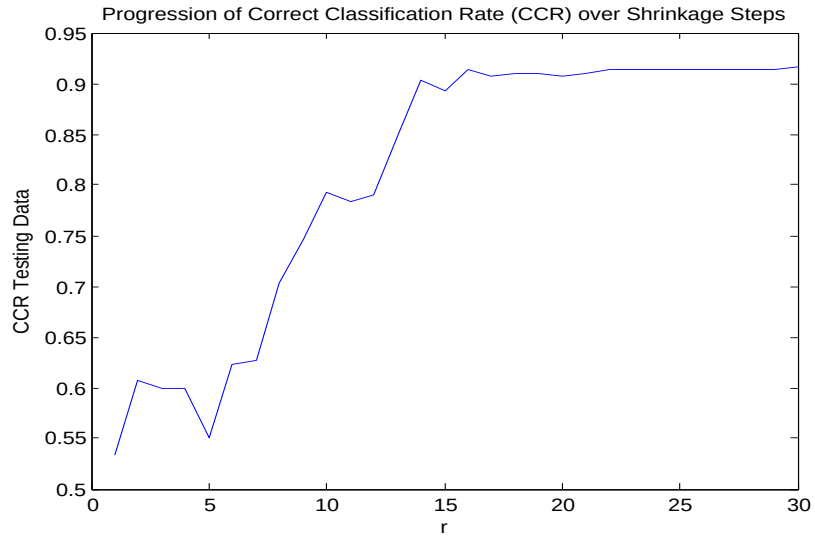

Figure 3.24: CCR Progression Bounded Inverse Pairwise Distance Measure for the AR face database; h=23.9, d=10

3.3.3 Gaussian Kernel

Perhaps the most widely used kernel in the statistical learning literature is the Gaussian kernel given in Equation (2.7).

The Gaussian kernel provides a smooth, bounded dissimilarity measure between two points that is perfectly suited as a weight function in the SAFDA framework. One aspect of the Gaussian kernel that differs from the similarity measures discussed in Sections 3.3.1 and 3.3.2 is that its bandwidth parameter is more tied to the scale of the data. For the measures discussed in the previous two sections, the parameter was selected via cross validation which is a very time-consuming process (this will be more closely examined in Section 3.6), as that parameter was not related to the scale of the data. The bandwidth in the Gaussian kernel is integrally related to the scale of the data as it provides a measuring stick for determining the importance of the surrounding points. If the bandwidth is chosen too small, close interpoint distances overwhelm all other considerations and lead to the same problems of the inverse distance similarity measures. If the bandwidth is chosen too large, all pairwise distances are 'washed out' and a uniform weight is imposed thereby mirroring LDA. More importantly, the bandwidth choices *must change throughout the shrinking steps* to account for the shrinking of the pairwise distances simply due to downweighting of the dimensions to be discarded. While the curse of dimensionality states that distances between points increase as more dimensions are added, the opposite is also true in that the downweighting of dimensions (via our fractional scheme) brings the means closer together. This is why a weighting function with an *adaptive* parameter is introduced.

Using a percentile of the pairwise distances permits the bandwidth to automatically *adapt* to the progression of the data through shrinkage iterations. We will examine the percentiles of the pairwise mean differences to see if there is an optimal percentile to use in

the adaptive scheme. As time considerations are of primary concern in the implementation of the SAFDA algorithm, the adaptive algorithm is much more attractive as it only needs to run just one time (one pass algorithm) assuming a percentile has been chosen *a priori*. Of course, the best percentile could be chosen via cross-validation to fit a given problem, but our adaptive procedure seeks a percentile that generalizes well to most situations, thereby removing the need of CV. To determine the best percentile for the choice of the bandwidth parameter, h, a range of percentiles was examined for the case studies. Intuitive reasoning suggests that the minimum would be a good choice as we desire relatively large pairwise distances to have little impact on the learned space. The larger choices of bandwidth lead to giving measurable weight to larger pairwise distances.

Case Study 1: MNIST database

Figure 3.25 shows the effects of using different percentiles of pairwise distances for the Gaussian kernel bandwidth for MNIST data. The *best* choice of percentile for the given training/test split is the 19^{th} percentile. However, the minimum seems to perform about as well as the 19^{th} percentile (less than half a percent of CCR lower).

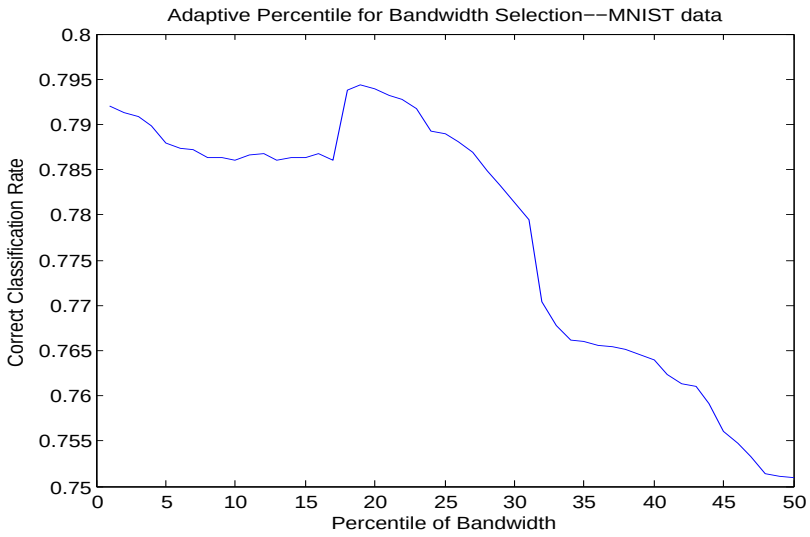

Figure 3.25: Bandwidth selection using percentiles MNIST data; d=4

The minimum is actually a more generalizable choice for an adaptive parameter in the case of a large number of classes when examining this phenomenon on a few large datasets. Because the minimum gives one of the best adaptive parameter choices (as seen by the upward path of CCR near the minimum), the bandwidth in the plots that follow is the minimum pairwise distance of the shrunken data at each step.

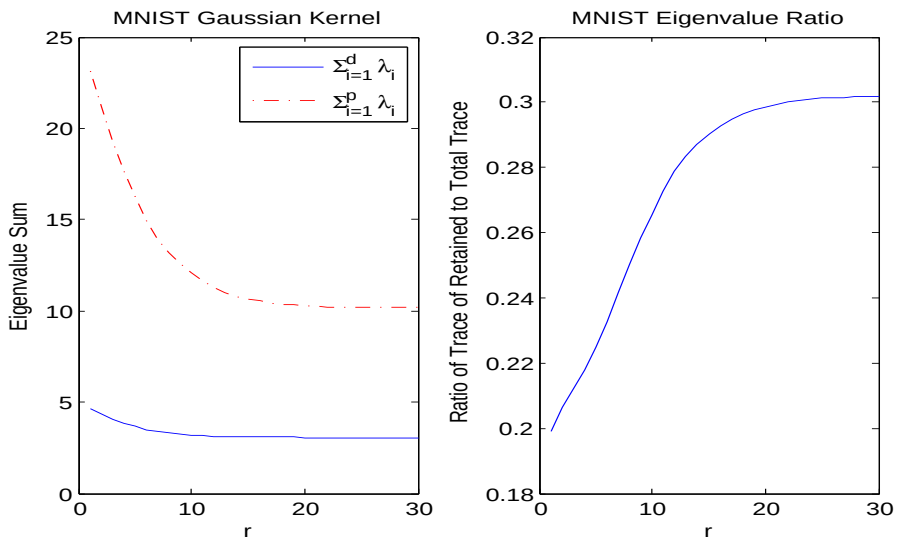

Figure 3.26: Eigenvalue Separation for MNIST data using Gaussian Kernel; d=4

Figure 3.26 clearly shows a much smoother progression of the sum of the sets of retained eigenvalues and total trace than the inverse pairwise distance weighting functions (Figures 3.4 and 3.18) due to the smoother bounded kernel.

As seen in Figure 3.5, the inverse pairwise distance measure allows for a single pair to dominate the weighted between class scatter matrix. This is because the inverse pairwise distance measure has a very small 'region of influence' (i.e., classes must be very close together, compared to the closest pair of class centroids, to have any appreciable impact on the weights, especially for large values of h). However, the upper bound of one and larger region of influence in the Gaussian kernel weight measure makes it very difficult for a single pair of classes to determine the best directions. Figure 3.27 shows smooth convergence of the weights in which each step in the shrinkage process updates the best rotation by a small amount without completely changing the projection matrix to account for a single pair of

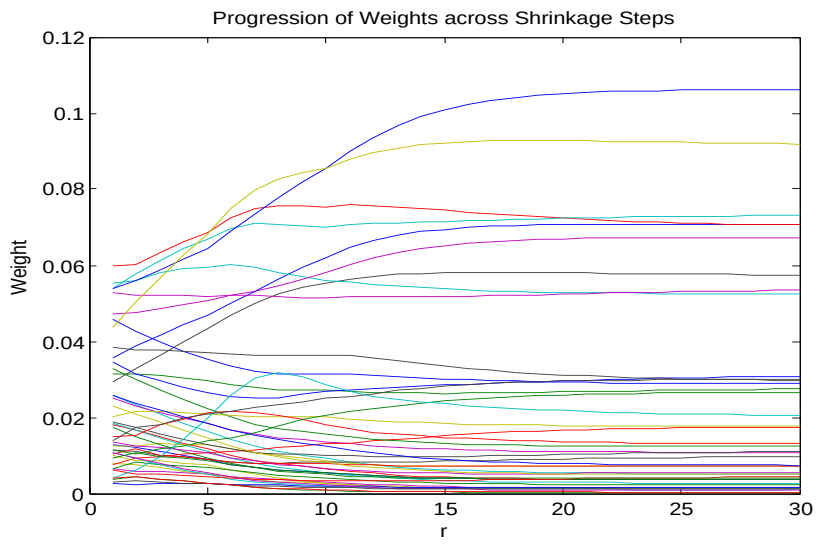

Figure 3.27: Progression of Weights using Gaussian Kernel for MNIST data; d=4

confused classes. Another detail of the plot in Figure 3.27 that differentiates the Gaussian kernel from the other weight functions is the dispersal of weights to a broad range with no overwhelmingly dominant direction. This shows that while some directions are more important than others, all of the directions have some import in the best direction, whereas the inverse pairwise distance measures put almost all the weight on very few directions and more or less ignore the rest.

The progression of classification on the test set as the shrinkage occurs (Figure 3.28) shows a slight degradation of performance over the first two shrinkage steps, followed by a marked increase in performance until a plateau is reached about halfway through the shrinkage steps. The early dropoff in classification is most likely due to tuning the best rotation which is much more fluid in the early shrinkage steps (larger 'chunks' removed within each step). The dropoff in the first steps is hypothesized to be caused by the tuning

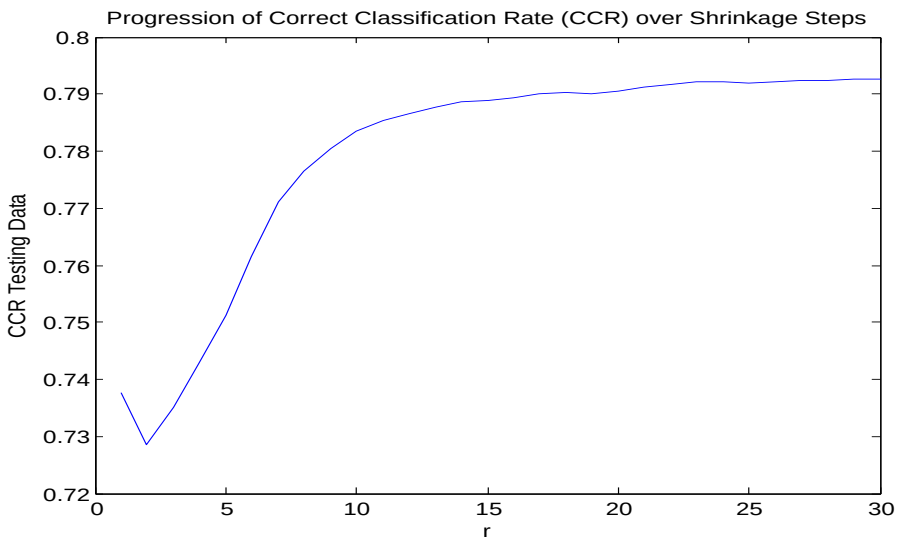

Figure 3.28: CCR Progression Gaussian Kernel for MNIST data; d=4

of the algorithm across shrinkage steps. The directed search *a la* projection pursuit seeks to maximize the separation criterion of the training means, but the early stages overcorrect to more closely match the training data.

Case Study 2: AR database

Figure 3.29 shows that using the minimum pairwise distance for the bandwidth yields the best result across a broad range of percentiles (0^{th} to 50^{th}) for the AR data. The minimum pairwise distance (0^{th} percentile) is the best bandwidth choice because the smallest distances are those that need the most focus. When the distances get much larger than the minimum, they are not nearly as important in the construction of the weighted between class scatter matrix. It should be noted that a finer grid of percentiles near the minimum has been explored, as earlier results suggested the lower percentiles may yield better CCR results.

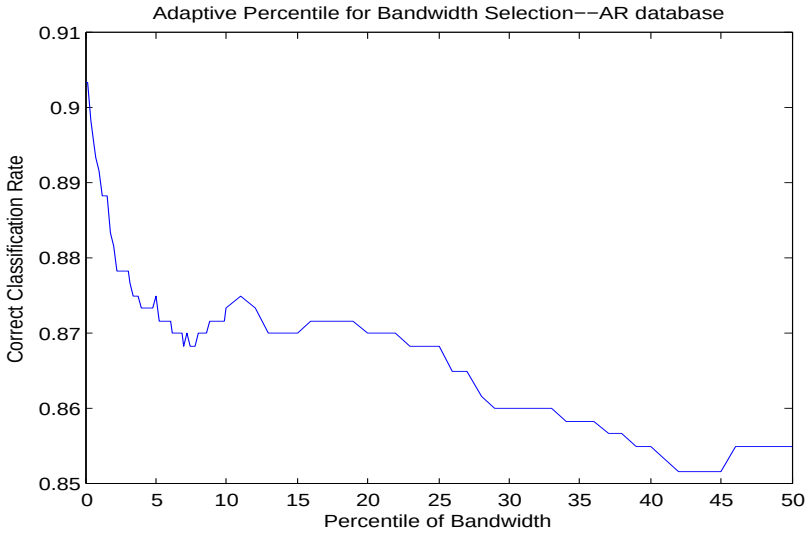

Figure 3.29: Bandwidth selection using percentiles AR face database; d=10

Figure 3.30 shows the performance in a small neighborhood of the minimum pairwise distance for the bandwidth to see if there would be any benefit to using a bandwidth

slightly below or above the minimum pairwise distance. This plot shows the effects of using a varying fraction of the minimum ($\frac{1}{2}$ to $\frac{3}{2}$) as the bandwidth parameter on 10 different training/test splits. Results show that there is no consistent change by using a fraction of the minimum pairwise distance for the bandwidth (i.e., different splits produce different best choices of the best multiplicative coefficient of the minimum). Across the 10 different splits, the optimal choice of minimum fraction ranged from 0.6 to 1.3 with no clear dominant fraction. Therefore, for simplicity, the minimum will be used for the bandwidth of the Gaussian kernel.

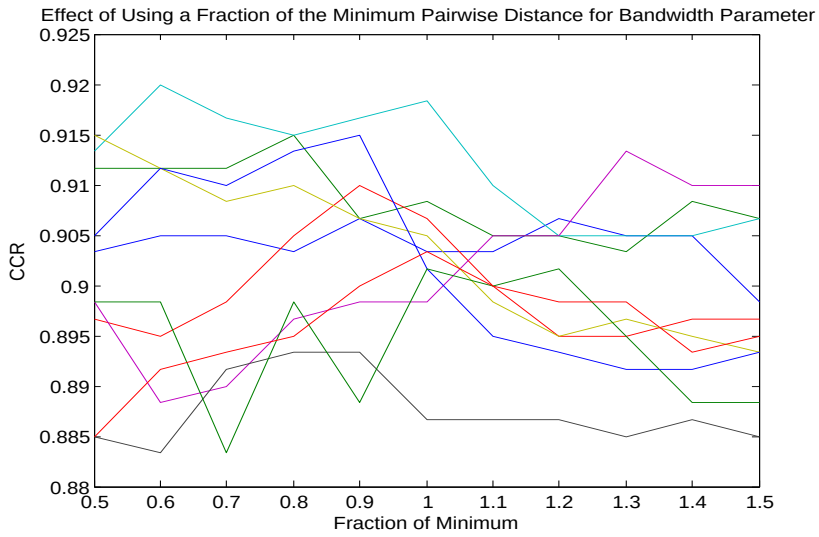

Figure 3.30: Fraction of Minimum Pairwise Distance examined for Bandwidth; d=10

Many similar phenomena are discovered in the database with a large number of classes under the Gaussian kernel as in the MNIST database. A much smoother convergence of

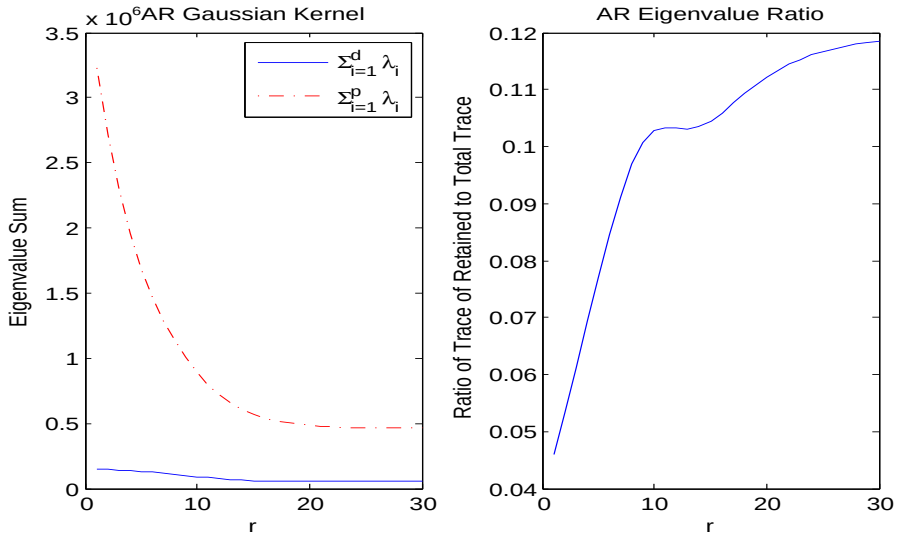

Figure 3.31: Eigenvalue Separation for AR data using Gaussian Kernel; d=10

the trace of the retained and the trace of $\tilde{S}_B$ (Figure 3.31), a very smooth weight progression (Figure 3.32), and a slight degradation in performance followed by a large improvement across shrinkage steps (Figure 3.33) are all once again present as seen in the MNIST database.

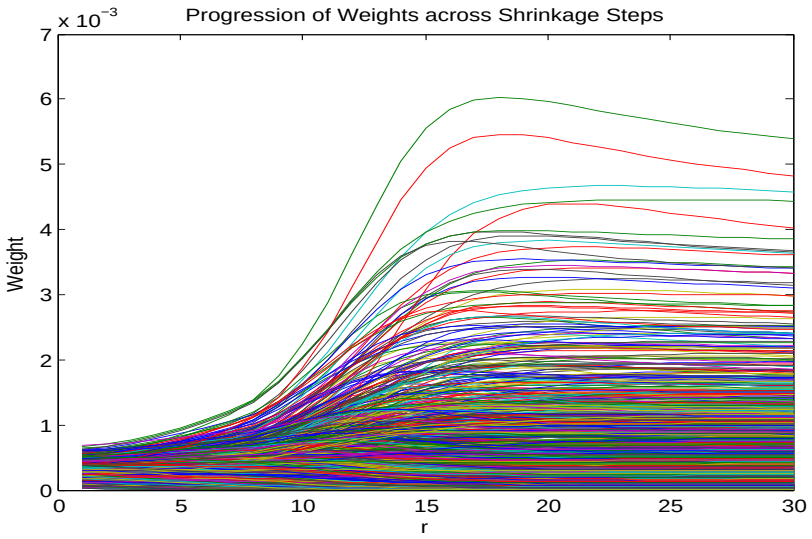

Figure 3.32: Progression of Weights using Gaussian Kernel for AR data; d=10

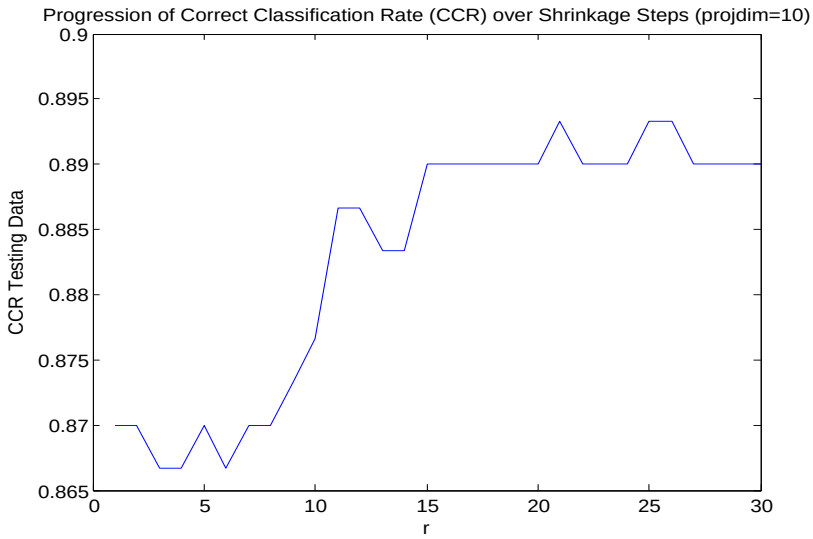

Figure 3.33: CCR Progression Gaussian Kernel for AR data; d=10

Discussion

The Gaussian kernel lends itself perfectly to the SAFDA procedure because it is:

- Simple to understand,
- A smooth, bounded kernel that yields a smooth rotation/convergence, and
- Easily suited to an adaptive framework, thus removing the computationally costly need of CV for parameter selection.

While the size of the database (number of classes) shows little connection to which similarity measure is the best with respect to these convergence criterion, the weighting functions chosen clearly lead to a differing progression of eigenvalues and weights as the best projection is discovered. The Gaussian kernel's smooth transition from one rotation to the next is beneficial by not allowing the algorithm to overcorrect within intermediate steps no matter the number of classes. Also, the adaptive nature is quite useful to remove the time prohibitive need of cross validation in the large number of classes problem.

It is useful to compare Figures 3.31, 3.32, and 3.33 to their analogues in Section 3.3.1 (Figures 3.14, 3.15, and 3.16) and note many similar features. In fact, the CCR for the inverse pairwise distance similarity measure is slightly higher than that of the Gaussian (about a 1% difference). This can be explained by the inverse pairwise distance measure using CV to find the best choice of parameter, or it may simply be due to the random training/validation/test splits chosen. While this case shows slightly *better* performance out of the inverse pairwise distance measure, it is important to remember the additional computational cost incurred by that method through CV. The Gaussian achieved near identical results to the inverse pairwise distance measure in this case (and will be shown in Section 3.6 to outperform the CV measure in most cases), yet the computational time requirement

was much lower (only one-pass needed). Overall, the Gaussian kernel exhibits many properties that make it ideal for this framework of weighted LDA and, more specifically, make the fractional LDA paradigm feasible in real time through SAFDA.

If the best projection was learned prior to completion of the r_{max} shrinkage steps, the algorithm can be stopped early to save additional calculations that are redundant. Due to the smooth behavior of the weight progression when using the Gaussian kernel, it is unlikely that great changes in the learned projection space would arise in the later stages if a stable point had been achieved. We will examine this issue in Section 3.4.

3.4 Stopping Criterion

The F-LDA (and by extension, yet to a much lesser extent, SAFDA) procedure involves a great deal of computational effort by recomputing the weighted between class scatter matrix at every shrinkage step and realigning the best projection direction with the shrunken space. These computations contribute to the hefty time cost of the algorithm. While SAFDA significantly reduces the number of steps by simultaneously shrinking all dimensions to be dropped, there may be additional time savings by stopping the shrinkage steps early if the 'best' rotation had been found prior to the completion of the r_{max} prespecified shrinkage steps.

As the shrinkage steps apply a geometric scaling in the traditional downweighting scheme, the earlier steps induce a larger impact between successive pairwise differences than the later steps. A stopping criterion was examined that would allow the shrinking portion of the algorithm to reach an early termination once some sense of convergence of weight vectors had been achieved. To measure convergence of the pairwise distances, a

divergence measure between the normalized weight vectors at successive steps was calculated. When that measure decreased to a fixed value (or fixed percentage of the original distance metric), the algorithm would terminate and the projection matrix at that step would be used for classification.

Common divergence metrics used to compare successive weights are:

1. Kullback-Leibler divergence–$KL(\mathbf{w}^{(l)}, \mathbf{w}^{(l+1)}) = \sum_{i=1}^{C(C-1)/2} ln\left(\frac{\mathbf{w}_i^{(l+1)}}{\mathbf{w}_i^{(l)}}\right)\mathbf{w}_i^{(l+1)}$

2. Hellinger distance–$H(\mathbf{w}^{(l)}, \mathbf{w}^{(l+1)}) = \sqrt{2\left(1 - \sum_{i=1}^{C(C-1)/2}\sqrt{\mathbf{w}_i^{(l)}\mathbf{w}_i^{(l+1)}}\right)}$

Both divergence measures are useful for measuring divergence between the weight vectors. The primary benefit to using the Hellinger metric over the KL divergence is the bounded nature of the Hellinger distance. Kullback-Leibler is sensitive to very small values (since the weight of the previous step is in the denominator) which leads to a more volatile metric that is not on a consistent scale from one training set to the next.

To select an appropriate stopping criterion, we examine the progression of Hellinger distances of successive weight vectors in conjunction with the corresponding classification performance on the training data at each step. Empirical evidence based on many such plots suggests that the near-maximum achievable CCR occurs by the time the Hellinger distance between successive weight vectors has decreased to one third of the Hellinger distance between the initial two weights. This is heuristic, but it has shown promising results. At times, waiting until the weight vectors have come that close together (Hellinger distance decreased to one third of the initial distance) can be waiting too long (i.e., the plateau being reached earlier). It was determined to set the threshold low enough to allow for some fluctuations in the Hellinger distance at earlier iterations without stopping at a local minimum and to ensure that further iterations would not significantly improve classification.

3.4.1 Case Study 1: MNIST database

Figures 3.34 and 3.35 show progression of the weight vector divergence measures across shrinkage steps for the MNIST data. Clearly, they look similar and converge to 0 (as the shrinkage steps have less and less affect in each step due to geometric progression), but the interesting feature is the scale of the respective measures. KL divergence scale is dependent on the number of classes (length of the weight vectors), whereas the Hellinger distance metric is in a consistent range regardless of the number of classes. This can be seen by comparing Figures 3.34 and 3.35 with Figures 3.37 and 3.38.

A threshold for the Hellinger distance was chosen to be one third of the initial distance between successive weights. This led to the SAFDA procedure terminating (for the MNIST data) after 12 shrinkage steps. Figure 3.36 shows the threshold on the Hellinger plot, as well as the associated CCR at the derived stopping step. There is very little additional performance achieved after the 12^{th} step (as seen in the lower plot), therefore, we conclude that the threshold of one third initial weight distance is reasonable for a small number of classes problem.

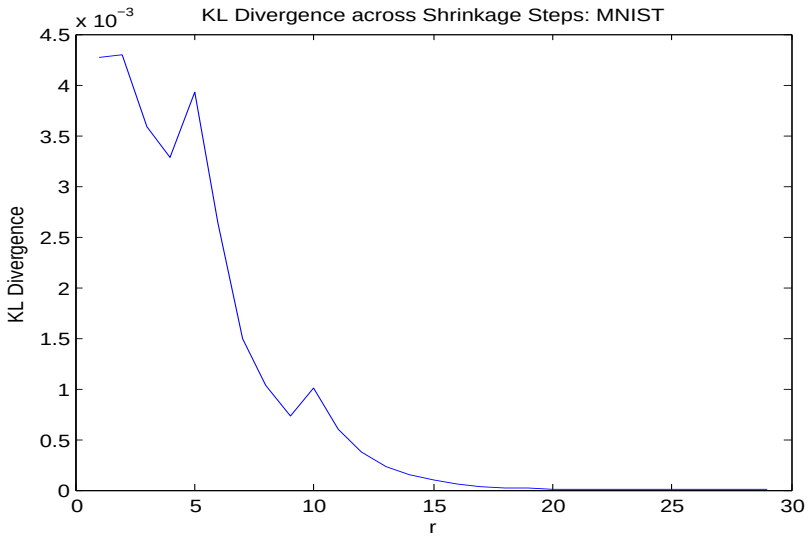

Figure 3.34: Kullback-Leibler progression on MNIST; d=4

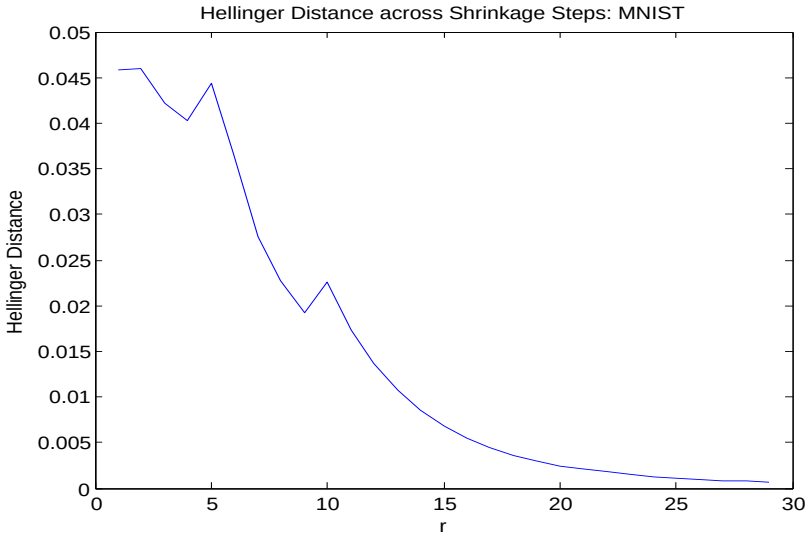

Figure 3.35: Hellinger distance progression on MNIST; d=4

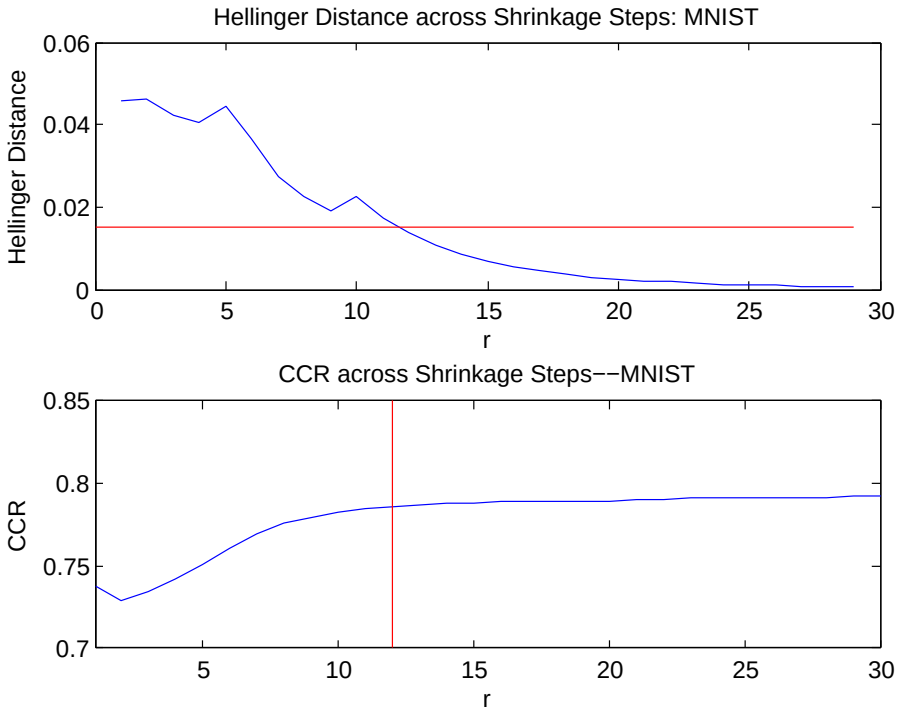

Figure 3.36: Hellinger distance progression with CCR on MNIST; d=4

3.4.2 Case Study 2: AR database

Figures 3.37 and 3.38 show the two metrics and their progression across shrinkage steps on the AR database using the Gaussian kernel on *ten* different random splits of the training/test datasets. They look similar to each other, but the scale is more consistent with that of the MNIST data with fewer dimensions/classes under the Hellinger distance metric.

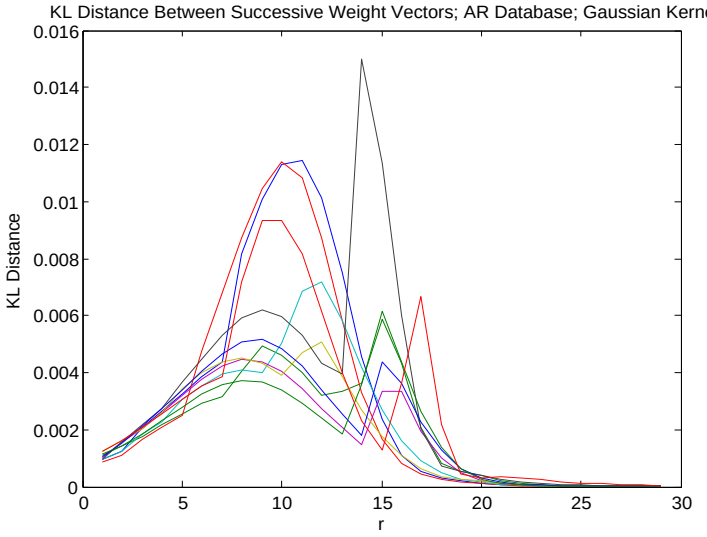

Figure 3.37: Kullback-Leibler progression on AR; d=10

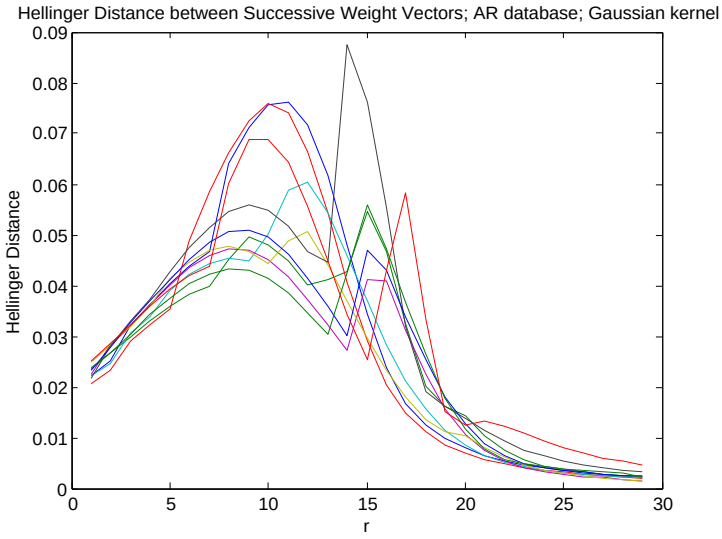

Figure 3.38: Hellinger distance progression on AR; d=10

Figure 3.39 shows an example of Hellinger distances concurrent with the associated CCR for a random test set. In this figure, the maximum classification accuracy was indeed reached at an earlier time, but that is not necessarily the case in general. Once the Hellinger distance has decreased below the threshold line on the graph (one third of the initial Hellinger distance), the correct classification rate shows little to no improvement. This pattern was observed for many splits of the AR database.

The SAFDA procedure was carried out many times on different random splits of the AR data, and the performance results are shown in Tables 3.6 and 3.7. The results show the early stop actually yields a higher classification rate in some cases (although there is a difference, it is minimal and not appreciable in practice). The average CCR using this stopping criterion across 10 different random splits of the AR data to d=10 dimensions

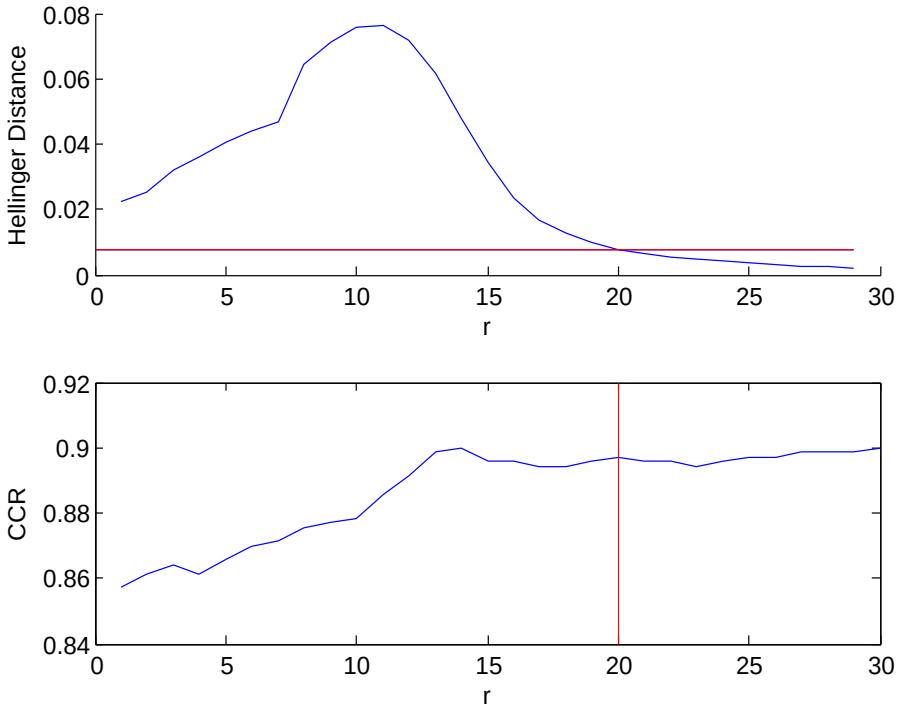

Figure 3.39: Hellinger distance progression with CCR on AR; d=10

was 88.7% (sd=0.93%), whereas carrying out the SAFDA procedure throughout all shrinkage steps yielded an average CCR of 88.9% (sd=0.77%). Table 3.7 shows a mean time differential of 3.19 seconds which seems rather insignificant. However, 3.19 seconds is a savings of approximately 25% of the SAFDA portion of the algorithm. This savings would be much more pronounced in a larger database (more classes). The preprocessing time is a function of both number of dimensions and number of classes, whereas the SAFDA algorithm is purely based on the number of classes (assuming $C < p$). The stopping criterion for the AR database yields a 25% time savings over the full algorithm with no significant

loss of classification performance (when examining SAFDA's contribution separate from preprocessing) .

The stopping criterion using Hellinger distance was also used under the bounded inverse distance kernel. Tables 3.4 and 3.5 (in Section 3.6) show similar results of SAFDA (both early stopping and full runs) being more or less the same in CCR as F-LDA and still dominating LDA and wLDA with little extra cost in time. Clearly, the time commitment of the procedure is far larger when cross-validation is utilized (even with the stopping criterion added). The non-adaptive kernel can be improved using a stopping criterion, but is still not feasible in real time for large number of classes to search a reasonable grid of parameters via CV.

3.5 Downweighting Schemes

The approach used in the previous sections was to equally downweight each dimension to be dropped by a common value of α. This corresponds to scheme 1 mentioned in Section 2.3.1. This scheme is the most similar to the F-LDA paradigm, but other schemes may provide a way to converge to the best solution more quickly.

Scheme 2 downweights the dimensions associated with the smallest eigenvalues of the weighted between class scatter matrix more than those associated with larger eigenvalues. This scheme downweights eigenvalues based solely on their order. The dimension to be dropped associated with the largest eigenvalue is downweighted by the traditional α (based on r_{max}) while successive dimensions are downweighted via a geometric progression of weights to .5 (to ensure no dimension is completely removed in one step). This scheme is implemented in a similar manner to scheme 1 with a modified scaling matrix (S) that accounts for the larger downweighting factors as the eigenvalues become smaller.

Scheme 3 departs from this theme because the scaling matrices are no longer constant at different shrinkage steps (i.e., one dimension may be nearly ignored in one step and yet become important in a later step due to the shrinking process introducing class confusion). The magnitude of the eigenvalue at a given dimension (after downweighting but *before* the weight measure is applied) is used to determine the size of the scaling (with a suitable lower bound to prevent dimension removal too quickly).

Downweighting Scheme	ccrSAFDA (stop)	time SAFDA (stop)
Common α	0.8922	153.6 s
Order Based	0.8800	146.7 s
Magnitude Based	0.8889	1687.8 s
Downweighting Scheme	ccrSAFDA (full)	timeSAFDA (full)
Common α	0.8889	157.2 s
Order Based	0.8800	156.8 s
Magnitude Based	0.8822	6801.7 s

Table 3.1: CCR and Time for SAFDA Algorithms under Different Downweighting Schemes: AR Database

Table 3.1 shows that the more sophisticated downweighting schemes (2 and 3) offer no improvement in CCR when compared to scheme 1 (common α). In fact, there is a slight degradation in classification but the CCR of SAFDA under any downweighting scheme is still much better than LDA and wLDA with practical equivalence to F-LDA. It is interesting to note that scheme 2 (order based) achieves 'convergence' much more quickly than scheme 1 for the same train/test sets. There is a slight drop in performance but it may be beneficial in cases with a large number of classes (trade-off of correct classification for tractable time).

Scheme 3 as it stands is clearly not feasible in a timely manner. Scheme 3 took even longer than the F-LDA problem because the additional eigenvalue decomposition that is performed at every intermediate step is always in the full dimension $((C-1)\times(C-1))$, whereas the F-LDA decomposition is on a smaller matrix as each dimension is peeled off. There may be a benefit to forming the scaling matrix based on the initial decomposition of the between class scatter matrix and using a common matrix for scaling to remove the need of recomputing at every step, but it is highly improbable that this scheme would be better than the order based nature of scheme 2.

3.6 Performance and Computation Time Comparison for Different Algorithms

3.6.1 Case Study 1: MNIST database with 10 classes

As discussed in Section 3.2, the MNIST data is pre-split into training and test data. Table 3.2 shows the correct classification rates using the given training/test split for the variety of LDA techniques (LDA, wLDA, F-LDA, and SAFDA) using the adaptive Gaussian kernel. It is clear that the SAFDA techniques are slightly superior in terms of error rates to the simpler LDA and wLDA methods while more or less matching the F-LDA procedure.

LDA	wLDA	F-LDA	SAFDA (early stop)	SAFDA (full)
70.260%	73.780%	78.900%	78.730%	79.210%

Table 3.2: CCR for MNIST database for Different Procedures using Gaussian Kernel Weight Function

Run	LDA	wLDA	F-LDA	SAFDA (early stop)	SAFDA (full)
1	57.482	57.422	57.436	57.350	57.459
2	50.082	50.086	50.182	49.991	50.010
3	48.569	48.508	48.601	48.435	48.453
4	49.030	49.040	49.205	48.964	49.030
5	47.474	47.498	47.642	47.408	47.428

Table 3.3: Time (sec) for MNIST database for Different Procedures using Gaussian Kernel Weight Function

Table 3.3 shows the amount of time (in seconds) that the various algorithms take. There is very little difference in the times because of the small number of classes. F-LDA is not too computationally expensive when the number of classes is small as most of the work is performed in $(C-1)$ dimensional space, but we can see that the full SAFDA procedure is slightly better than F-LDA (due to the interdependence of the dimensions). These procedures were run multiple times on the *same* data (same training and test splits) to compare the variability in time due to the computer. Since the data was identical for all five runs, classification results were identical as expected. For the case of a small number of classes, the differences in the algorithm times is actually smaller than the underlying variability in computation time. In fact, some runs of this process had the time for wLDA less than that for LDA (see, e.g., runs 1 and 3) and all runs showed both SAFDA (early stop) and SAFDA (full) converging to a solution more quickly than LDA! This, of course, is not possible if the processes occurred on a machine devoted to this task, as wLDA and SAFDA are enhanced versions of LDA. The only explanation for multiple runs showing wLDA as the quicker procedure is variability in the allocation of computer resources. For

a small C problem, F-LDA and SAFDA are nearly indistinguishable. The advantages are more clear in Section 3.6.2 for an example with a large number of classes.

3.6.2 Case Study 2: AR Database with 100 classes

For a problem involving a large number of classes (e.g., face recognition), the F-LDA algorithm becomes computationally too expensive to be useful in practice, especially when some of the parameters need to be determined via cross validation. For example, non-adaptive weighting functions (i.e., inverse pairwise distance) have additional costs in terms of cross validation. This is evident from the implementation on the AR database with non-adaptive weighting functions in Tables 3.4 and 3.5. Because of the time intensive nature of the F-LDA procedure, a very coarse grid of parameters was examined via cross validation $(h = 4, 6, 8, 10, 12)$, chosen in the 'stable' region of h. When using these weighting functions that require CV, the algorithm will be referred to as SFDA to distinguish the need for cross validation and adaptive parameter searches. Parameter choices were examined in Section 3.3. It is clear that the cross validation adds a great deal of time to the algorithm (even when using a very coarse grid for the parameter choices) as seen in Table 3.5.

Run	LDA	wLDA	F-LDA	SFDA (early stop)	SFDA (full)
1	79.667%	83.667%	87.667%	87.167%	88.833%
2	85.333%	87.833%	89.000%	89.000%	90.333%
3	79.833%	83.833%	88.167%	88.167%	88.333%
4	83.333%	86.000%	89.500%	89.167%	88.667%
5	78.667%	84.667%	89.000%	87.167%	90.000%
6	81.000%	87.833%	87.167%	87.833%	88.833%
7	82.167%	84.667%	88.167%	86.167%	86.500%
8	81.333%	87.333%	87.167%	90.000%	90.000%
9	82.167%	86.500%	88.333%	89.833%	90.000%
10	79.500%	86.167%	87.500%	88.833%	89.167%
Mean	81.300%	85.850%	88.167%	88.333%	89.067%
StdDev	2.0273%	1.5722%	0.80890%	1.2522%	1.1364%

Table 3.4: CCR for 10 random splits of AR database for Different Procedures using Bounded Inverse Pairwise Distance Weighting Function with coarse CV

Run	LDA	wLDA	F-LDA	SFDA (early stop)	SFDA (full)
1	146.56	149.97	4165.2	177.79	203.24
2	145.91	149.19	4158.6	178.31	202.70
3	146.90	149.94	4153.0	177.24	203.36
4	146.41	149.73	4143.8	176.59	202.62
5	146.79	149.77	4151.6	177.90	203.59
6	145.15	148.32	4149.7	175.13	201.47
7	145.63	148.70	4167.4	176.85	202.37
8	148.87	151.55	4153.7	179.70	205.47
9	146.88	149.55	4145.7	178.38	203.56
10	146.54	149.47	4145.6	177.21	202.94
Mean	146.57	149.62	4153.4	177.51	203.13
StdDev	0.99411	0.86569	8.1033	1.2255	1.0405

Table 3.5: Time (sec) for 10 random splits of AR database for Different Procedures using Bounded Inverse Pairwise Distance Weighting Function with coarse CV

Table 3.6 shows the results of the SAFDA procedure compared to LDA and wLDA, while Table 3.7 shows the massive savings in time over F-LDA when using the adaptive Gaussian kernel. These tables show 10 separate random splits of AR data into training and test data with final projection space of 10 dimensions for classification. The parameter, h, for wLDA and F-LDA was chosen to be the minimum pairwise distance of means in the initial configuration, and the parameter for the SAFDA was chosen adaptively at each step to be the minimum of the pairwise mean differences (see discussion in Section 3.3.3). Also, downweighting scheme 1 (common α) was used for these tables. The training/test set split was 17 images per person for training and the remaining 9 for testing which yields 1700 training images and 900 test images. The full run of SAFDA and F-LDA involves 30 shrinkage steps ($r_{max} = 30$) and the early stop permits the algorithm to terminate when data suggests the best rotation has been achieved. Details on the early stopping criteria were given in Section 3.4.

Run	LDA	wLDA	F-LDA	SAFDA (early stop)	SAFDA (full)
1	80.667%	85.556%	87.222%	89.222%	88.889%
2	84.667%	87.333%	88.444%	89.667%	89.889%
3	80.222%	85.111%	88.556%	88.111%	88.333%
4	82.889%	85.778%	88.111%	87.667%	88.667%
5	79.111%	86.000%	87.444%	87.889%	88.333%
6	81.111%	86.778%	86.556%	89.444%	89.556%
7	81.556%	86.000%	86.778%	88.000%	88.556%
8	81.889%	87.444%	87.333%	90.444%	90.556%
9	81.556%	85.667%	87.444%	88.111%	88.333%
10	80.333%	86.222%	86.333%	88.444%	88.556%
Mean	81.400%	86.189%	87.422%	88.700%	88.967%
StdDev	1.5478%	0.76811%	0.76048%	0.92970%	0.77167%

Table 3.6: CCR for 10 random splits of AR database for Different Procedures using Gaussian Kernel

Run	LDA	wLDA	F-LDA	SAFDA (early stop)	SAFDA (full)
1	146.98	147.61	976.00	155.11	158.56
2	146.15	146.67	973.66	154.52	157.60
3	144.97	145.33	974.20	153.07	156.52
4	146.01	146.31	967.10	153.85	157.41
5	146.24	146.70	970.77	154.69	157.86
6	146.53	146.98	971.84	154.47	157.91
7	147.47	147.73	973.98	155.75	158.87
8	146.84	147.17	969.75	155.59	158.28
9	147.53	147.85	972.33	156.24	159.05
10	146.88	147.23	972.73	155.18	158.37
Mean	146.56	146.96	972.24	154.85	158.04
StdDev	0.76015	0.75615	2.5443	0.93761	0.75198

Table 3.7: Time (sec) for 10 random splits of AR database for Different Procedures using Gaussian Kernel

Classification results are not substantially different between the two kernels as shown in Tables 3.4 and 3.6, but the non-adaptive weighting function's need of cross validation takes an excessive amount of time when compared to the SAFDA's adaptive procedure. The real power of the SAFDA algorithm is shown in Table 3.7. The SAFDA procedure is clearly much more computationally efficient than the F-LDA algorithm. While there is a significant amount of time/effort in preprocessing (approximately 145 seconds), the additional cost of F-LDA (sequential removal) is significant. Standard LDA is very quick as is wLDA once the parameter is chosen for the Gaussian kernel. Since there are 100 classes in the AR database, all post-preprocessing is performed on 99 dimensional matrices. The SAFDA algorithm does indeed take longer than LDA and wLDA, but the additional cost is minimal (especially considering the gain in CCR–8 seconds for approximately 7.5% bump in correct classification rate over LDA). The time for F-LDA is well beyond that of the

standard techniques. LDA only takes about 1 second beyond preprocessing whereas, F-LDA, on average, takes about 827 seconds for this dataset. In contrast, SAFDA matches or betters the F-LDA CCR but only takes approximately 8 seconds after preprocessing. This difference is exacerbated significantly if cross-validation needs to be done for a different kernel (i.e., inverse pairwise distance) as the entire process needs to be replicated many times to find the optimal h as seen in Tables 3.4 and 3.5. In addition, if more classes are in the database (realistic scenarios in face recognition would have much more than 100 subjects in the database), the extra costs for F-LDA are even larger. While larger databases contribute to longer time for training for LDA, wLDA, and SAFDA, the extra time is proportional to the algorithmic complexity (much smaller than that of F-LDA) rather than the number of classes or the number of dimensions.

3.6.3 Configurations of Projected Centroids through the Fractional Shrinking Iterations and Confusion Matrices

Case Study 1: MNIST data

First, the MNIST data was projected into $d = 2$ dimensional space to observe the effects on the class means of the SAFDA procedure as the shrinkage steps occurred. Figure 3.40 gives the optimal projection via the wLDA procedure ($r = 1$ in the SAFDA procedure) and Figure 3.41 shows the class centroids projected into the optimal space learned via SAFDA. Clearly, the SAFDA algorithm has induced a much more separated space to improve classification. The labelling scheme for various classes is consistent in all plots so the legends are only shown in these first two to save space.

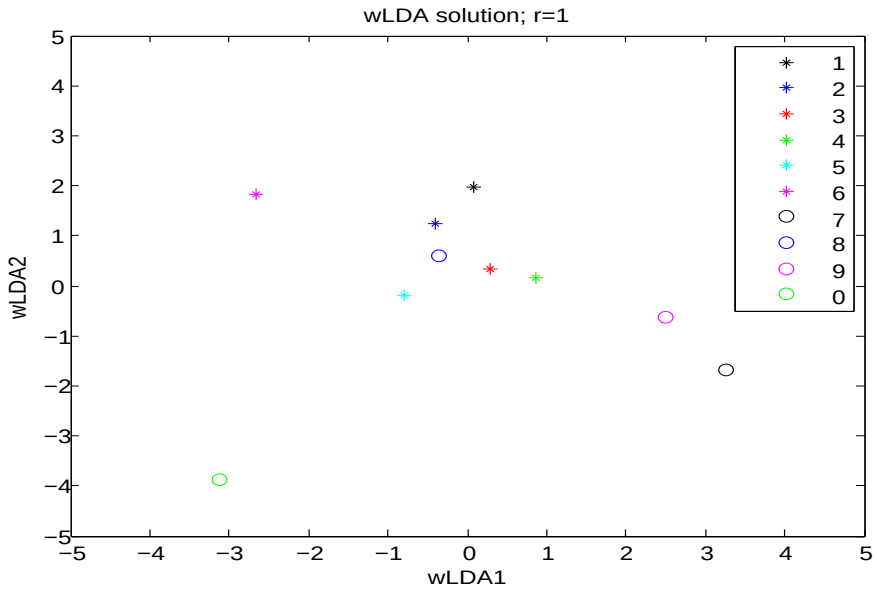

Figure 3.40: MNIST Class centroids projected into optimal wLDA space; d=2

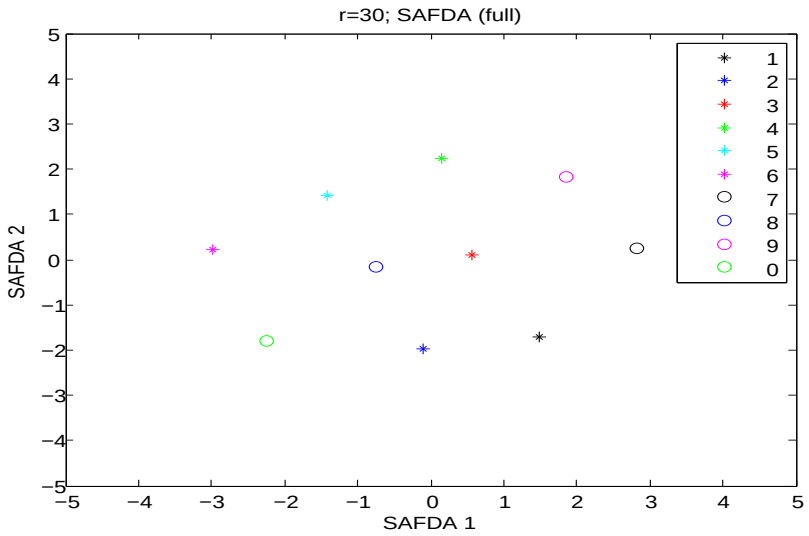

Figure 3.41: MNIST Class centroids projected into optimal SAFDA space; d=2

Figure 3.42 shows snapshots of the 'best' projection direction across a range of r shrinkage steps. An .avi movie of the progression of means is also presented on a DVD alongside the dissertation. For a two-dimensional projection, the early stopping criterion ends at $r = 16$. It is clear from Figure 3.42 that the mean separation does not improve substantially after the 16^{th} shrinkage step. More importantly, the wLDA solution given at $r = 1$ shows significant overlap between classes 1 and 2 as well as 5 and 8.

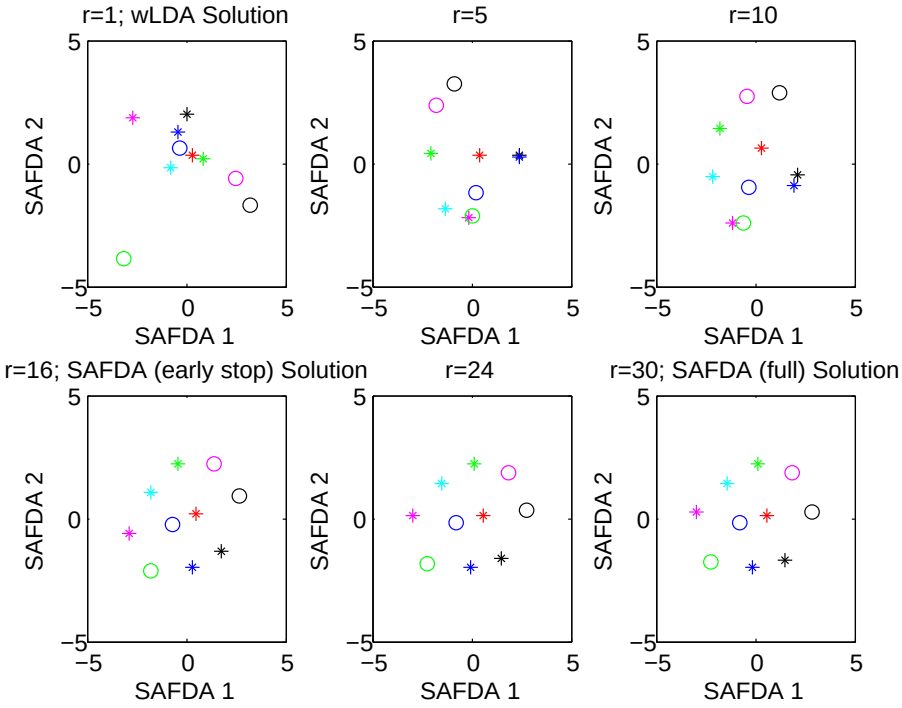

Figure 3.42: MNIST Class centroids progression across shrinkage steps; d=2

From these plots, we can see that the SAFDA procedure finds a much better *two-dimensional* space in which the means are separated. The first subplot (upper left) shows

the optimal wLDA two-dimensional projection which is achieved in the first step (no shrinkage). The progression of the SAFDA clearly allows the learned space to *adapt* to the data in such a way that the means are quite separated in the projected two-dimensional space. This can be thought of as a space-filling design of the maximin variety (maximize the minimum distance between class centroids).

Two-dimensional projections ($d = 2$) above were used only for a visual progression of the means graphically. All of the previous sections focus on a four-dimensional projection so the remainder of this subsection will focus on $d = 4$. The MNIST database yielded the following confusion matrices (Tables 3.8 and 3.9) when the Optimal LDA and SAFDA algorithms were used to reduce the dimension to $d = 4$ respectively.

	Predicted Class									
Truth	1	2	3	4	5	6	7	8	9	0
1	91%	5%	2%	0%	0%	0%	0%	1%	0%	0%
2	30%	50%	8%	1%	1%	5%	1%	2%	1%	1%
3	5%	14%	43%	3%	5%	0%	2%	23%	3%	1%
4	0%	1%	6%	77%	4%	1%	0%	1%	10%	0%
5	1%	2%	6%	8%	68%	2%	1%	9%	3%	1%
6	0%	3%	0%	3%	5%	86%	0%	1%	0%	2%
7	3%	3%	2%	1%	0%	0%	78%	0%	13%	0%
8	3%	4%	17%	6%	23%	1%	1%	41%	3%	1%
9	0%	1%	5%	11%	1%	0%	9%	0%	73%	0%
0	0%	0%	1%	0%	2%	1%	0%	0%	0%	94%

Table 3.8: MNIST Confusion Matrix under Optimal LDA for d=4

	Predicted Class									
Truth	1	2	3	4	5	6	7	8	9	0
1	96%	1%	0%	0%	0%	0%	0%	2%	0%	0%
2	13%	67%	5%	1%	1%	3%	3%	3%	1%	3%
3	2%	10%	75%	2%	5%	0%	2%	2%	1%	1%
4	1%	0%	1%	78%	2%	1%	1%	2%	13%	0%
5	1%	2%	5%	7%	72%	2%	1%	7%	1%	3%
6	1%	1%	0%	3%	3%	81%	1%	5%	0%	6%
7	3%	3%	1%	2%	0%	0%	80%	0%	9%	0%
8	6%	1%	2%	3%	10%	2%	0%	71%	3%	0%
9	0%	0%	1%	10%	1%	1%	5%	1%	79%	0%
0	0%	1%	2%	0%	2%	3%	1%	0%	0%	91%

Table 3.9: MNIST Confusion Matrix under SAFDA for d=4

We can see common classification errors under the LDA projection confusing Classes 4 and 9 as well as Classes 3, 5, and 8. Class 2 points are also often misclassified as Class

1 (30%) and Class 7 as class 9. All of these errors make sense intuitively because of the similarity of the shapes of these numbers. Class 0 is the farthest outlier class in the original data as shown in the optimal LDA projection in Figure 3.43. Class 1 most likely has a high CCR (even though its class centroid is near many others) due to a low inherent variability (difficult to draw the number one in many different fashions), whereas the class centroid for Class 2 lies very nearby leading to a high percentage of Class 2 being misclassifed as Class 1. There is an interesting caveat to the SAFDA procedure that, while most classes yield an improved classification rate, a few classes' performances are degraded under SAFDA. This can be seen in the MNIST data for Classes 0 and 6 (performance was slightly better under LDA). This is because Classes 0 and 6 were outlier classes in the original space and, therefore, easy to distinguish. While they drove the solution of the one-pass method, SAFDA focused more on the confused classes and fit the data to better separate those classes. This introduced some additional confusion in the outlier classes, but the trade-off in overall performance was worth the slight loss in CCR in those two classes.

Comparing Figures 3.43 and 3.44 shows why the four-dimensional projection of SAFDA outperforms the LDA projection. We can see that any classes that are near each other in the first two dimensions of SAFDA are always reasonable separated in dimensions 3 and 4. In the LDA projection, Classes 1 and 2 as well as Classes 3 and 8 are near each other in both plots. SAFDA enhances the LDA procedure by finding projections that separate those easily confused classes.

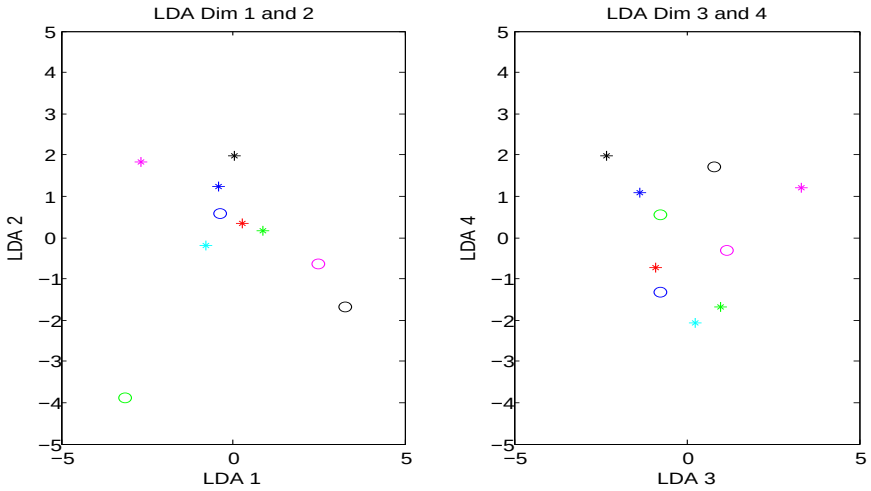

Figure 3.43: MNIST Class centroids Optimal LDA space; d=4

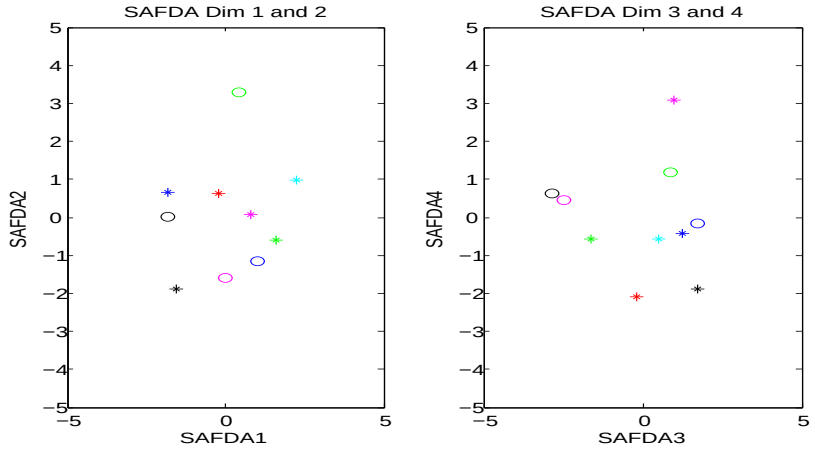

Figure 3.44: MNIST Class centroids Optimal SAFDA space; d=4

Figure 3.45 shows the first two dimensions of the space learned via the SAFDA procedure for the MNIST data. The stopping criterion was triggered slightly earlier ($r = 12$ for $d = 4$ versus $r = 16$ for $d = 2$), most likely due to the comparative ease in separating classes in a larger dimension. It is important to remember that only the progression of the first two SAFDA dimensions is shown in Figure 3.45. The centroids are closer after the final step in SAFDA when compared to LDA *only in the first two dimensions*. Visualizing the centroid progression in four-dimensional space through a 'G-Gobi'-type tool would show the centroids to be much more clearly separated (space filling) as in Figure 3.42.

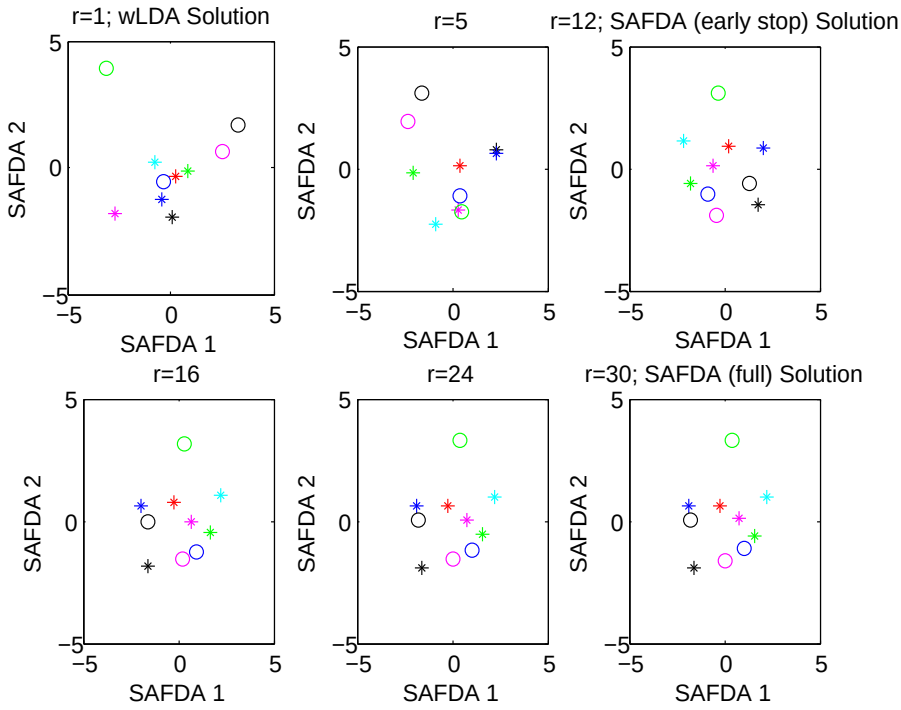

Figure 3.45: MNIST Class centroids progression (first two dimensions shown) across shrinkage steps; d=4

Case 2: AR database

For the AR data with 100 classes, the confusion matrix is displayed in the form of a heat plot. Also, as the number of test points is low (7 per person), 10 separate training/test set splits were run separately to simulate the confusion matrix for a larger test set. This was done to get an idea of common misclassifications, if any do occur.

First, we present the ECDF's of the CCRs for the 100 classes for LDA, wLDA, SAFDA (full), and SAFDA (early stop) in Figure 3.46. Clearly, SAFDA outperforms the one-pass methods (more CCRs near 1). Also, SAFDA (full) and SAFDA (early stop) exhibit almost identical behavior.

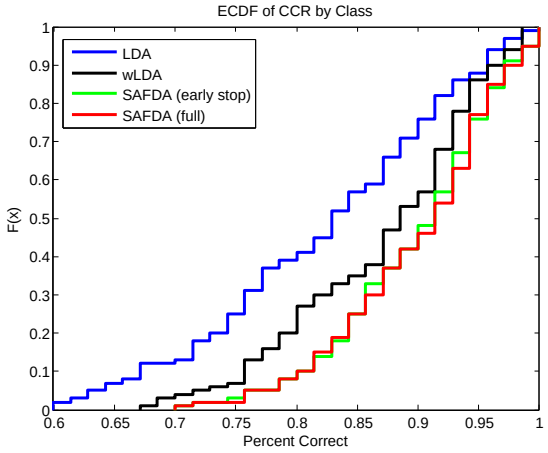

Figure 3.46: ECDF of AR CCR by Class, $d = 10$

Figure 3.47 shows the results of Optimal LDA classification in 10 dimensions when compared to SAFDA on the same data. Clearly, the SAFDA procedure outperforms the LDA technique as seen by the fewer off diagonal errors in the SAFDA confusion matrix. The additional lines drawn on the matrix were drawn to demarcate the males and females.

The males were subjects 1-50 and the females were 51-100. Common misclassification errors occur within the same gender because most incorrectly identified individuals are predicted as the same gender (upper-left and lower-right).

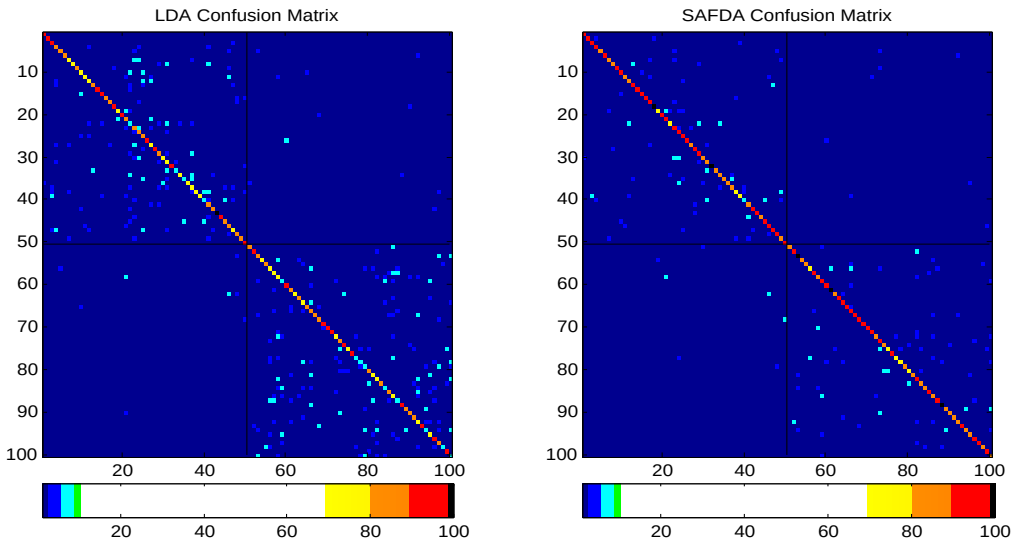

Figure 3.47: AR–LDA vs. SAFDA d=10

The same technique was used to compare the full SAFDA procedure to the SAFDA procedure with early termination. Figure 3.48 shows that there is little appreciable difference between the two, suggesting the early stopping criterion could be used (at least on this data) with little detriment to CCR.

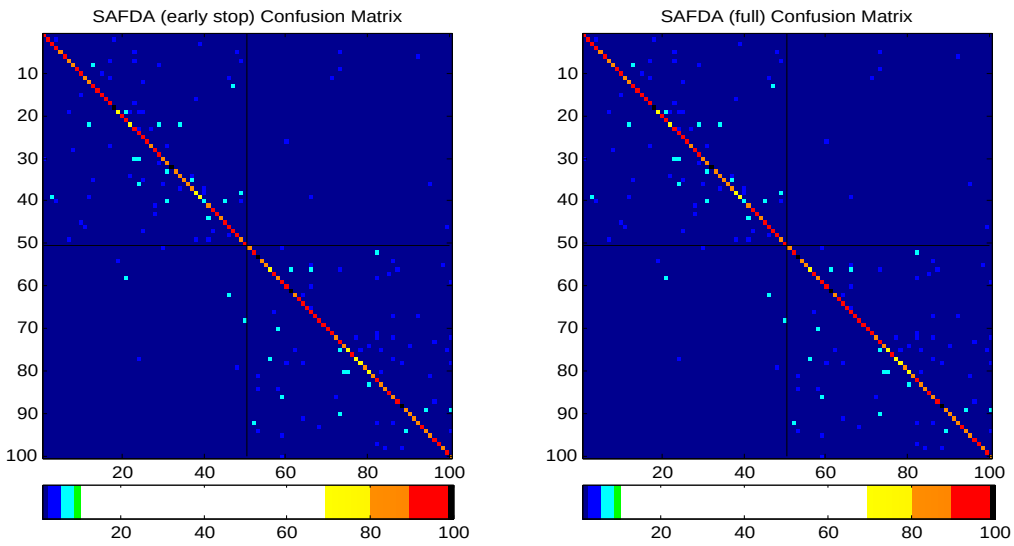

Figure 3.48: AR–SAFDA (full) vs. SAFDA (stop) d=10

3.6.4 Discussion

While the Gaussian kernel uses an adaptive parameter selection, the bounded pairwise inverse distance kernel has to select the tuning parameter via cross validation. Results shown in Tables 3.4 and 3.5 demonstrate SAFDA's efficacy when classification rate is compared to standard LDA techniques and the time savings when compared to F-LDA. The SAFDA procedure dominates the LDA and wLDA procedures and, on average, even defeats the much more computationally intensive F-LDA procedure. When the SAFDA (early stop) procedure does not beat the F-LDA procedure in CCR (i.e., run 4), it is not significantly different in performance. In fact, the full run always outperformed F-LDA for all test runs in our research. On average, the SAFDA is a nice improvement over the other techniques.

As we can see, there is a *huge* savings in time when using the bounded inverse distance measure where cross-validation is used. There is also a moderate time savings (that becomes more apparent as C increases) when using the adaptive Gaussian kernel. The results of the different kernels show no appreciable difference under classification performance across the different splits in the AR database but there is a sizable time savings of SAFDA over F-LDA (more so under the non-adaptive CV kernel). The CCR of the LDA procedure shows nearly equivalent results under both the adaptive and non-adaptive kernel selection (81.4% with the Gaussian vs. 81.3% with the bounded inverse pairwise distance kernel) as expected because there is no weighting in traditional LDA. This small difference in mean CCR is likely due to random train/test splits in LDA, while the choice of weight function can affect classification results in the weighted LDA variants (i.e., wLDA, F-LDA, SAFDA). Overall, the Gaussian kernel seems to be a better choice based on time and performance.

Chapter 4: Contributions and Future Work

4.1 Discussion and Conclusion

The SAFDA procedure was developed to incorporate projection pursuit ideas into a linear discriminant framework while yielding a directed search algorithm that seeks to maximize class separation in the reduced space for classification. This directed search is quite useful as a time saver for large-dimensional problems, as the random search used in standard projection pursuit (PP) techniques can take a great deal of computational effort before reaching a stable solution. While PP seeks interesting directions (in our case, directions which maximize class separation) in the reduced space via a random search, shrinking dimensions of relatively low importance by larger and larger amounts allows the procedure to adapt itself to the resulting data space, while seeking to maximize traditional (weighted) LDA criteria. PP ideas are slowly being merged into supervised learning methods, and the SAFDA procedure is a testament to its utility in this field.

SAFDA seeks to find a low-dimensional space in which the class centroids essentially 'fill the space'. The underlying motivation for the technique was to make the fractional LDA (F-LDA) procedure (Lotlikar and Kothari, 2000) feasible in a large-dimensional framework with a large number of classes. SAFDA not only substantially improves upon the time requirements of F-LDA, but in some cases, also serves to slightly improve class

separation by simultaneously contracting/compressing dimensions, thereby exploiting the interrelationships of the many features of the data (a major aspect of image data).

As well as introducing a directed search to maximize the LDA criterion, SAFDA improves upon classification rates of many standard linear techniques through a weighted framework that focuses the search for discriminative directions on classes that are the most confused. While F-LDA suggests using a simple weight function of inverse pairwise distances, cross validation must be performed to find the best parameter for the given weight function. SAFDA introduced the Gaussian kernel in which the best parameters can be learned from the data with simple rules that seem to permit an *adaptive* scheme in which the weight function (kernel) adapts to the scale of the data at any given step of the algorithm, when the data lie in a 'fractional dimension' (somewhere between the full p dimensional space and the final d dimensional space). While SAFDA can use any weighting function that is a monotone decreasing function of pairwise distances between class centroids, the Gaussian Kernel based weight function seems to be the most promising in both speed (adaptive parameter) and classification performance.

The number of shrinkage iterations within each dimension removal was fixed at r_{max} (specified *a priori*) in F-LDA, which may lead to performing additional extraneous shrinkage steps after a stable solution has already been reached. A heuristic threshold was introduced on the Hellinger distances of successive weight vectors to determine whether convergence had been achieved. When the Hellinger distance dropped below the given threshold, the algorithm terminated (prior to completion of the prespecified r_{max} iterations) to prevent using additional time that does not necessarily improve the algorithm's performance markedly. There is a chance that this stopping point may be a local minimum in the search

process and force a premature stop to the directed search algorithm, but the threshold was chosen low enough to prevent this in most situations.

Various downweighting schemes were also examined to see if a more data dependent scheme (downweighting based on the eigenvalue magnitudes or sizes within a 'fractional' or 'shrunken' dimension) would yield a quicker convergence to the best class separation in the reduced dimension, but the more complex schemes did not yield markedly better results than shrinking each dimension by a constant rate. The constant α downweighting function for all dropped dimensions is suggested as it was the simplest and easiest to understand/relate to the F-LDA methodology.

The results on the MNIST and AR datasets suggest that the SAFDA procedure can successfully be implemented in image analysis problems. The time requirement of SAFDA is substantially lower than that of F-LDA and other LDA variants that require cross-validation while still yielding similar classification results. As the dimension of the data grows (larger or higher resolution images) and/or the number of classes becomes large (common occurrence in large face databases), the computational cost of SAFDA in terms of time is manageable after the initial whitening step, which is also needed in F-LDA. The time required for F-LDA was dependent on the number of iterations within each dimension reduction step as well as the number of dimensions to remove, $(p-d)$, because each dropped dimension was determined sequentially.

While the focus of SAFDA in this work is centered on the face recognition/image classification problem, there are no specific aspects of the algorithm that limit its use to these applications. The SAFDA procedure can be applied to any discrimination problem, but the obvious time intensive nature of high-dimensional images makes this an ideal application area for this methodology. Results on the MNIST digit recognition and AR face databases

show that the SAFDA procedure yields comparable results to F-LDA while dramatically reducing the time commitment. For these case studies, SAFDA dominates standard LDA as well as wLDA with only slightly more computational effort, due to its goal of maximizing class centroid separability *in the reduced space*.

The results of this work demonstrate that the SAFDA procedure shows great promise in improving classification rates of high-dimensional data in a large multi-class framework, while still allowing the dimension reduction to be performed in a relatively short amount of time. By utilizing the sequential shrinkage of the dimensions to be dropped while *simultaneously* shrinking all dimensions that will eventually be removed from consideration, the algorithm finds space filling projections of class centroids. This can be achieved much more quickly than the sequential dimension reduction employed by F-LDA. The simultaneous shrinking also permits the interdependence of the features to be incorporated into the search. F-LDA ignores such interdependence by removing dimensions one at a time.

Overall, the SAFDA procedure's performance is quite promising for both situations with a large and small number of classes. It provides an easy to use, tractable, and directed projection pursuit procedure for dimension reduction that produces very good classification results.

4.2 Future Work

Most of the additional work on the SAFDA algorithm lies in six directions:

1. Improving the directed search algorithm.

2. Removing dimensions in multiple 'chunks' (rather than dropping all dimensions at once) based on the eigenvalues of the data

3. Exploring SAFDA based dimension reduction as input vectors to SVM methodology.

4. Comparing SAFDA with boundary based techniques (time and accuracy) like SVM.

5. Extracting contextual information from the final space (especially important in the analysis of face recognition).

6. Develop a theoretical framework for this procedure based on a minimax criterion for pairwise distances.

Item 1 can possibly be approached by examining the effect of recomputing the within class scatter matrix throughout the algorithm (no whitening) at every substep. Obviously, the additional computation of the within class scatter matrix at every 'fractional dimension', as well as inverting it, leads to a large increase in calculation time. The additional computation time would probably not be worth a potential increase in classification rate, however, improvement in correct classification rate (CCR) may be possible in some situations.

Also involved with improving the algorithm could be the search for a more suitable weighting function (kernel). It is well known that no weight function (kernel) is optimal for the general classification problem. This particular procedure focuses on the Gaussian kernel due to its pervasiveness in the literature, strong properties, adaptive parameter potential, and pure dependence on pairwise distances. While this kernel is very suited to our goals, there may be other similarity measures that can be used in certain databases that could improve classification. Kernel selection is key in any classification problem, so this is a very general topic for research, and it can be useful in the SAFDA framework as well.

The algorithm as stated only suggests a one-stage reduction from p dimensions to d, but there is no restriction that the SAFDA procedure must remove all dimensions at once. A data determined step size can be determined via the percent of variability measured by the

eigenvalues to allow for a multistage process that removes multiple dimensions at each step. For example, the user can choose *a priori* to drop dimensions that explain the lowest k% of the total variability in the data at each stage. This procedure allows for the time savings via SAFDA as multiple dimensions are removed at a time, but it prevents a dimension from being shrunken too quickly.

Often, dimension reduction techniques are not applied singularly. A common technique when dealing with large p small n problems and attempting to apply LDA-like methods is to first reduce the very large dimension p to a dimension in which the S_W and S_B are both full rank using another technique (i.e., PCA). However, combining SAFDA with PCA is not conducive to better classification. PCA, when used in conjunction with LDA, is simply an unsupervised learning procedure which can detract from the SAFDA mission that is built to seek out dimensions of high class-separability in a time efficient manner. SAFDA can also be used to first find a lower dimensional space in which the centroids are 'optimally' separated before applying a more sophisticated boundary based technique in the lower dimensional space such as SVM (SAFDA+SVM).

This work focuses on comparing SAFDA with other LDA variants, more specifically, mimicking the F-LDA results while speeding up the process. Boundary based kernel techniques (e.g., SVM) are quite popular in classification literature but have some downsides in interpretation when compared to linear techniques. However, SVM often produces better classification results than linear methods because of its 'black-box' methods. Comparing the performance of the SAFDA procedure to SVM on datasets with a small number of classes could yield insights into whether the SAFDA procedure can improve linear techniques to be on par (from a classification standpoint) with boundary based methods. While the real benefits of SAFDA are apparent on classification problems with a large number of

classes, SVM has not been examined in depth for these situations. Comparisons between the techniques are restricted to a lower C problem until MSVM techniques are extended to the large C problem.

Item 5 is focused on the *interpretability* of the features selected in the final step under the SAFDA paradigm. This is of special importance in the face recognition problem when one wants to learn facial features that are the most crucial for face identification. One could use sparse SVD techniques to achieve a model that only selects a few features or examine the projection matrix to determine which combination of features was indicated as important via SAFDA. By understanding which features of the face itself are leading to the highest levels of discrimination, we can more closely adapt this procedure (and others) to fit under general conditions (nonaligned images, different camera, different resolution, etc.). The features selected by SAFDA based on a given database are numerically tied to the data collection system (i.e., orientation, lighting, background) unless these extrapersonal variants are controlled for in preprocessing stages. The interpretability of linear systems can be exploited to generalize which pixels, groups of pixels, or facial features are the most discriminative when determining identity.

F-LDA (and SAFDA by extension) were based on some intuitive techniques that yielded a directed search to a lower dimensional space in which the class centroids are separated. A final goal for the future is to develop a theoretical framework in which this procedure can be housed. Preliminary results suggest that SAFDA finds a projection that yields some 'maximin' space-filling configuration of the class centroids. Perhaps, a cost function based on this maximin criterion can be used to further streamline the process, as well as explain the improvement in separation via statistical theory. While SAFDA is based in an applied

ideal, a better theoretical understanding can yield new insight into its inner workings and lead to more sophisticated improvements to this procedure.

Bibliography

Abdi, H. and Williams, L. (2010). "Principal component analysis." *Wiley Interdisciplinary reviews: Computational Statistics*, 2, 4, 433–459.

Barker, M. and Rayens, W. (2003). "Partial least squares for discrimination." *Journal of Chemometrics*, 17, 3, 166–173.

Bavaud, F. (2010). "On the Schoenberg Transformations in Data Analysis: Theory and Illustrations." *Arxiv preprint arXiv:1004.0089*.

Belhumeur, P., Hespanha, J., Kriegman, D., et al. (1997). "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." *IEEE Transactions on pattern analysis and machine intelligence*, 19, 7, 711–720.

Belkin, M. and Niyogi, P. (2003). "Laplacian eigenmaps for dimensionality reduction and data representation." *Neural computation*, 15, 6, 1373–1396.

Bensmail, H. and Celeux, G. (1996). "Regularized Gaussian Discriminant Analysis through Eigenvalue Decomposition." *Journal of the American statistical Association*, 91, 436.

Beveridge, J., She, K., Draper, B., and Givens, G. (2001). "A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition."

In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE Computer Society; 1999.

Bickel, P. and Levina, E. (2004). "Some theory for Fisher's linear discriminant function,naive Bayes', and some alternatives when there are many more variables than observations." *Bernoulli*, 10, 6, 989–1010.

— (2008). "Regularized estimation of large covariance matrices." *The Annals of Statistics*, 36, 1, 199–227.

Breiman, L. (1984). *Classification and regression trees*. Chapman & Hall/CRC.

— (2001). "Random forests." *Machine learning*, 45, 1, 5–32.

Bruce, V. and Young, A. (1986). "Understanding face recognition." *British journal of psychology*, 77, 3, 305–327.

Brunelli, R. and Poggio, T. (1993). "Face recognition: Features versus templates." *IEEE transactions on pattern analysis and machine intelligence*, 15, 10, 1042–1052.

Cai, D., He, X., Han, J., and Zhang, H. (2006). "Orthogonal laplacianfaces for face recognition." *Image Processing, IEEE Transactions on*, 15, 11, 3608–3614.

Cevikalp, H., Neamtu, M., Wilkes, M., and Barkana, A. (2005). "Discriminative common vectors for face recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4–13.

Chen, L., Liao, H., Ko, M., Lin, J., and Yu, G. (2000). "A new LDA-based face recognition system which can solve the small sample size problem." *Pattern recognition*, 33, 10, 1713–1726.

Chen, S., Liu, J., and Zhou, Z. (2004). "Making FLDA applicable to face recognition with one sample per person." *Pattern recognition*, 37, 7, 1553–1555.

Chien, J. and Wu, C. (2002). "Discriminant waveletfaces and nearest feature classifiers for face recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1644–1649.

Chung, F. (1997). *Spectral graph theory*. No. 92. Amer Mathematical Society.

Comon, P. et al. (1994). "Independent component analysis, a new concept?" *Signal processing*, 36, 3, 287–314.

Croux, C., Filzmoser, P., and Joossens, K. (2008). "Classification efficiencies for robust linear discriminant analysis." *Statistica Sinica*, 18, 2, 581–599.

Dai, G., Yeung, D., and Qian, Y. (2007). "Face recognition using a kernel fractional-step discriminant analysis algorithm." *Pattern recognition*, 40, 1, 229–243.

Dayal, B. and MacGregor, J. (1997). "Improved PLS algorithms." *Journal of chemometrics*, 11, 1, 73–85.

De Ridder, D. and Duin, R. (2002). "Locally linear embedding for classification." *Pattern Recognition Group, Dept. of Imaging Science & Technology, Delft University of Technology, Delft, The Netherlands, Tech. Rep. PH-2002-01*.

Duda, R., Hart, P., and Stork, D. (2001). *Pattern classification*. Citeseer.

Etemad, K. and Chellappa, R. (1997). "Discriminant analysis for recognition of human face images." In *Audio-and video-based biometric person authentication: first International*

Conference, AVBPA'97, Crans-Montana, Switzerland, March 12-14, 1997: proceedings, 127. Springer Verlag.

Fisher, R. (1936). "The use of multiple measures in taxonomic problems." *Ann. Eugenics*, 7, 179–188.

Friedman, J. (1989). "Regularized discriminant analysis." *Journal of the American statistical association*, 84, 405, 165–175.

Friedman, J. and Tukey, J. (1974). "A projection pursuit algorithm for exploratory data analysis." *Computers, IEEE Transactions on*, 100, 9, 881–890.

Gao, H. and Davis, J. (2006). "Why Direct LDA is not Equivalent to LDA." *Pattern Recognition*, 39, 5, 1002–1006.

Gribonval, R. (2005). "From projection pursuit and CART to adaptive discriminant analysis?" *Neural Networks, IEEE Transactions on*, 16, 3, 522–532.

Hamsici, O. and Martinez, A. (2008). "Bayes optimality in linear discriminant analysis." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30, 4, 647.

Hastie, T., Buja, A., and Tibshirani, R. (1995). "Penalized discriminant analysis." *The Annals of Statistics*, 23, 1, 73–102.

Hastie, T. and Tibshirani, R. (1996). "Discriminant analysis by Gaussian mixtures." *Journal of the Royal Statistical Society. Series B (Methodological)*, 58, 1, 155–176.

Hastie, T., Tibshirani, R., and Buja, A. (1994). "Flexible Discriminant Analysis by Optimal Scoring." *Journal of the American statistical association*, 89, 428.

Hastie, T., Tibshirani, R., and Friedman, J. J. H. (2001). *The elements of statistical learning*. Springer.

He, X., Yan, S., Hu, Y., Niyogi, P., and Zhang, H. (2005). "Face recognition using laplacianfaces." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 328–340.

Hong, L. and Jain, A. (1998). "Integrating faces and fingerprints for personal identification." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 12, 1295–1306.

Huang, G., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). "Labeled faces in the wild: A database for studying face recognition in unconstrained environments." *University of Massachusetts, Amherst, Technical Report 07*, 49, 1.

Huang, Y., Liu, Q., and Metaxas, D. (2011). "A Component-Based Framework for Generalized Face Alignment." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41, 1, 287–298.

Hyvarinen, A. and Oja, E. (2000). "Independent component analysis: a tutorial." *Neural Networks*, 13, 4-5, 411–430.

Jain, A., Murty, M., and Flynn, P. (1999). "Data clustering: a review." *ACM computing surveys (CSUR)*, 31, 3, 264–323.

Jain, A., Ross, A., Pankanti, S., et al. (2006). "Biometrics: a tool for information security." *IEEE transactions on information forensics and security*, 1, 2, 125–143.

Johnson, R. and Wichern, D. (1988). "Multivariate statistics, a practical approach."

Kouropteva, O., Okun, O., and Pietikäinen, M. (2003). "Classification of handwritten digits using supervised locally linear embedding algorithm and support vector machine." In *Proc. 11th European Symp. Artificial Neural Networks*, 229–234. Citeseer.

Laud, P., Berliner, L., and Goel, P. (1992). "A stochastic probing algorithm for global optimization." *Journal of Global Optimization*, 2, 2, 209–224.

Law, M. and Jain, A. (2006). "Incremental nonlinear dimensionality reduction by manifold learning." *IEEE transactions on pattern analysis and machine intelligence*, 28, 3, 377–391.

LeCun, Y. and Cortes, C. (1998). "The MNIST database of handwritten digits."

Lee, E. and Cook, D. (2010). "A projection pursuit index for large p small n data." *Statistics and Computing*, 20, 3, 381–392.

Lee, E., Cook, D., Klinke, S., and Lumley, T. (2005). "Projection pursuit for exploratory supervised classification." *Journal of Computational and Graphical Statistics*, 14, 4, 831–846.

Lee, Y., Lin, Y., and Wahba, G. (2004). "Multicategory support vector machines." *Journal of the American Statistical Association*, 99, 465, 67–81.

Leino, A. (2004). "Independent Component Analysis: An Overview." *26th April*.

Leroy, B., Herlin, I., and Cohen, L. (1996). "Face identification by deformation measure." In *INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION*, vol. 13, 633–637. Citeseer.

Li, S. and Lu, J. (1999). "Face recognition using the nearest feature line method." *IEEE Transactions on Neural Networks*.

Liu, Y., Schmidt, K., Cohn, J., and Mitra, S. (2003). "Facial asymmetry quantification for expression invariant human identification." *Computer Vision and Image Understanding*, 91, 1-2, 138–159.

Loog, M., Duin, R., and Haeb-Umbach, R. (2001). "Multiclass linear dimension reduction by weighted pairwise Fisher criteria." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 7, 762–766.

Lotlikar, R. and Kothari, R. (2000). "Fractional-step dimensionality reduction." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 623–627.

Lowe, D. (1999). "Object recognition from local scale-invariant features." In *International Conference on Computer Vision*, vol. 2, 1150–1157.

Lu, J., Plataniotis, K., and Venetsanopoulos, A. (2003a). "Boosting linear discriminant analysis for face recognition." In *Proceedings of the IEEE International Conference on Image Processing*, 657–660. Citeseer.

— (2003b). "Face recognition using LDA-based algorithms." *IEEE Transactions on Neural Networks*, 14, 1, 195–200.

Lu, J. and Tan, Y. (2010). "Regularized locality preserving projections and its extensions for face recognition." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40, 3, 958–963.

Lu, X., Jain, A., and Colbry, D. (2006). "Matching 2.5 D face scans to 3D models." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 1, 31–43.

Lu, X., Wang, Y., and Jain, A. (2003c). "Combining classifiers for face recognition." In *IEEE International Conference on Multimedia & Expo*, vol. 3, 13–16. Citeseer.

Mardia, K., Kent, J., Bibby, J., et al. (1979). *Multivariate analysis*. Academic press London.

Martinez, A. (1998). "The AR face database." *CVC Technical Report*, 24.

Martínez, A. (2002). "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class." *IEEE Transactions on Pattern analysis and machine intelligence*, 748–763.

Martinez, A. and Kak, A. (2001). "Pca versus lda." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 2, 228–233.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Muller, K. (1999). "Fisher discriminant analysis with kernels." *Neural networks for signal processing IX*, 41–48.

Mitra, S., Lazar, N., and Liu, Y. (2007). "Understanding the role of facial asymmetry in human face identification." *Statistics and Computing*, 17, 1, 57–70.

Mitra, S., Savvides, M., and Brockwell, A. (2006). "The Role of Statistical Models in Biometric Authentication." *Lecture notes in computer science*, 3832, 581.

Mitra, S., Savvides, M., and Kumar, B. (2005). "Facial Asymmetry: A New Robust Biometric in the Frequency Domain." *Lecture notes in computer science*, 3656, 1065.

Moghaddam, B., Wahid, W., and Pentland, A. (1998). "Beyond eigenfaces: Probabilistic matching for face recognition." In *Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998. Proceedings*, 30–35.

Murphy, K. (2006). "Naive Bayes classifiers." Tech. rep., Technical Report, October.

Niu, B., Yang, Q., Shiu, S., and Pal, S. (2008). "Two-dimensional Laplacianfaces method for face recognition." *Pattern Recognition*, 41, 10, 3237–3243.

Niyogi, X. (2004). "Locality preserving projections." In *Advances in neural information processing systems 16: proceedings of the 2003 conference*, vol. 16, 153. The MIT Press.

Pamplona, S., Silva, L., Bellon, O., and Queirolo, C. (2010). "Automatic face segmentation and facial landmark detection in range images." *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society*, 40, 5, 1319.

Pan, Y., Ge, S., and Al Mamun, A. (2009). "Weighted locally linear embedding for dimension reduction." *Pattern Recognition*, 42, 5, 798–811.

Polzehl, J. (1995). "Projection pursuit discriminant analysis* 1." *Computational statistics & data analysis*, 20, 2, 141–157.

Posse, C. (1992). "Projection pursuit discriminant analysis for two groups." *Communications in Statistics-Theory and Methods*, 21, 1, 1–19.

Price, J. and Gee, T. (2005). "Face recognition using direct, weighted linear discriminant analysis and modular subspaces." *Pattern Recognition*, 38, 2, 209–219.

Queirolo, C., Silva, L., Bellon, O., and Segundo, M. (2010). "3D face recognition using simulated annealing and the surface interpenetration measure." *IEEE transactions on pattern analysis and machine intelligence*, 206–219.

Rao, C. (2002). "Linear Statistical Inference and Its Applications, paperback."

Saul, L. and Roweis, S. (2000). "An introduction to locally linear embedding." *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html*.

Savvides, M., Kumar, B., and Khosla, P. (2004). "Eigenphases vs eigenfaces." In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3.

Schervish, M. (1984). "Linear discrimination for three known normal populations." *Journal of statistical planning and inference.*, 10, 2, 167–175.

Sirovich, L. and Kirby, M. (1987). "Low-dimensional procedure for the characterization of human faces." *Journal of the Optical Society of America A*, 4, 3, 519–524.

Sivic, J., Russell, B., Efros, A., Zisserman, A., and Freeman, W. (2005). "Discovering objects and their location in images." In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, 370–377. IEEE.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Verlag.

Sznitman, R. and Jedynak, B. (2010). "Active testing for face detection and localization." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32, 10, 1914–1920.

Tang, E., Suganthan, P., Yao, X., and Qin, A. (2005). "Linear dimensionality reduction using relevance weighted LDA." *Pattern recognition*, 38, 4, 485–493.

Torkkola, K. (2001). "Linear discriminant analysis in document classification." In *IEEE ICDM Workshop on Text Mining*, 800–806. Citeseer.

Tu, C. and Lien, J. (2010). "Automatic location of facial feature points and synthesis of facial sketches using direct combined model." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40, 4, 1158–1169.

Turk, M. and Pentland, A. (1991). "Eigenfaces for recognition." *Journal of cognitive neuroscience*, 3, 1, 71–86.

Vasilescu, M. and Terzopoulos, D. (2002). "Multilinear image analysis for facial recognition." In *International Conference on Pattern Recognition*, vol. 16, 511–514.

Wold, S., Sjostrom, M., and Eriksson, L. (2001). "PLS-regression: a basic tool of chemometrics." *Chemometrics and intelligent laboratory systems*, 58, 2, 109–130.

Xu, Z., Lu, L., and Shi, P. (2008). "A hybrid approach to gender classification from face images." In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 1–4.

Yang, J., Jin, Z., Yang, J., Zhang, D., and Frangi, A. (2004). "Essence of kernel Fisher discriminant: KPCA plus LDA." *Pattern Recognition*, 37, 10, 2097–2100.

Yu, H. and Yang, J. (2001). "A direct LDA algorithm for high-dimensional datawith application to face recognition." *Pattern Recognition*, 34, 10, 2067–2070.

Yu, W., Teng, X., and Liu, C. (2006). "Face recognition using discriminant locality preserving projections." *Image and Vision computing*, 24, 3, 239–248.

Zhou, D., Yang, X., Peng, N., and Wang, Y. (2006). "Improved-LDA based face recognition using both facial global and local information." *Pattern Recognition Letters*, 27, 6, 536–543.

Zhu, L. and Zhu, S. (2007). "Face recognition based on orthogonal discriminant locality preserving projections." *Neurocomputing*, 70, 7-9, 1543–1546.