UNIVERSITY OF SOUTHAMPTON

SCHOOL OF MATHEMATICS

A Class of Distance-Preserving Matrix Optimization Models in Data Mining

by

Sohana Jahan

Thesis for the degree of Doctor of Philosophy

January 2017

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

SCHOOL OF MATHEMATICS

Mathematics

Doctor of Philosophy

A CLASS OF DISTANCE-PRESERVING MATRIX OPTIMIZATION MODELS IN DATA MINING

by Sohana Jahan

In this thesis we are concerned to work on a class of matrix optimization problems. A matrix optimization problem (MOP) involves optimizing the sum of a linear function and a proper closed simple convex function subject to affine constraints in the matrix space. Many important optimization problems in various applications such as data mining, network localization, etc arising from a wide range of fields such as engineering, finance and so on, can be cast in the form of MOPs. This thesis is focused on the application of MOPs in data mining specially on data visualization, regression and classification. Data mining is the process of discovering interesting patterns and knowledge where different approaches (eg. dimension reduction) are applied to pre-process the data smoothing out noises. Dimensionality reduction is a traditional problem in pattern recognition and machine learning. A wide number of methods are used to project high dimensional data into low dimensional space so that the result performs better for further processing such as regression, classification, clustering etc.

The classical Multi-Dimensional Scaling (cMDS) is an important method for data dimension reduction and therefore for assigning them into fixed number of classes. Nonlinear variants of cMDS have been developed to improve its performance. One of them is the MDS with Radial Basis Functions (RBF). A key issue that has not been well addressed in MDS-RBF is the effective selection of its centers. Proper selection of centers leads to better classification of the data. This research treats this selection problem as a multi-task learning problem, which leads us to employ the $(2,1)$-norm to regularize the original MDS-RBF objective function. Two reformulations: Diagonal and spectral reformulations have been studied. Both can be effectively solved through an iterative block-majorization method. Numerical experiments show that the regularized models can improve the original model significantly. Though working very fast for small data set, these models are little time consuming for large data set. So we were seeking for a model that will project the large data efficiently.

Supervised distance preserving projection method (SDPP) is a very efficient method proposed recently for dimension reduction in supervised settings. Basic formulation of SDPP aims to preserve distances locally between data points in the projected space

(reduced feature space) and the output space. In our work we proposed a modification of SDPP which incorporates the total variance of the projected co-variates to the SDPP problem. We formulated the proposed optimization problem as a Semidefinite Least Square (SLS) SDPP. The SLS-SDPP maximizes the total variance of the projected co-variates and preserves the local geometry of the output space as well. A two block Alternating Direction Method of Multipliers have been developed to learn the transformation matrix solving the SLS-SDPP which can easily handle out of sample data. The projections of testing data points in low dimensional space are further used for regression or classifying them into different classes. The experimental evaluation on both synthetic and real world data demonstrates that SLS-SDPP improves SDPP significantly, outperforms some other state-of-the-art approaches and can be applied to any higher dimensional large data set. Finally SLS-SDPP is applied on some very well known face recognition problems. Satisfactory performance of our proposed dimension reduction method compared to some leading approaches in this area signify the applicability of our model to a wide range of image recognition problems.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, Sohana Jahan , declare that the thesis entitled *A Class of Distance-Preserving Matrix Optimization Models in Data Mining* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: an article entitled "'Regularized Multidimensional Scaling with Radial Basis Functions"' in the Journal of Industrial and Management Optimization(JIMO), doi:10.3934/jimo.2016.12.543 .

Signed:

Date: 19^{th} July 2016.

Acknowledgements

I would like to take this opportunity to express my sincere appreciation to those who have contributed to this thesis and supported me in one way or the other during this amazing journey.

Firstly, I would like to express my sincere gratitude to my supervisor $Dr\ Hou-Duo,\ Qi$ for his continuous guidance and all the useful discussions and brainstorming sessions, especially during the difficult conceptual development stage. I also remain indebted for his patience, understanding, motivation and support during the times when I was really down and depressed due to personal family problems.

A special acknowledgement goes to $Dr\ Tri-Dung\ Nguyen$ for his insightful comments and encouragement which incented me to widen my research from various perspectives.

I am also hugely appreciative to $Professor\ Jörg\ Fliege$ for invaluable suggestions and encouraging my research.

Heartfelt thanks goes to the Commonwealth Scholarship Commission for giving me the opportunity to carry out my doctoral research by giving me the financial support.

Special mention goes to my office mate *Shuanghua* for all his useful suggestions.

I will forever be thankful to my former research advisor $Professor\ Md.\ Ainul\ Islam.$ He has always been very helpful and supportive in providing me advice and numerous opportunities to learn and develop as a researcher.

Finally, Words cannot express how grateful I am to my Father, mother and my mother-in law for all of the sacrifices they have made on my behalf . I would also like to thank all of my friends who supported me mentally to strive towards my goal.

At the end I would like to acknowledge two most important persons in my life my husband *Imtiaz* and my beloved two years old son *Sameeh* for almost unbelievable support and being a constant source of strength and inspiration during this journey.

Nomenclature

$\mathcal{S}^n$	The space of all real $n \times n$ symmetric matrices.
$\mathcal{S}^n_+$	The cone of positive semidefinite matrices in $\mathcal{S}^n$.
$\mathcal{S}^n_{++}$	The set of all positive definite matrices in $\mathcal{S}^n$.
$\mathcal{O}_n$	The set of all $n \times n$ orthonormal matrices
$\Re^{m \times n}$	The space of all $m \times n$ matrices
$Z^\dagger$	The Moore-Penrose pseudoinverse of $Z \in \Re^{m \times n}$
$\mathrm{Tr}(C)$	The trace of the matrix C, (sum of the eigenvalues (or the diagonal elements) of C)
$\|X\|_F$	The Frobenius norm defined by $\|X\|_F = sqrt(sum_(i = 1)^m sum_(j = 1)^n \|x_(ij)\|$
$\Pi_{\mathcal{S}^n_+}(X)$	The projection of a given matrix $X \in \mathcal{S}^n$ onto $\mathcal{S}^n_+$
$\mathcal{A}^*$	Conjugate of a linear operator $\mathcal{A}$, ($\mathcal{A}^* y = A_1 y_1 + \ldots + A_n y_n$, A_i is the ith row of $\mathcal{A}$)
$\langle \mathcal{A}x, y \rangle = \langle x, \mathcal{A}^* y \rangle$	For any linear operator $\mathcal{A} : X \mapsto Y$, $\mathcal{A}x = (< A_1, x >, < A_2, x >, \ldots, < A_n, x >)$
All further notations	are either standard or defined in the text.

Chapter 1

Introduction

1.1 Data Mining:

Data mining is the computational process of discovering useful and interesting patterns and knowledge from large amount of data. The kinds of patterns or knowledge that can be mined from a data set include characterization and discrimination; associations and correlations; classification and regression; cluster analysis; and outlier detection. Real-world databases are mostly noisy and with missing and inconsistent data due to their huge size and being collected from multiple, heterogenous sources. Low-quality data leads to low-quality mining performance. So preprocessing of data plays a vital role to improve the quality of the data and, consequently, of the mining results. There are several data preprocessing techniques such as data cleaning (smoothing noise, filling in missing values), data reduction by eliminating redundant features, data transformations (e.g., normalization) where data are scaled to fall within a smaller range like 0.0 to 1.0. These preprocessing techniques can improve the accuracy and efficiency of mining algorithms involving distance measurements. Among these techniques several dimension reduction algorithms are being developed by researchers to obtain a representation of the data set which is much smaller in size, yet produces almost the same analytical results. The data mining process can be described in many ways and Fig. 1.1. is one of them that emphasize the distance preserving information. After representing data in a lower dimensional space, different data mining techniques such as support vector

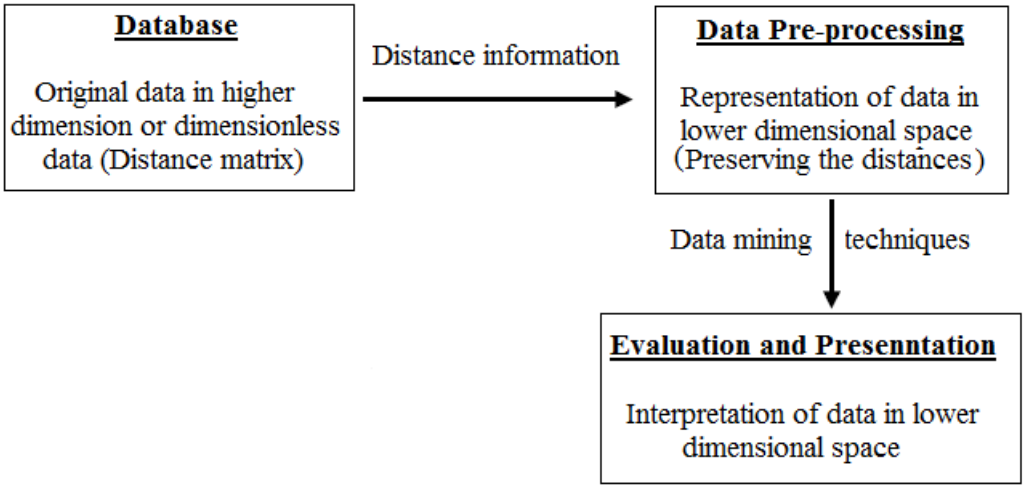


Figure 1.1: Basic steps for Data mining.

machine , k-nearest neighbor (for classification), k-means algorithm (for clustering), etc. are then applied on the reduced data set to understand the pattern of the data. A detailed theoretical explanation is beyond the scope of this thesis. So for a complete literature review one can see [46, 98, 97]. In our research we have used support vector machine (SVM) and k-nearest neighbor (k-nn) methods to study the classification of the data which is documented in section 2.4.2.

The most common dimension reduction method is Principal Component Analysis which searches for a set of orthogonal vectors that can best be used to represent the data. Recently much research is being devoted to dimension reduction techniques that preserve the interpoint distances of the data in the original space. Matrix optimization techniques are found to be very efficient in preserving these distances by determining the Euclidean distance matrix (EDM) that matches the original distance matrix as close as possible. In our research we have incorporated matrix optimization techniques to obtain the representation of data in a lower dimensional space which is discussed briefly in next sections.

1.2 Matrix Optimization Problem:

Matrix optimization problem involves optimizing the sum of a linear function and a proper closed simple convex function subject to affine constraints in the matrix space.

A standard form of a matrix optimization problem is given by

(MOP)

$$\begin{array}{rcl} \min\limits_{X} & \langle \Psi, X\rangle + \Phi(X) & \\ \text{s.t.} & \mathcal{A}X & = \quad b, \\ & X & \in \quad \mathcal{X}, \end{array} \tag{1.1}$$

where $\mathcal{X}$ is the Cartesian product of several finite dimensional real (symmetric or non-symmetric) matrix spaces given by $\mathcal{X} = \mathcal{S}^{m_1} \times \mathcal{S}^{m_2} \times \ldots \times \mathcal{S}^{m_k} \times \Re^{l_1 \times n_1} \times \ldots \Re^{l_t \times n_t}$ where $m_1, m_2, ..., m_k, n_1, n_2, ..., n_t, l_1, l_2, ..., l_t, k, t$ are positive integers. Without loss of generality, assume that $l_t \leq n_t, t = 1, \ldots, t$. $\langle ., . \rangle$ is the natural inner product of $\mathcal{X}$ and $\|.\|$ is the induced norm (e.g. Frobenius norm.),. $\Psi \in \mathcal{X}$ and $\mathcal{A} : \mathcal{X} \to \Re^k$ is a linear operator and $b \in \Re^k$. $\Phi : \mathcal{X} \to (-\infty, \infty]$ is a closed proper convex function with its Fenchel conjugate Φ^* defined by $\Phi^*(Z) := sup_{X \in \mathcal{X}}\{\langle Z, X\rangle - \Phi(X)\}$. The dual of MOP is given by

$$\begin{array}{rcl} \min\limits_{\{y,Z\}} & \langle b, y\rangle - \Phi^*(Z) & \\ \text{s.t.} & \mathcal{A}^* y - \Psi & = \quad Z, \end{array} \tag{1.2}$$

where $y \in \Re^k$ and $Z \in \mathcal{X}$ are dual variables and $\mathcal{A}^*$ is conjugate linear operator.

Matrix variables naturally arise from a number of optimization problem formulations from various areas such as engineering, neuroscience, bio informatics, finance, scientific computing, applied mathematics, etc. In such applications, the convex function Φ is simple. For example, if $\mathcal{X} = \mathcal{S}^n$ is a real symmetric matrix's space and $\Phi = \delta_{\mathcal{S}^n_+}(.)$ is the indicator function where $\mathcal{S}^n_+$ is the cone of real positive semidefinite matrices in $\mathcal{S}^n$, then $X \succeq 0$, i.e X is constrained to be a semidefinite matrices. Therefore the corresponding MOP

$$\begin{array}{rcl} \min\limits_{X} & \langle \Psi, X\rangle & \\ \text{s.t.} & \mathcal{A}X & = \quad b, \\ & X & \in \quad \mathcal{S}^n_+, \end{array} \tag{1.3}$$

is said to be the semidefinite programming (SDP), which has many interesting applications. If the function Φ is quadratic then the MOP can be defined as quadratic semidefinite programming (QSDP).

In our research we have worked on a class of matrix optimization problems which can be applied on data mining. Specially we have focused on data visualization, classification and regression. Data mining is an essential process where different approaches (eg. dimension reduction) are applied to extract data patterns. Dimensionality reduction is a traditional problem in pattern recognition and machine learning. A wide number of methods are used to project high dimensional data into low dimensional spaces so that the result performs better for further processing such as regression, classification, clustering, etc. Data projection using dimension reduction has been found fundamental to image recognition, short text classification and in many applications in both social and engineering sciences. For example, an image can be thought as a point in a high dimensional space (e.g. $64 \times 64 = 4096$). Although the input dimension is very high, a very small number of features may be used to recognize the image. So dimension reduction is further required to determine the best relevant features.

In the next sections we will show some examples to demonstrate how a matrix variable arises in the formulation of our dimension reduction problems.

1.3 Regularized Multidimensional Scaling

Multidimensional Scaling (MDS) is a set of data analysis techniques that analyze similarities and dissimilarities of data. Using MDS, data in a higher dimensional space can be projected into a lower dimensional space that preserves essential information in the data, smoothing out noise, to understand the structure of data easily. The classical Multi-Dimensional Scaling (cMDS) employs Euclidean distance to model dissimilarities. Suppose we have N data points $\{\mathbf{x}_i\}_{i=1}^N$ in the input space $\Re^n$ and their associated Euclidean distances d_{ij} is defined to be $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, where $\|\cdot\|$ is the Euclidean norm in $\Re^n$. Due to practical reasons, the original data contains noises and they can be represented in a lower-dimensional space $\Re^m$ ($m \ll n$). cMDS and its nonlinear variants have found many applications in both social and engineering sciences and are well documented in the books by Cox and Cox [23], Borg and Groenen [10], and Pękalaska and

Duin [76]. At the first part of this research, we have worked on one of the important nonlinear variants involving Radial Basis Functions (RBF) that was first proposed by Webb [108, 109] in the context of MDS. In [108] Webb proposed the following nonlinear methodology.

Firstly, the data set is mapped to another space called feature space $\Re^\ell$ through nonlinear function $\Phi : \Re^n \mapsto \Re^\ell$. For example, Φ can be radial basis functions. A very common RBF used by the researchers is Gaussian kernel $\Phi(r) = \exp(-r^2)$. The dimension of the feature space is determined by the number of RBFs used and is equal to the number of centers used in RBFs.

Secondly, the form of data representation in $\Re^m$, denoted as $\mathbf{f}$, is assumed to be a linear function of the feature vector Φ and takes the following form:

$$\mathbf{f}(x) = W^T \Phi(x), \qquad \forall\ x \in \Re^n \tag{1.4}$$

where $W \in \Re^{\ell \times m}$. Finally, the method seeks the best transformation matrix W that minimizes the raw `STRESS` (i.e., loss function):

$$\sigma^2(W) = \sum_{i,j=1}^{N} \alpha_{ij} (q_{ij}\,(W) - d_{ij})^2\,, \tag{1.5}$$

where for $i, j = 1, \ldots, N$, $\alpha_{ij} > 0$ are known weights and

$$q_{ij}(W) = \|\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_j)\| = \|W^T(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))\|. \tag{1.6}$$

Hence, the optimization problem of Webb's model is to determine the transformation matrix $W \in \Re^{\ell \times m}$ that minimizes $\sigma^2(W)$.

One of the key components of Webb's model is computing the feature vector $\Phi(x)$, which depends on its centers $\mathbf{c}_i$, $i = 1, \ldots, \ell$, since selection of important centres leads to better projection of the data points into lower dimensional space. Moreover the number of centres should be as small as possible because working with more centres require higher computational complexity and more CPU time and also may lead to over fitting of data..

Webb [108] suggests to randomly choose the centers and to use a cross-validation scheme to pick the best one. However, the cross-validation scheme is often very expensive to run. We will consider the selection of the centers for RBFs as a kind of multi-task learning problem, which has been widely studied in machine learning (see [2, 3]). We will introduce $(2,1)$-norm which works as a regularizer to control the selection of centers for RBFs. Therefore the objective of our model becomes to minimize the $(2,1)$-norm of W together with the original objective function $\sigma^2(W)$. Thus the optimization problem of our model is:

$$\min_{W \in \Re^{\ell \times m}} \; P(W) = \sigma^2(W) + \gamma \|W\|_{2,1}^2, \tag{1.7}$$

where $\gamma > 0$ is the regularization parameter and $\|W\|_{2,1}$ is the regularization term in the model defined by

$$\|W\|_{2,1} = \|W_{1:}\| + \ldots + \|W_{\ell:}\|,$$

where $W_{i:}$ is the ith row of W. The above problem is converted to a convex optimization problem and then an alternating minimization algorithm is proposed to solve it efficiently which is discussed in chapter 3.

1.4 Semidefinite Least Square Model

Semidefinite least square (SLS) model is very important and arises in diverse applications of finance, engineering, machine learning etc. Consider the following semidefinite least square (SLS) programming problem:

$$\begin{array}{rll} \min\limits_{X} & \frac{1}{2}\|X - G\|_F^2 & \\ \text{s.t.} & \mathcal{A}X & \geq \; b, \\ & \mathcal{B}X & = \; d, \\ & X & \succeq \; 0, \end{array} \tag{1.8}$$

where $\mathcal{A} : \mathcal{S}^n \to \Re^k$ and $\mathcal{B} : \mathcal{S}^n \to \Re^p$ are linear operators defined by $\mathcal{A}X = (tr(A_1X), tr(A_2X), \ldots, tr(A_kX))^T$ and $\mathcal{B}X = (tr(B_1X), tr(B_2X), \ldots, tr(B_pX))^T$ with $A_i, B_j \in \mathcal{S}^n$, $b \in \Re^k$ and $d \in \Re^p$. This model arises in many important applications.

A simple example is the problem of finding the nearest correlation matrix subject to additional linear constraints, $X_{jj} = 1$ for all $i = 1, ..., n$ studied in [39, 79, 53] which can be modeled as

$$\begin{array}{lrcl} \min\limits_{X} & \frac{1}{2}\|X - G\|_F^2 & & \\ \text{s.t.} & X_{jj} & = & 1, \\ & X & \succeq & 0. \end{array} \tag{1.9}$$

Another form of SLS model is

$$\begin{array}{lrcl} \min\limits_{X} & \langle \Psi, X \rangle + \frac{1}{2}\|\mathcal{A}X - b\|_F^2 & & \\ \text{s.t.} & \mathcal{B}X & = & d, \\ & X & \succeq & 0, \end{array} \tag{1.10}$$

where $\Psi \in \mathcal{S}^n$. Euclidean embedding problem is an interesting example of this form of SLS recently studied in [56] where the goal is to find an Euclidean distance matrix that is nearest to a given incomplete possibly noisy dissimilarity matrix.

In this thesis we have formulated supervised distance preserving projection (SDPP) [117] problem as an SLS model.

Suppose we have n data points $\{x_1, x_2, ..., x_n\}, x_i \in \Re^m$ and their responses $\{y_1, y_2, ..., y_n\}$, $y_i \in \Re^k$. Assuming that the mapping $X \to Y$ is continuous and X is well sampled, the idea is to project high dimensional data $\{x_1, x_2, ..., x_n\}$ in a lower dimensional space Z with dimensionality $r << m$ by $Z = f(X) = W^T X$ in such a way that the projection preserve distances locally between data points in the projected space (reduced feature spaces) and the output space. In [117], SDPP seeks for the transformation matrix W that minimizes

$$F(W) = \frac{1}{n}\sum_{i=1}^{n}\sum_{x_j \in N(x_i)} (d_{ij}^2(W) - \delta_{ij}^2)^2$$

where $N(x_i)$ denotes a neighborhood of x_i, $d_{ij}^2(W) = \|z_i - z_j\|^2 = \|W^T(x_i - x_j)\|^2$ and $\delta_{ij}^2 = \|y_i - y_j\|^2$. Locality around any point x_i is controlled by its k nearest neighbors in $N(x_i)$. In this thesis we incorporate the total variance $\sum_{i=1}^{n} \|z_i\|^2$ to the stress $F(W)$ and maximize the objective function,

$$\max \sum_{i=1}^{n} \|z_i\|^2 - \frac{\nu}{n} \sum_{i=1}^{n} \sum_{x_j \in N(x_i)} (d_{ij}^2(W) - \delta_{ij}^2)^2$$

which can be reformulated (described in section 3.4) as the optimization problem:

$$\begin{aligned} \min_{X} \quad & \langle \Psi, X \rangle - \tfrac{\nu}{n} \|U\|_F^2 \\ \text{s.t.} \quad & \mathcal{A}X - U = b, \\ & X \succeq 0, \end{aligned} \tag{1.11}$$

where $\nu > 0$ is the penalty parameter. To put equal emphasis on both the terms we choose the value $\nu = 1$. $X = WW^T$,$\Psi = \sum_{i=1}^{n} \Psi_{ii} = \sum_{i=1}^{n} x_i x_i^T$, $\mathcal{A}X = \langle \Phi_{ij}, X \rangle =$ ($d_{ij}^2(W)$ and $b = \delta_{ij}^2$.

The goal of our model SLS-SDPP is to determine the positive sefidefinite matrix X from which the transformation matrix W can be obtained to get the projection of the data points in a lower dimensional space. The detail of this model will be discussed in chapter 4.

1.5 Thesis organization

The remainder of this thesis is divided as follows:

In Chapter 2 at first we will review the RBF-MDS model introduced by Webb [108] and single out the problem of choosing centers for the RBFs used. We will then introduce the $(2,1)$-norm as a regularizer to the model. On the way, we will also highlight the major differences as well as relationships between our model and the multi-task learning model in [3]. In 2.3, we will study two reformulation models: diagonal and spectral. We will then develop an iterative block-majorization method for our model. Numerical results on three commonly used data sets are reported and explained in Section 2.4, where we demonstrate that the regularized models can significantly improve the original model of Webb [108].

Our research on Supervised Distance Preserving Projection(SDPP) is documented in Chapter 3. At first we will discuss SDPP introduced by Zhu et al. [117]. We will incorporate the variance of projected points to the SDPP and formulate the modified SDPP problem as a semidefinite least square (SLS-SDPP) which is a QSDP problem. We will develop a two-block ADMM [95, 66] in section 3.6 to solve the SLS-SDPP problem. Several synthetic and real world data are considered to demonstrate the performance of our model in compared to five other methods SDPP, supervised Principal Component Analysis (SPCA) [4], Partial Least Square (PLS) [111], Kernel Dimension Reduction (KDR) [35] and Fishers Discriminant Analysis (FDA) [31]. Experimental evaluation shows that our algorithm can learn the transformation matrix efficiently which can easily handle out of sample data and SLS-SDPP significantly improves the performance of SDPP and outperforms some other leading methods in most of the cases. In Chapter 4 we will demonstrate the efficiency of our proposed algorithm on some face recognition problems by conducting experiments on some well known face data set and comparing the performance of SLS-SDPP with two leading approach Eigenface and Fisherface. The findings of this thesis are discussed and directions for future work are presented in Chapter 5.

Chapter 2

Data Classification using Radial Basis Function

2.1 Introduction

Multidimensional Scaling (MDS) is a set of data analysis techniques that analyse similarities and dissimilarities of data. MDS has its origin in psychometric where it was proposed to help understand people's judgment of the similarity of members of a set of objects.Torgerson [100] proposed the first MDS method and coined the term. MDS has now become a general data analysis technique used in a wide variety of fields. Application of MDS in diverse fields as marketing, sociology, physics, political science and biology are presented in the book on theory and application on MDS by Young and Hamer [114].

Suppose we have N objects with pairwise dissimilarities d_{ij} between objects i and j. The main purpose of multidimensional scaling is to map these objects into N points $x_1, x_2..., x_N$ in a low-dimensional metric space (usually 2 or 3 for visualization purpose) such that the metric distance between x_i and x_j matches the dissimilarity d_{ij} as closely as possible. A large number of ways to achieve this purpose are discussed in Cox and Cox [23] and Borg and Groenen [10]. If the metric space is Euclidean and the match is exact (i.e. $\|x_i - x_j\| = d_{ij}$ for all i, j), then the dissimilarity matrix $D := (d_{ij})$ is said to

have an exact Euclidean representation. But due to various reasons such as non-metric measurement in d_{ij}, often D doesn't have the Euclidean representation. In such case the idea is to determine a Euclidean distance matrix (EDM) in a lower dimensional space that approximates the distance matrix D in the original space. The classical Multi-Dimensional Scaling (cMDS) performs well if the distance matrix D is close to a true Euclidean distance matrix with a low-embedding dimension. Otherwise, certain corrections have to be made on the distance matrix.

The use of cMDS as a data dimension-reduction method (or data visualization method when the embedding dimension is 2 or 3) can be traced to the seminal work of Schoenberg [91] and the independent work of Young and Householder [115]. The method was made popular by Torgerson [101] and later by Gower [45] (see [70, Chapter 14] for details).

Early methods include adding a same positive constant to every pairwise distance, which results in the additive constant or the partial additive constant problems (see [71, 21, 16, 7, 81]). More advanced corrections are obtained through optimizing certain loss functions. The `STRESS` function first proposed by Kruskal [59] is one of the most often used loss functions (some other `STRESS` type functions are discussed in [10, Chapter 3]). The resulting optimization problems based on `STRESS` functions can be efficiently solved by the *majorization method* introduced by de Leeuw [63] ([10, Chapter 8] for a detailed description of the method). We have also used a majorization procedure in the proposed algorithm. Another class of corrections can be obtained through computing the nearest Euclidean distance matrix from the known distance matrix (see [43, 44, 82, 83, 84]). All of these methods make nonlinear corrections on the pairwise distances and therefore can be regarded as nonlinear variants of cMDS.

The classical Multi-Dimensional Scaling (cMDS) and its nonlinear variants have found many applications in both social and engineering sciences and are well documented in the books by Cox and Cox [23], Borg and Groenen [10], and Pękalaska and Duin [76]. In this chapter we have studied one of the important nonlinear variants involving Radial Basis Functions (RBF) that was first proposed by Webb [108, 109] in the context of MDS. The key issue in employing RBFs in MDS is to decide their centers. This includes the number of the centers to be used and then what they are. This issue has not been

well addressed in existing literature. For example, Webb [108] suggests to randomly choose the centers and then use an expensive cross-validation procedure to decide what they are. Here, we take a completely different route and regard the selection of the centers as a multi-task learning problem that has been widely studied in machine learning, see Argriou et al. [2, 3]. This will lead us to an optimization model that can be solved efficiently.

The nonlinear variant introduced by Webb [107] differs from those mentioned above in the following way. It regards the space where the original data lies the input space (also see [108]). The first stage of Webb's method is to map the data from the input space to a feature space through nonlinear functions such as RBFs. The dimension of the feature space is determined by the number of RBFs used and is equal to the number of centers used in RBFs. Assuming the first stage task is settled, the second stage is to find the *best* linear function that maps the feature space data to a low-dimensional embedding space (2 or 3 if the purpose is to visualize the data). Webb's method actually focuses on the second stage and suggests using a (potentially very expensive) cross-validation procedure to furnish the task in the first stage.

The purpose of this research is to propose a computational model that deals with the two stages. The key viewpoint here is to regard the selection of the centers for RBFs as a kind of multi-task learning problem, which has been widely studied in machine learning (see [2, 3]). We would like to emphasize that there are major differences between our learning problem and that in [3]. Roughly speaking, We have a non-convex optimization model while [3] has a convex one. But the principal idea of choosing the common tasks via minimizing the $(2,1)$-norm of the learning matrix in [3] is carried over to our model. This $(2,1)$-norm works as a regularizer to control the selection of centers for RBFs. We have studied two reformulation models: diagonal and spectral. We have developed an iterative block-majorization method for the resulting model. Numerical results on three commonly used data sets are reported and explained in Section 2.4, where we demonstrate that the regularized models can significantly improve the original model of Webb

[108].

2.2 The Problem of Learning Centers

In this section, we first introduce the RBF-MDS model of Webb [108]. We then treat the center selection problem in the model as a multi-task learning problem.

2.2.1 RBF-MDS Model

Suppose we have N data points $\{\mathbf{x}_i\}_{i=1}^N$ in the input space $\Re^n$ and their associated Euclidean distances d_{ij} is defined to be $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, where $\|\cdot\|$ is the Euclidean norm in $\Re^n$. Due to practical reasons, the original data contains noises and they can be represented in a lower-dimensional space $\Re^m$ ($m \ll n$). For example, when it is for visualization, m is often chosen to be 2 or 3. The representation is often done through nonlinear dimension reduction methodologies.

In [108] Webb proposed the following methodology. Firstly, the data set is mapped to another space called feature space $\Re^\ell$ through nonlinear function $\Phi : \Re^n \mapsto \Re^\ell$. For example, Φ can be radial basis functions defined as follows:

Definition 2.1. A radial basis function (RBF) is a real-valued function whose value depends on the distance of each point from some other point c, called a center. Thus the RBF denoted by $\phi(\mathbf{x}, \mathbf{c})$ is defined by

$$\phi(\mathbf{x}, \mathbf{c}) = \mathbf{f}(\|\mathbf{x} - \mathbf{c}\|)$$

In computational applications, multivariate functions often need to be approximated by other single univariate functions which are not known or only known at a finite number of points. Radial basis functions are one efficient, frequently used way to do this. Applications of RBF include finite element or spectral methods for the solution of partial differential equations, neural network with radial basis functions and machine learning,

approximations on spheres, statistical approximations, where positive definite kernels are very important, geophysical research and many engineering applications

The greatest advantage of using RBFs is that they can be applied in almost any dimension. Moreover their high accuracy and fast convergence to the approximated target function make them the most useful.

Examples of Radial Basis Functions

A good choice of ϕ is important for the quality of the approximation. Some popular forms of RBFs are given in the table.

Table 2.1: Examples of Radial Basis Functions

	Mathematical Form $\phi(r)$
Linear	r
Multiquadratics	$\sqrt{r^2 + c^2}$
Gaussian Kernel	$\exp(-r^2)$
Inverse Multiquadratic	$\frac{1}{\sqrt{r^2+c^2}}$
The Thin Spline	$r^2 \ln r$

For a detailed literature of each of the above functions one can see [15, 78]. Webb in [108] used Gaussian kernel as a RBF which is discussed briefly as follows:

Gaussian Kernel:

Gaussian kernel is widely used RBF for regression and discrimination. The general form is $\phi(r) = \exp(-r^2)$ where $r = \frac{\|\mathbf{x}-\mathbf{c}\|}{h}$, c is a center, h is smoothing parameter. In fitting data with distributed noise on the inputs Gaussian form is the optimal basis function in a least square sense. The basis functions are continuously differentiable and integrable. This property is useful for RBF to solve differential equations.

Empirical evidence suggests that in low dimensions, Gaussian kernel offers better performance. Therefore, (for the lower-dimension) Gaussian kernel is used by many researchers

for approximating data in an arbitrary number of dimensions.

Choice of smoothing parameter:

One of the most computational demanding parts of RBF is to choose the smoothing parameter. Approximately optimal performance is achieved by using over a wide range of smoothing parameter values and it is to be expected that cross validation or some of the simple heuristics can be used to determine the smoothing parameter for acceptable performance.

Now, let $\Phi(x) = (\phi_1(x), \ldots, \phi_\ell(x)) \in \Re^\ell$, with

$$\phi_i(x) = \exp\left\{-\|\mathbf{x} - \mathbf{c}_i\|^2/h^2\right\}, \qquad i = 1, \ldots, \ell$$

where h is the bandwidth and $\mathbf{c}_i$ is the center of ϕ_i. Secondly, the form of data representation in $\Re^m$, denoted as $\mathbf{f}$, is assumed to be a linear function of the feature vector Φ. In terms of the original input space data, $\mathbf{f}$ is a nonlinear function from $\Re^n$ to $\Re^m$ and takes the following form:

$$\mathbf{f}(x) = W^T\Phi(x), \qquad \forall\ x \in \Re^n \tag{2.1}$$

where $W \in \Re^{\ell\times m}$. In other words, $\mathbf{f}$ is a composite of a linear function (represented by the matrix W) and the radial basis function Φ. Finally, the method seeks the best transformation matrix W that minimizes the raw `STRESS` (i.e., loss function):

$$\sigma^2(W) = \sum_{i,j=1}^{N} \alpha_{ij}(q_{ij}\,(W) - d_{ij})^2\,, \tag{2.2}$$

where for $i, j = 1, \ldots, N$, $\alpha_{ij} > 0$ are known weights and

$$q_{ij}(W) = \|\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_j)\| = \|W^T(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))\|. \tag{2.3}$$

Hence, the optimization problem of Webb's model is

$$\min_{W \in \Re^{\ell \times m}} \sigma^2(W). \tag{2.4}$$

A majorization method is then used to solve (2.4). Let $\mathbf{v}^{ij} = \Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)$. We assume that the data set is rich enough such that the vectors

$$\left\{\mathbf{v}^{ij} : \ i < j = 2, \ldots, N\right\} \text{ span the feature space } \Re^{\ell}. \tag{2.5}$$

It is obvious that one of the key components of Webb's model is computing the feature vector $\Phi(x)$, which depends on its centers $\mathbf{c}_i$, $i = 1, \ldots, \ell$. There are two natural questions to be asked here. How many centers should be used (i.e., how to decide ℓ)? What are the best choices of those centers? In [108], Webb suggests to randomly choose the centers and to use a cross-validation scheme to pick the best one. However, the cross-validation scheme is often very expensive to run. In the following, we try to answer those questions from a fresh viewpoint of multi-task learning.

2.2.2 Centre Selection as a Multi-Task Learning Problem

A general setting up for multi-task learning problems is described in [3, Sect. 2]. In this section, we will relate the center choosing problem to a multi-task learning problem. Suppose there are ℓ factors represented by $\phi_i(x)$, $i = 1, \ldots, \ell$ and there are m tasks. Each task in our problem can be represented as a linear regression of the ℓ factors:

$$\mathbf{f}_i(x) = \langle W_{:i}, \Phi(x) \rangle = \sum_{j=1}^{\ell} W_{ji} \phi_j(x), \qquad i = 1, \ldots, m \tag{2.6}$$

where $W_{:i}$ (`Matlab` type of notation) is the ith column of W and $\langle \cdot, \cdot \rangle$ is the standard inner product in $\Re^{\ell}$. The purpose is to learn the common factors (most effective centres) (out of the ℓ factors) among all m tasks, which is explained below.

Suppose $\phi_1(x)$ is not a common factor, then the corresponding coefficients W_{1i}, $i = 1, \ldots, m$ should be all zero. In other words, the factor $\phi_1(x)$ can be removed from the linear regression model (2.6). This corresponds to the 1st row of W being zero. Now,

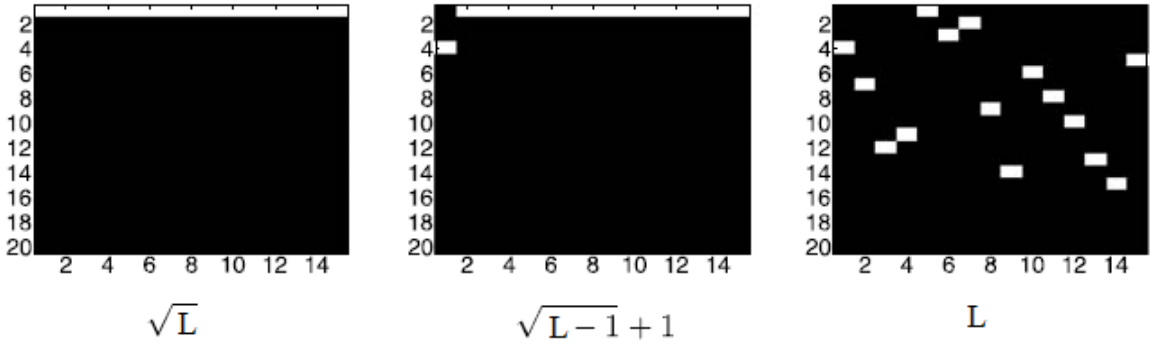


Figure 2.1: values of the $(2,1)$-norm matrix containing only L nonzero entries, equal to 1. When the norm increases, the level of sparsity along the rows decreases.

the problem of learning common factors is equivalent to finding the zero rows of W. This can be well achieved by minimizing the $(2,1)$-norm of W together with the original objective function $\sigma^2(W)$. The $(2,1)$-norm of W is obtained by first computing the 2-norms of the rows $W_{i:}$ and then the 1-norm of the vector $\|W_{1:}\|, \|W_{2:}\|, \ldots, \|W_{ell:}\|$.

$$\|W\|_{2,1} = \|W_{1:}\| + \ldots + \|W_{\ell:}\|,$$

where $W_{i:}$ is the ith row of W. The $(2,1)$-norm favors a small number of nonzero rows in the matrix W, therefore ensuring that the common features (most effective centers) will be selected.

A simple example can be considered for further illustration represented in Fig. 2.1. Consider the matrix W whose entries are taken binary values and there are only L entries which are equal to 1. The minimum value of $(2,1)$-norm equals $\sqrt{L}$ and is obtained when all the 1 entry are placed in the same row. The maximum valu is L and is obtained when each 1 entry is placed in a different row. This example implies that minimization of (2,1)-norm forces most of the rows of the matrix to become zero.

Therefore, the optimization model that we are trying to solve becomes

$$\min_{W \in \Re^{\ell \times m}} \; P(W) = \sigma^2(W) + \gamma \|W\|_{2,1}^2, \tag{2.7}$$

where $\gamma > 0$ is the regularization parameter and $\|W\|_{2,1}$ is the regularization term in the model. Through (2.7), we can get rid of the centers that are less important in terms of their contributions to $\|W\|_{2,1}$, leading to effective selections of important centers. We should point out that in [3], the number of tasks (m) is larger than the number of factors (ℓ). Here, we have the opposite ($m < \ell$). Furthermore, the objective function corresponding to the raw stress $\sigma^2(W)$ in [3] is convex with respect to W. Here, $\sigma^2(W)$ is nonconvex. We shall see that we can nicely combine the majorization strategy and the techniques in handling the $(2,1)$-norm developed in [3] to solve problem (2.7).

2.3 Iterative Block-Majorization Methods

This section is devoted to numerical methods for solving problem (2.7). The $(2,1)$-norm is nonsmooth (not differentiable) and the `stress` function $\sigma^2(W)$ is not convex. Hence, problem (2.7) is difficult to solve. We will relate problem (2.7) to that of [3] in order to spare us from giving very involved technical proofs. This led us to two reformulations that are conducible to developing majorization methods later on.

2.3.1 Diagonal and Spectral Reformulations

Let $\mathcal{S}^\ell$ denote the space of $\ell \times \ell$ symmetric matrices with the standard inner product $\langle \cdot, \cdot \rangle$. Let $\mathcal{S}^\ell_+$ denote the cone of positive semidefinite matrices in $\mathcal{S}^\ell$ and $\mathcal{S}^\ell_{++}$ denote the set of all positive definite matrices in $\mathcal{S}^\ell$. Let $\mathscr{O}^\ell$ denote the set of all $\ell \times \ell$ orthonormal matrices. That is, $U \in \mathscr{O}^\ell$ if and only if $U^T U = I$. For $C \in \mathcal{S}^\ell_+$, we let $C^\dagger$ denote the pseudo-inverse of C.

Definition 2.2. Given an $m \times n$ matrix C, the Moore-Penrose pseudoinverse is a unique $n \times m$ matrix $C^\dagger$ satisfying the following equalities:

- $CC^\dagger C = C$, ($CC^\dagger$ need not be the general identity matrix),
- $C^\dagger CC^\dagger = C^\dagger$,
- $(CC^\dagger)^H = CC^\dagger$, ($CC^\dagger$ is Hermitian. C^H is the conjugate transpose of C)

- $(C^\dagger C)^H = C^\dagger C$, ($C^\dagger C$ is Hermitian)

If the inverse of $(C^H C)$ exists, then $C^\dagger$ can be determined by

$$C^\dagger = (C^H C)^{-1} C^(H).$$

For a constant $a \in \Re$, $a^\dagger = 1/a$ if $a \neq 0$ and $a^\dagger = 0$ otherwise.

We let $\mathrm{Tr}(C)$ denote the trace of C.

Suppose $C \in \mathcal{S}_+^\ell$ has the following spectral decomposition

$$C = U\mathrm{Diag}(\lambda_1, \ldots, \lambda_\ell)U^T,$$

where $\lambda_1 \geq \ldots \geq \lambda_\ell \geq 0$ are the eigenvalues of C in nonincreasing order, $\mathrm{Diag}(\lambda_1, \ldots, \lambda_\ell)$ is the diagonal matrix with λ_i being on its diagonal, and $U \in \mathcal{O}^\ell$. The pseudo-inverse of C is then given by

$$C^\dagger = U\mathrm{Diag}(\lambda_1^\dagger, \ldots, \lambda_\ell^\dagger)U^T.$$

Define the function

$$Q(W, C) = \sigma^2(W) + \gamma\langle WW^T, C^\dagger\rangle. \tag{2.8}$$

By following the proof of [3, Thm. 1 and Cor. 2], we can obtain the following result.

Theorem 2.3. *Problem (2.7) is equivalent to the problem*

$$\begin{array}{ll} \min & Q(W, \mathrm{Diag}(\lambda)) \\ s.t. & \lambda = (\lambda_1, \ldots, \lambda_\ell) \geq 0, \quad \sum_{i=1}^\ell \lambda_i \leq 1 \\ & \lambda_i \neq 0 \ \textit{whenever} \ W_{i:} \neq 0, \ i = 1, \ldots, \ell. \end{array} \tag{2.9}$$

Moreover, if $(\widehat{W}, \widehat{\lambda})$ is the optimal solution of (2.9), it holds

$$\widehat{\lambda}_i = \frac{\|\widehat{W}_{i:}\|}{\|\widehat{W}\|_{2,1}}, \quad i = 1, \ldots, \ell. \tag{2.10}$$

Because of this theorem, we call (2.9) the diagonal reformulation of (2.7). We now present what we call the spectral reformulation, which has better numerical performance than the diagonal reformulation. We start from a simple observation.

$$W \in \Re^{\ell \times m} \text{ if and only if } W = UA \tag{2.11}$$

for some $U \in \mathscr{O}^{\ell}$ and $A \in \Re^{\ell \times m}$. The `stress` function $\sigma^2(W)$ can then be written as

$$\begin{aligned} \sigma^2(W) &= \sigma^2(UA) \\ &= \sum_{i,j=1}^{N} \alpha_{ij}(q_{ij}(UA) - d_{ij})^2 \\ &= \sum_{i,j=1}^{N} \alpha_{ij} \left(\|A^T U^T (\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\| - d_{ij} \right)^2 . \end{aligned}$$

We consider the following problem:

$$\min_{A \in \Re^{\ell \times m},\ U \in \mathscr{O}^{\ell}} E(A, U) = \sigma^2(UA) + \gamma \|A\|_{2,1}^2. \tag{2.12}$$

We note that problem (2.12) is not equivalent to problem (2.7) under the transformation in (2.11). But they have a common term of the `stress` function. Since our main target is to minimize the stress function so we can minimize (2.12) instead of (2.7). This time, $\|A\|_{2,1}$ is the regularizer instead of $\|W\|_{2,1}$. The benefit in using $\|A\|_{2,1}$ is that problem (2.12) has a nice characterization, which allows us to develop a majorization method. Problem (2.12) is similar in structure to [3, Problem (4)] and is equivalent to the following problem.

Theorem 2.4. *Problem (2.12) is equivalent to the problem*

$$\inf \left\{ Q(W, D) :\ W \in \Re^{\ell \times m}, D \in \mathcal{S}_{++}^{\ell}, \operatorname{Tr}(D) \leq 1 \right\}. \tag{2.13}$$

In particular, any minimizing sequence of problem (2.13) is bounded and converges to a minimizer of problem (2.12). Moreover, if $(\widehat{W}, \widehat{D})$ is any limit of a minimizing sequence, then any $(\widehat{A}, \widehat{U})$ such that the columns of $\widehat{U}$ forms an orthogonormal basis of eigenvectors

of $\widehat{D}$ and $\widehat{A} = \widehat{U}^T\widehat{W}$, is an optimal solution of problem (2.12) (and therefore $\widehat{W}$ is the minimizer of (2.7) by (2.11)).

Proof. Let ν_s denote the infimum of (2.13). Suppose $\{W^k, D^k\}$ is a minimizing sequence. Then

$$\nu_s = \lim_{k\to\infty} Q(W^k, D^k) \geq \lim_{k\to\infty} \sigma^2(W^k) \geq 0, \tag{2.14}$$

because the regularization term in (2.8) is always nonnegative. Due to the constraint $\mathrm{Tr}(D) \leq 1$ in (2.13), $\{D^k\}$ is bounded. Suppose that the sequence $\{W^k\}$ is unbounded. Without loss of generality, we assume that

$$\frac{W^k}{\|W^k\|} \to \overline{W} \neq 0. \tag{2.15}$$

Dividing both sides of (2.14) by $\|W^k\|^2$ and taking limits, we obtain

$$0 = \sum_{i,j=1}^{N} \|\overline{W}^T \mathbf{v}^{ij}\|,$$

which implies

$$\overline{W}^T \mathbf{v}^{ij} = 0 \qquad \forall\ i < j = 2, \ldots, N.$$

Assumption (2.5) forces $\overline{W} = 0$, which contradicts (2.15). Hence, the sequence $\{W^k\}$ is bounded. This proves that any minimizing sequence is bounded. The remaining proof can be similarly constructed as in [3, Thm. 1 and Cor. 1]. □

Note that in (2.13) , $D \in \mathcal{S}^{\ell}_{++}$ otherwise if we set $D \in \mathcal{S}^{\ell}_{+}$ then it is possible to get a matrix D which will lead to the term $\langle WW^T, D^{\dagger}\rangle = 0$ in (2.8).

It is because that $\widehat{U}$ is a normalized eigenvector matrix of $\widehat{D}$ and it can be obtained through a spectral decomposition of $\widehat{D}$, we refer to problem (2.13) as the spectral reformulation model. The next result shows that the spectral reformulation model (2.13) is a generalization of the diagonal reformulation model (2.9).

Proposition 2.5. *Let ν_d be the optimal objective value of problem (2.9) and ν_s be the infimum of problem (2.13). Then we have*

$$\nu_d \geq \nu_s.$$

Moreover, if D is restricted to be diagonal in (2.13), the equality holds.

Proof. Suppose (W, λ) is an optimal solution of problem (2.9). Let $\mathscr{I}$ denote the indices of positive λ_i:

$$\mathscr{I} = \{i \mid \lambda_i > 0,\ i = 1, \ldots, \ell\} \quad \text{and} \quad \bar{\mathscr{I}} = \{1, \ldots, \ell\} \setminus \mathscr{I}.$$

Let $\ell_0 = |\mathscr{I}|$, the cardinality of $\mathscr{I}$. Define

$$\lambda_{\min} = \min_{i \in \mathscr{I}} \lambda_i.$$

Obviously $\lambda_{\min} > 0$. Define the sequence $\lambda^k \in \Re^\ell$, $k = 1, 2...$ by

$$\lambda_i^k = \begin{cases} \lambda_i - \frac{1}{2k}\lambda_{\min} & \text{if } i \in \mathscr{I} \\ \frac{\ell_0}{2k(\ell - \ell_0)}\lambda_{\min} & \text{if } \bar{\mathscr{I}} \neq \emptyset \text{ and } i \in \bar{\mathscr{I}}. \end{cases}$$

It is easy to verify that $\lambda^k > 0$ for all $k = 1, 2, \ldots$, and

$$\sum_{i=1}^{\ell} \lambda_i^k = \sum_{i=1}^{\ell} \lambda_i \leq 1.$$

Let $D^k = \mathrm{Diag}(\lambda^k)$. Then, the sequence (W, D^k) satisfies the constraints in (2.13).

Now we compute the respective objective function values. We first note that

$$\begin{aligned} Q(W, \mathrm{Diag}(\lambda)) &= \sigma^2(W) + \gamma \langle WW^T, (\mathrm{Diag}(\lambda))^\dagger \rangle \\ &= \sigma^2(W) + \gamma \sum_{i=1}^{\ell} \left(\|W_{i:}\|^2 \lambda_i^\dagger \right) \\ &= \sigma^2(W) + \gamma \sum_{i \in \mathscr{I}} \left(\|W_{i:}\|^2 / \lambda_i \right). \end{aligned}$$

It also follows from the constraints in (2.9) that

$$i \in \mathscr{I} \quad \text{whenever} \quad W_{i:} \neq 0.$$

This property yields

$$\begin{aligned} Q(W, D^k) &= \sigma^2(W) + \gamma \langle W W^T, (D^k)^\dagger \rangle \\ &= \sigma^2(W) + \gamma \sum_{W_{i:} \neq 0} \left(\|W_{i:}\|^2 / \lambda_i^k \right) \\ &\leq \sigma^2(W) + \gamma \sum_{i \in \mathscr{I}} \left(\|W_{i:}\|^2 / \lambda_i^k \right). \end{aligned}$$

Taking limits on both sides, we have

$$\begin{aligned} \liminf_{k \to \infty} Q(W, D^k) &\leq \sigma^2(W) + \gamma \lim_{k \to \infty} \sum_{i \in \mathscr{I}} \left(\|W_{i:}\|^2 / \lambda_i^k \right) \\ &= \sigma^2(W) + \gamma \sum_{i \in \mathscr{I}} \left(\|W_{i:}\|^2 / \lambda_i \right) \\ &= Q(W, \mathrm{Diag}(\lambda)) = \nu_d. \end{aligned}$$

As stated before, (W, D^k) is a feasible sequence of problem (2.13). It is obvious that being the infimum of (2.13)

$$\nu_s \leq \lim_{k \to \infty} Q(W, D^k).$$

This proves $\nu_s \leq \nu_d$.

The above proof actually shows that if D is restricted to be diagonal, we must have $\nu_s \leq \nu_d$. Now suppose that D is restricted to be diagonal. Let $\{W^k, D^k\}$ be a minimizing sequence of (2.13). That is

$$\nu_s = \lim_{k \to \infty} Q(W^k, D^k). \tag{2.16}$$

Denote D^k by $D^k = \mathrm{Diag}(\lambda^k)$ and $\lambda^k > 0$ for $k = 1, 2, \ldots$. By Thm. 2.4, the sequence $\{W^k, D^k\}$ is bounded. Without loss of any generality, we assume that

$$W^k \to W \qquad \text{and} \qquad \lambda^k \to \lambda.$$

Obviously, $\lambda \geq 0$ and $\sum_{i=1}^{\ell} \lambda_i \leq 1$. The sequence $\{\langle W^k(W^k)^T, (D^k)^\dagger\rangle\}$ is also bounded because $\{W^k, D^k\}$ is a minimizing sequence of (2.13) and $\sigma(W^k) \geq 0$ for all k. Assume that $W_{i:} \neq 0$ for some i. Then $W_{i:}^k \neq 0$ for sufficiently large k. We further have

$$\begin{aligned} \infty &> \lim_{k\to\infty} \langle W^k(W^k)^T, (D^k)^\dagger\rangle \geq \lim_{k\to} \|W_{i:}^k\|^2 (\lambda_i^k)^\dagger \\ &= \begin{cases} \|W_{i:}\|^2(\lambda_i)^\dagger & \text{if } \lambda_i > 0 \\ \infty & \text{if } \lambda_i = 0. \end{cases} \end{aligned}$$

This can only happen when $\lambda_i > 0$. Thus we have proved that $\lambda_i \neq 0$ whenever $W_{i:} \neq 0$. In other words, (W, λ) is feasible with respect to the constraints in (2.9) and

$$\lim_{k\to\infty} \langle W^k(W^k)^T, (D^k)^\dagger\rangle = \langle WW^T, C^\dagger\rangle,$$

where $C = \mathrm{Diag}(\lambda)$. By continuity of $\sigma^2(\cdot)$, (2.16) implies

$$\nu_s = \sigma^2(W) + \gamma\langle WW^K, C^\dagger\rangle \geq \nu_d.$$

Combining the first part, we have $\nu_s = \nu_d$. □

Although problem (2.9) is not exactly a special case of problem (2.13), Prop. 2.5 allows us to treat it as if it was obtained through restricting D to be positive diagonal matrices in (2.13). Comparing to (2.9), the matrix D has more freedom to move in (2.13), hence leading to the lower objective function value ν_s. This is likely to contribute to a lower objective function of $\sigma^2(W)$. This possibility has been confirmed by our extensive numerical experiments.

2.3.2 Iterative Block-Majorization Method

In this section, we develop an algorithm for the spectral model problem (2.13). It can be straightforwardly applied to the diagonal model (2.9) with simple modifications.

As we mentioned before, problem (2.13) is not attainable but the infimum is finite. Argyrion et. al [3] proved that such kind of problem is equivalent to the following

problem, which is attainable:

$$\min \left\{ Q(W,D): \begin{array}{l} W \in \Re^{\ell \times m} \\ D \in \mathcal{S}_+^\ell, \mathrm{Tr}(D) \leq 1 \\ \mathrm{Range}(W) \subseteq \mathrm{Range}(D) \end{array} \right\}. \tag{2.17}$$

The optimal objective value of (2.17) equals the infimum of (2.13). An interesting result about (2.17) is that when W is fixed, minimizing $Q(W,D)$ over D in the feasible set of (2.17) has a closed-form solution:

$$D = \frac{\sqrt{WW^T}}{\mathrm{Tr}\sqrt{WW^T}}. \tag{2.18}$$

When W is not fixed then for a given D the value of W can be obtained from the relation (2.23). In (2.18), the square root $\sqrt{D}$ of a matrix $D \in \mathcal{S}_+^\ell$ is defined to be the unique matrix $C \in \mathcal{S}_+^\ell$ such that $D = C^2$. The result (2.18) is stated below [3, Eq. (23)]. This is the key result that we are going to use in our block majorization method.

Formula (2.18) immediately suggests alternatively minimizing $Q(W,D)$ with respect to W and D. However, it is well known that the stress function, which is part of $Q(W,D)$, is a very complicated function (nonsmooth, nonconvex) to minimize. A widely adopted method is majorization method discussed in the following section.

2.3.2.1 Majorization method for solving MDS

Iterative majorization is an elegant minimization method which is based on the work of De Leeuw (1977) and well documented in the books by Borg and Groenen [10]. One of the main features of iterative majorization (IM) is that it generates a monotonically nonincreasing sequence of function values. If the function is bounded from below, we usually end up in a stationary point that is a local minimum. The central idea of the majorization method is to replace iteratively the original complicated function $h(x)$ by an auxiliary function $g(x,z)$, where z in $g(x,z)$ is some fixed value. The function g has to meet the following requirements to call $g(x,z)$ a majorizing function of $h(x)$.

- The auxiliary function $g(x,z)$ should be simpler to minimize than $h(x)$.

- The original function must always be smaller than or at most equal to the auxiliary function; that is, $h(x) \leq g(x, z)$.
- The auxiliary function should touch the surface at the so-called supporting point z; that is, $h(z) = g(z, z)$.

The iterative majorization algorithm minimizes the stress function that measures the deviance of the distances between points in a geometric space and their corresponding dissimilarities to determine the EDM such that the distances between the transformed points are close to the distances in the original space.

Now we will approximate the function $Q(W, D)$ by a simpler majorization function, which is less expensive to minimize.

For a given $V \in \Re^{\ell \times m}$ and $i, j = 1, \ldots, N$ define

$$c_{ij}(V) = \begin{cases} \alpha_{ij} d_{ij}/q_{ij}(V) & \text{if } q_{ij}(V) > 0 \\ 0 & \text{otherwise,} \end{cases}$$

and

$$B(V) = \sum_{i,j=1}^{N} c_{ij}(V)(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T \in \mathcal{S}^{\ell}.$$

Let

$$C = \sum_{i,j=1}^{N} \alpha_{ij}(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T.$$

Finally, let

$$\sigma_m^2(W, V) = \mathrm{Tr}(W^T C W) - 2\mathrm{Tr}(V^T B(V) W) + \sum_{i,j=1}^{N} \alpha_{ij} d_{ij}^2.$$

Then, $\sigma_m^2(W, V)$ satisfies the following relaxation properties:

$$\sigma^2(W) \leq \sigma_m^2(W, V) \qquad \forall\ W, V$$

and

$$\sigma^2(W) = \sigma_m^2(W, W).$$

Because of those properties, $\sigma_m^2(W,V)$ is called a majorization function of σ^2 at W. We note that $\sigma_m^2(W,V)$ is quadratic in W.

Now, define

$$Q_m(W,V,D) = \sigma_m^2(W,V) + \gamma\langle WW^T, D^\dagger\rangle.$$

Then Q_m is a majorization function of $Q(W,D)$ in the sense that

$$Q_m(W,V,D) \geq Q(W,D), \qquad \forall\ W,V,D \tag{2.19}$$

and

$$Q_m(W,W,D) = Q(W,D). \tag{2.20}$$

We note that

$$\begin{aligned} Q_m(W,V,D) &= \langle WW^T, C+\gamma D^\dagger\rangle - 2\langle V^T B(V), W\rangle \\ &\quad + \sum_{i,j=1}^{N} \alpha_{ij} d_{ij}^2. \end{aligned}$$

We are ready to present our block-majorization algorithm.

Algorithm 2.6. **Iterative Block-Majorization Method**

(S.0) Initialization: Choose $W^0 \in \Re^{\ell\times m}$ and $D^0 \in \mathcal{S}_+^\ell$. Let $k=0$.

(S.1) Set $V = W^k$ and update W^k by

$$W^{k+1} = \arg\min_{W\in\Re^{\ell\times m}} Q_m(W,V,D^k). \tag{2.21}$$

(S.2) Update D^k by

$$D^{k+1} = \arg\min_{D\in\mathcal{S}_+^\ell} Q(W^{k+1},D). \tag{2.22}$$

The following remarks are useful in understanding this algorithm.

(i) We note that the update D^{k+1} in (2.22) also satisfies

$$\begin{aligned} D^{k+1} &= \arg\min_{D\in\mathcal{S}_+^\ell} \gamma\langle W^{k+1}(W^{k+1})^T, D^\dagger\rangle \\ &= \arg\min_{D\in\mathcal{S}_+^\ell} Q_m(W^{k+1}, W^k, D). \end{aligned}$$

This view puts Algorithm 2.6 in the general framework of the block majorization method studied by de Leeuw[64] when specialized to (2.17). This justifies why we call the algorithm the iterative block-majorization method. General convergence properties of Alg. 2.6 can be similarly stated as in [64], which is discussed in section (2.3.3).

(ii) D^{k+1} can be computed through formula (2.18) with $W = W^{k+1}$. The computation of W^{k+1} is equivalent to solving the following equation:

$$\left(C + \gamma(D^k)^\dagger\right) W = B(W^k)W^k \tag{2.23}$$

with the positive semidefinite coefficient matrix $(C + \gamma(D^k)^\dagger)$.

(iii) In our implementation, we terminated the algorithm whenever there was no significant change in W or in $P(W)$. That is, whenever

$$\frac{\|W^{k+1} - W^k\|}{l^2} \leq \epsilon$$

or

$$\frac{|P(W^{k+1}) - P(W^k)|}{|P(W^k)|} \leq \epsilon$$

for a small tolerance $\epsilon > 0$, we stop the algorithm.

(iv) Alg. 2.6 can be straightforwardly applied to (2.9) by replacing D by $\mathrm{Diag}(\lambda)$ and updating λ by formula (2.10).

2.3.3 Convergence Analysis

As remarked in point (i) in the preceding section, Alg. 2.6 fits in the framework of the block majorization method of de Leeuw [64]. Hence, we can state its convergence in the

style of [64]. We include the basic convergence results with a brief proof.

Theorem 2.7. *We assume that condition (2.5) holds. Then following statements hold.*

(i) The sequence $\{W^k, D^k\}$ *is bounded.*

(ii) Suppose (W^∞, D^∞) *is an accumulation point of the sequence* $\{W^k, D^k\}$*. Then* (W^∞, D^∞) *is a minimizer of (2.13). Then we have*

$$\lim_{k\to\infty} Q(W^k, D^k) = Q(W^\infty, D^\infty)$$

Proof. (i) We have the following chain of inequalities

$$\begin{aligned} Q(W^{k+1}, D^{k+1}) &= Q_m(W^{k+1}, W^{k+1}, D^{k+1}) \text{ (by (2.20))} \\ &\leq Q(W^{k+1}, D^k) \quad \text{(by (2.22))} \\ &\leq Q_m(W^{k+1}, W^k, D^k) \quad \text{(by (2.19))} \\ &\leq Q_m(W^k, W^k, D^k) \quad \text{(by (2.21))} \\ &= Q(W^k, D^k). \quad \text{(by (2.20))} \end{aligned}$$

Hence the sequence of the function values $\{Q(W^k, D^k)\}$ is decreasing and is bounded below by 0 because of (2.8). By following the proof for Thm. 2.3, we can prove under the assumption (2.5) that $\{W^k\}$ is bounded. The boundedness of $\{D^k\}$ follows from the update formula (2.18) for D. We proved (i).

(ii) The claimed result is the straightforward consequence of the continuity of $Q(W, D)$ as the sequence of $\{Q(W^k, D^k)\}$ is monotonically decreasing. □

We conclude the section by noticing that [64] assumes that the feasible set of their problem is compact, whereas we need to prove the boundedness of the sequence under the assumption (2.5).

Though it seems hard to get a general condition that ensures the full span of the vectors to the whole space in (2.5) .

A simple example can be illustrated:

Suppose that there are $N = 10$ points, e.g., $x_1 = (1, 0, ..., 0), ..., x_{10} = (0, ..., 0, 1)$. If we choose $ell = 1$, then the vectors v^{ij}s easily satisfy the assumption (2.5).

2.4 Numerical Experiments

In this section, we first present a practical two-stage algorithm that utilizes Alg. 2.6. We then test the algorithm against three well-known benchmarking dataset *iris data set*, *cancer data set* and *seeds data set*, all from UCI machine learning repository[1]. We will demonstrate the effectiveness of our algorithm against the approach in [107] by projecting the dataseets in to a 2-dimensional space. We will also take a further step to apply existing support vector machine (SVM) algorithms in [97] (which is briefly discussed later of this section) to the obtained 2-dimensional datasets to show the significant improvement over Webb's model.

2.4.1 A Two-Stage Algorithm

The strong motivation in using the $(2, 1)$-norm $\|W\|_{2,1}$ in problem (2.7) is that the more important a center $\mathbf{c}_i$ is, the farther away of the ith row of W should be from origin. In other words, if the center $\mathbf{c}_i$ is more important than the center $\mathbf{c}_j$, it is then expected from the $(2, 1)$-norm regularization that

$$\|W_{i:}\| > \|W_{j:}\|.$$

This immediately suggests the following heuristic procedure for selecting the most important centers. Suppose $W \in \Re^{\ell \times m}$ is the final iterate of Alg. 2.6. We compute the length of each row of W: $\{\|W_{1:}\|, \ldots, \|W_{\ell:}\|\}$. We sort the sequence in decreasing order and denote the resulting sequence by

$$\{t_1,\ t_2, \ldots, t_\ell\} \quad \text{and} \quad T = \sum_{j=1}^{\ell} t_i,$$

[1] http://archive.ics.uci.edu/ml/

where T is the total length of the sequence.

Without loss of generality, we denote the corresponding sequence of centers by $\mathbf{c}_1, \ldots, \mathbf{c}_n$. The interpretation is that the centers are arranged in the order of decreasing importance. We then compute the cumulated percentage of the total length by the leading centers in the sequence:

$$p_i = \frac{\sum_{j=1}^{i} t_j}{T}, \quad i = 1, \ldots, \ell.$$

Obviously, $\{p_i\}$ is increasing and $p_\ell = 1$. Let p be a pre-set high percentage (e.g., $p = 95\%$) and Choose

$$\ell_0 = \min \{i : \ p_i \geq p, \ i = 1, \ldots, n\}. \tag{2.24}$$

We may think that the first ℓ_0 centers $\{\mathbf{c}_1, \ldots, \mathbf{c}_{\ell_0}\}$ account at least p percentage of the total effectiveness contributed by the ℓ centers. We expect that ℓ_0 would be much less than ℓ.

Having selected the ℓ_0 effective centers by (2.24), we proceed to solve the following optimization problem:

$$\min_{W \in \Re^{\ell_0 \times m}} \ \sigma^2(W). \tag{2.25}$$

We note that problem (2.25) is of the type of Webb's problem (2.4), but in a reduced dimension because $\ell_0 < \ell$. We summarize this two-stage algorithm as follows.

Algorithm 2.8. **Two-Stage Algorithm**

S.1 Apply Alg. 2.6 to get its final iterative matrix $W \in \Re^{\ell \times m}$. Use (2.24) to select the most important ℓ_0 centers.

S.2 Apply the iterative block majorization algorithm of Webb[107] to solve problem (2.25).

Note that, at the 1st stage the stress function is already minimized and it gives a good projection of data. As the initial value for the second stage is the final value of the 1st

stage and the stress function is decreasing so the use of 2nd stage minimizes the stress more and therefore makes the projection better.

2.4.2 Classifier of data

In pattern recognition, one of the most important data mining techniques is classification. In a classification task, a pattern is given and the task is to classify it into one out of c classes. The number of classes, c, is assumed to be known a priori. Each pattern is represented by a set of feature values, $x_i, i = 1, 2, \ldots, n$ which make up the n-dimensional feature vector $x = [x_1, x_2, \ldots, x_n]^T \in \Re^n$. It is assumed that each pattern is represented uniquely by a single feature vector and that it can belong to only one class. Different authors have worked with several techniques to determine the classifiers based on different approach such as Bayes decision theory, Gaussian probability density function, minimum distance classifier, the expectation maximization algorithm, learning from neighbors (eg. K-nearest neighbor), classifier based on cost function optimization (eg. support vector machine), etc. First part of our research is concerned with the classifiers based on cost function optimization. In the second part we worked with k-nearest neighbor rule.

The idea of the classifier based on cost function optimization is to design a discriminant function/decision surface that separates the classes in some optimal sense. The classifier can be linear or non-linear. Depending on the data set, where the classes may be totally separable or non-separable, a proper decision surface can be determined that separates the classes. The decision surface in the n-dimensional space is a hyperplane. Such a hyperplane may or may not be unique. There are different types of linear classifiers such as

- The Perceptron algorithm
- Least square method
- Logistic discrimination

- Support vector machine etc.

In this thesis we have mainly focused on Support Vector Machine (SVM) to separate the data of different classes. The advantage of using SVM over other methods is that it converges to the best possible hyperplane, i.e., the optimal hyperplane classifier of a support vector machine is unique, whereas other methods converge to any of the possible solutions.

2.4.3 Support Vector Machine

Support vector machine (SVM) is a supervised learning model with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. For any two classes of data, first step of SVM is to transform the original data (for learning) into a higher dimensional space using a nonlinear mapping. Next, it searches a hyperplane within this new dimension that leaves maximum margin from both classes so that datapoints can move freely. The original SVM algorithm was first established by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963 [105]. Theodoridis et al. in [98] discussed different type of SVMs with examples. In this section we have documented a brief description of SVMs.

Linear SVM

Separable classes: If the classes are separable then the goal is to design a hyperplane $g(x) = \mathbf{w}^T x + \mathbf{w}_0 = 0$ that classifies correctly all the training vectors where the vector $\mathbf{w}$ determines the direction and w_0 determines the position of the hyperplane. The distance of a point x from a hyperplane with direction $\mathbf{w}$ is $z = \frac{|g(x)|}{\|w\|}$. For any two classes w_1 and w_2 of points the value of w and w_0 can be scaled so that the value of $g(x)$ at the nearest points in the w_1 and w_2 is equal to 1 for w_1 and -1 for w_2. This is equivalent to

- Having a margin of length $\frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|}$
- Requiring that

$$w^T x + w_0 \geq 1, \ \forall x \in w_1$$

$$w^T x + w_0 \leq -1, \ \forall x \in w_2$$

Since we are looking for the direction **w** of the hyperplane that gives the maximum possible margin, note that minimizing the norm $\|w\|$ will maximize the margin. Thus for each feature vector x_i if we denote the corresponding class indicator by y_i ($+1$ for class w_1 and -1 for class w_2) then the mathematical form to get the optimal hyperplane using SVM becomes a nonlinear (quadratic) optimization problem given by:

$$\begin{aligned} \min \quad & J(w, w_0) = \tfrac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(w^T x + w_0) \quad \geq \quad 1 \ , \quad i = 1, 2, \ldots, N, \end{aligned} \tag{2.26}$$

with a set of linear inequality constraints. This problem can be solved using Lagrangian function. In the optimal solution the vector parameter w is a linear combination of $N_s \leq N$ feature vectors known as support vectors and the optimal hyperplane classifier is known as support vector machine (SVM).

Non-Separable classes: If the classes are nonseparable then the optimal hyperplane always contains some data points in the class separation band. The training feature vectors of two nonseparable classes belong to one of the three categories:

- Vectors that fall outside the band and classified correctly satisfying the inequality $y_i(w^T x + w_0) \geq 1$
- Vectors that fall inside the band and classified correctly satisfying the inequality $0 \leq y_i(w^T x + w_0) \leq 1$
- Vectors that are misclassified obey the inequality $y_i(w^T x + w_0) < 0$.

Now the above three inequalities can be described by a single constrain $y_i(w^T x + w_0) \geq 1 - \mu_i$ where $\mu_i = 0$ for the first category, $0 < \mu_i \leq 1$ for the second one and for the third category $\mu_i > 1$. The goal for nonseparable classes of data is to maximize the margin but at the same time to keep the number of points with $\mu_i > 0$ as small as possible.

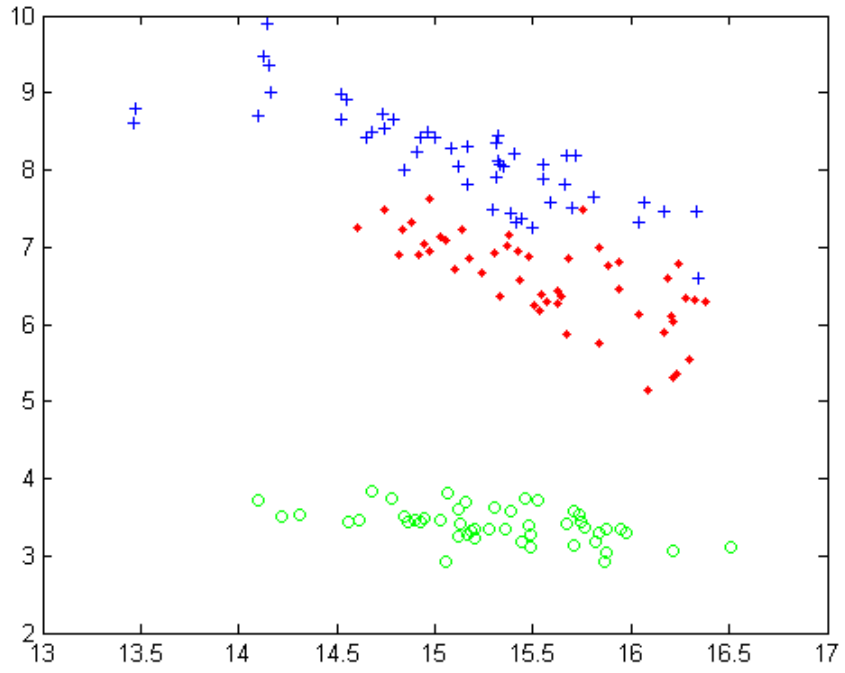


Figure 2.2: Iris data projected in 2-dimensional space, The data consists of 3 classes, one class represented by "o" is completely separated from the other two, represented by "+" and "⋄".

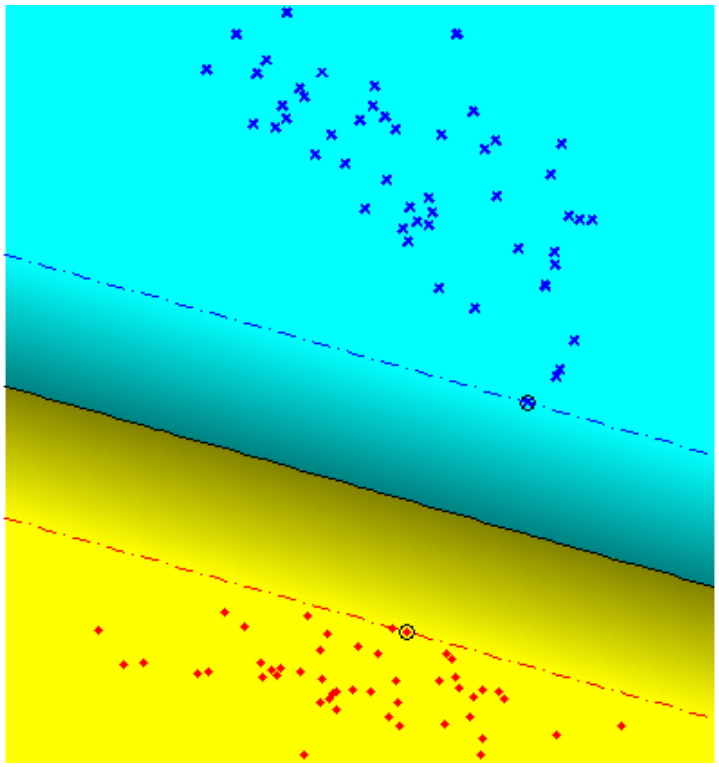
Thus the mathematical formulation of such an optimization problem becomes

$$\begin{aligned} \min \quad & J(w, w_0, \mu) = \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \mu_i \\ \text{s.t.} \quad & y_i(w^T x + w_0) \geq 1 - \mu_i \ , \quad i = 1, 2, \ldots, N, \\ & \mu_i \geq 0 \ , \quad i = 1, 2, \ldots, N, \end{aligned} \tag{2.27}$$

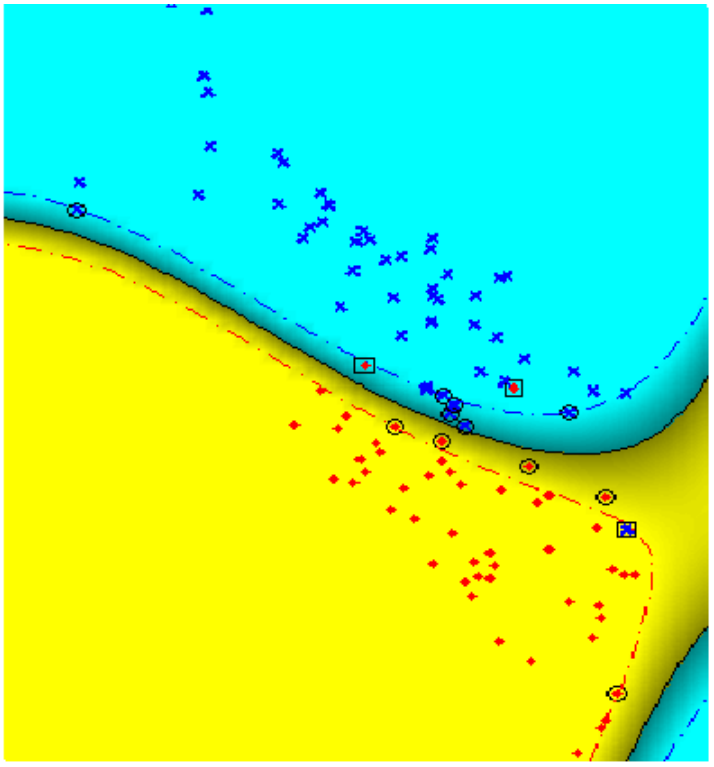
which is also a convex programming problem where C is a positive constant that controls the relative influence of the two competing terms.

Non Linear SVM

In the case of nonlinear SVM, to determine the hyperplane the feature vectors are mapped into a higher dimensional space, where the classes are expected to be linearly separable. The mapping is $x \mapsto \phi(x) \in H$, where the dimension of H is higher than the dimension of the feature vectors. The function $\phi(x)$ is chosen such that $\langle \phi(x), \phi(y) \rangle = K(x, y)$, where $\langle ., . \rangle$ denotes the inner product operation in H and $K(., .)$ is a function



(a) Linear SVM on separable classes of Iris data.



(b) Nonlinear SVM on nonseparable classes of Iris data.

Figure 2.3: (a) The separation of the two separable classes by a linear SVM. (b) The separation of the two nonseparable classes by a non linear SVM. Support vectors are bounded by "O" and misclassified points are bounded by □.

known as kernel function. Some examples of nonlinear kernels used in nonlinear SVM are as follows:

- The polynomial function of degree p: $K(x, y) = (\langle x, y\rangle + 1)^p$.
- The radial basis function: $K(x, y) = \exp(-\|x - y\|^2/2\sigma^2)$.
- Hyperbolic tangent: $K(x, y) = \tanh(\beta\langle x, y\rangle - \delta)$.

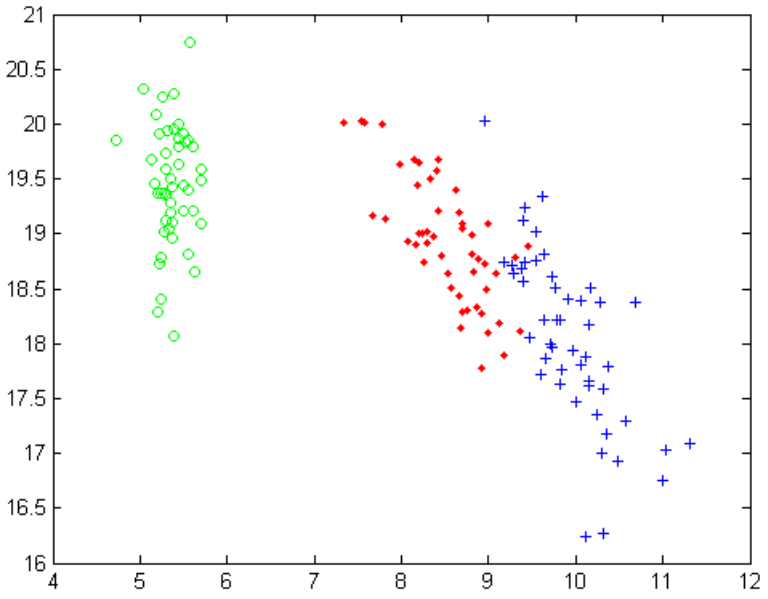
where σ in radial basis function and β and δ in the hyperbolic tangents function are user defined parameters. Solving the linear problem in higher dimensional space is equivalent to solve the nonlinear problem in original space. In our work, we have used nonlinear SVM with radial basis kernels.

Multiclass Clasifier: Although the SVM classiffers was mainly designed for binary classes, they are easily combined to handle the multiclass case. A simple, effective combination trains N one-versus-rest classifiers (say, one positive, rest negative) for the N-class case and for a test point the class corresponds to the largest positive distance. For details literature on SVM one can consult [98].

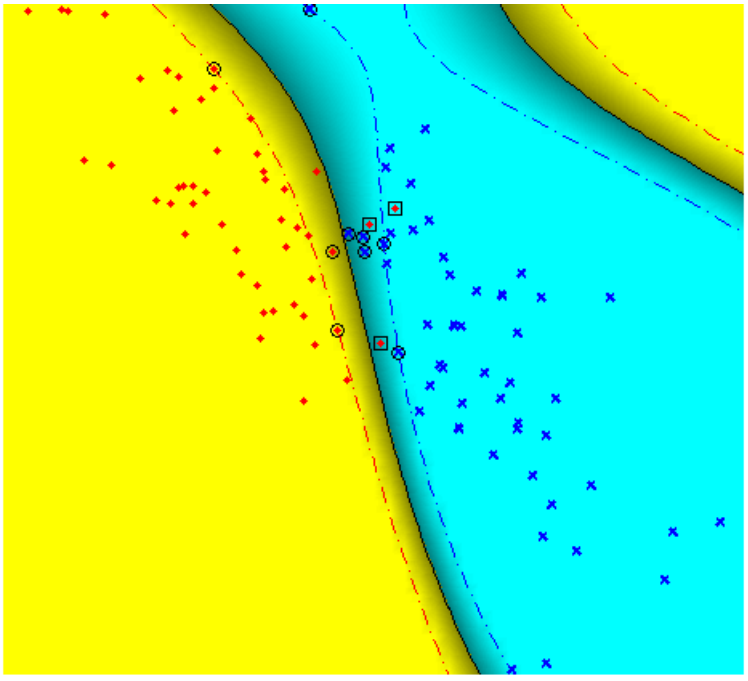
Example 2.1. *Consider the Iris data set. It consist of 150 data from three classes, each class containing* 50 *samples. Each data item consists of four different real values and each value represents an attribute of each instance such as length and width of sepal or petal. One class is known to be linearly separable from the other two, which are not linearly separable from each other.*
The projection of this 4*-dimensional data set into a* 2*-dimensional space is given in Fig. 2.2. Linear SVM is applied on the separable classes and Nonlinear SVM is used on the nonseparable classes to determine the missclassified points as shown in Fig. 2.3.*

2.4.4 Parameter Setting and Performance Indicators

In the numerical experiment, the weight matrix W was initialized with random values, where W_{ij} are distributed uniformly over the range $[0, 1]$. The tolerance $\epsilon = 10^{-4}$ is

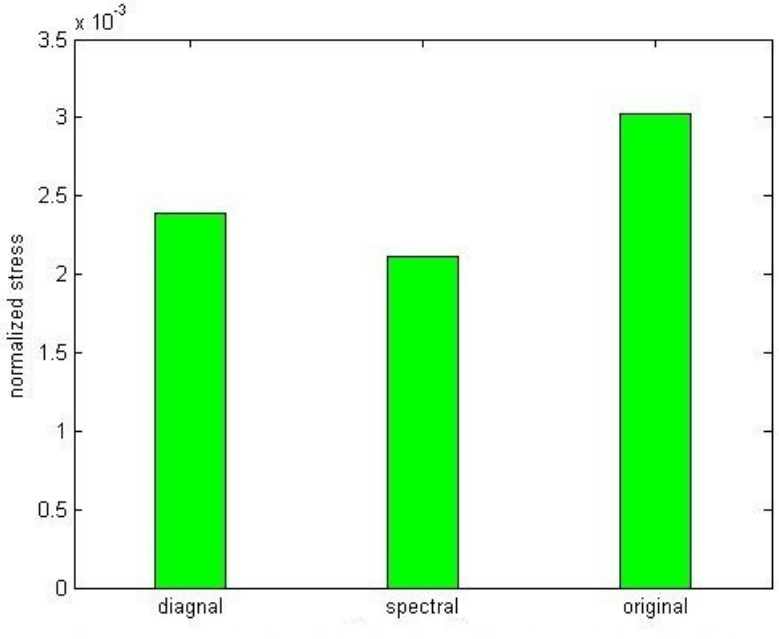


(a) 2-dimensional projection of Iris data using `RMDS-S`

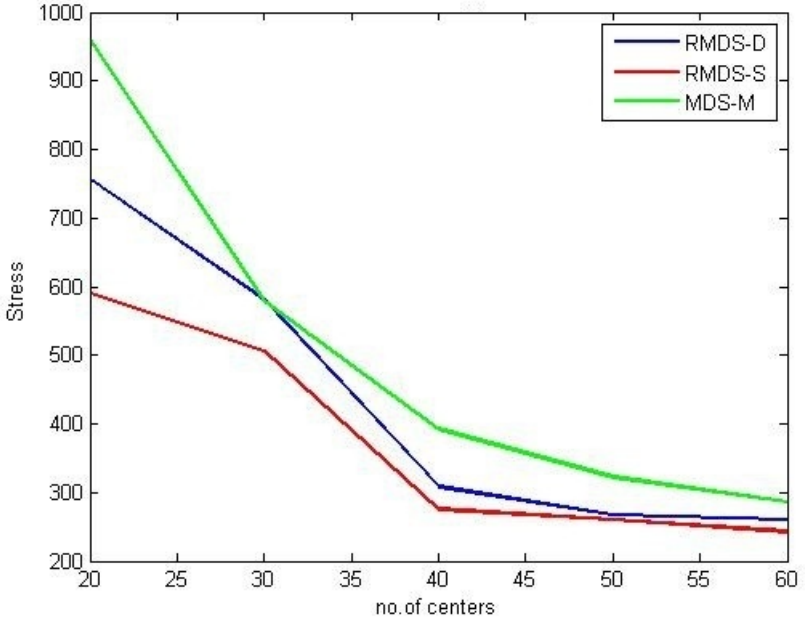


(b) SVM on Iris data projected by `RMDS-S`

Figure 2.4: (a) Projected 2-dimensional Iris data, consisting of 3 classes. One class represented by "o" is completely separated from the other two, represented by "+" and "$\diamond$". (b) Separation of the nonseparable two classes by a support vector machine algorithm. Over 100 runs, our model (e.g., `RMSD-S`) yielded about an average of 12 support vectors (bounded by "O") and 3 misclassified points (bounded by $\square$), while the corresponding numbers for Webb's model are 18 and 6 respectively.



(a) Average normalized stress over 100 runs on Iris data



(b) Comparison of stress values of Iris data against selected centers

Figure 2.5: (a) Comparison of the average normalized stress values for the three models `RMSD-D`, `RMSD-S` and `MDS-M` over 100 random runs with 30 selected centers. (b) Comparison of stress values when the number of centers (ℓ) varies.

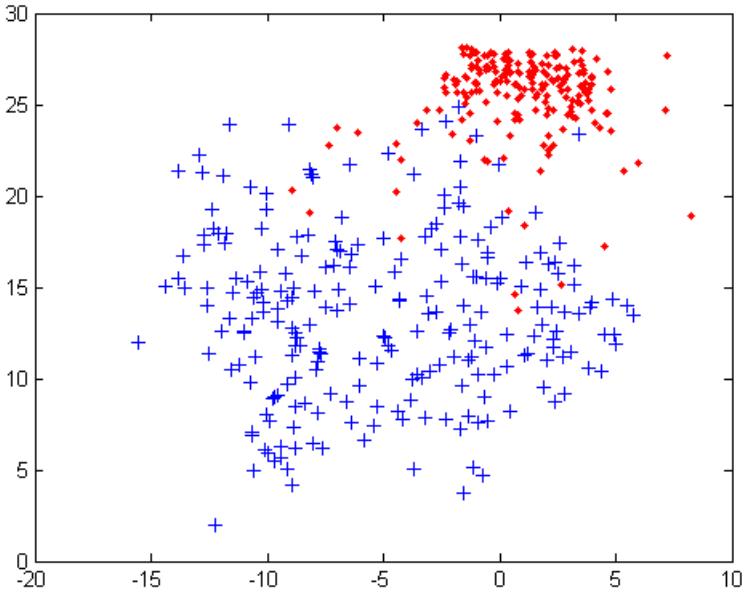
chosen for terminating the both stages of Alg. 2.8 by the rules in Remark (iii) on Alg. 2.6. The bandwidth parameter $h^2 = 10.0$ is taken from [107]. α_{ij} were taken to be unity. The penalty parameter γ is 1. Singular Value Decomposition is used to calculate the pseudoinverse of the matrices. We set $p = 95\%$ in (2.24). For each of the data sets, a random of 20% of the data was initially selected as centers. In order to speed up our algorithm, the maximum number of iterations for the first stage in Alg. 2.8 is set at $\lfloor 0.2N \rfloor$, where N is the number of data samples in the data set and $\lfloor 0.2N \rfloor$ is the largest integer not greater than $0.2N$. Throughout, we set $m = 2$, which means that the original data was scaled to a data set in 2 dimensions.

Two versions of Alg. 2.8 were compared with the majorization algorithm of [107], which is denoted by `MDS-M` for ease of comparison. One version refers to the case when the diagonal model (2.9) is used in (S.1) of Alg. 2.8. We denote this version by `RMDS-D`. The other version refers to the case when the spectral model (2.13) is used and is denoted by `RMDS-S`. We applied the three algorithms to each of the data sets. The results presented below were the average results on 100 runs, each of which had independent random initialization of the parameters (i.e., W and centers) involved. Four quantities were calculated: `It` (number of iterations), σ^2 (the final stress), σ_n^2 (the final normalized stress), and `cpu` (time used). The normalized stress is widely used and its definition can be found in [10, p.42, Eq. (3.10)] (see the comments therein for justification of this quantity in explaining data):

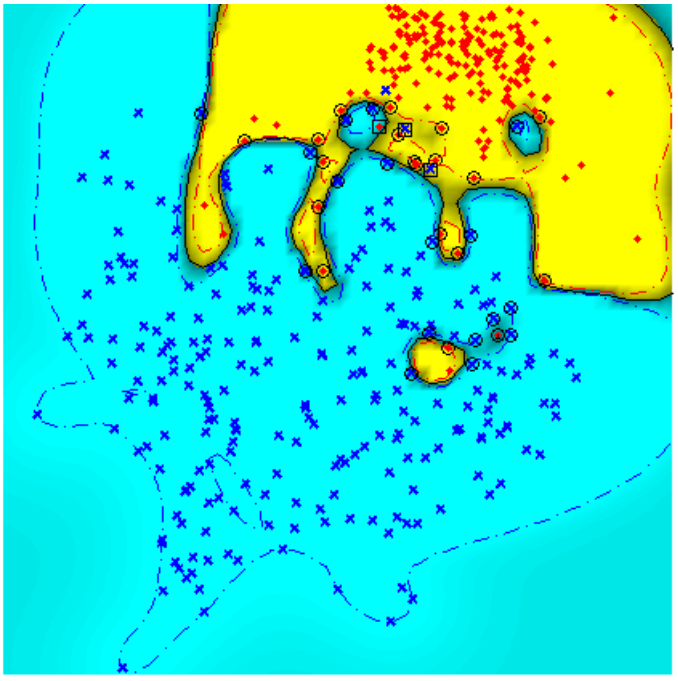
$$\sigma_n^2(W) = \frac{\sum_{i,j=1}^{N} \alpha_{ij}(q_{ij}\,(W) - d_{ij})^2}{\sum_{i,j=1}^{N} d_{ij}^2}.$$

2.4.5 Numerical Performance

In this subsection, we will demonstrate the good performance of Alg. 2.8 on the selected datasets, each of which will be projected to a 2-dimensional dataset (i.e., $m = 2$). We are going to use a number of graphs to show its behavior in `CPU` time, normalized stress as well as stress values. We will also take a further step to apply existing support vector

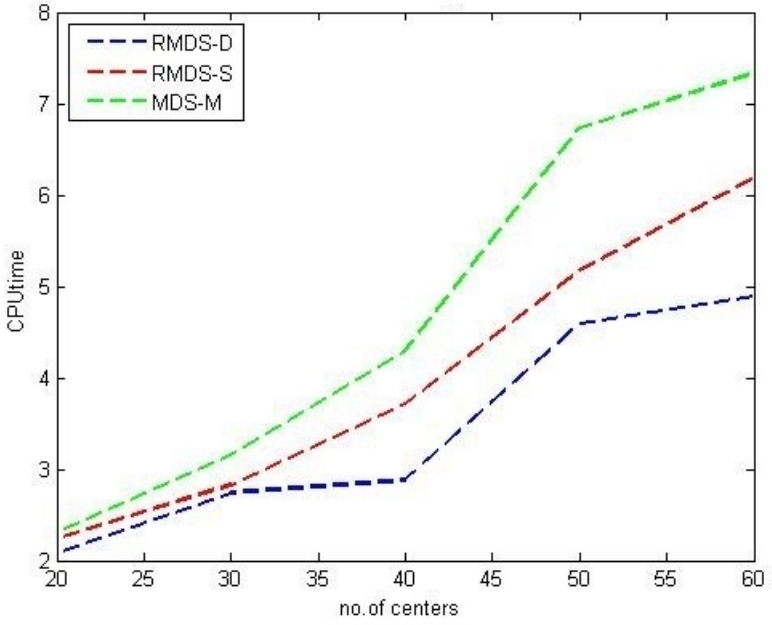


(a) 2-dimensional projection of Cancer data by `RMSD-S`

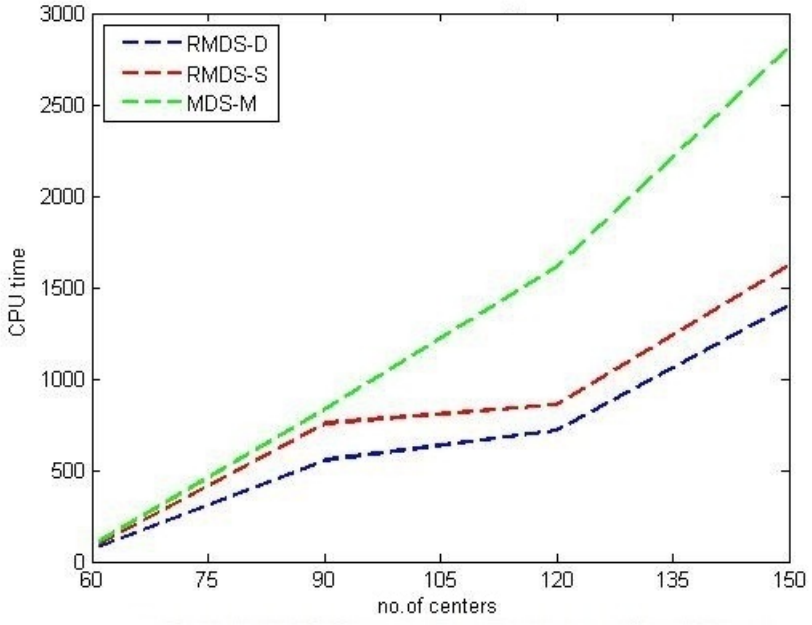


(b) SVM on Cancer data projected by `RMDS-S`

Figure 2.6: (a) Cancer data set projected in two dimensional space by `RMDS-S`. (b) shows the SVM separation on the projected Cancer data.Over 100 runs, our model (e.g., `RMSD-S`) yielded about an average of 5 misclassified points (bounded by □), while the corresponding numbers for Webb's model are 9.

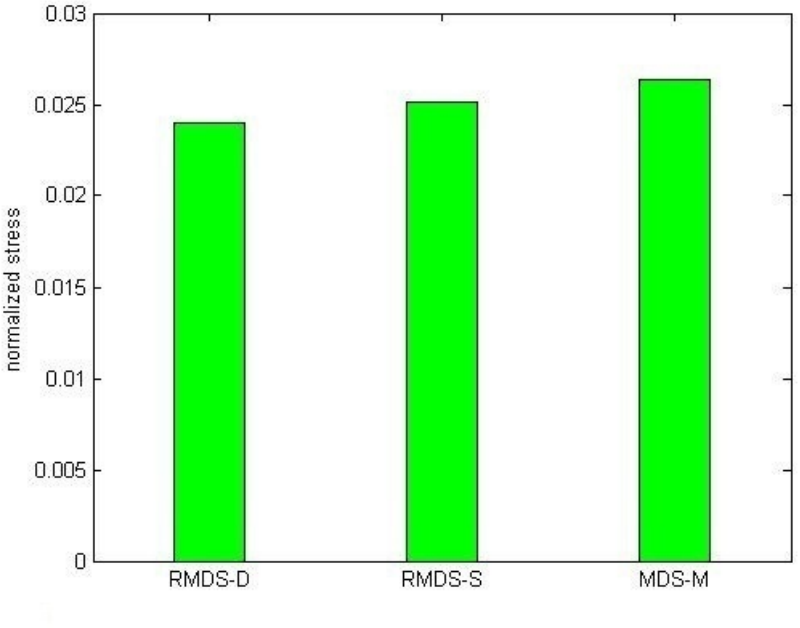


Total CPU time taken by three algorithms for Iris data

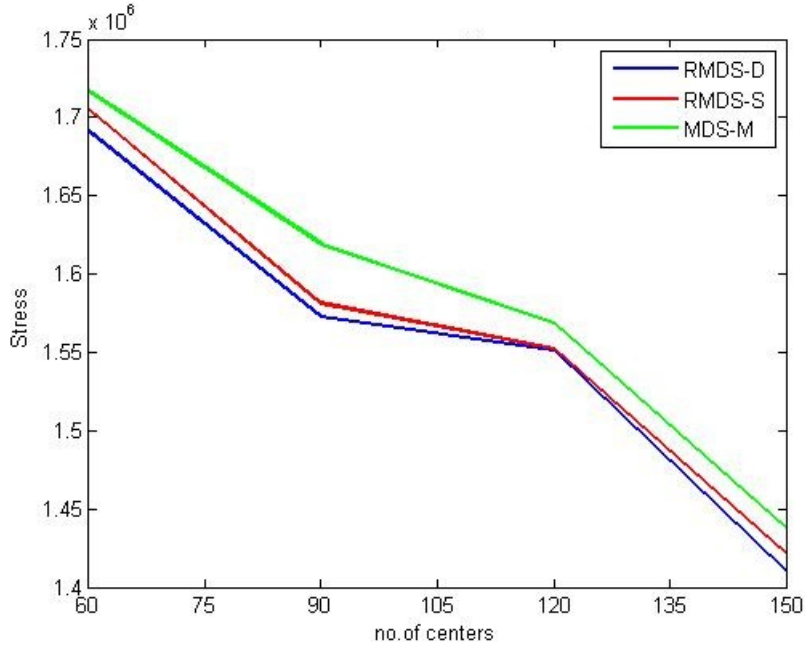


Total CPU time taken by three algorithms for Cancer data

Figure 2.7: CPU time comparison by `RMDS-D`, `RMDS-S`, and `MDS-M` on Iris and Cancer datasets when the number of centers varies.



(a) Average normalized stress over 100 runs on Cancer data



(b) Comparison of stress values of Cancer data when the number of centers changes

Figure 2.8: (a) Comparison of the average normalized stress values for the three models `RMSD-D`, `RMSD-S` and `MDS-M` over 100 random runs with 60 selected centers. (b) is the comparison of stress values when the number of centers (ℓ) varies.

machine (SVM) algorithms in [97] to the obtained 2-dimensional datasets to show the significant improvement over Webb's model. All tests were carried out using the 64-bit version of MATLAB R2013a on a Windows 7 desktop with 64-bit operating system having Intel(R) Core(TM) 2 Duo CPU of 3.16GHz and 4.0GB of RAM.

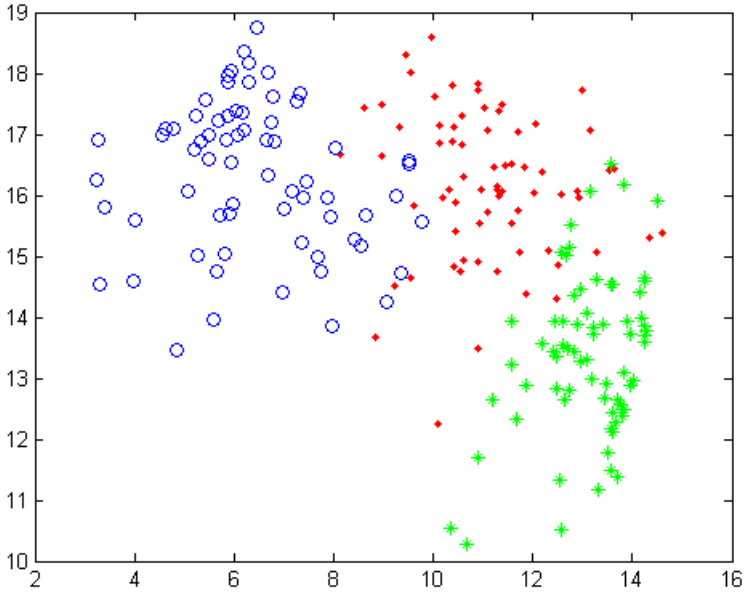
(a) Iris Data. It is a very known data set used in pattern recognition literature. This data set consists of data from three classes, each has 50 samples. Each data item consists of four different real values and each value represents an attribute of each instance such as length and width of sepal or petal. One class is known to be linearly separable from the other two, which are not linearly separable from each other. Our purpose is to represent this 4-dimensional dataset as a 2-dimensional dataset.

For this purpose, we started with randomly selected 30 initial centers. At the first stage, our methods `RMDS-D` and `RMDS-S` select an average of 22 centers. 2-dimensional projection of Iris data is shown in Fig. 2.4(a), which clearly shows that one class is totally separable from other two classes. SVM algorithm [98, Sect. 18, Chap. 4] is applied to the two non-separable classes. Our models yielded an average of 13 support vectors and 3 misclassified points, while for the original model the number of support vectors is between 18 and 6 points are misclassified. Fig. 2.4(b) illustrates SVM classification of Iris data obtained by `RMDS-S`. General performance information on 100 random run on the dataset can be found in Table 2.2.

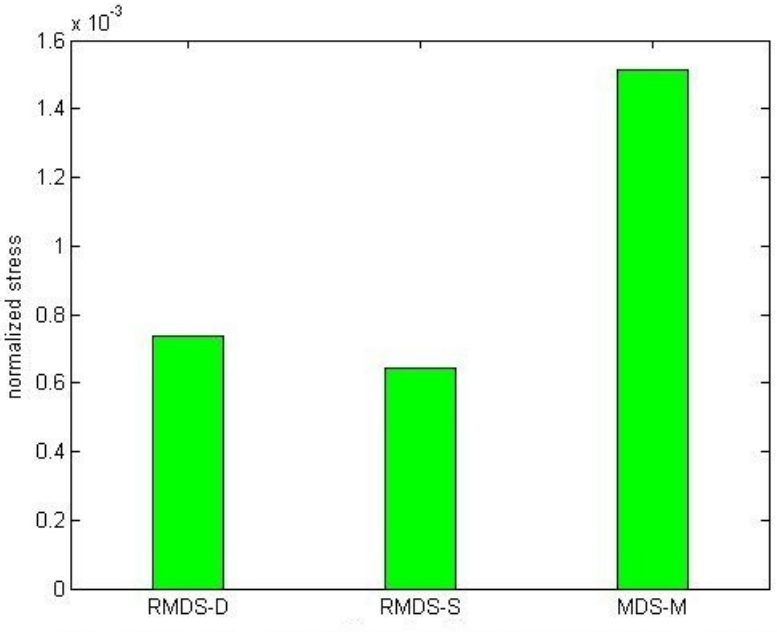
Table 2.2: Average performance of 100 runs for Iris data

Method	CPU Time (sec)	Iteration	Stress	Normalized stress
RMDS-D	3.28	71.50	487.67	0.0024
RMDS-S	4.03	92.30	432.52	0.0021
MDS-M	4.02	112.10	617.15	0.0030

In Fig. 2.5(a), The mapping quality of the constructed configurations of Iris data by `RMDS-D`, `RMDS-S` and `MDS-M` is compared in terms of the average normalized stress values



(a) 2-dimensional projection of Seeds data by `RMSD-S`



(b) Average normalized stress over 100 runs on Seeds data

Figure 2.9: (a) 2-D projection of Seeds data. (b) Comparison of the average normalized stress values for the three models `RMSD-D`, `RMSD-S` and `MDS-M` over 100 random runs with 40 selected centers.

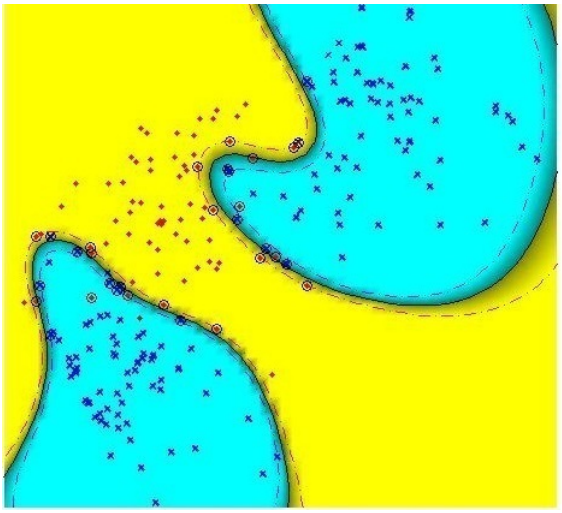
among 100 random runs each selecting 30 centers out of 60 random data points. Numerically, `RMDS-D` and `RMDS-S` improve mapping quality by 20% and 30% over `MDS-M` respectively in terms of the average stress value, which can be verified from Table 2.2.

Fig. 2.5(b) illustrates that the proposed methods outperformed `MDS-M` in terms of stress value when the same number (ℓ) of centers were selected from 100 random data points. The stress value decreases as the number of center increases for each of the three methods. CPU times taken by the three algorithms were plotted in Fig. 2.7.

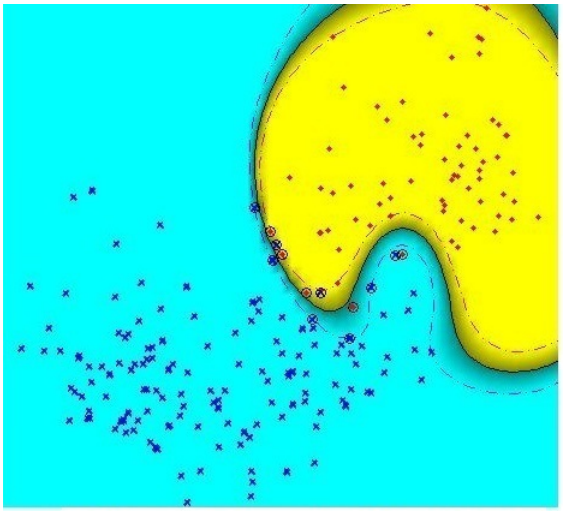
(b) Cancer Data. The cancer data set is another well-known data set used by many researchers. It has two classes (benign and malignant). Each data item consists of 11 columns and the first and the last column respectively represents ID number and class information of the item. The remaining 9 columns are attribute values described in integer from 1 to 10. It contains 699 data items and 16 of them have some missing values. So we used 683 data items which have every attribute values. For this data set, the proposed algorithm selects an average of 52 effective centers from 60 randomly selected centers. The two dimensional projection of the 9 dimensional dataset using `RMDS-S` is given in Fig. 2.6.

The number of support vectors for this dataset projected by proposed methods is an average of 54 whereas for the original model this number is 64 and the number of misclassified points are respectively 5 and 9. This shows that our methods improves the projection of the data and can separate the points of different classes better than the original model would do. Table 2.3 compares the average performance of 100 runs of the three methods for the cancer data using 60 centers out of 100 randomly selected points.

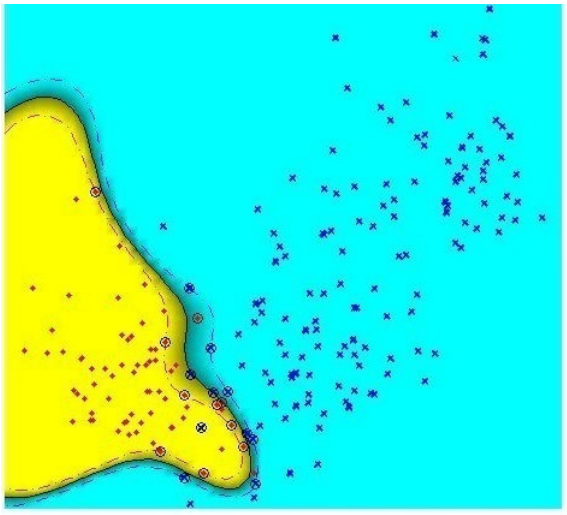
It can be seen that though the proposed methods take a little more time than the original method, both the stress and the normalized stress values (Fig. 2.8) by `RMDS-D` and `RMDS-S` are lower than that by the original method.



(a) SVM on Seeds data projected by RMDS-S. Class 1 - against - Class 2 and Class 3.



(b) SVM on Seeds data projected by RMDS-S. Class 2 - against - Class 1 and Class 3.



(c) SVM on Seeds data projected by RMDS-S. Class 3 - against - Class 1 and Class 2.

Figure 2.10: SVM on Seeds data projected in 2 dimensional space by RMDS-S is shown in these figures. Where the separation of the classes are shown using multiclass classifier.

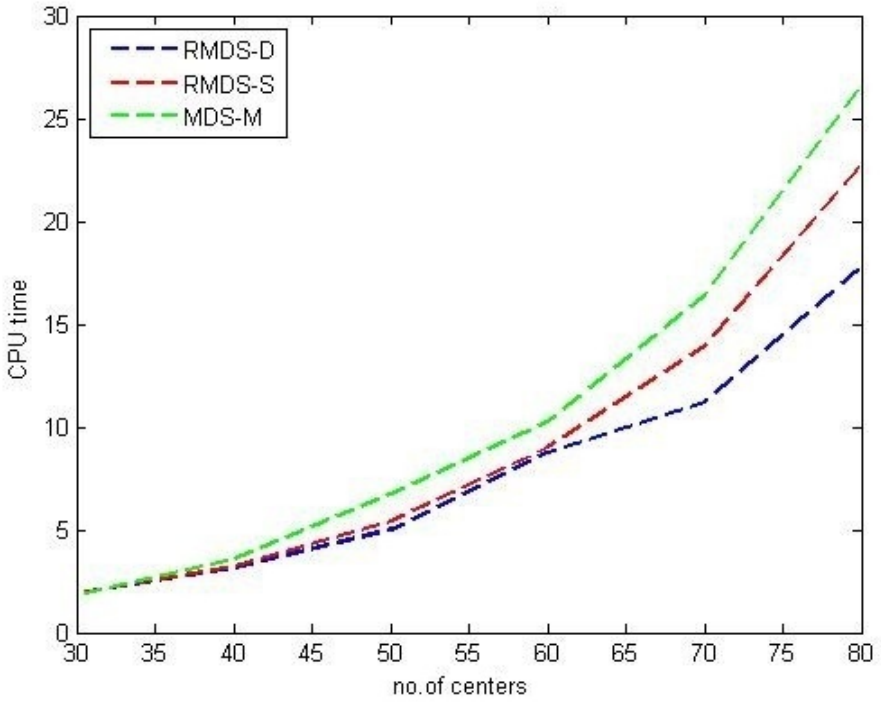


Figure 2.11: CPU time comparison by `RMDS-D`, `RMDS-S`, and `MDS-M` on Seeds datasets when the number of centers varies.

Table 2.3: Average performance of 100 runs for Cancer data

Method	CPU Time (sec)	Iteration	Stress	Normalized stress
RMDS-D	264.58	105.4	1.5962 e^{06}	0.0240
RMDS-S	231.14	89.3	1.6636 e^{06}	0.0251
MDS-M	217.90	103.0	1.7446 e^{06}	0.0264

(c) Seeds Data The seed data set is composed of 210 entities and each entity is represented by 7 real-valued attributes in addition to the class level contained in the last column. There are three classes, 70 points in each, representing three different varieties of wheat: Kama, Rosa and Canadian. We have selected 40 centers initially and the number of effective centers selected by our algorithm is 33.

As there are three classes, we applied *one-against-all* support vector machine algorithm as shown in Fig. 2.10 to determine the misclassified data. Over 100 runs the average numbers of support vectors obtained by our algorithm are respectively 35, 16, 20 and

that of misclassified points are 10, 3 and 5. The corresponding numbers for the original model are 42, 20, 24 and 12,5 and 8. The normalized stress value comparison is illustrated in Fig. 2.9(b). The bar graph illustrates the average normalized stress value of 100 runs with 40 selected centers from 80 random initial points obtained by `RMDS-D`, `RMDS-S` and `MDS-M`. Our methods improve about 54-60% over the original model, which can also be verified from Table 2.4. We note that for each of the tested data sets, as the number of center increases, our methods with a high percentage of selections (e.g., 95%) are less time consuming than `MDS-M`. This is demonstrated in Fig. 2.7 and Fig. 2.11.

Table 2.4: Average performance of 100 runs for Seeds data

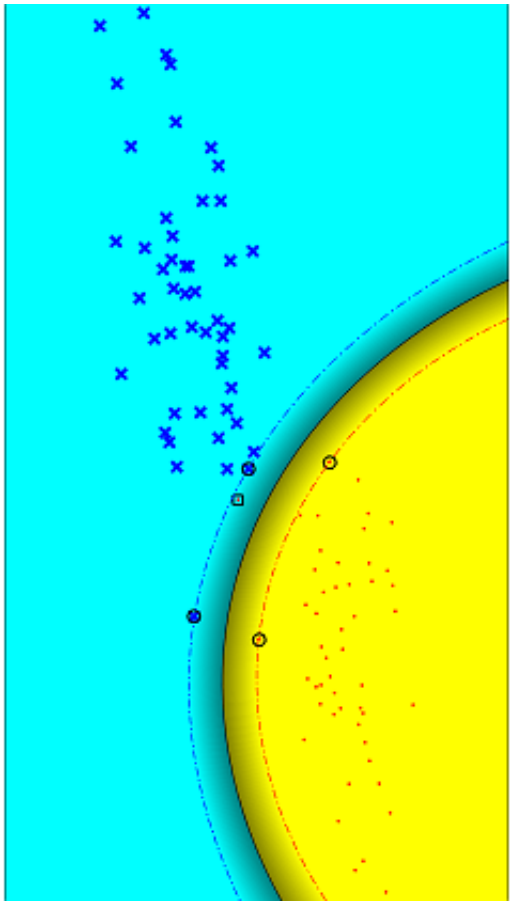
Method	CPU Time (sec)	Iteration	Stress	Normalized stress
RMDS-D	11.54	62.00	843.7	0.0007
RMDS-S	12.35	68.00	732.7	0.0006
MDS-M	8.72	59.20	1727.1	0.0015

2.5 Discriminant Analysis:

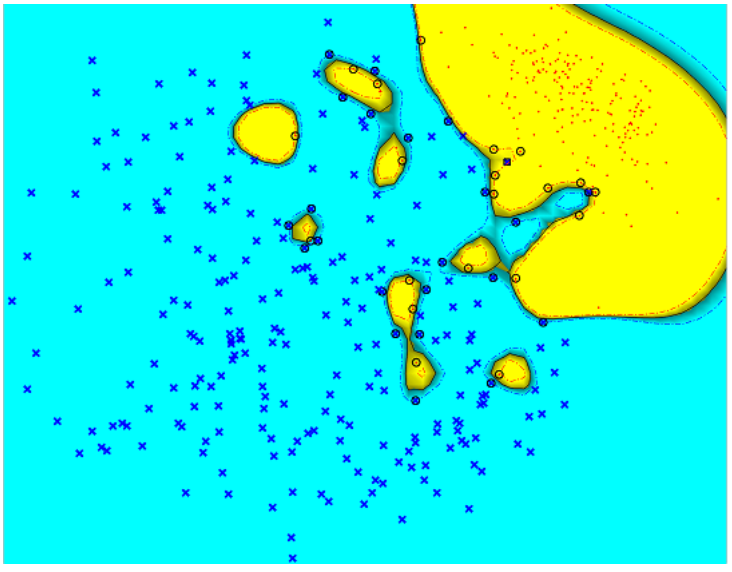
We note that our models discussed in this chapter do not take any advantages of some priori information concerning the data sets. For example, some data points may be known beforehand to belong to certain class. Hence, it would be interesting to include a discriminate analysis in our models. Multidimensional scaling techniques to discriminant analysis have been considered by several authors [107, 58, 24, 68]. Webb [107] defined an optimization criterion that is the sum of two terms: a class separability criterion and a structure-preserving stress term which is similar to that of Koontz and Fukunaga [58]

In this section we will discuss the improvement our models with discriminant analysis in the objective function.The objective function with the class separability term can be defined by :

$$J = (1 - \lambda)J_{SE} + \lambda J_{SP}$$



(a) SVM on Iris data projected using discriminant analysis



(b) SVM on Cancer data projected using discriminant analysis

Figure 2.12: SVM on iris data and cancer data projected in 2 dimensional space using discriminant analysis . Each of these datasets have just one misclassified point (square bordered))

where J_{SE} is a class separability criterion, J_{SP} is a structure-preserving stress term and $\lambda(0 \leq \lambda < 1)$ determines the relative effects of these two terms. A value of $\lambda = 1.0$ gives the standard multidimensional scaling criterion with no class information. At the other extreme, $\lambda = 0$ means that emphasis is on class separability. Define the separability criterion

$$J_{SE} = \sum_{i,j=1}^{N} \delta(i,j)\alpha_{ij}q_{ij}^2 \tag{2.28}$$

where q_{ij} are the distances in the transformed space defined by equation 2.3. Define $\delta(i,j)$ by

$$\delta(i,j) = \begin{cases} 1 & \text{if } i \sim j(x_i \textit{ and } x_j \textit{ belongs to same class}) \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\alpha_{ij} = \frac{\frac{1}{d_{ij}(X)}}{\sum_{i,j=1}^{N}(\frac{1}{d_{ij}(X)})} \tag{2.29}$$

We define the second term J_{SP} by

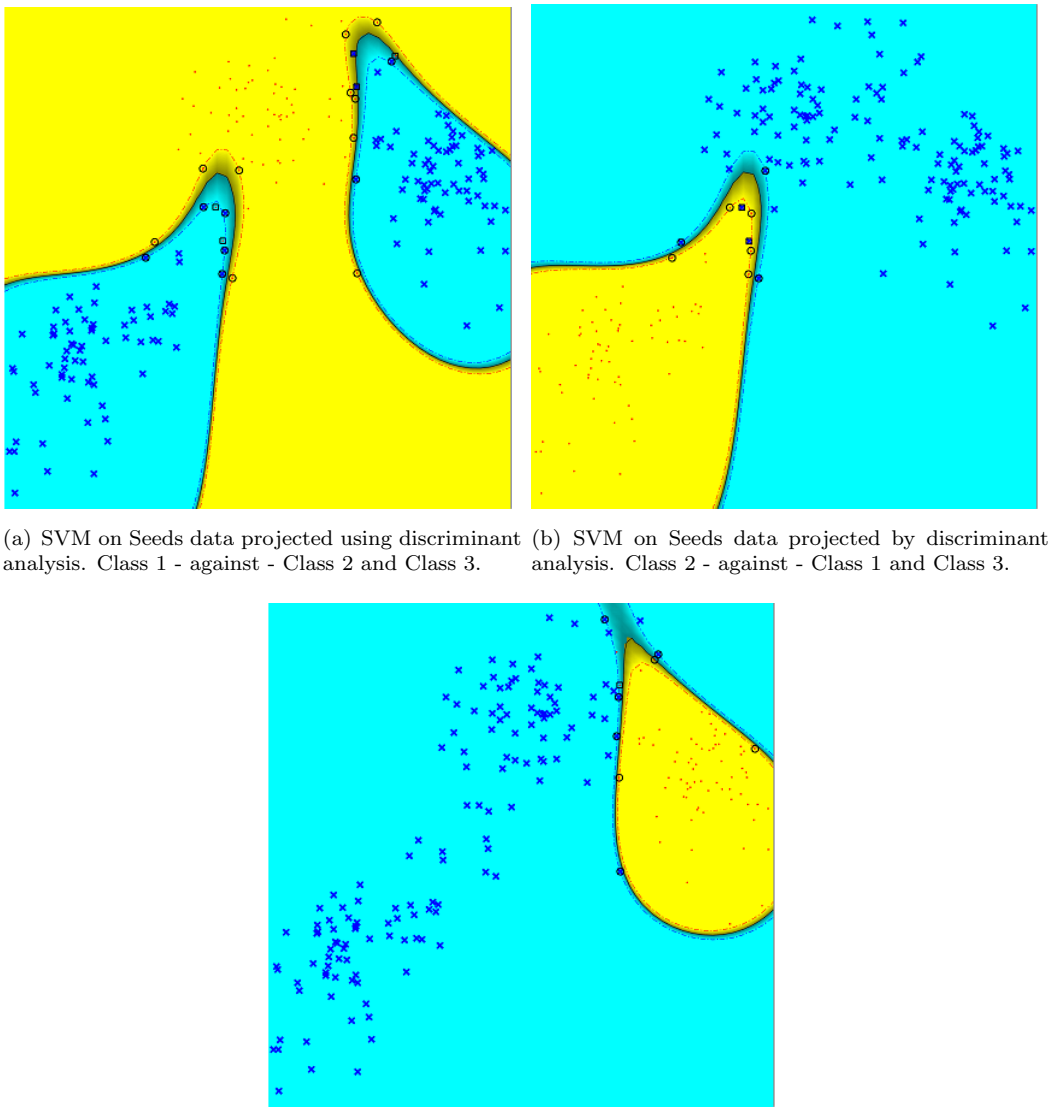
$$J_{SP} = Q(W,D) = \sigma^2(W) + \gamma\langle WW^T, D^\dagger\rangle$$

where $\sigma^2(W) = \sum_{i,j=1}^{N} \alpha_{ij}(q_{ij}(W) - d_{ij})^2$ is identical to the loss function (2.2) apart from the weights $\alpha_{ij} = \alpha_{ij}(X)$ given by equation α. The parameter λ controls the relative importance of the structure preserving term to the class separability criterion. Therefore the objective function J takes the form

$$J(W) = \lambda(\sigma^2(W) + \gamma\langle WW^T, D^\dagger\rangle) + (1-\lambda)(\sum_{i,j=1}^{N} \delta(i,j)\alpha_{ij}q_{ij}^2)$$

for $\alpha_{ij} > 0$. This can be written as

$$\begin{aligned} J &= \sum_{i,j=1}^{N} \alpha_{ij}((1-\lambda)\delta(i,j)+\lambda) \times \left(q_{ij} - \frac{\lambda}{(1-\lambda)\delta(i,j)+\lambda} d_{ij}(X)\right)^2 \\ &+ \lambda \sum_{i,j=1}^{N} \alpha_{ij} \left(1 - \frac{\lambda}{((1-\lambda)\delta(i,j)+\lambda)^2}\right) d_{ij}^2 + \lambda\gamma\langle WW^T, D^\dagger\rangle \end{aligned}$$



(a) SVM on Seeds data projected using discriminant analysis. Class 1 - against - Class 2 and Class 3.

(b) SVM on Seeds data projected by discriminant analysis. Class 2 - against - Class 1 and Class 3.

(c) SVM on Seeds data projected by discriminant analysis. Class 3 - against - Class 1 and Class 2.

Figure 2.13: SVM on Seeds data projected in 2 dimensional space by discriminant analysis is shown in these figures, where the separation of the classes are shown using multiclass classifier.

Denoting $\bar{\alpha}_{ij} = \alpha_{ij}((1-\lambda)\delta(i,j)+\lambda)$ and $\bar{d}_{ij} = \frac{\lambda}{(1-\lambda)\delta(i,j)+\lambda} d_{ij}$ and ignoring the second summation (as it is independent of q_{ij}), we have, the minimization of J is equivalent to the minimization of

$$J_e = \sum_{i,j=1}^{N} \bar{\alpha}_{ij}(q_{ij} - \bar{d}_{ij})^2 + \lambda\gamma\langle WW^T, D^\dagger\rangle \tag{2.30}$$

The first term of (2.30) is of the same form as the stress term σ^2 and thus J_e has the same form of $Q(W, D)$.

Therefore we can minimize J_e by our proposed algorithm with different definition of the matrices C and $B(V)$. The matrix $B(V)$, for a given $V \in \Re^{\ell \times m}$ is:

$$B(V) = \sum_{i,j=1}^{N} c_{ij}(V)(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T \in \mathcal{S}^{\ell}.$$

with

$$c_{ij}(V) = \begin{cases} \lambda \alpha_{ij} d_{ij}/q_{ij}(V) & \text{if } q_{ij}(V) > 0 \\ 0 & \text{otherwise,} \end{cases}$$

and C is defined by

$$C = \sum_{i,j=1}^{N} \alpha_{ij}((1-\lambda)\delta(i,j) + \lambda)(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j))^T.$$

Table 2.5: Numerical results obtained by applying SVM on three datasets projected using discriminant analysis.

Dataset	Support vector	Missclassified points	Improvment
Iris	5	1	66%
Cancer	47	1	80%
Seeds (C1)	23	5	50%
Seeds (C2)	10	2	33%
Seeds (C3)	9	2	60%

Finally, the mejorizing function $\sigma_m^2(W, V)$ of $\sigma^2(W, V)$ becomes

$$\sigma_m^2(W, V) = \mathrm{Tr}(W^T C W) - 2\mathrm{Tr}(V^T B(V) W) + \sum_{i,j=1}^{N} \alpha_{ij} d_{ij}^2.$$

Therefore we have

$$Q_m(W, V, D) = \sigma_m^2(W, V) + \lambda\gamma\langle WW^T, D^{\dagger}\rangle.$$

Here Q_m is the majorization function of $Q(W, D)$ in the sense that

$$Q_m(W, V, D) \geq Q(W, D), \qquad \forall\ W, V, D \tag{2.31}$$

and

$$Q_m(W, W, D) = Q(W, D). \tag{2.32}$$

Applying Alg. 2.6 we obtain the value of W that minimizes Q_m satisfying equation (2.23).

Here we will apply this approach on the data sets iris, cancer and seeds. Numerical experiments show that with the choices of the matrices B and C that incorporate class information , the projection quality improves $60 - 70\%$ in terms of the misclassified points that we obtain applying SVM on the projected dataset as shown in Fig. 2.12(a), Fig. 2.12(b) and Fig. 2.13 and also reported in Table 2.5
Note that the choices for α_{ij} and δ are quite general and the procedure may be used for other forms than those given in this section.

2.6 Summary

In this chapter we have studied one of the important nonlinear variants of classical multidimensional scaling involving Radial Basis Functions (RBF) that was first proposed by Webb [108, 109] in the context of MDS. The key issue in employing RBFs in MDS is to decide their centers. Webb [108] suggests to randomly choose the centers and then uses an expensive cross-validation procedure to decide what they are. In our research, we took a completely different route and regard the selection of the centers as a multi-task learning problem that has been widely studied in machine learning. This approach has led us to introduce the $(2, 1)$-norm as a regularization term to the stress function used by Webb [107]. Two reformulations, namely the diagonal and the spectral, are developed that aim to ease the difficulties in solving the $(2, 1)$-norm minimization problem. An iterative block majorization algorithm is developed to solve our models. The performance of our models were then compared to the original model in [107] on three well-known

data sets. Discriminant analysis of the methods are also discussed. Numerical experiments on three benchmarking data set illustrate significant improvement of our models over the original one in terms of projection quality and CPU time.

We note that the spectral model `RMDS-S`, when compared to the diagonal model `RMDS-D`, is less sensitive to the choice of the regularization parameter γ. For example, when $\gamma = 10$, there appeared a significant level of failure in `RMDS-D` in all three data sets, while `RMDS-S` worked almost same as we reported here. That means spectral model is more robust than the diagonal model, but with higher computational complexity. To get better projection using both RMDS-S and RMDS-D, the value of γ should be chosen from the interval $(0, 1]$. In our experiment, we simply use $\gamma = 1$.

We applied both models on several datasets and observed that both models work very well for small data set but for large data set they may be time consuming. To overcome these we were trying to develop new models that will project large dataset in less time with acceptable accuracy. Among a great number of dimension reduction methods, recently proposed supervised distance preserving projection method (SDPP) showed promising results on regression data but couldn't show that much convincing result for classification problems. We proposed a modification of SDPP in the next chapter that significantly improves SDPP data in classification and also can handle large dataset very well.

Chapter 3

Supervised Distance Preserving Projection using Alternating Direction Method of Multipliers

3.1 Introduction

Supervised Distance Preserving Projection (SDPP) is a dimension reduction method in supervised setting proposed recently by Zhu et al [117] which showed very promising result in regression problems. The basic formulation of SDPP aims to preserve distances locally between data points in the projected space (reduced feature space) and the output space. The method learns a linear mapping from the input space to the reduced feature space that leads to an efficient regression design. A drawback of SDPP approach is, for classification problems the preservation of local structure approach forces data of different classes to project very close to one another in the projected space which ends up with low classification rate.

To avoid the crowdedness of SDPP approach we have proposed a modification of SDPP which deals both regression and classification problem and significantly improves the performance of SDPP.

In our research, we incorporated the total variance of the projected co-variates to the SDPP problem which prevents data of different classes to stay together and therefore preserves the global structure. Thus the purpose of our proposed model is to keep the distance relation with neighbors (local structure) and at the same time to preserve the global structure by maximizing the total variance. This approach not only facilitates efficient regression like SDPP but also successfully classifies data into different classes.

In the last chapter, we have worked with classification task of data. We proposed two models that worked very well for small dataset but a little time consuming for large data set. So a new model was necessary to explore that can handle large dataset. Note that the last chapter is concerned with data set on unsupervised settings, that is, the models do not take any advantages of some prior information concerning the data sets. For example, some data points may be known beforehand to belong to certain class. Since the main intention is to identify the classes of test sets so using the class information of the training data points in determining the transformation matrix may increase the classification rate.

Based on this idea we proposed the modification of SDPP that works with dataset on supervised settings and can easily handle large datasets. We formulated the proposed optimization problem as a Semidefinite Least Square (SLS) SDPP problem. A two block Alternating Direction Method of Multipliers have been developed to learn the transformation matrix solving the SLS-SDPP which can easily handle out of sample data. The projections of testing data points in low dimensional space are further used for regression or assigning them into fixed number of classes. The experimental evaluation on both synthetic and real world high dimensional large data is conducted to compare the performance of SLS-SDPP with some state-of-the-art approaches.

3.2 Previous Studies

Most of the research on dimension reduction are focused on unsupervised settings i.e. handles data without labels. Among the great number of unsupervised dimensionality reduction methods available, the most well known is principal component analysis and

its nonlinear extension kernel PCA which is well documented in [92]. PCA and kernel PCA maximizes the data variance. Locally Linear Embedding (LLE) [90] which considers symmetries of locally linear constructions and ISOMAP [96] which incorporates pairwise geodesic distance based on k-nearest neighbor graph are examples of manifold learning that analyze the local geometric structure of the data. Maximum variance unfolding [110] is another unsupervised dimension reduction method that learns a kernel matrix by defining a neighborhood graph and retaining pairwise distance in the resulting graph. For extensive research on unsupervised dimension reduction methods one can consult [62, 69].

On the other hand, in supervised learning, each data is labeled. In regression task the label takes continuously varying real values whereas in classification task the labels are discrete numbers that indicates which input data belongs to which class. The most widely used supervised dimension reduction method for classification task is

Fishers discriminant analysis (FDA) and its kernalized form kernel FDA [72]. These methods maximizes the ratio of between-class and within-class covariances for a good projection of data in separate classes. For a general C class problem, FDA maps the data into a (C-1) dimensional space.

Other than FDA, some very well known supervised DR methods on regression are sufficient dimensionality reduction [35, 65, 113], kernel dimension reduction (KDR)[35], partial least square (PLS)[111, 112], supervised principal component analysis (SPCA) [4] and recently proposed supervised distance preserving projection method (SDPP)[117]. In order to handle nonlinear projection , kernalized versions KFDA, KPLS,KSPCA,KSDPP of all of these methods are also proposed.

Sufficient dimensionality reduction (SDR) [35, 65, 113] method seeks for a central subspace which is the intersection of all such subspaces containing the orthogonal transformation U such that the output Y and input co-variates X are conditionally independent and no information about the regression is lost in reducing the dimension. But unfortunately for this approach to be successful strong assumptions have to be made

on the existence of U. **Kernel dimension reduction (KDR)** is a new methodology for SDR that overcomes this problem. This method doesn't impose particular assumption on the underlying joint distribution of X and Y. KDR maximizes conditional dependence by a positive definite ordering of the expected covariance operators in the probability determining reduced kernel Hilbert spaces. However KDR is computationally highly demanding.

Classical **Partial Least Square (PLS)** [111, 112] is a linear DR method for regression task that involves a family of techniques to analyze the relationship between blocks of data by constructing a low dimensional subspace with orthogonal latent components. PLS is an iterative procedure. At each iteration it extracts latent vectors by maximizing the covariance between the projected co-variate and the output responses.

Supervised principal component analysis (SPCA)[4] is a generalization of Principal Component Analysis (PCA). It is based on Hilbert-Schmidt independence criterion and aims to estimate sequence of principal components that have maximal dependence on response variable. SPCA is solved by eigen decomposition of the weighted covariance matrix enhanced by the kernel of responses. Thus similar to PCA , SPCA has closed form solution and doesn't suffer from high computational complexity. But SPCA doesn't consider local structure of data. Consequently this method cannot extract the intrinsic dimensionality of the data.

3.3 Supervised Distance Preserving Projection

The Supervised Distance Preserving Projection (SDPP) is a dimensionality reduction method that minimizes the differences between distances among projected co-variates and distances among responses locally. It also preserves the continuity of the response space. In other words the low dimensional space is optimized in a way that the local

geometrical structure of the low dimensional subspace preserves the geometrical characteristics of the response space.

Suppose we have n data points $\{x_1, x_2, ..., x_n\}$, $x_i \in \Re^m$ and their responses $\{y_1, y_2, ..., y_n\}$. The response space Y can be multidimensional.

In [117] Zhu et al. proposed the following methodology. Assuming that the mapping $h : X \to Y$ is continuous and provided X is well sampled. It is assumed that for each point $x \in X$ and for every $\epsilon_y > 0$ there exists an $\epsilon_x > 0$ such that $d(x, x') < \epsilon_x \Rightarrow \delta(h(x), h(x')) < \epsilon_y$. where $d(,)$ and $\delta(,)$ are distance functions in X and Y respectively.

The idea of SDPP is to represent high dimensional data $\{x_1, x_2, ..., x_n\}$ in a lower dimensional space Z with dimensionality $r << m$

The form of data representation in $\Re^r$, denoted as $\mathbf{f}$, is assumed to be a linear function of the feature vector x in the original input space, defined by

$$\mathbf{f}(x) = W^T x, \qquad \forall\ x \in \Re^m \tag{3.1}$$

where the transformation matrix $W \in \Re^{r \times m}$.

Thus the method seeks for the transformation matrix W that minimizes

$$F(W) = \frac{1}{n} \sum_{i=1}^{n} \sum_{x_j \in N(x_i)} (d_{ij}^2(W) - \delta_{ij}^2)^2$$

where $N(x_i)$ denotes a neighborhood of x_i and Euclidean metric is used to characterize the pairwise distances; that is $d_{ij}^2(W) = \|z_i - z_j\|^2$ and δ_{ij} takes one of the following form:

- Simply $\delta_{ij} = \|x_i - x_j\|$
- In data classification task:

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \sim j (x_i \textit{ and } x_j \textit{ belongs to same class}) \\ 1 & \text{otherwise.} \end{cases}$$

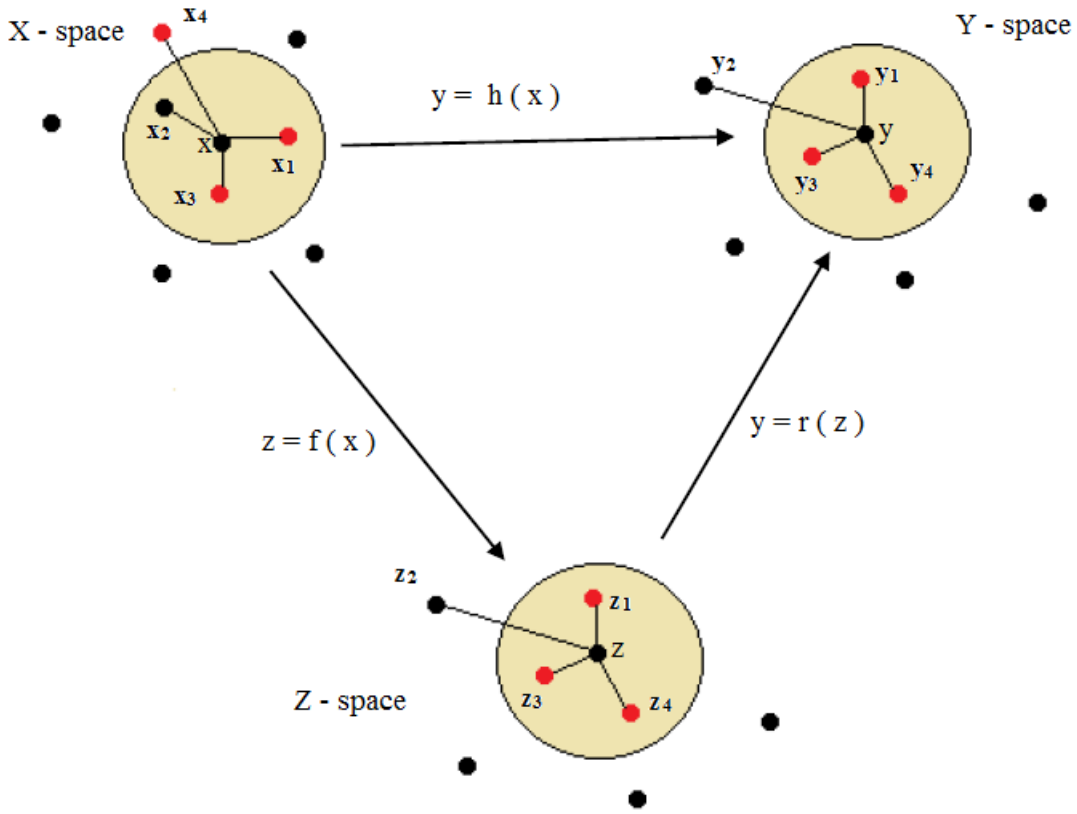


Figure 3.1: SDPP: Solid lines indicate connection between neighbors

- Continuous values for regression:

  $\delta_{ij} = \|y_i - y_j\|$, with $y = h(x) = a_1x_1 + a_2x_2 + ... + a_mx_m$ (Linear regression model).

In [117] the third form of δ_{ij} is considered because the method is mainly developed for regression tasks.

Locality around any point x_i is controlled by its k nearest neighbors in $N(x_i)$ where the number k is hyper-parameter of SDPP that has to be set beforehand or tuned from data. In [117] the value of k selected by a continuity measure that is discussed briefly in section 3.3.1.

The schematic illustration of SDPP [117] is given in Fig. 3.1. For a point x in input space, consider three nearest neighbor $N(x) = \{x_1, x_2, x_3\}$. Suppose in output space the neighborhood of y is $\{y_1, y_3, y_4\}$ ie. y_2 is outside of the neighborhood of y. SDPP seeks for the transformation matrix W for which $z_2 = f(x_2)$ is moved outside the neighborhood in the Z-space while z_4 is moved inside to match the local geometry in the Y-space as

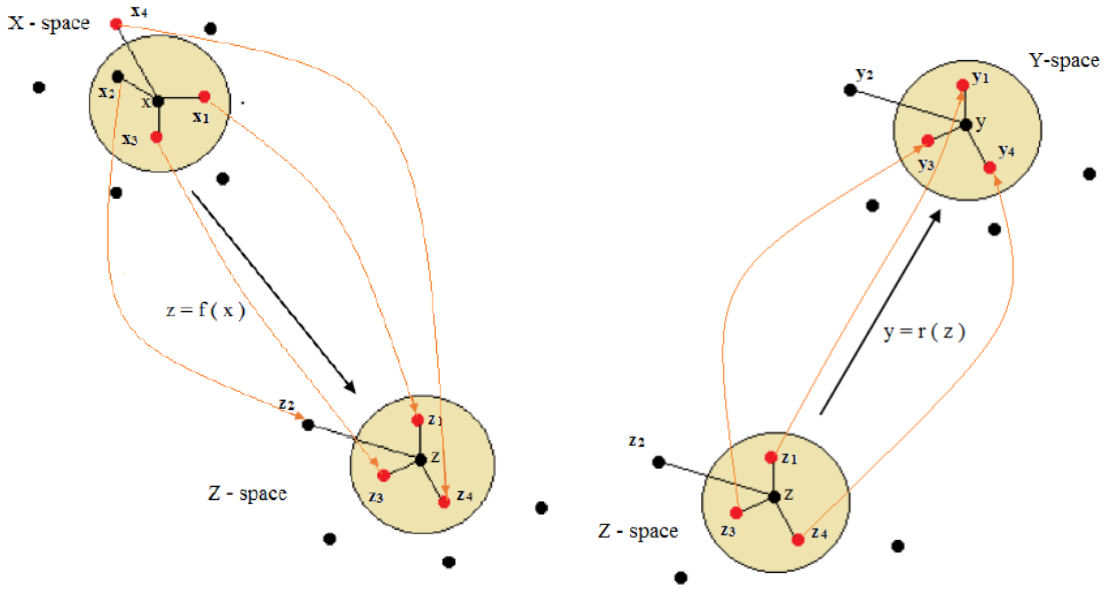


Figure 3.2: Preservation scheme of the local geometry by SDPP.

shown in Fig. 3.1 and Fig. 3.2. Thus SDPP incorporates a neighborhood graph G_{ij} in the objective function defined as follows:

$$G_{ij} = \begin{cases} 1 & \text{if } i \sim j(k - NNneighbor, x_j \in N(x_i)) \\ 0 & \text{otherwise.} \end{cases}$$

Thus the objective of SDPP is to minimize

$$F(W) = \frac{1}{n} \sum_{i=1}^{n} \sum_{x_j \in N(x_i)} (d_{ij}^2(W) - \delta_{ij}^2)^2$$

which can be written equivalently as follows:

$$F(W) = \frac{1}{n} \sum_{ij} G_{ij} (d_{ij}^2(W) - \delta_{ij}^2)^2. \tag{3.2}$$

The rest of this chapter is focused on solving the model (3.2) with the neighborhood graph G_{ij}. In [117] two different strategies have been designed to efficiently optimize the objective function (3.2)of SDPP:

- Semidefinite quadratic linear programming (SQLP).
- Conjugate Gradient (CG) optimization

But due to some limitations of SQLP, CG method is used throughout the research of SDPP in [117].

3.3.1 Continuity Measure

The continuity measure, $M_{cont}^{Z\mapsto Y}$, of the mapping $r : Z \mapsto Y$ is defined in [117, 104] by

$$M_{cont}^{Z\mapsto Y}(k_r) = 1 - C(k_r)\sum_{i=1}^{n}\sum_{j\in V_{k_r}(i)}(r_{ij} - k)$$

where $V_{k_r}(i)$ is the set of points that are in the k_r-neighborhood of point z_i in the projection space Z but not in the response space Y and r_{ij} be the rank of y_j in the ordering based on its distance from y_i. $C(k_r)$ is defined by

$$C(k_r) = \begin{cases} \frac{2}{nk_r(2n-3k_r-1)} & \text{if } k_r < \frac{n}{2} \\ \frac{2}{n(n-k_r)(n-k_r-1)} & \text{if } k_r \geq \frac{n}{2}, \end{cases}$$

3.3.2 Selection of the parameter

The value of the width k of neighborhood can be selected in several ways as discussed in [22]. The continuity measure 3.3.1 can be used to determine the value of the hyper parameter k . Firstly different SDPP projection matrices $W(k)$ are learned using different locality widths k, in order to obtain different low-dimensional representations. Secondly, the testing input observations X_t that are unseen are projected and then for each projection $Z_t(k)$ the corresponding continuity measures $M_{cont(kr)}^{Z\mapsto Y}$ is calculated against the corresponding outputs y_t, for a sequence of region sizes k_r. The value of k with jointly the highest continuity over the range of k_r is then used to learn the final model with all the data. In [14], one of the analysis of the connectivity of nearest-neighbor graphs suggests that k can be also be selected heuristically by setting k to be in the order of $log(n)$. For small sample sizes ($n \geq 100$), the value of k can be selected as 10% of the available learning points ; that is ($k \approx 0.1n$) .

3.4 SDPP as Semidefinite Least Square (SLS-SDPP)

We reformulate the SDPP problem as semidefinite matrix least squares with linear equality constraints. First we rewrite the square of the pairwise distance in the Z-space as

$$d_{ij}^2(W) = \|W^T(x_i - x_j)\|^2 = \langle (x_i - x_j)(x_i - x_j)^T, WW^T \rangle$$

Let $\Phi_{ij} = (x_i - x_j)(x_i - x_j)$ and $X = WW^T$. Then the objective function $F(W)$ in (3.2) takes the form

$$F(W) = \frac{1}{n} \sum_{i,j} G_{ij}(\langle \Phi_{ij}, X \rangle - \delta_{ij}^2)^2 \tag{3.3}$$

where G_{ij} is defined before in SDPP. Here $G \in \Re^{n \times n}, X \in \Re^{m \times m}$,
$\Phi_{ij} \in \Re^{m \times m}$,$(i,j) \in \xi$ where ξ is the set of all possible pairs (i,j)
Let $u_{ij} = \langle \Phi_{ij}, X \rangle - \delta_{ij}^2$. Then, the optimization model that of SDPP can be written as

$$\min \ f(W) = \frac{1}{n} \sum_{i,j} G_{ij} \|U\|_F^2 \tag{3.4}$$

such that$\langle \Phi_{ij}, x \rangle - u_{ij} = \delta_{ij}^2$, $(i,j) \in \xi$.

3.4.1 Reformulation as SLS-SDPP

Suppose $\sum_{i=1}^n x_i = 0$ i.e $\{x_i\}_{i=1}^n$ is already centralized. Then $z_i = W^T x_i$, $for\ i = 1, 2, ..., n$ is also centralized. We incorporate the total variance $\sum_{i=1}^n \|z_i\|^2$ to the objective function 3.4 where

$$\sum_{i=1}^n \|z_i\|^2 = \sum_{i=1}^n \|W^T x_i\|^2 = \sum_{i=1}^n \langle x_i x_i^T, WW^T \rangle = \sum_{i=1}^n \langle \Psi_{ii}, WW^T \rangle.$$

Denoting $X = WW^T$, the optimization model is reformulated as follows:

$$\begin{aligned} \max \langle \sum_{i=1}^n \Psi_{ii}, X \rangle - \frac{\nu}{n} \sum_{i,j} G_{ij} \|U\|_F^2 \\ s.t. \ \langle \Phi_{ij}, X \rangle - u_{ij} = \delta_{ij}^2 \\ X \succeq 0, \end{aligned}$$

where $\nu > 0, (i,j) \in \xi$.

Now denote $\Psi = \sum_{i=1}^{n} \Psi_{ii}, \mathcal{A}X = \langle \Phi_{ij}, X \rangle$ and $b = \delta_{ij}^2$. Let $H = (G_{ij}) > 0$ where $(i,j) \in \xi^\star \subset \xi$. Here $\|\xi^\star\| = p = k * n$. Therefore $H \in \Re^p$.

For a vector $\mathbf{v} \in \Re^p$, we define $\|\mathbf{v}\|_H^2 = \sum_{i=1}^{p} H_i v_i^2$, our objective function can be rewritten as

$\max \langle \Psi, X \rangle - \frac{\nu}{n}\|\mathcal{A}X - b\|_H^2 = \langle \Psi, X \rangle - \frac{\nu}{n}\|U\|_H^2$

According to the definition of $G = (G_{i,j}), H_i = 1, \forall i$. We consider the value of penalty parameter $\nu = 1$ to put equal emphasis on both the terms of the objective function. Therefore our goal is to find the best value of the matrix X which solves the following Semidefinite Least Square (SLS) problem:

$$\begin{aligned} (P)\ max\ \langle \Psi, X \rangle - \frac{1}{n}\|U\|_F^2 \\ s.t.\ \mathcal{A}X - U = b \\ X \in \mathcal{S}_+^m. \end{aligned}$$

Note that a similar type problem is previously studied by Jiang et al in [56] where they developed a Partial Proximal Point algorithm to solve the problem. In the next section we will study a two block ADMM method to solve SLS-SDPP problem (P).

3.5 Alternating Direction Method of Multipliers

Alternating Direction Method of Multipliers (ADMM) is a simple but powerful algorithm that is well suited to convex optimization problem in particular to problems arising in applied statistics and machine learning. It takes the form of a decomposition-coordination procedure, in which a large global problem is solved by breaking it into smaller and easier subproblems. ADMM can be viewed as an attempt to blend the benefits of two earlier approaches dual decomposition and augmented Lagrangian methods for constrained optimization. It is also equivalent or closely related to many other algorithms, such as Douglas-Rachford splitting from numerical analysis, Spingarns method of partial inverses, Dykstras alternating projections method, Bregman iterative algorithms for l_1

problems in signal processing, proximal methods and many others. The fact that it has been re-invented in different fields over the decades underscores the intuitive appeal of the approach. The algorithm is a natural fit for more complicated problems in areas like graphical models. In addition, although we are interested on statistical learning problems, the algorithm is readily applicable in many other important areas such as engineering design, time series analysis, network flow, multi-period portfolio optimization or scheduling.

A general q-block convex optimization problem involves optimizing sum of q-convex functions with non overlapping variables. Thus the objective function is of the following form :

$$\min \left\{ \sum_{i=1}^{q} \theta_i(z_i) | \sum_{i=1}^{q} \beta_i^\star(z_i) = b \right\}, \tag{3.5}$$

where for each $i \in 1, 2, ..., q$, Z_i is finite dimensional real Euclidean space equipped with an inner product $\langle ., . \rangle$ and its induced norm $\| \cdot \|$, $\beta_i : X \mapsto Z_i$ is a linear map and $b \in X$ is given, the functions θ_i are closed proper and convex and without overlapping variables.

For a given $\sigma > 0$ the augmented Lagrangian function for (3.5) is defined by:

$$L_\sigma(z_1, z_2, ..., z_q; x) := \sum_{i=1}^{q} \theta_i(z_i) + \langle x, \sum_{i=1}^{q} \beta_i^\star(z_i) - b \rangle + \frac{\sigma}{2} \| \sum_{i=1}^{q} \beta_i^\star(z_i) - b \|^2, \tag{3.6}$$

where $z_i \in Z_i$ and $x \in X$ is the dual variable. The problem (3.5) can be minimized by classical Augmented Lagrangian Function Method (ALFM) introduced by Heston-Powell-Rockfeller in [52, 77, 86]. For a chosen point $(z_1^0, z_2^0, ..., z_q^0, x^0)$, the successive iterations of ALFM are as follows:

$$(z_1^{k+1}, z_2^{k+1}, ..., z_q^{k+1}) \in \arg\min \; L_\sigma(z_1, z_2, ..., z_q; x^k), \tag{3.7}$$

$$x^{k+1} = x^k + \tau\sigma(\sum_{i=1}^{q} \beta_i^\star(z_i) - b), \tag{3.8}$$

where $\tau > 0$ is a constant that controls the step length. The non-separability of the quadratic penalty term in L_σ makes the joint minimization problem (3.7) more challenging task to solve exactly or approximately with high accuracy.

Alternation direction method of multipliers (ADMM) is a variant of augmented Lagrangian function method which can be used to solve the above challenging problem easily and efficiently. ADMM breaks (3.7) into q small problems and uses Gauss-Seidal approach to update the variables in an alternating or sequential fashion, which accounts for the term alternating direction. ADMM for the q-block problem consists of the following iterations:

$$\begin{aligned}
z_1^{k+1} &\in \arg\min \; L_\sigma(z_1, z_2^k, ..., z_q^k; x^k) \\
z_2^{k+1} &\in \arg\min \; L_\sigma(z_1^{k+1}, z_2, ..., z_q^k; x^k) \\
&\vdots \\
z_i^{k+1} &\in \arg\min \; L_\sigma(z_1^{k+1}, ..., z_{i-1}^{k+1}, z_i, z_{i+1}^k, ..., z_q^k; x^k) \\
&\vdots \\
z_q^{k+1} &\in \arg\min \; L_\sigma(z_1^{k+1}, z_2^{k+1}, ..., z_{q-1}^{k+1}, z_q; x^k) \\
x^{k+1} &= x^k + \tau\sigma(\sum_{i=1}^{q} \beta_i^\star z_i^{k+1} - b).
\end{aligned}$$

This section is devoted to an extensive discussion of 2-block ADMM and its convergence properties for a better understanding of the 2-block ADMM developed in chapter 4 to solve our proposed model.

Classical 2-block ADMM was first introduced by GLowinski and Marrocco [41] and Gabay and Mercier [38]. Its several applications are well documented in the article of Boyd et al. [12] and Eckstein and Yao [29].The general form of 2-block convex optimization problem is :

$$\min \{\theta_1(z_1) + \theta_2(z_2) | \beta_1^\star(z_1) + \beta_2^\star(z_2) = b\}. \tag{3.9}$$

The dual of (3.9) is given by

$$\max \{\langle -b, x\rangle - \theta_1^* S - \theta_2^* t | \beta_1 x + S = 0, \beta_2 x + T = 0\}. \tag{3.10}$$

For given $\sigma > 0$, the augmented Lagrangian associated with (3.9) is given as follows

$$L_\sigma(z_1, z_2; x) := \theta_1(z_1) + \theta_2(z_2) + \langle x, \beta_1^\star(z_1) + \beta_2^\star(z_2) - b\rangle + \frac{\sigma}{2}\|\beta_1^\star(z_1) + \beta_2^\star(z_2) - b\|^2. \quad (3.11)$$

Therefore for chosen $\tau > 0$ and $(z_1^0, z_2^0, x^0) \in dom(\theta_1) \times dom(\theta_2) \times X$ the successive iteration of classic 2-block ADMM is as follows:

$$\begin{aligned} z_1^{k+1} &\in \arg\min\ L_\sigma(z_1, z_2^k; x^k), \\ z_2^{k+1} &\in \arg\min\ L_\sigma(z_1^{k+1}, z_2; x^k), \\ x^{k+1} &= x^k + \tau\sigma(\beta_1^\star z_1^{k+1} + \beta_2^\star z_2^{k+1} - b). \end{aligned}$$

3.5.1 Convergence of ADMM

In this section we include some convergence result of two block ADMM discussed in [12].

Assumption 3.1. The (extended-real-valued) $\theta_i : Z_i \mapsto (-\infty, +\infty]$, $\forall i$ are proper, closed and convex functions.

Definition 3.2. A function θ is closed if its epigraph

$$epi(\theta) = \{(z, t) \in \Re^n \times \Re | \theta(z) \leq t\} \ \textit{is closed}$$

Therefore the assumption 3.1 can be expressed compactly using the epigraphs of the functions: The function θ_i satisfies assumption 3.1 if and only if its epigraph

$$epi(\theta_i) = \{(z_i, t) \in \Re^{n_i} \times \Re | \theta_i(z_i) \leq t\} \ \textit{is closed and convex.}$$

Assumption 3.1 implies that the subproblems arising in the updates of z_i are solvable, i.e., we can find $\{z_1, z_2\}$, not necessarily unique (without further assumptions on $\beta_1^\star$ and $\beta_2^\star$), that minimize the augmented Lagrangian.

It is important to note that assumption 3.1 allows θ_1 and θ_2 to be nondifferentiable and to assume the value $+\infty$. For example, we can take θ_i to be the indicator function of a closed nonempty convex set $\mathcal{C}$, i.e., $\theta_i(c) = 0$ for $c \in \mathcal{C}$ and $\theta_i(c) = +\infty$ otherwise.

Assumption 3.3. The unaugmented Lagrangian L_0 has a saddle point.

Explicitly, there exist $(\hat{z}_1, \hat{z}_2, \hat{x})$, not necessarily unique, for which

$$L_0(\hat{z}_1, \hat{z}_2, x) \leq L_0(\hat{z}_1, \hat{z}_2, \hat{x}) \leq L_0(z_1, z_2, \hat{x})$$

holds for all (z_1, z_2, x). By assumption 3.1, it follows that $L_0(\hat{z}_1, \hat{z}_2, \hat{x})$ is finite for any saddle point $(\hat{z}_1, \hat{z}_2, \hat{x})$. This implies that $(\hat{z}_1, \hat{z}_2)$ is a solution to (3.9), so $\beta_1^\star(\hat{z}_1) + \beta_2^\star(\hat{z}_2) = b$ and $\theta_i(\hat{z}_i) < \infty, \forall i$. It also implies that $\hat{x}$ is dual optimal, and the optimal values of the primal and dual problems are equal, i.e., that strong duality holds.

Under assumptions 3.1 and 3.3, the ADMM iterates satisfy the following:

- **Residual convergence:** Defining the residual $r = \beta_1^\star(z_1) + \beta_2^\star(z_2) - b$, we have, $r^k \to 0$ as $k \to \infty$, i.e., the iterates approach feasibility.
- **Objective convergence:** $\theta_1(z_1^k) + \theta_2(z_2^k) \to \hat{p}$ as $k \to \infty$, i.e., the objective function of the iterates approaches the optimal value.
- **Dual variable convergence:** $x^k \to \hat{x}$ as $k \to \infty$, where $\hat{x}$ is a dual optimal point.

A proof of the residual and objective convergence results is given in appendix A which follows from the proofs given in [12]. Note that z_1^k and z_2^k need not to converge to optimal values, although such results can be shown under additional assumptions.

In practice, for any $\tau \in (0, 2)$, convergence of 2-block ADMM has been proven first by Gabay and Mercier [38] when θ_1 is strongly convex, $\beta_1^\star$ is the identity mapping and $\beta_2^\star$ is injective. Glowinski in [40] and Fortin and Glowinski in [33] proved the convergence for $\tau \in (0, (1 + \sqrt{5})/2)$ if θ_2 is a general nonlinear convex function.

Note that the convergence of ADMM is not highly accurate. The algorithm produce acceptable results with modest accuracy within few iterations which is sufficient for

many real life applications. This behavior makes ADMM similar to algorithms like the conjugate gradient method. So ADMM can be combined with other methods for producing a high accuracy solution from a low accuracy solution. In general case ADMM is practically useful in most cases when modest accuracy is sufficient. Fortunately, this is usually the case for the kind of large-scale problems we consider.

3.5.2 Optimality Conditions

The primal feasibility

$$\beta_1^\star(\hat{z}_1)^{k+1} + \beta_2^\star(\hat{z}_2)^{k+1} - b = 0 \tag{3.12}$$

and dual feasibility

$$0 \in \partial\theta_1(z_1) + \beta_1 x \tag{3.13}$$

$$0 \in \partial\theta_2(z_2) + \beta_2 x \tag{3.14}$$

are the necessary and sufficient optimality conditions for the ADMM problem (3.9). Here, ∂ is the subdifferential operator; (When each θ_i is differentiable, the subdifferentials can be replaced by the gradients and $\in$ can be replaced by $=$). Define the residual $r = \beta_1^\star z_1 + \beta_2^\star z_2 - b$. Since z_2^{k+1} minimizes $L_\sigma(z_1^{k+1}, z_2, x^k)$ by definition, we have

$$\begin{aligned} 0 &\in \partial\theta_2 z_2^{k+1} + \beta_2 x^k + \sigma\beta_2(\beta_1^\star z_1^{k+1} + \beta_2^\star z_2^{k+1} - b) \\ &= \partial\theta_2 z_2^{k+1} + \beta_2 x^k + \sigma\beta_2 r^{k+1} \\ &= \partial\theta_2 z_2^{k+1} + \beta_2 x^{k+1}. \end{aligned}$$

This means that z_2^{k+1} and x^{k+1} always satisfy (3.14) which implies the optimality will be obtained if equation (3.12) and (3.13) are satisfied . This implies that the iterates of the method are always dual feasible.

Now, since z_1^{k+1} minimizes $L_\sigma(z_1, z_2^k, x^k)$ by definition, we have

$$\begin{aligned} 0 &\in \partial\theta_1 z_1^{k+1} + \beta_1 x^k + \sigma\beta_1(\beta_1^\star z_1^{k+1} + \beta_2^\star z_2^k - b) \\ &= \partial\theta_1 z_1^{k+1} + \beta_1(x^k + \sigma r^{k+1} + \sigma\beta_2^\star(z_2^k - z_2^{k+1})) \\ &= \partial\theta_1 z_1^{k+1} + \beta_1 x^{k+1} + \sigma\beta_1\beta_2^\star(z_2^k - z_2^{k+1}). \end{aligned}$$

Equivalently,

$$\sigma\beta_1(\beta_2^\star(z_2^{k+1} - z_2^k)) \in \partial\theta_1(z_1)^{k+1} + \beta_1 x^{k+1}.$$

This means that the quantity $s^{k+1} = \sigma\beta_1(\beta_2^\star(z_2^{k+1} - z_2^k))$ is residual for the dual feasibility condition (3.13). Therefore we refer s^{k+1} as the dual residual at iteration $k+1$, and $r^{k+1} = \beta_1^\star z_1^{k+1} + \beta_2^\star z_2^{k+1} - b$ as the primal residual at iteration $k+1$.

The above optimal criterion can be summarized as follows:

The optimality conditions for the ADMM problem consist of three conditions (3.12 - 3.14). $(z_1^{k+1}, z_2^{k+1}, x^{k+1})$ always satisfy the equation (3.14); the residuals for the other two, (3.12) and (3.13), are the primal and dual residuals r^{k+1} and s^{k+1} respectively which converge to zero as the ADMM proceeds. Convergence proof of ADMM is included in appendix A .

3.5.3 Stopping Criteria

A reasonable termination criterion [12] for ADMM is that the primal and dual residuals must be small which can be observed from the inequality (proof included in appendix A)

$$\theta_1 z_1^k + \theta_2 z_2^k - \hat{p} \leq -(x^k)^T r^k + (z_1^k \hat{z}_1)^T s^k. \tag{3.15}$$

This shows that when the residuals r^k and s^k are small, the objective suboptimality also must be small. Since $\hat{z}_1$ is not known, if we guess or estimate that $\|z_1^k - \hat{z}_1\|_2 \leq d$, we have

$$\theta_1 z_1^k + \theta_2 z_2^k - \hat{p} \leq -(x^k)^T r^k + d\|s^k\|_2 \leq \|x^k\|_2\|r^k\|_2 + d\|s^k\|_2.$$

The middle or righthand terms can be used as an approximate bound on the objective suboptimality (which depends on the guess of value of d).

Therefore the stopping criteria can be

$$\|r^k\|_2 \leq \epsilon^{pri} \; and \; \|s^k\|_2 \leq \epsilon^{dual}, \tag{3.16}$$

where $\epsilon^{pri} > 0$ and $\epsilon^{dual} > 0$ are feasibility tolerances for the primal and dual feasibility conditions (3.12) and (3.13) respectively which can be chosen using an absolute and relative criterion, such as

$$\epsilon^{pri} = \sqrt{p}\epsilon^{abs} + \epsilon^{rel} \max\{\|\beta_1^\star z_1^k\|_2, \|\beta_2^\star z_2^k\|, \|b\|\},$$

$$\epsilon^{dual} = \sqrt{n}\epsilon^{abs} + \epsilon^{rel}\|\beta_1 x^k\|_2,$$

where $\epsilon^{abs} > 0$ is an absolute tolerance and $\epsilon^{rel} > 0$ is a relative tolerance where the factors $\sqrt{p}$ and $\sqrt{n}$ come from the fact that the l_2 norms are in $\Re^p$ and $\Re^n$ respectively. The value of the relative tolerance might be $\epsilon^{rel} = 10^{-3}$ or 10^{-4}, depending on the application. The absolute stopping criterion depends on the scale of the typical variable values.

Classic ADMM and many of its variations have been explored in the literature. In our proposed algorithm, introduced in chapter 4, we have used different penalty parameter to improve the convergence that we described here briefly.

Tuning penalty parameter σ:

The convergence of the algorithm can be improved by adjusting the parameter σ at each iteration based on the previous iteration progress with the goal of improving the convergence in practice. This also makes the performance of the algorithm less dependent on the initial choice of the penalty parameter. In [87], it is shown that if $\sigma_k \to \infty$ then

the convergence of ADMM can be superlinear. Though the convergence of ADMM with σ as a variable can be difficult to prove but the fixed- σ theory can be applied if one just assumes that σ becomes fixed after a finite number of iteration. A simple scheme to update the penalty parameter studied in [48, 106] is as follows:

$$\sigma_{k+1} = \rho\sigma_k \ or \ \sigma_{k+1} = \sigma_k. \tag{3.17}$$

Other variations of ADMM are, more general augmenting terms, over-relaxation of the feasible conditions, update ordering of primal and dual variables etc. Some of these methods can give superior convergence in practice compared to the standard ADMM presented above. For more properties of the algorithm one can see [34, 37, 33, 42, 99, 36, 28, 17]. In particular, the convergence of ADMM can be explored in [12] including [37] and [27].

3.6 ADMM for SLS-SDPP

In this section we studied a two block ADMM to determine the best transformation matrix W which is described step by step as follows.

First step is to obtain the dual (D) of the primal problem (P). The next step is to determine the augmented Lagrange function of (D).

Now , Consider the Lagrangian function of the primal problem (P)

$$\begin{aligned} L(X,U,z) &= \langle \Psi, X\rangle - \frac{1}{n}\|U\|_F^2 + \langle z, \mathcal{A}X - U - b\rangle + \delta_{\mathcal{S}_+^m}(X) \\ &= \langle -b, z\rangle + \langle \mathcal{A}^* z + \Psi, X\rangle + \langle z, -U\rangle - \frac{1}{n}\|U\|_F^2 + \delta_{\mathcal{S}_+^m}(X). \end{aligned}$$

Therefore the dual function is

$$\min_{z\in\Re^p}\left\{\Theta(X,U,z) = \max_{X\in\Re^{m\times m}, U\in\Re^p} L(X,U,z)\right\}.$$

The dual problem of (P) thus obtained is:

$$\begin{aligned}(D)\ min\ \Theta(z) = -\langle b, z\rangle + \frac{n}{4}\|z\|^2 \\ s.t.\ \mathcal{A}^* z + \Psi + S = 0 \\ S \in \mathcal{S}_+^m.\end{aligned}$$

For the convergence of 2 block ADMM , we need the following assumption which is a simpler version of assumption 3.3 stated in (section 3.5.1).

Assumption 3.4. a) There exists a feasible solution $\hat{X} \in \mathcal{S}_+^m$ of problem (P) such that $\mathcal{A}X - U = b, \hat{X} \in int(\mathcal{S}_+^m)$.

b) There exists a feasible solution $\{\hat{S}, \hat{z}\} \in \mathcal{S}_+^m \times \Re^p$ of problem (D) such that $\mathcal{A}^* z + \Psi + S = 0, \hat{S} \in int(\mathcal{S}_+^m)$.

From convex analysis [13, sec. 5.5.3] [11, Cor. 5.3.6] it is known that under assumption 3.4 the strong duality for (P) and (D) holds and the following Karush-Kuhn-Tucker (KKT) conditions has nonempty solution

$$\mathcal{A}X - U = b,\ \mathcal{A}^* z + \Psi + S = 0, \langle X, S\rangle = 0, X \in \mathcal{S}_+^m, S \in \mathcal{S}_+^m. \tag{3.18}$$

For $\sigma > 0$, the augmented Lagrange function for (D) is defined by

$$\begin{aligned}L_\sigma(z, S; X) &= -\langle b, z\rangle + \frac{n}{4}\|z\|^2 + \langle X, \mathcal{A}^* z + \Psi + S\rangle + \frac{\sigma}{2}\|\mathcal{A}^* z + \Psi + S\|_F^2 \\ &= -\langle b, z\rangle + \frac{n}{4}\|z\|^2 + \frac{\sigma}{2}\|\mathcal{A}^* z + \Psi + S + \frac{X}{\sigma}\|_F^2 - \frac{\|X\|_F^2}{2\sigma},\end{aligned}$$

where $(z, S, X) \in \Re^p \times \mathcal{S}_+^m \times \mathcal{S}_+^m$.

Now we are ready to introduce the ADMM algorithm for our SLS-SDPP problem.

Algorithm 3.5. **Alternating Direction Method of Multipliers**

Given parameters $\tau \in (0, \infty)$.

(S.0) Choose $\sigma_0 > 0$, $S^0 \in \mathcal{S}^m_+$, $X_0 = W_0 W_0^T \in \mathcal{S}^m_+$. Set $z^0 = (\frac{n}{2}I + \sigma AA^*)^{-1}\left(b - \mathcal{A}X^0 + \sigma\mathcal{A}(-\Psi - S^0)\right)$. Let $k = 0$.

(S.1) Update S^k by

$$S^{k+1} \in \arg\min L_{\sigma_k}(z^k, S; X^k) = \Pi_{\mathcal{S}^m_+}\left(-\Psi - \mathcal{A}^* z^k - \frac{X^k}{\sigma_k}\right). \tag{3.19}$$

(S.2) Update z^k by

$$z^{k+1} \in \arg\min L_{\sigma_k}(z, S^{k+1}; X^k) = (\frac{n}{2}I + \sigma_k \mathcal{A}\mathcal{A}^*)^{-1}(b - \mathcal{A}X^k + \sigma_k\mathcal{A}(-\Psi - S^{k+1})). \tag{3.20}$$

(S.3) Update X^k by

$$X^{k+1} \in \arg\min L_{\sigma_k}(z^{k+1}, S^{k+1}; X) = X^k + \tau\sigma_k(\ \mathcal{A}^* z^{k+1} + S^{k+1} + \Psi). \tag{3.21}$$

(S.4) Update σ_k by

$$\sigma_{k+1} = \rho\sigma_k (\rho > 0)\ or\ \sigma_{k+1} = \sigma_k. \tag{3.22}$$

Note that

- In (S.1), the projection $\Pi_{\mathcal{S}^n_+}(X_1)$ of a given matrix $X_1 \in \mathcal{S}^n$ onto $\mathcal{S}^n_+$ is the optimal solution of the problem

$$\begin{aligned} min\ & \frac{1}{2}\|Y - X_1\|_F^2 \\ s.t.\ & Y \in \mathcal{S}^n_+. \end{aligned}$$

  Now let $P \in \mathcal{O}_n$ be such that $X_1 = PDiag(\lambda_1(X_1), \lambda_2(X_1), ..., \lambda_n(X_1))P^T$. Then $\Pi_{\mathcal{S}^n_+}(X_1)$ has the closed form
  $\Pi_{\mathcal{S}^n_+}(X_1) = PDiag(\max\{\lambda_1(X_1), 0\}, \max\{\lambda_2(X_1), 0\}, ..., \max\{\lambda_n(X_1), 0\})$.

- In (S.2), to update z we need to solve linear systems involving the operator AA^*. The computation of AA^* and its (sparse) Cholesky factorization, which only needs to be done once, can be done at a moderate cost.

In [95] Sun et al. discussed a similar type 3-block semi-proximal ADMM for conic optimization problem.

The convergence of algorithm 3.5 for solving problem (D) follows from the following theorem established in [95].

Theorem 3.6. *If the assumption 3.4 holds and if $\mathcal{A}$ is surjective. Then the sequence (S^k, z^k, X^k) generated by the algorithm 3.5 is well defined. Furthermore under the condition that either (a) $\tau \in (0, 2)$ or (b) $\tau \geq 2$ but $\sum_{k=0}^{\infty} \|S^{k+1} + \mathcal{A}^* z^{k+1} + \Psi\|_F^2 < \infty$ the sequence (S^k, z^k, X^k) converges to unique limit say $(S^\infty, z^\infty, X^\infty)$ satisfying the KKT conditions (3.18).*

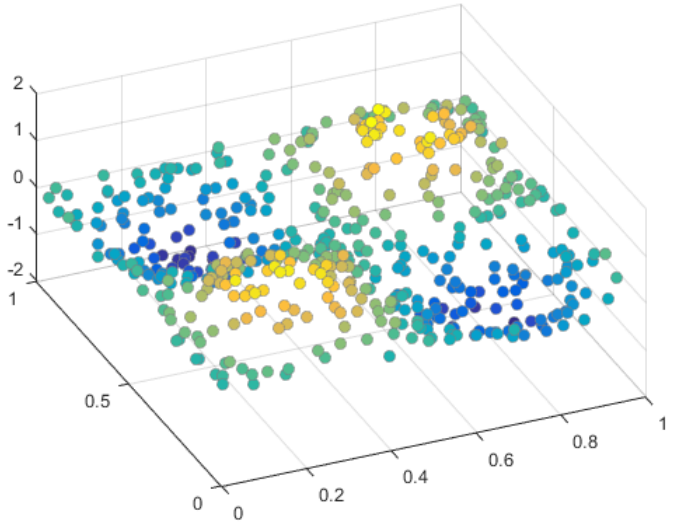
Since the objective functions in (P) and (D) are closed, proper and convex so by assumption 3.1 and 3.3 , the assumption 3.4 holds.

It is important to note that the algorithm doesn't optimizes the projection matrix W directly. It optimizes the positive semidefinite matrix $X = WW^T$. The projection matrix W can be computed as the square root of X or alternatively can be computed applying singular value decomposition (SVD) on X. In our numerical part we have applied SVD on X. The ith column of W is calculated as $\sqrt{\lambda_i} p_i$ where λ_i and p_i are the ith eigenvalue and eigenvector respectively.

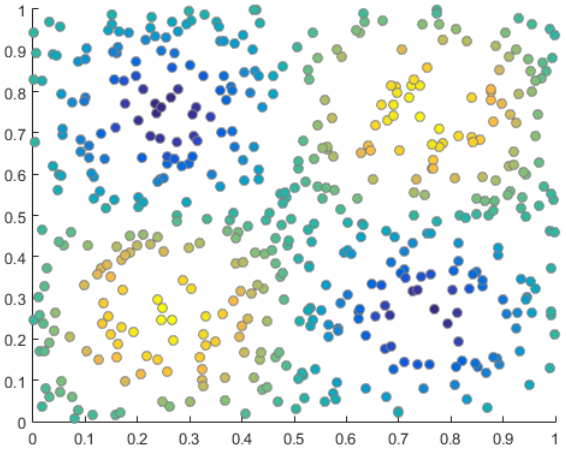
3.7 Numerical Experiments

In this section, we will demonstrate the performance of Alg. 3.5 on several synthetic and real world data sets. We will compare our results with SDPP [117], SPCA [4], PLS [111, 112], KDR [35] and FDA [72] methods.

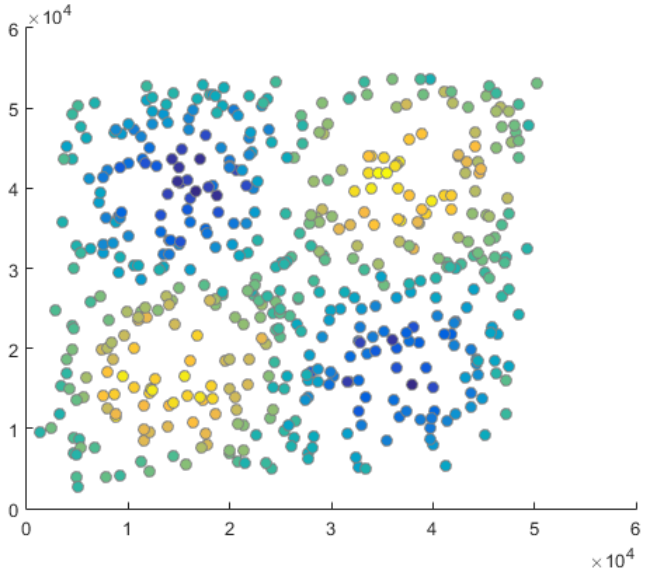
All tests have been carried out using the 64-bit version of MATLAB R2015a on a Windows 7 desktop with 64-bit operating system having Intel(R) Core(TM) 2 Duo CPU of 3.16GHz and 4.0GB of RAM.



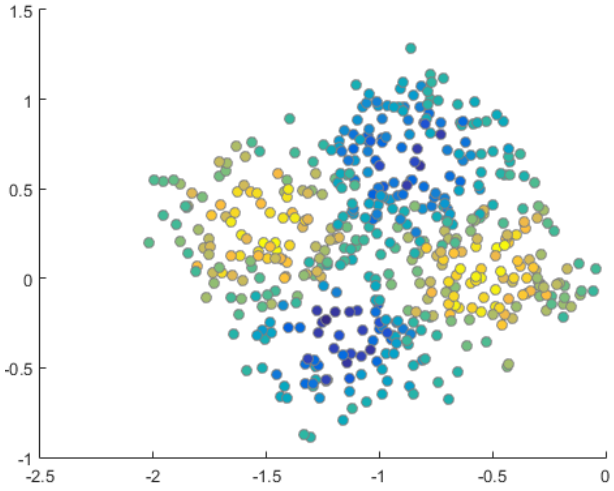
(a) 3D plot of test points with two effective features.



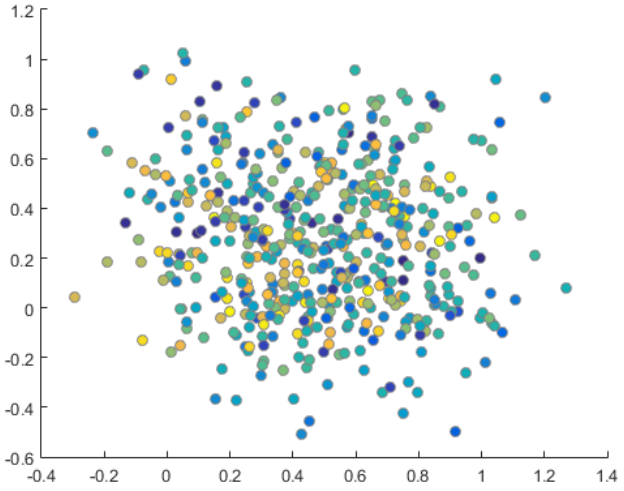
(b) True Projection



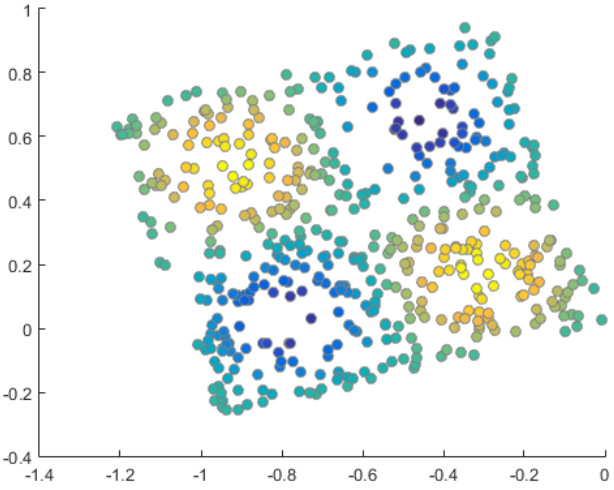
(c) Projection by SLS-SDPP



(d) Projection by SDPP



(e) Projection by SPCA



(f) Projection by KDR

Figure 3.3: Smoothed Parity. (a) 3D plot of test points with two effective features. (b) True projection of two most effective features. (c)-(f) Represents two-dimensional projection by SLS-SDPP, SDPP, SPCA and KDR respectively. SLS-SDPP, SDPP and KDR successfully extracted the intrinsic structure.

For our algorithm, we measure the accuracy of an approximate optimal solution (X, S, z) by using the following relative residual [95]

$$\eta = \max\{\eta_P, \eta_D, \eta_{\mathcal{S}_+^n}, \eta_C\}$$

where $\eta_P = \frac{\|AX-b+\frac{z}{2}\|}{1+\|b\|}$, $\eta_D = \frac{\|A^*z+S+\Psi\|}{1+\|\Psi\|}$, $\eta_{\mathcal{S}_+^n} = \frac{\|\Pi_{\mathcal{S}_+^m}(-X)\|}{1+\|X\|}$ and $\eta_C = \frac{|\langle X,S\rangle|}{1+\|X\|+\|S\|}$. We compute the duality gap $\eta_g = \frac{\langle\Psi,X\rangle-\langle b,z\rangle}{1+|\langle\Psi,X\rangle|+|\langle b,z\rangle|}$ as well. We terminate the algorithm when $\eta < 10^{-6}$.

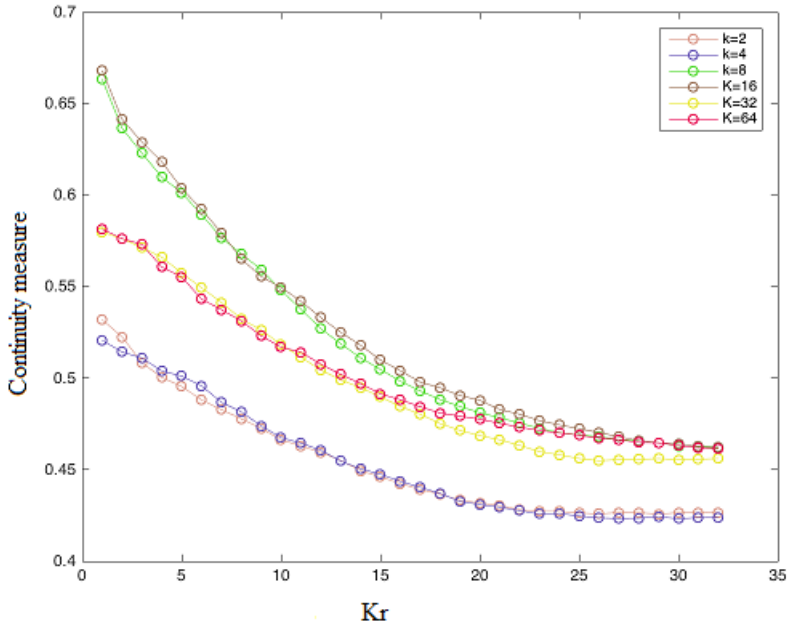
For the classification task, we have used k-Nearest Neighbor rule discused in the following section.

3.7.1 K-Nearest Neighbor:

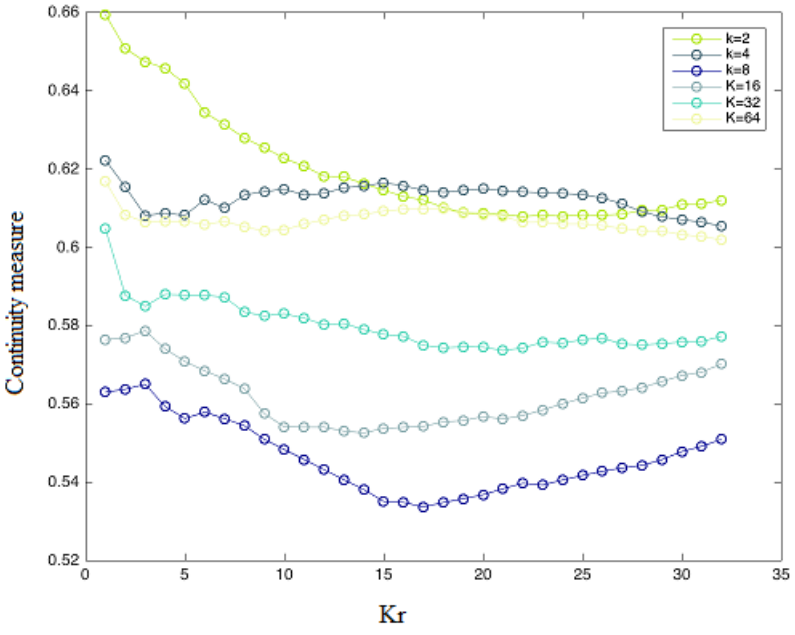
The K-Nearest Neighbor algorithm is a non-parametric method used for classification. It is among the simplest of all machine learning algorithms. K-NN is based on learning by analogy, ie., by comparing a given test point with training points that are similar to it. The algorithm for the nearest neighbor rule is summarized as follows: For an unknown feature vector x,

- First identify the k nearest neighbors out of the N training vectors, regardless of class label. For a two class problem k is chosen to be odd and in general not to be a multiple of the number of classes M.
- Next, identify the number of vectors k_i from these k neighbors , that belong to class w_i, for $i = 1, 2, ..., M$
- Finally assign x to the class w_i with the maximum number k_i. If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.

The nearest neighbor rule exhibits a good performance if the number of training samples is large. But the performance may degrade dramatically when the value of N is relatively small. To cope with the performance degradation associated with small values of N, one can see a number of techniques proposed in [97].



(a) Continuity measure with respect to different k and k_r for Smooth parity data



(b) Continuity measure with respect to different k and k_r for Swissroll data

Figure 3.4: Continuity measure with respect to different k and k_r for (a) Smooth parity data: Highest continuity measure achieved at $k = 8$ and $k = 16$ which suggest to choose the neighborhood size $k \in [8, 16]$, (b) Swissroll data: Highest continuity measure, obtained at $k = 2$, suggests to choose the neighborhood size $k = 2$.

In the second part of our research, we have used k-nearest neighbor rule to classify the test data points where the training data set is large enough.

3.7.2 Parameter Setting and Performance Indicators

For each of the data sets, randomly 60% of the data were initially selected as training data. In the numerical experiment, the weight matrix W in $X = WW^T$ is initialized using PCA on the training data set. The value of the neighborhood k is selected as $k \leq 10$ using the continuity measure 3.3.1. τ is set to 1.618 as suggested in [95]

The maximum number of iterations is set at $\lfloor 0.2N \rfloor$, where N is the number of data samples in the data set and $\lfloor 0.2N \rfloor$ is the largest integer not greater than $0.2N$.

We have conducted a number of experiments to illustrate the behavior of our algorithm on several regression and classification problems. For regression problems, the test samples of synthetic datasets are projected on lower dimensional space to visualize the underlying structure of the data. For real world data sets root mean squared error (RMSE) and mean absolute error (MAE) are calculated to compare the regression accuracy of our algorithm with some other methods mentioned above.
For the classification problems, 1-Nearest Neighbor rule is used on the projected low dimensional dataset to assign them into different classes. For each of the dataset, the classification error rate is calculated as the ratio of number of misclassified points to the total number of test samples.

3.7.3 Regression:

In this section first we will consider two synthetic dataset Smooth parity [117] and Swiss-sroll (http://isomap.stanford.edu/datasets.html). The mapping quality of constructed configuration of these datasets by SLS-SDPP will be compared with that of SDPP, SPCA and KDR.

Smoothed Parity: This is a synthetic 5 dimensional dataset with two wffwctive features constructed by $y = sin(2\pi X_1)sin(2\pi X_2) + \epsilon$ by Zhu et al. [117], where the noise term $\epsilon \in N(0, 0.1^2)$. 1000 points have been constructed and half of them is used for the

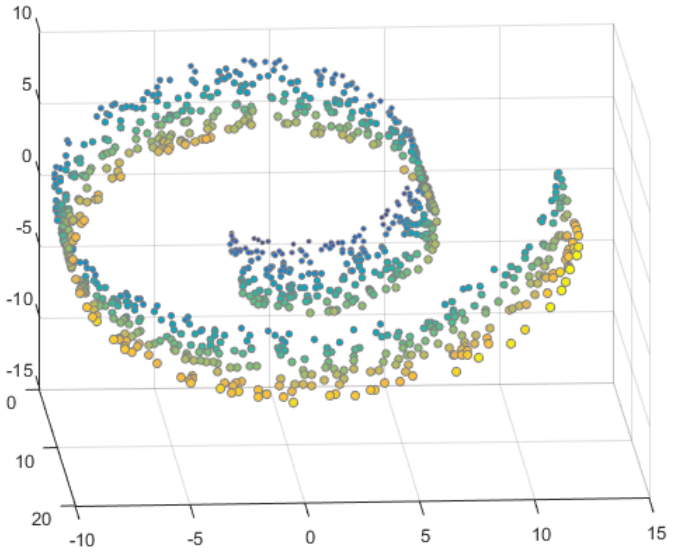
training. We have applied four methods ADMM, SDPP, SPCA and KDR to obtain a two dimensional projection of the data set which is shown in Fig. 3.3 and compared with the true two dimensional projection presented in Fig. 3.3(b). In the learning purpose, suitable neighborhood size lies in the range [8,16] determined by continuity measure described in 3.3.1 which is shown in Fig. 3.4(a). From Fig. 3.3 it is clear that SPCA is incapable of obtaining a good projection but ADMM, SDPP and KDR successfully projected the data that preserves the two most effective features. However, KDR requires a much longer time for training than ADMM and SDPP.

SwissRoll Data:

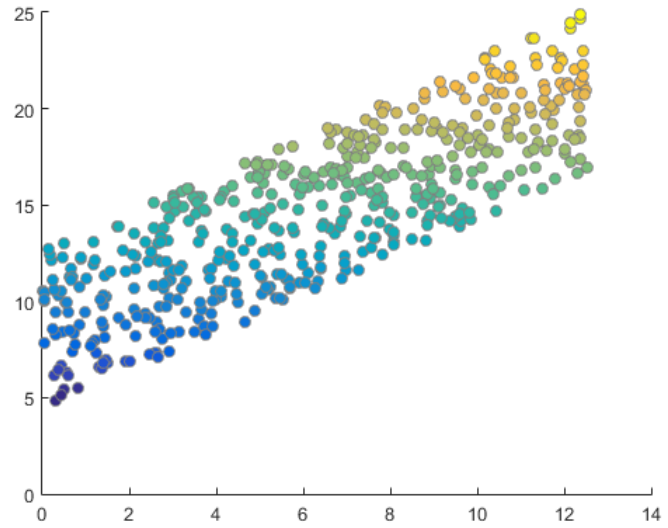
Swissroll data is benchmarking 3 dimensional data which corresponds to two dimensional patterns distributed uniformly on a plane and embedded non-linearly in 3D. We consider 1000 data points where half of them is used for learning and rest of the data is used for the testing. For the training purpose, 2 nearest points are considered as neighborhood points as suggested by continuity measure shown in figure Fig. 3.4(b). All the 4 methods are applied to get a two dimensional projection of the testing points. Fig. 3.5(a) presents a 3D plot of test points with all the features. Observing the true projection 3.5(b) and (c)-(f) of Fig. 3.5 it can be concluded that SPCA and KDR fail to obtain the most effective feature whereas ADMM and SDPP project the best informative features correctly.

In the next experiments we will evaluate the performance of proposed method on some real world regression problems obtained from UCI repository and compare with other methods in terms of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). We will use a number of graphs to show the improvement of SLS-SDPP over SDPP and some other leading methods.

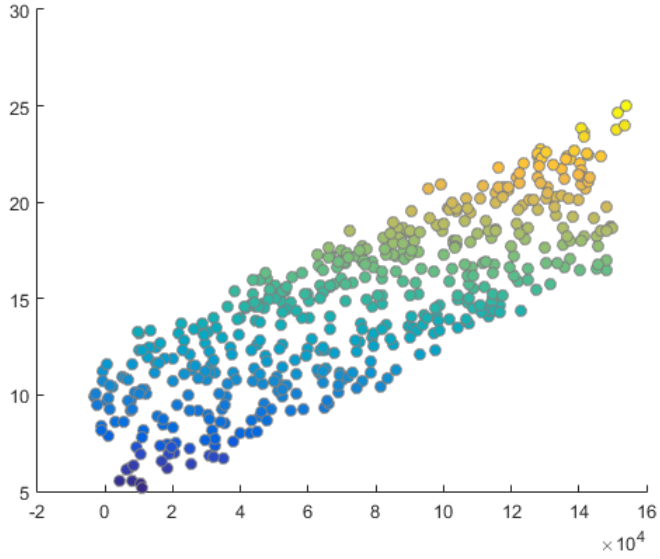
Two very important and well known data set Parkinsons Telemonitoring and Concrete Compressive Strength Data Set are considered for experimental evaluation. Both of the datasets are preprocessed by mean centering and normalized to unit variance. After the dimension reduction step, a simple linear model is used for regression. Randomly 60% of



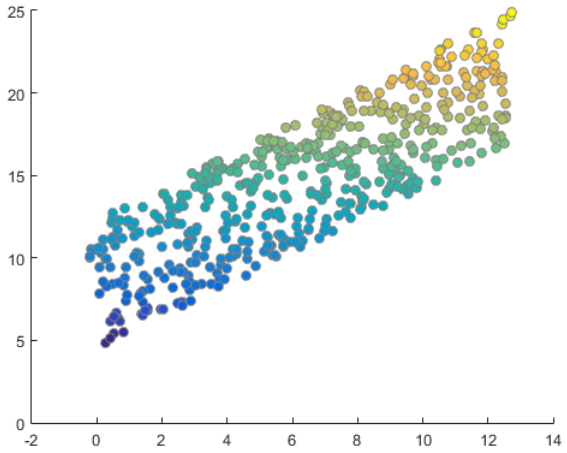
(a) Scatter plot of swissroll data



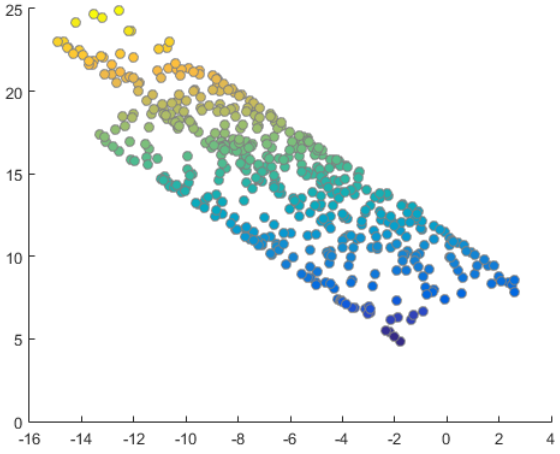
(b) True one dimensional projection



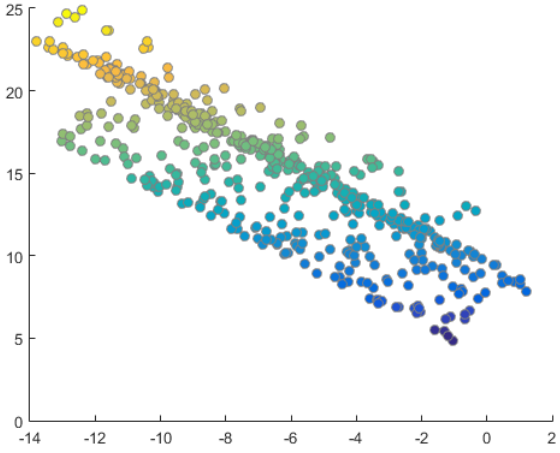
(c) SLS-SDPP projection



(d) SDPP projection



(e) SPCA projection



(f) KDR projection

Figure 3.5: SwissRoll data. (a) Scatter plot of 3 dimensional Swissroll data. (b) True projection of test data points, (c)-(f) Represents two-dimensional projection by ADMM, SDPP, SPCA and KDR respectively. SLS-SDPP and SDPP correctly projects the most effective features.

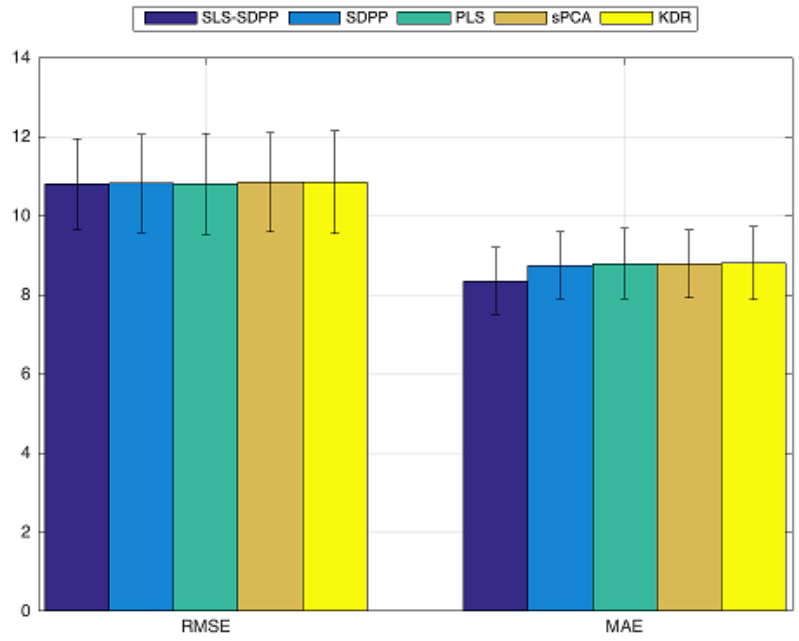


Figure 3.6: Average Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) with error bars for prediction of test set of Parkinsons Telemonitoring Data Set obtained by SLS-SDPP, SDPP, PLS, SPCA and KDR. The bar diagram represents almost same performance for all the methods in terms of RMSE. In terms of MAE, SLS-SDPP outperforms all other methods.

the total data are used for training and 40% for testing. 100 such samples are evaluated and their average results are shown in Table 3.1 and 3.2.

Table 3.1: Average RMSE and MAE for test set prediction of Parkinson Telemonitoring dataset

Method	RMSE (mean±std)	MAE (mean±std)
SLS-SDPP	**10.6781±1.1481**	**8.3503±0.8525**
SDPP	10.7934±1.2371	8.7459±0.8378
PLS	10.8133±1.2806	8.7822±0.8883
SPCA	10.8006± 1.2449	8.7714 ±0.8555
KDR	10.8478± 1.3032	8.8008±0.9139

Parkinson's Telemonitoring Data Set : Parkinson's disease is a condition in which parts of the nervous system become progressively damaged. Symptoms of Parkinson's disease includes: involuntary shaking of particular parts of the body (tremor), slow movement and stiff, change in speech, inflexible muscles, depression and anxiety, balance problems, writing changes, anosmis: loss of sense of smell, insomnia, memory problem etc. Parkinson's Telemonitoring Data Set is composed of a range of biomedical voice measurements from 42 people with early-stage Parkinson's disease recruited to a six-month trial of a telemonitoring At-Home-Testing-Device (AHTD) for remote

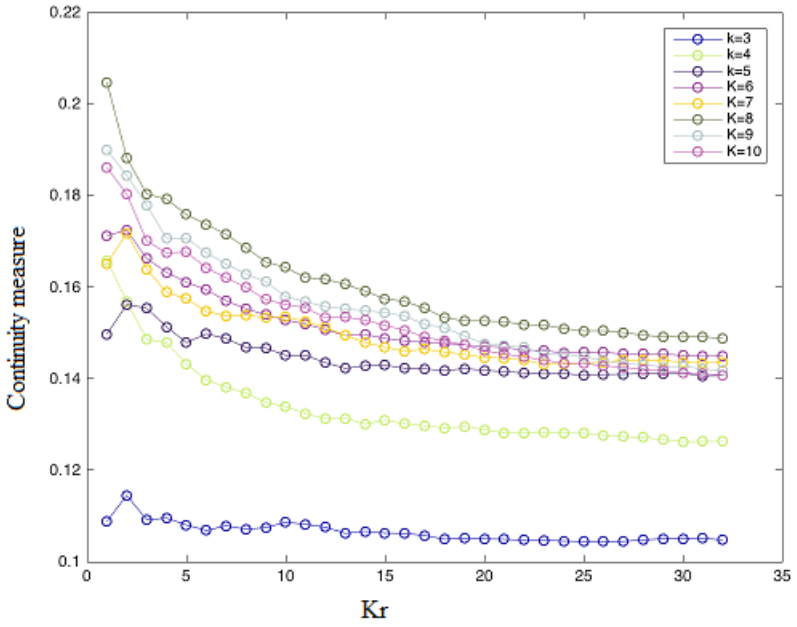
symptom progression monitoring. The recordings were automatically captured in the patient's homes, transmitted to a dedicated server at the clinic through the internet, and calculation of the speech signal processing (dysphonia) measures. The dataset was created by A. Tsanas and M. Little [102] of the University of Oxford, in collaboration with 10 medical centers in the US and Intel Corporation who developed the telemonitoring device to record the speech signals. The original study used a range of linear and nonlinear regression methods to predict the clinician's Parkinson's disease symptom score on the unified Parkinson's disease rating scale (UPDRS).

In this dataset there are around 5875 voice recording from individuals. The main aim of the data is to predict the total UPDRS scores from the 16 voice measures. In [102], the data set is verified to be well fitted at 6 dimensional space. So here we used our approach to obtain the best 6 relevant features and compare the performance with the other DR methods. The value of the parameter k is chosen to be 8 using continuity measure shown in Fig. 3.7(a). In Table 3.1 average root mean square error (RMSE) and mean absolute error (MAE) are given for each of the five methods. The red colored value indicates the minimum error in fitting the data which implies that performance of SLS-SDPP is better compared to other 4 methods which is also verified from Fig. 3.6. The small value of std indicates the stability of our algorithm.

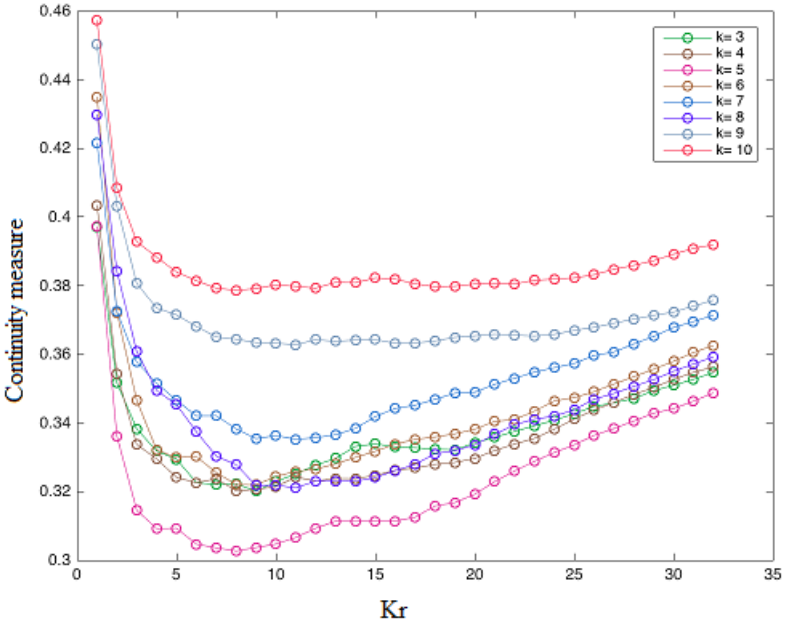
Table 3.2: Average RMSE and MAE (mean±std) for the test set prediction on Concrete Compressive Strength Data Set

Error	Dim	SLS-SDPP	SDPP	PLS	SPCA	KDR
RMSE	1	**10.4649±1.2072**	10.5241±1.4220	12.7666±2.1539	12.8090±2.0429	13.7423±2.7046
	2	10.4540±1.1356	**10.4075±1.5903**	11.8629±1.1837	12.8712±1.9074	11.6399±2.6641
	3	10.4629±1.1648	**10.4079±1.5915**	10.9379±1.1096	12.8370±1.9316	11.5163±2.2178
	4	**10.4450±1.2848**	10.5890±1.3303	10.5108±1.1943	12.8674±1.8353	11.0320±1.7599
	5	**10.2932±0.7866**	10.7247±1.0038	10.4432±1.1883	12.9647±1.5969	**10.2933±1.6597**
	6	10.9910±0.7200	10.4893±0.7228	**10.4359±1.1480**	10.4770±1.1142	10.4473±1.1485
	7	10.4495±0.7052	10.5313±0.7228	**10.4480±1.0746**	10.4520±1.0604	10.4514±1.0952
	8	10.4349±1.1134	**10.4342±1.0034**	**10.4342±1.0034**	**10.4342±1.0034**	16.3019±1.8078
MAE	1	8.3879±1.0882	**8.3687±1.0994**	10.3995±1.7948	10.4451±1.6868	10.8884±2.3565
	2	8.2339±1.2642	**8.2338±1.2918**	9.3977±0.8294	10.4484±1.5463	9.2285±2.3954
	3	**8.2288±1.3385**	8.2342±1.3380	8.3771±0.6452	10.4125±1.5656	8.9818±1.6882
	4	8.5898±1.1262	8.3935±1.1461	**8.2199±0.7943**	10.4553±1.4677	8.6622±1.3565
	5	**8.0177±0.7413**	8.5133±0.8507	8.1563±0.8538	10.5221±1.3204	**8.0167±1.2904**
	6	8.3434±0.7236	8.2713±0.5728	**8.1612±0.8389**	8.1860±0.7925	**8.1659±0.8225**
	7	8.2747±0.5815	8.2820±0.5787	**8.1869±0.7809**	8.1872±0.7735	**8.1868±0.7903**
	8	8.1852±0.7413	**8.1842±0.7477**	**8.1842±0.7477**	**8.1842±0.7477**	13.1814±1.5359

Concrete Compressive Strength Data Set : Concrete is the most commonly used



(a) Continuity measure with respect to different k and k_r for Parkinsons Telemonitoring Data



(b) Continuity measure with respect to different k and k_r for Concrete Compressive Strength Data

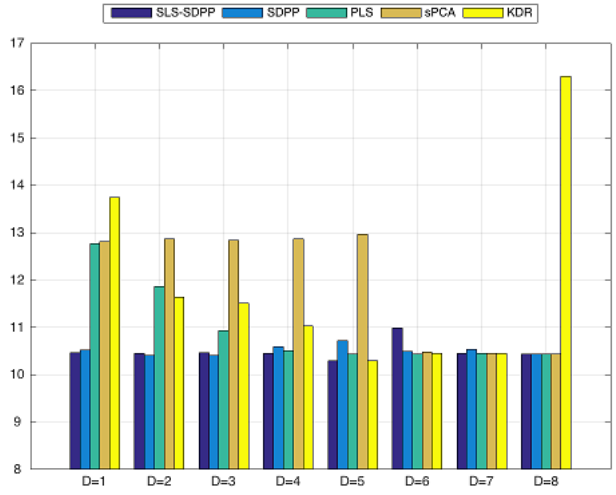
Figure 3.7: Continuity measure with respect to different k and k_r for (a) Parkinsons Telemonitoring Data: Figure suggests to choose the neighbprhood size $k = 8$ since highest continuity measure is obtained at $k = 8$ (b) Concrete Compressive Strength Data: Highest continuity measure is obtained at $k = 10$ therefore $k = 10$ is chosen as the neighborghood size.

structural material with a non-linear mechanical behavior difficult to predict. It is important to understand the compressive strength of concrete for activities like construction arrangement as well as proportioning new mixtures and for the quality assurance. Concrete Compressive Strength Data Set is obtained from UCI repository contains 1030 instance with 9 attributes which presents the compression strength of concrete depending on 7 potentially influential components (in $kg/m3$) (the cement, the blast furnace slags, the fly ashes, the water, the superplasticizers, the coarse aggregates, the fine aggregates) and the age of the material. Table 3.2 reports the regression accuracy of this data set for each of the four methods in terms of RMSE and MAE where red colored value indicates the minimum error in fitting the data, blue colored values indicate best estimation of each method and bold numbers at each row indicates lowest error along that dimension. The data set is best fitted by SLS-SDPP at dimension 5 which can be observed from the table (red colored). KDR showed the same performance in terms of MAE which can also be verified from Fig. 3.8. SDPP achieved its best estimation at lowest dimension ($D = 2$) which is useful for visualization purpose.

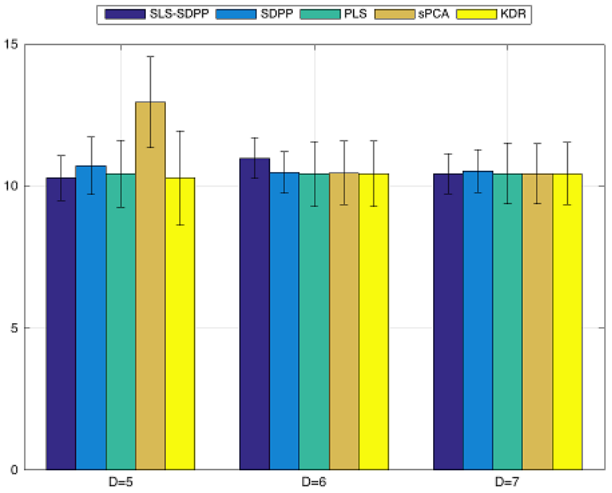
3.7.4 Classification:

This section is focused on classification problems. Several synthetic and real world benchmarking data set from UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets.html) are considered to illustrates the performance of SLS-SDPP compared to other supervised dimensionality reduction methods SDPP, SPCA and KDR in classification task. In addition, we also compared our algorithm with Fishers Discriminant Analysis (FDA). For each of the dataset 60% of the data is used to calculate the transformation matrix which is used to predict the class of remaining 40% data using nearest- neighbor classifier.

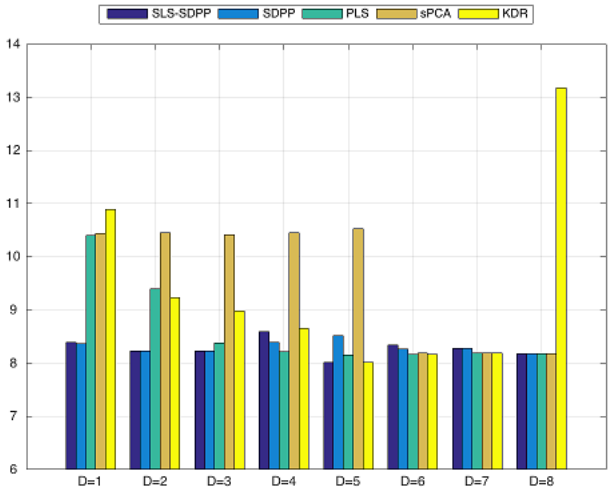
TaiChi data: TaiChi is a well known symbol in Asian culture. It represents two opposing entities Yin and Yang which is shown by black and white region in Fig. 3.9(a). This symbol provides intellectual framework of the scientific development especially in fields like biology and traditional medical sciences in ancient China. The basic structure of Tai Chi is formed by drawing one large circle, two medium half-circles and two small circles. The two small Yin and Yang circles, located at the centers of the Yang and Yin



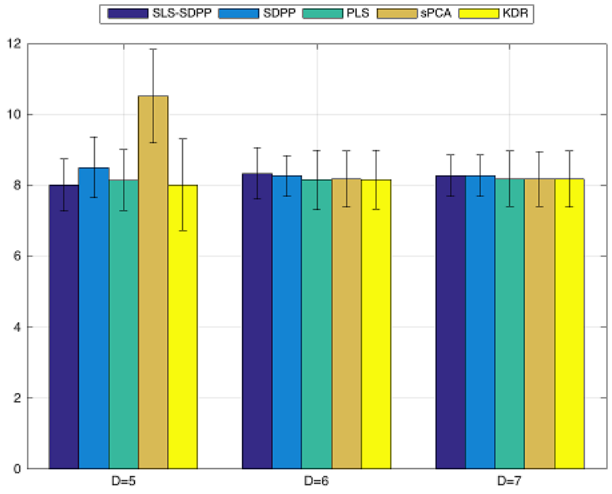
(a) Average RMSE along different dimension.



(b) Average RMSE with error bar along different dimension.



(c) Average MAE along different dimension.

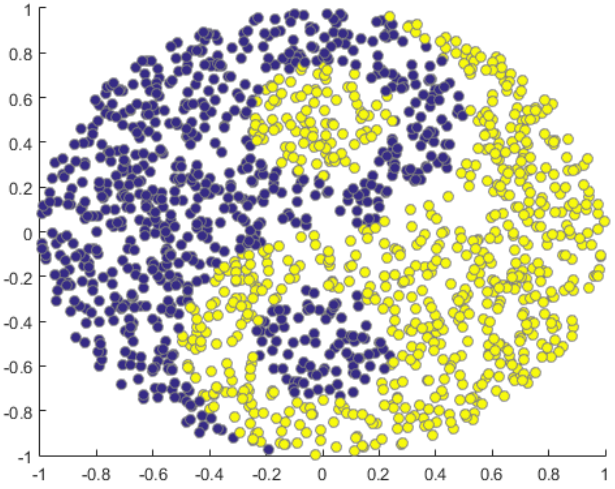


(d) Average MAE with error bar along different dimension.

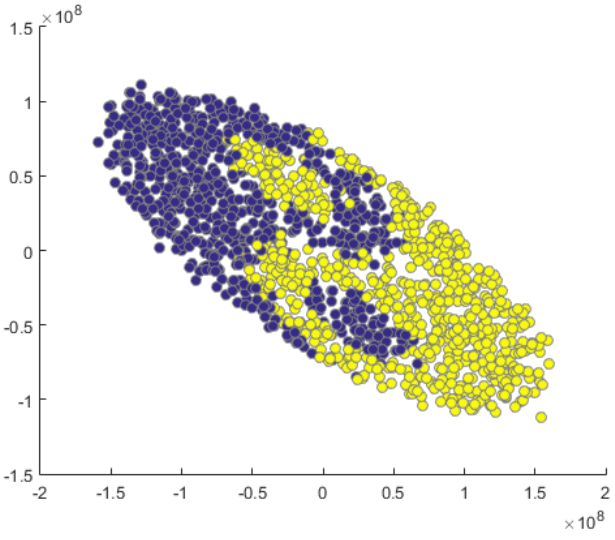
Figure 3.8: Average RMSE and MAE for test data prediction of Concrete Compressive Strength Data Set along different dimension obtained by SLS-SDPP, SDPP, PLS, SPCA and KDR. The diagrams show, best performance achieved by SLS-SDPP at D=5. The small error bar implies the stability of our method regardless of training data.

half- circles that are tangent to each other and also to the large circle. Tai Chi symbol is used previously in [117], to define a toy classification problem, where Yin and Yang are two distinct classes. We hereby used the TaiChi data which is a discretized version of the symbol simulated by Zhu et al in [117]. The original data set contains 2000 5-dimensional vector. The goal is to identify the first two effective directions for a correct classification of Yin and Yang. All the four methods ADMM, SDPP, SPCA and KDR are considered to check whether they can obtain the first two features. 1000 points are

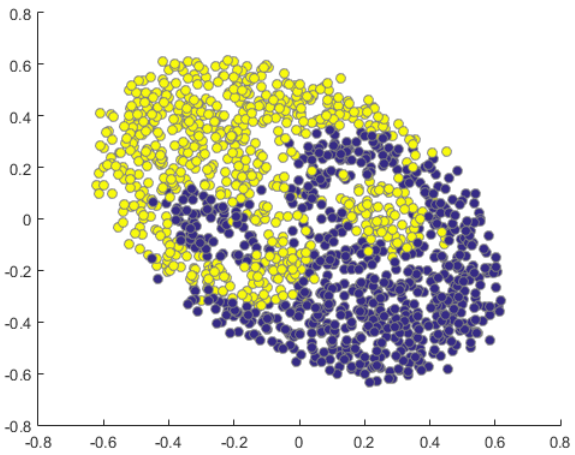
(a) Scatter plot of TaiChi data



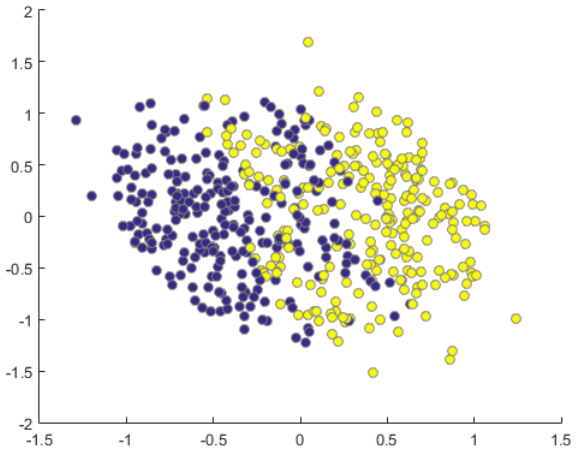
(b) Simulation of TaiChi model



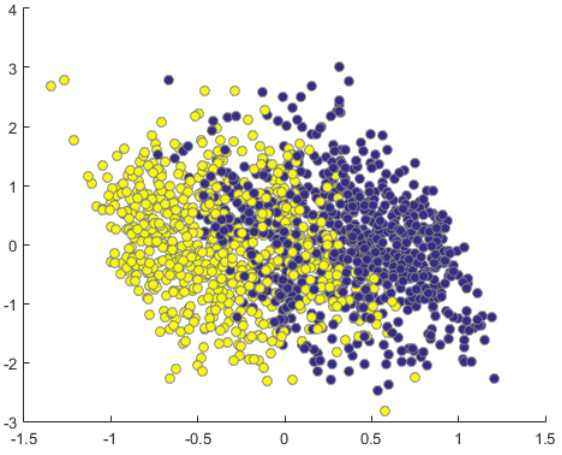
(c) SLS-SDPP projection



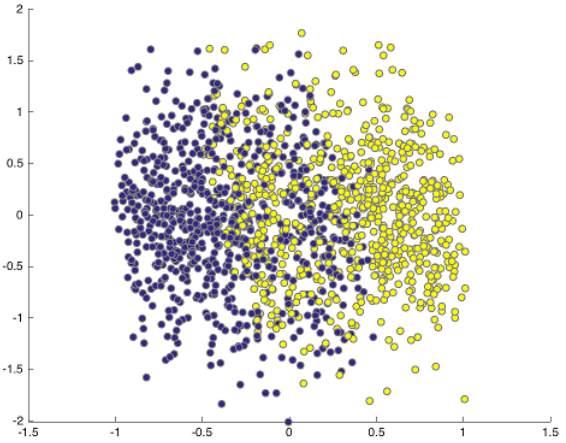
(d) SDPP projection



(e) sPCA projection



(f) KDR projection

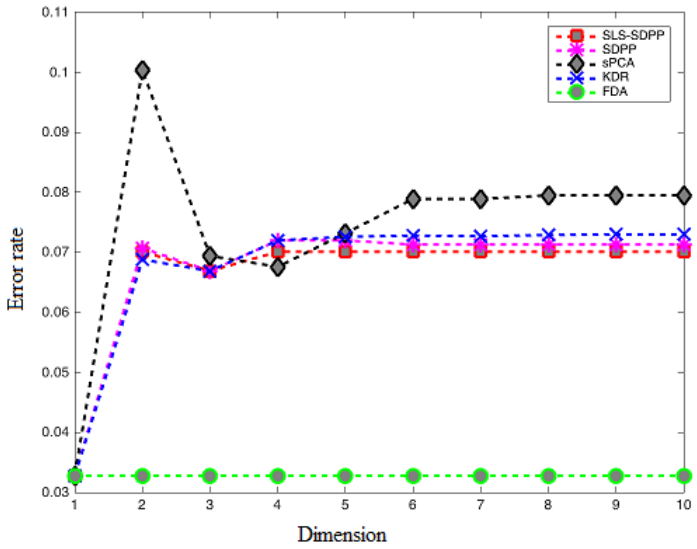


(g) FDA projection

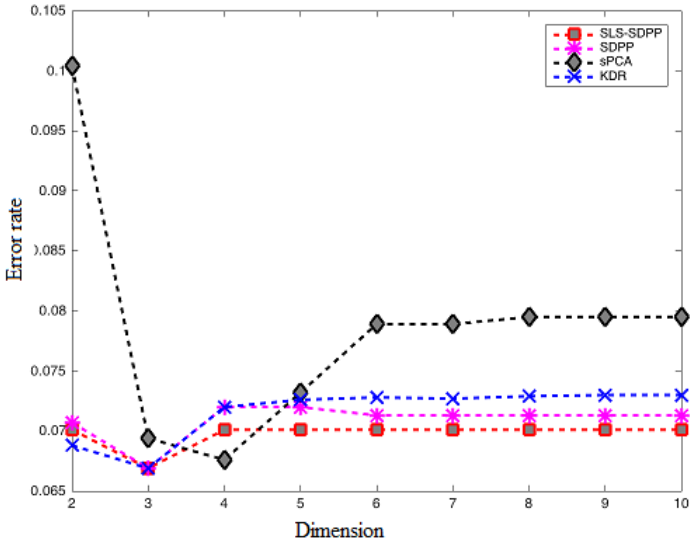
Figure 3.9: TaiChi data. (a) TaiChi model (b) Simulation of TaiChi . (c)-(g) presents the projection by ADMM, SDPP, SPCA, KDR and FDA respectively. Figures show that only SLS-SDPP and SDPP classified the data points successfully and projected correctly.

considered for training purpose. Fig. 3.9 presents the two-dimensional projections for the test set which shows that SPCA and KDR are incapable of separating the classes But ADMM and SDPP appear to be fully successful to separate the small circles and the big half circles.

Seismic bump data: Mining activity is connected with the occurrence of dangers which are commonly called mining hazards. A special case of such threat is a seismic hazard which frequently occurs in many underground mines. Seismic hazard is the hardest detectable and predictable of natural hazards and in this respect it is comparable



(a) Error Rate among different projection dimension



(b) Error Rate among different projection dimension

Figure 3.10: Seismic bump data. (a) Classification error rates for different projection dimension computed by algorithm ADMM, SDPP, SPCA, KDR and FDA. (b) Classification error rates for different projection dimension computed by SLS-SDPP, SDPP, SPCA and KDR. Figures show that minimum classification error rate is obtained at D=1 by all the methods.

to an earthquake. Seismic bump dataset, collected from UCI repository, contains 2584

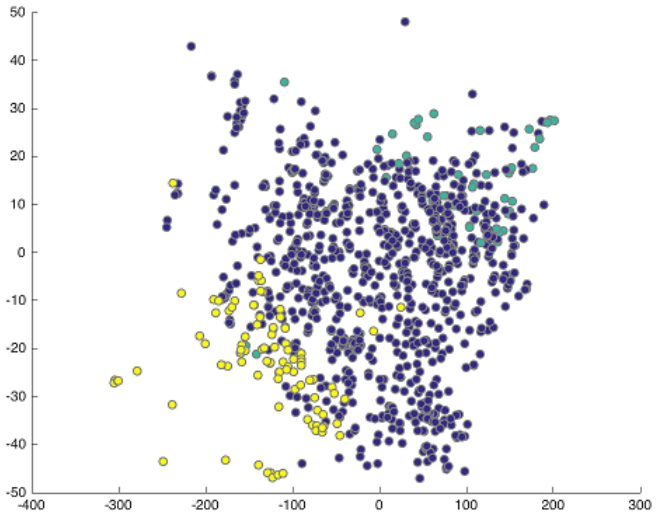
Table 3.3: Average classification error rate of test set for Seismic bump data

Error	Dim	SLS-SDPP	SDPP	SPCA	KDR	FDA
Error Rate	1	**0.0328**	**0.0328**	**0.0328**	**0.0328**	**0.0328**
	2	**0.0701**	0.0707	0.1004	0.0688	0.0328
	3	0.0701	**0.0669**	0.0694	**0.0669**	0.0328
	4	0.0701	0.0720	0.0676	0.0720	0.032
	5	**0.0701**	0.0720	0.0732	0.0726	0.0328
	6	**0.0701**	0.0713	0.0789	0.0728	0.0328
	7	**0.0701**	0.0713	0.0789	0.0727	0.0328
	8	**0.0701**	0.0713	0.0795	0.0729	0.0328
	9	**0.0701**	0.0713	0.0795	0.0730	0.0328
	10	**0.0701**	0.0713	0.0795	0.0730	0.0328

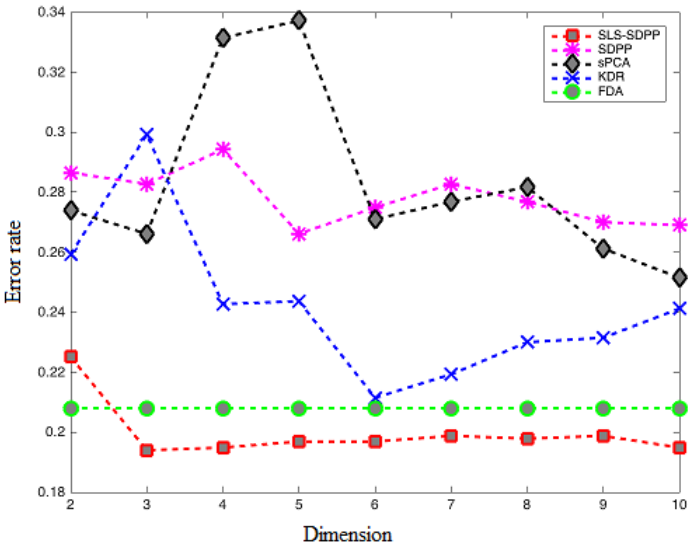
data having 19 attributes. The task of seismic prediction can be defined in different ways, but the main aim of all seismic hazard assessment methods is to predict (with given precision relating to time and date) of increased seismic activity which can cause a rockburst. That task of hazards prediction bases on the relationship between the energy of recorded tremors and seismoacoustic activity with the possibility of rockburst occurrence. Hence, such hazard prognosis is not connected with accurate rockburst prediction. Moreover, with the information about the possibility of hazardous situation occurrence, an appropriate supervision service can reduce a risk of rockburst (e.g. by distressing shooting) or withdraw workers from the threatened area. Good prediction of increased seismic activity is therefore a matter of great practical importance.

We have applied SLS-SDPP, SDPP, SPCA, KDR and FDA to project the seismic bump data into lower dimensional space. 60% data is considered for training and 40% for testing. Fig. 3.10 depicts the error rate of testing data along different projecting dimension which shows that lowest error rate 3.28% is obtained at dimension 1 and all the five methods were successful to achieve this accuracy. The next best estimation is achieved at dimension 3 obtained by SLS-SDPP, SDPP AND KDR and it is notable that error rate obtained by SLS-SDPP doesn't change much with the projection dimensionality and remained lower than all other methods which can also be seen from Table 3.3.

Cardiotocography Data Set: cardiotocography (CTG) is a technical means of recording the fetal heartbeat and the uterine contractions during pregnancy. This dataset



(a) Scatter plot of CTG data



(b) Error Rate among different projection dimension

Figure 3.11: CTG data. (a) 2D projection of data (b) Classification error rates for different projection dimension computed by algorithm SLS-SDPP, SDPP, SPCA, KDR and FDA. Figure (b) illustrates that best performance is achieved at $D = 3$ by SLS-SDPP. Also the performance of SLS-SDPP is consistently better than all other methods.

contains the diagnostic features of 2126 fetal cardiotocograms (CTGs) processed automatically. The CTGs were also classified by three expert obstetricians and a consensus classification label assigned to each of them. 21 attributes are measured and one of the three fetal states Normal (N), Suspect (S) and Pathologic (P) is concluded . 60% of the data are considered for training and the remaining data are used for testing. Fig. 3.11 shows testing data classification error rate along different projection dimension. It is easily visible from Fig. 3.11 as well as from Table 3.4 that SLS-SDPP projected the data with lowest error rate 0.1940 at dimension 3 and the error rate remained much lower than all other methods in the following dimensions . SDPP produces its best estimation at 5 dimensional space with error rate 0.2661 and for SPCA and KDR the minimum error rate is 0.2515 and 0.2115 obtained at projection dimension 9 and 7 respectively which can also be observed from Table 3.4. Since this data set is of 3 classes, for FDA the solution rank is 2. Therefore the error rate remains constant for any dimension $m \geq 2$. Similar to seismic data set problem , error rate obtained by SLS-SDPP doesn't change much with the projection dimensionality which is beneficial for practical applications where most of the time the projection dimensionality is determined by cross validation.

Table 3.4: Average error rate of class prediction of test set for Cardiotocogram data

Error	Dim	SLS-SDPP	SDPP	SPCA	KDR	FDA
Error Rate	2	**0.2251**	0.2865	0.2739	0.2593	0.208
	3	**0.1940**	0.2827	0.2661	0.2992	0.208
	4	**0.1949**	0.2943	0.3314	0.2427	0.208
	5	**0.1969**	0.2661	0.3372	0.2437	0.208
	6	**0.1969**	0.2749	0.2710	0.2115	0.208
	7	**0.1988**	0.2827	0.2768	0.2193	0.208
	8	**0.1979**	0.2768	0.2817	0.2300	0.208
	9	**0.1988**	0.2700	0.2612	0.2315	0.208
	10	**0.1949**	0.2690	0.2515	0.2412	0.208

Diabetic Retinopathy data: The Diabetic Retinopathy dataset contains features extracted from the Messidor image set. Messidor database [1] [25] contains hundreds of eye fundus images. The fundus of the eye is the interior surface of the eye opposite the lens and includes the retina, optic disc, macula, fovea, and posterior pole. Diabetic

[1] Kindly provided by the Messidor program partners (see http://www.adcis.net/en/DownloadThirdParty/Messidor.html).

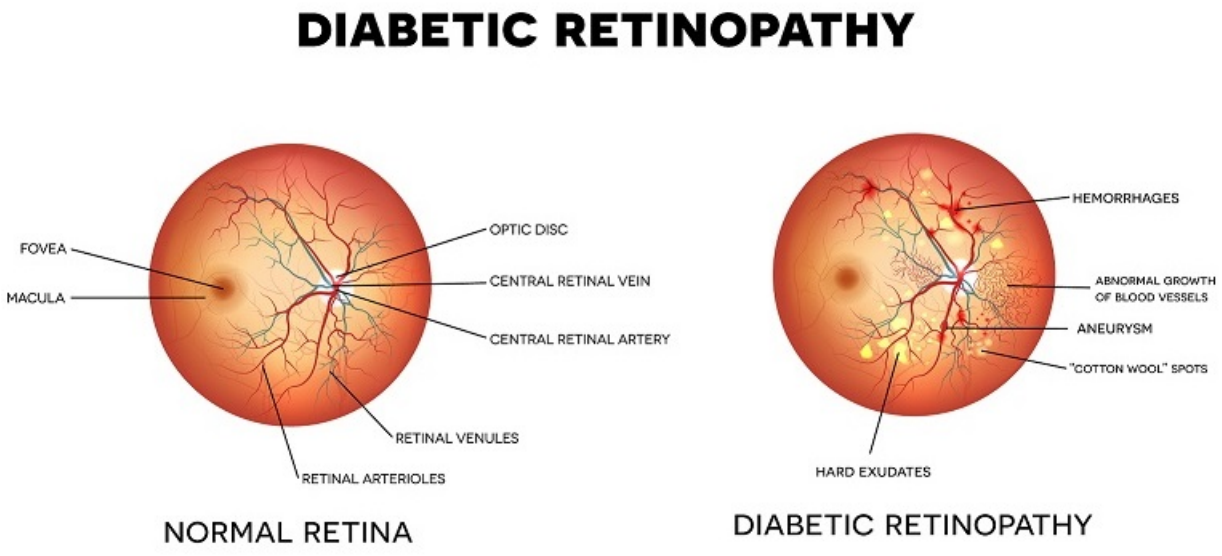
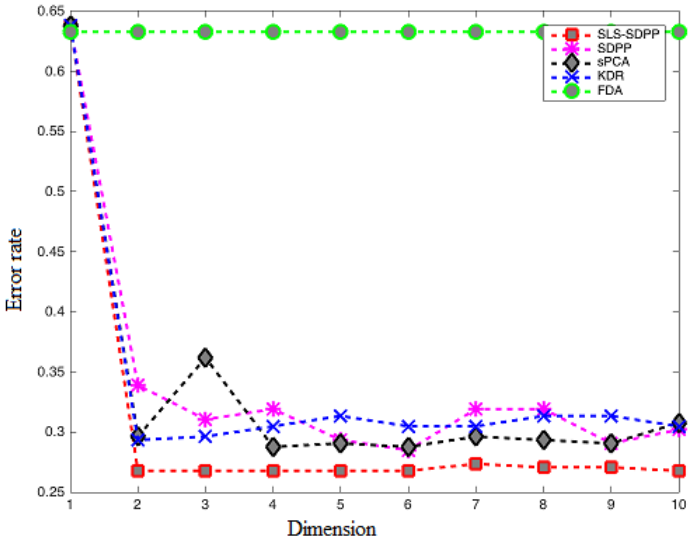


Figure 3.12: Condition of eyes of a person having diabetes



(a)

Figure 3.13: Diabetic-Retinopathy data. (a) Condition of eyes of a person having diabetes (b) Classification error rates for different projection dimension computed by SLS-SDPP, SDPP, SPCA, KDR and FDA. Figure suggests that lowest error rate is obtained by SLS-SDPP and the error rate for this method remained consistently lower then other methods.

Table 3.5: Average error rate of class prediction of test set for Diabetic Retinopathy data

Error	Dim	SLS-SDPP	SDPP	SPCA	KDR	FDA
Error Rate	1	0.6382	0.6382	0.6382	0.6382	0.6382
	2	**0.2678**	0.3390	0.2963	0.2934	0.6382
	3	**0.2678**	0.3105	0.3618	0.2963	0.6382
	4	**0.2678**	0.3191	0.2877	0.3048	0.6382
	5	**0.2678**	0.2934	0.2906	0.3134	0.6382
	6	**0.2678**	0.2849	0.2877	0.3048	0.6382
	7	**0.2735**	0.3191	0.2963	0.3048	0.6382
	8	**0.2707**	0.3191	0.2934	0.3134	0.6382
	9	**0.2707**	0.2906	0.2906	0.3134	0.6382
	10	**0.2678**	0.3020	0.3077	0.3048	0.6382

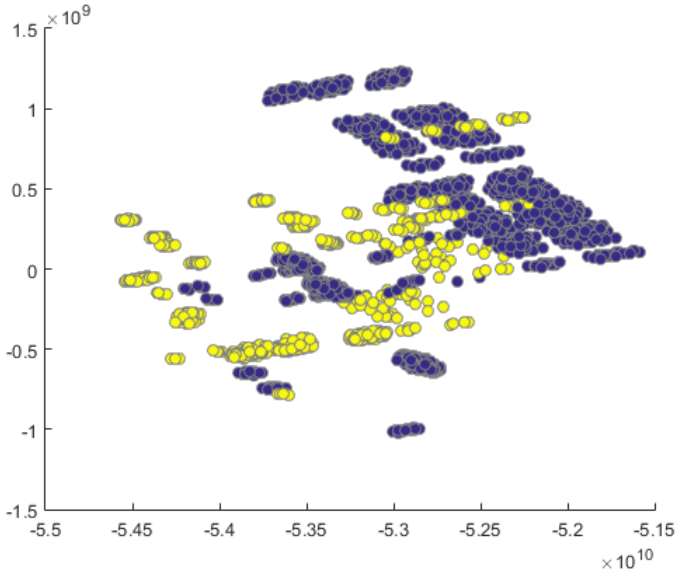
retinopathy is the eye condition that affect people with diabetes. Fig. 3.12 presents the image of a healthy eye and an eye of a diabetic patient.

The dataset contains 1115 eye fundus images each with 19 features set to predict whether an image contains signs of diabetic retinopathy or not. All features represent either a detected lesion, a descriptive feature of a anatomical part, or an image-level descriptor. Similar to previous experiments we considered 60% of the data for learning the transformation matrix and the remaining 40% for testing the accuracy of the classification task obtained by our proposed method as well as the other 4 methods.

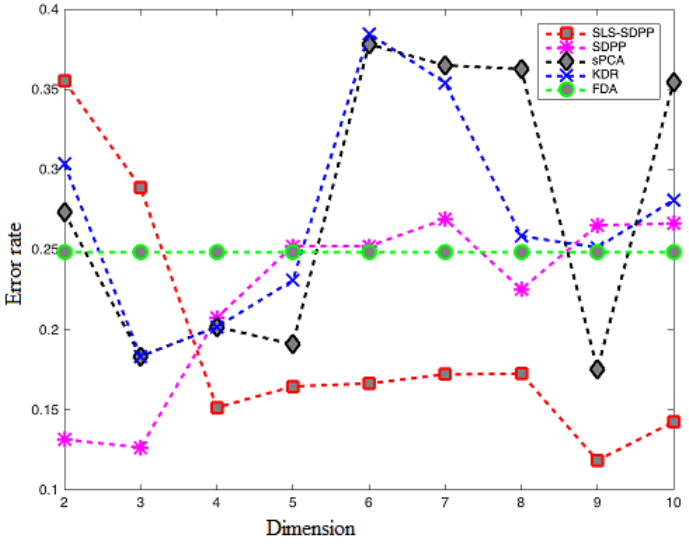
Table 3.5 reports the average error rate for the test set prediction along different projection dimension. For each method, the blue colored numbers are used to indicate their best performance. It can be observed that SLS-SDPP can classify the test data more accurately than all other methods . The next better performance is obtained by SPCA at $D = 4$ as shown in Fig. 3.13. It can also be observed from Fig. 3.13(a) that FDA failed to produce a convincing projection of the test data.

Mushroom data: This data set is a benchmarking data collected from UCI repository. This is 22 dimensional dataset with 8124 instance that includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family.

Each species is identified as definitely edible or definitely poisonous.We have used 60% points for learning and rest of the data for testing. Two dimensional projections of the



(a) Scatter plot of Mushroom data



(b) Error Rate among different projection dimension

Figure 3.14: Mushroom data. (a) Scatter plot of Mushroom data, (b) Classification error rates for different projection dimension computed by algorithm ADMM, SDPP, SPCA, KDR and FDA. Best performance is obtained at $D = 9$ by SLS-SDPP

Table 3.6: Average error rate of class prediction of test set for Mushroom data

Error	Dim	SLS-SDPP	SDPP	SPCA	KDR	FDA
Error Rate	1	0.2486	0.2486	0.2486	0.2486	0.2486
	2	0.3555	**0.1318**	0.2735	0.3037	0.2486
	3	0.2891	**0.1266**	0.1836	0.2741	0.2486
	4	**0.1516**	0.2076	0.2020	0.2018	0.2486
	5	**0.1648**	0.2524	0.1911	0.2311	0.2486
	6	**0.1667**	0.2524	0.3785	0.3815	0.2486
	7	**0.1723**	0.2693	0.3653	0.3544	0.2486
	8	**0.1728**	0.2255	0.3630	0.2587	0.2486
	9	**0.1186**	0.2655	0.1756	0.1615	0.2486
	10	**0.1427**	0.2665	0.3545	0.2812	0.2486

data is presented in Fig. 3.14. The average classification error rate of the test data along different projection dimension can be observed from table 3.6, which shows that SDPP obtains its best estimation at D=3. But from D=4, the error rate in SLS-SDPP remained consistently lower than all other methods and obtain the best estimation at D=9. SPCA and KDR also have their minimum error rate at D=9.

3.8 Summary

In this chapter we have worked on dimension reduction methods on supervised settings. Among different dimension reduction methods recently proposed Supervised Distance Preserving Projection (SDPP) method showed very promising result in data mining. The method learns a linear mapping from the input space to the reduced feature space in such a way that the local geometrical structure of the low dimensional subspace preserves the geometrical characteristics of the response space. For each data point, the local structure is preserved by keeping the distance of k nearest neighbors. The value of parameter k is chosen by a continuity measure. Though the methods works very well in regression task, its performance is not that convincing for classification problems.

In this chapter, we have proposed a modification of SDPP which deals the classification problem and significantly improves the performance of SDPP. We have incorporated the total variance of the projected co-variates to the SDPP problem that keeps the distance relation with neighbors (preserves local structure) and at the same time preserves the global structure by maximizing the total variance. This approach not only facilitates efficient regression like SDPP but also successfully classifies data into different classes.

We have formulated the proposed optimization problem as a Semidefinite Least Square (SLS) SDPP problem. A two block Alternating Direction Method of Multipliers have been developed to learn the transformation matrix solving the SLS-SDPP which can easily handle out of sample data. The projections of testing data points in low dimensional space are further used for regression or assigning them into fixed number of classes. Experimental evaluations on several synthetic and real world large data sets illustrate that in regression tasks our method exhibits an equivalent or better generalization performance compared to other methods and in classification problems, the error rate obtained by SLS-SDPP, is halted after a certain projection dimension and consistently remained lower than that of all other methods. Thus **SLS-SDP significantly improves SDPP and outperforms some other existing state-of-the-arts-approaches.**

Chapter 4

Application to Face Recognition

4.1 Introduction

Personal identication or verification is a very common requirement in modern society specially to access restricted area or resources, to travel abroad etc. Biometric identification systems are now being used almost everywhere as it is more secure and user-friendly. So this area is getting more focused from researchers. The most common biometric techniques are automated recognition of fingerprints, faces, iris, retina, hand print and voice. Also video surveillance system has become one of the most popular systems in terms of security. So face identification or recognition in a controlled or an uncontrolled scenario has become one of the most important and challenging area of research.

A general face recognition problem can be stated as follows:

Given a set of face images labeled with the persons identity (the training set) and an unlabeled set of face images from the same group of people (the test set). The aim is to **identify each person in the test set.**

In a face recognition problem, an image is considered as a high dimensional vector where each of the coordinates corresponds to a pixel value in the sample image.

This chapter is concerned with the application of our proposed method SLS-SDPP, discussed in chapter 4, on face recognition problems to reduce the dimension of face image data. The maximizing variance as well as the structure preserving approach of SLS-SDPP showed remarkable performance in comparison to two leading methods Eigenface

[103] and Fisherface [5] both in controlled and uncontrolled scenario. Numerical experiments are conducted on three very well known data set Human face data, Yale and ORL. For the classification task k-NN algorithm is applied on the reduced dimensional vectors.

4.2 Previous Studies

Many face recognition techniques have been developed over the past few decades. A complete survey can be found in [19, 116]. Though some of these systems successfully complete the job in constrained scenarios, the general task of face recognition still poses a number of challenges.

Among different techniques, one of the well-established and successful method is appearance-based method [73, 103, 93]. The appearance-based methods use the high dimensional ($n \times m$; eg. $64 \times 64 = 4096$) vector as input to a classifier. Though this technique works well for classifying frontal views of faces they are highly sensitive to pose variation. Therefore addition of an alignment stage (input face image is warped to a reference face image based on some correspondence like position of eyes, chin, nose, two corners of mouth etc) before the classification can help to avoid this problem. Another alternative to the appearance based approaches is to classify local facial components. The main idea is to match templates of different facial regions (both eyes, nose, mouth) independently. Some efficient appearance based (also known as global approach) and component base techniques are proposed in [61, 50, 57]. Despite the success of these techniques, working with high dimensional (eg. 4096) dataset lead to high computational and storage demands. A possible solution for reducing this storage amount and speeding-up the computations is to use feature extraction methods. Previous works have demonstrated that dimensionality reduction provides an efficient way to detect intrinsic structures of data as well as to extract a reduced number of variables that capture the most relevant features of the high-dimensional data. In the last decades different linear and nonlinear dimension reduction methods such as, principal component analysis (PCA) and an its

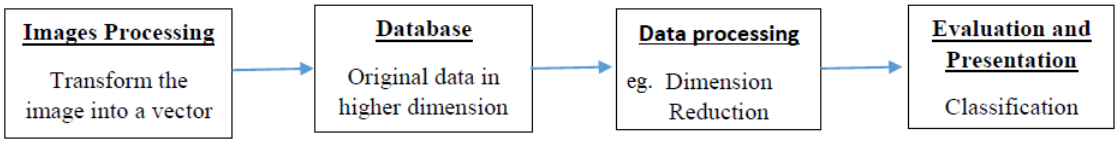


Figure 4.1: Basic steps of Face recognition procedure

nonlinear variants (KPCA) [54, 55, 5], local linear embedding (LLE), Curvilinear component analysis (CCA) are being used by several authors to reduce the dimension of face image vectors.

Turk and Pentland introduce the Eigenface method for face recognition [103]. A well known method Fisher face proposed by Belhumeur et al. [5] uses PCA for dimension reduction step and LDA for the classification. Zhuang et al [118] proposed to use Inverse Fishers discriminant criteria (IFFace) as Fisher Face method might fail for some dataset. Some other wellknown subspace learning algorithms are Locality Preserving Projection (LPP) [49], Neighborhood Preserving Embedding (NPE) [47], Local Discriminant Embedding (LDE) [18]. Cai et al [26] proposed regularized subspace learning model using a Laplacian penalty to constrain the coefficients to be spatially smooth. Kukharev and Forczmańnski used few variants of Karhunen-Loeve Transform (KLT) and Linear Discriminant Analysis (LDA) [60]. Manifold learning techniques such as ISOMAP [96], LLE [90] and Laplacian Eigenmap [6] consider nonlinear dimensionality reduction by investigating the local geometry of data. These techniques are good for representation, but only concern with the training data.

Most of the above dimensionality reduction techniques preserves the local structure or focused on the global structure. So building a method that preserves local structure as well as maximizes the global variance can be more reliable for a classification problem. Here we have applied our proposed method SLS-SDPP, to reduce the dimension of face image data. For the classification task k-NN algorithm is applied on the reduced dimensional vectors. Numerical experiments conducted on three very well known data set Human face data , Yale and ORL showed remarkable performance of SLS-SDPP in comparison to two leading methods Eigenface [103] and Fisherface [5] both in controlled

and uncontrolled scenario.

4.3 Problem Formulation

Given a set of N sample images $\{x_1, x_2, \ldots, x_N\}$ in an n-dimensional image space and assuming that each image belongs to one of the l classes $C_1, C_2, ..., C_l$, consider a linear transformation $z = W^T x$ from the original n-dimensional image space into an m-dimensional feature space, where $m < n$ and where $W \in \Re^{n \times m}$ is a matrix with orthonormal columns. Here W^T is the transformation matrix from higher dimensional image space to the lower dimensional space.
Different techniques have been used by several researchers to determine this transformation matrix W. Here we have briefly discussed the idea of two leading methods Eigenface and Fisherface.

4.3.1 Eigenface

Eigenface method uses Principal Component Analysis (PCA) to reduce the dimension of the image space by maximizing the total scatter of all projected samples. If the total variance matrix S is defined by:

$$S = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \tag{4.1}$$

where $\mu \in \Re^n$ is the mean image of all samples and $S \in \Re^{n \times n}$, then the basic idea of Eigenface method is to determine the transformation matrix W in such a way that the determinant of the total scatter matrix $W^T S W$ of the projected sample is maximized. Thus the objective function of Eigenface method is

$$\max_{W \in \Re^{n \times m}} |W^T S W| \tag{4.2}$$

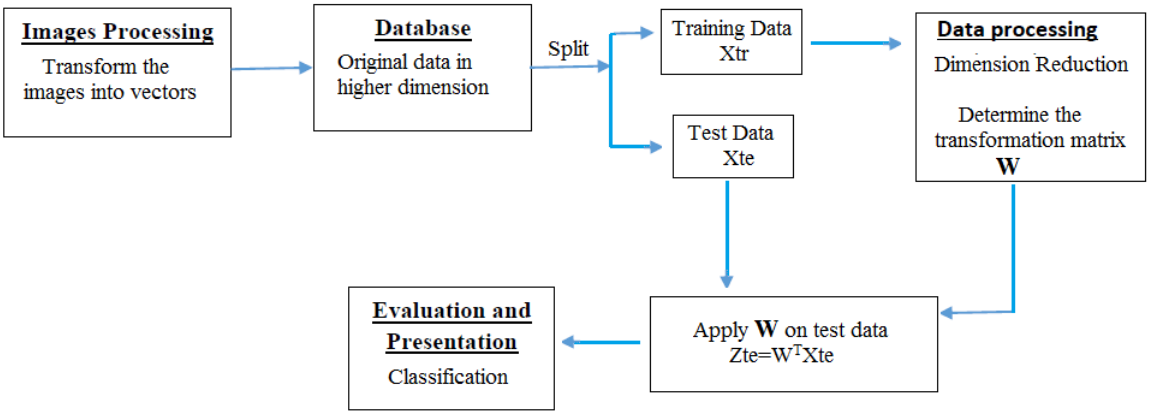


Figure 4.2: Overview of Face recognition method using dimension reduction.

The m columns of the optimum matrix W is the set of n dimensional eigenvectors corresponding to m largest eigenvalues of the matrix S. Each of this eigenvectors having the same dimension as the original images, referred to as Eigenpictures or Eigenfaces.

Note that the Eigenface method doesn't use the class information of the images to determine the projection matrix.

4.3.2 Fisherface

Since the main intention of face recognition problem is to identify the classes of test images, so using the class information of the training images in determining the transformation matrix W may increase the classification rate. Based on this idea Belhumeur proposed the Fisherface method in [5] which uses Fishers Linear Discriminant (FLD). This method selects W in in such a way that the ratio of the between-class scatter and the within class scatter is maximized.

Let S_b be the between class scatter matrix defined by

$$S_b = \sum_{i=1}^{l} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

and S_w be the within class scatter matrix defined by

$$S_w = \sum_{i=1}^{l} (x_i - \mu_i)(x_i - \mu_i)^T.$$

μ_i is the mean of sample images of class C_i and N_i is the number of images in class X_i. If S_w is nonsingular, the optimal projection W_{opt} is chosen as the matrix with orthonormal columns which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples. That is,

$$W_{opt} = arg \max_W \frac{|W^T S_b W|}{|W^T S_w W|}$$

The m columns $w_1, w_2, \ldots, w_m$ of optimum W are the generalized eigenvectors corresponding to the m largest generalized eigenvalues λ_i of S_b and S_w. That is, $S_b w_i = \lambda_i S_w w_i$.

Note that $m \leq l - 1$, where l is the number of classes, since the maximum number of nonzero generalized eigenvalues is $l - 1$ [5].

In the face recognition problem, one is confronted with the difficulty that the within-class scatter matrix $S_w \in \Re^{n \times n}$ is always singular. This stems from the fact that the rank of S_w is at most $N - l$, and, in general, the number of images in the learning set N is much smaller than the number of pixels in each image n. This means that it is possible to choose the matrix W such that the within-class scatter of the projected samples can be made exactly zero. In order to overcome the complication of a singular S_w , Belhumeur proposed the following alternative methodology in [5].

Fisherface method uses two steps to determine the optimum transformation matrix W_{opt}. First step is to reduce the dimension n of the original image space to $N - l$ using PCA so that the resulting within-class scatter matrix S_w is nonsingular. The second step is to apply the Fishers Linear Discriminant (FLD) (which uses the class information of the training images) on the transformed data to reduce the dimension $N - l$ to m.

Thus Fisherface method aims to determine the matrix

$$W_{opt}^T = W_{FLD}^T W_{PCA}^T$$

from the following two steps:

Step 1:

$$W_{PCA} = \arg\max_{W \in \Re^{n \times (N-l)}} |W^T S W|$$

Step 2:

$$W_{FLD} = \arg\max_{W \in \Re^{(N-l) \times m}} \frac{|W^T W_{PCA}^T S_b W_{PCA} W|}{|W^T W_{PCA}^T S_w W_{PCA} W|}$$

In step 1, S is the variance matrix of total sample and the $N-l$ columns of the optimum matrix W_{PCA} is the eigenvectors corresponding to $N-l$ largest eigenvalues of the matrix S.

In step 2, S_b, and S_w are the between class and withn class scatter matrix defined earlier.

In Fisherface method though PCA at the first step is used to avoid the nonsingularity of the within class scatter matrix S_w, this PCA step doesn't guarantee the nonsingularity of the transformed covariance matrix [118]. On the other hand, a drawback of Eigenface method is that the maximization of total variance not only maximizes the between class scatter but also the within class scatter which leads to lower classification rate.(Verified by numerical experiments.)

In view of these limitations, we propose to use our model SLS-SDPP that maximizes the variance of the total sample and preserves the distances of local points by minimizes the differences between distances among projected co-variates and distances among responses locally.

4.3.3 SLS-SDPP

Our proposed approach determines the matrix W in such a way that the local geometrical structure of the projected low dimensional subspace preserves the geometrical characteristics of the response space which means that the images of same class are clustered in projected space (Shown in Fig. 4.6(b),4.8(b)). At the same time SLS-SDPP maximizes the variance of total sample that prevents the different class of images to be

projected very close, therefore preserves the global structure. Thus the objective of our proposed approach SLS-SDPP is

$$max_{W\in\Re^{n\times m}} \sum_{i=1}^{n} \|W^T x_i\|^2 - \frac{\nu}{n} \sum_{ij} G_{ij}(d_{ij}^2(W) - \delta_{ij}^2)^2$$

where G is the neighborhood graph defined by

$$G_{ij} = \begin{cases} 1 & \text{if } i \sim j(k - NNneighbor) \\ 0 & \text{otherwise,} \end{cases}$$

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \sim j(x_i \textit{ and } x_j \textit{ belongs to same class}) \\ 1 & \text{otherwise,} \end{cases}$$

and $d_{ij}^2(W) = \|W^t(x_i - x_j)\|^2$

The detailed description of our method is discussed in Chapter 3. In this chapter we have applied our proposed algorithm Alg. 3.5 to determine the projection matrix W. NN- rule is further applied to identify the classes of the images.

An overview of face recognition methods using dimension reduction is shown in Fig. 4.2. All of the three methods discussed above follow these basic steps to identify the class of the test images.

4.4 Visualization of human face data:

At first we have applied SLS-SDPP to visualize the benchmark dataset of artificially generated human face images.

The human face data is collected from ISOMAP database (http://isomap.stanford.edu/datasets.html). This data consists of 698 synthesized face images observed under different poses and lighting directions. The input consists of a sequence of (64×64) 4096-dimensional vectors. Each one of these 4096 elements corresponds to the brightness of individual pixel.

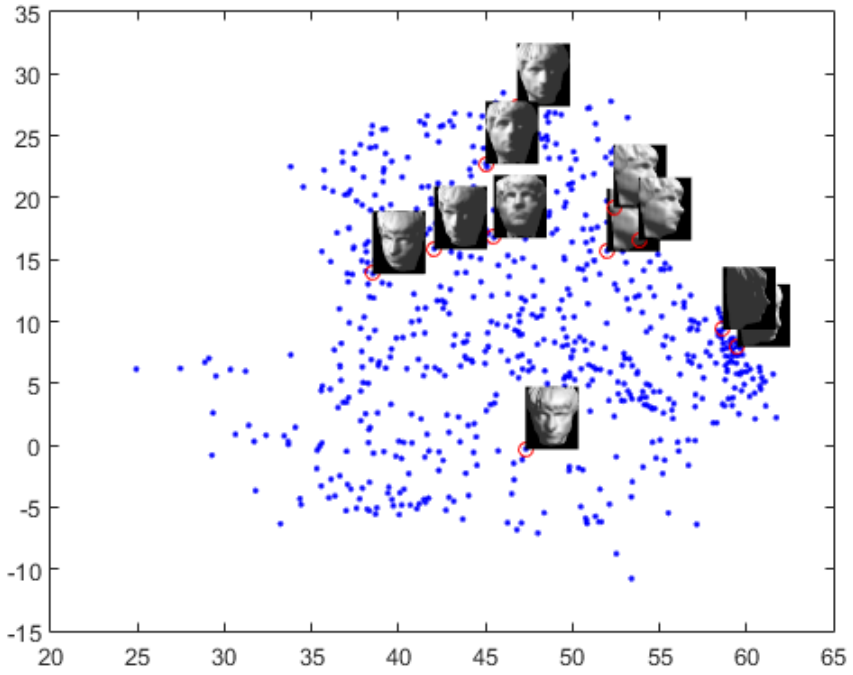


Figure 4.3: Projection of Human face data into 2D space; The x axis in Fig. 4.3 represents the left-right (right to left) poses and the y axis represents the up-down (down to up) poses of the faces.

The dataset is preprocessed by mean centering and normalized to unit variance.

For learning the transformation matrix W, we choose 60% of the total faces as training samples. SLS-SDPP is applied on the training data to determine the projection matrix $W \in \Re^{4096}$ which is used to project the test images into a two dimensional space to visualize the underlying structure. The x axis in Fig. 4.3 represents the left-right poses and the y axis represents the up-down poses of the faces. A sample of the original input images (red circles) superimposed on the corresponding data points (blue) which indicates that in most of cases the projection preserves the continuity of left-right and up-down poses significantly.

4.5 Recognition from gallery image:

In this section we will apply Eigenface, Fisherface and SLS-SDPP on two very well known face data set Yale and ORL. Yale data base is mainly generated by Computer

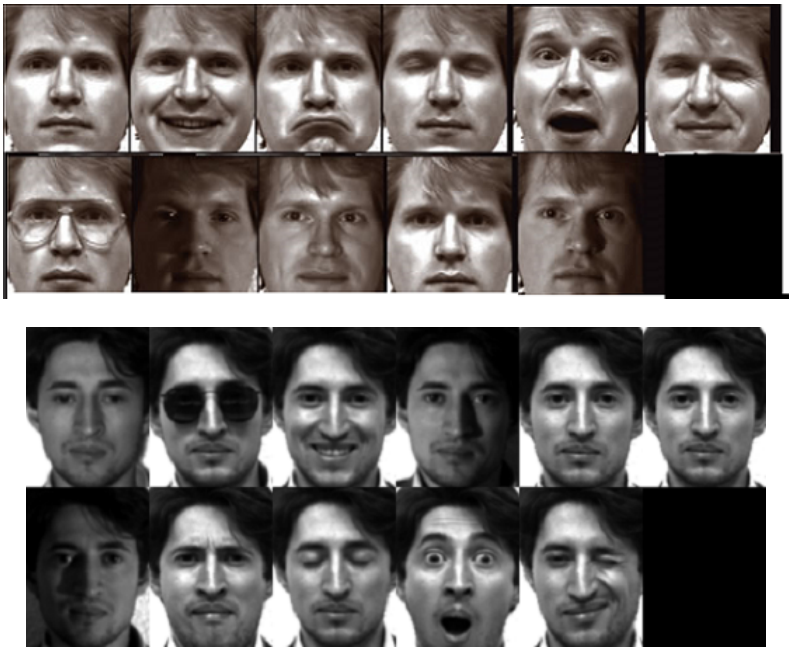


Figure 4.4: Illustration of face images with different lighting condition and facial expression of two individuals from Yale database.

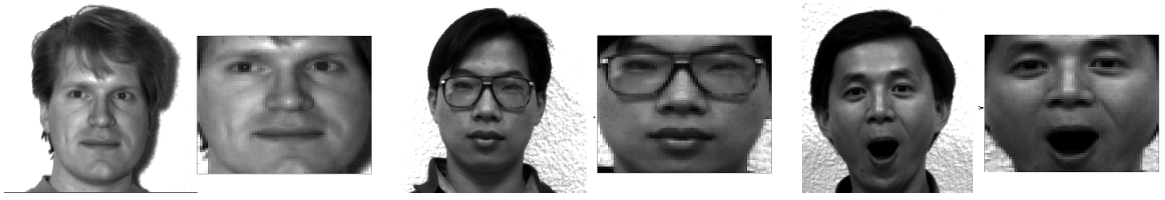
Vision Laboratory in the Computer Science and Engineering Department at University of California San Diego and the ORL database is constructed at AT&T laboratories Cambridge.

Yale Face Database

Yale face database contains 165 gray scale images of 15 individuals. There are 11 images per subject with size 243×320, with different facial expression or configuration: one normal image under ambient lighting, two with or without glasses, three images taken with different point light sources (centre, left, right), and five different facial expressions(happy, sad, sleepy, surprised and wink). Fig. 4.4 depicts total 22 samples of two individuals (11 samples of each individual) of Yale face data .

Olivetti Research Laboratory ORL database

The ORL contains 10 different images of each of the 40 distinct subjects of size 92×112. The images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an



(a) (b) (c)

(d) Mean Yale face

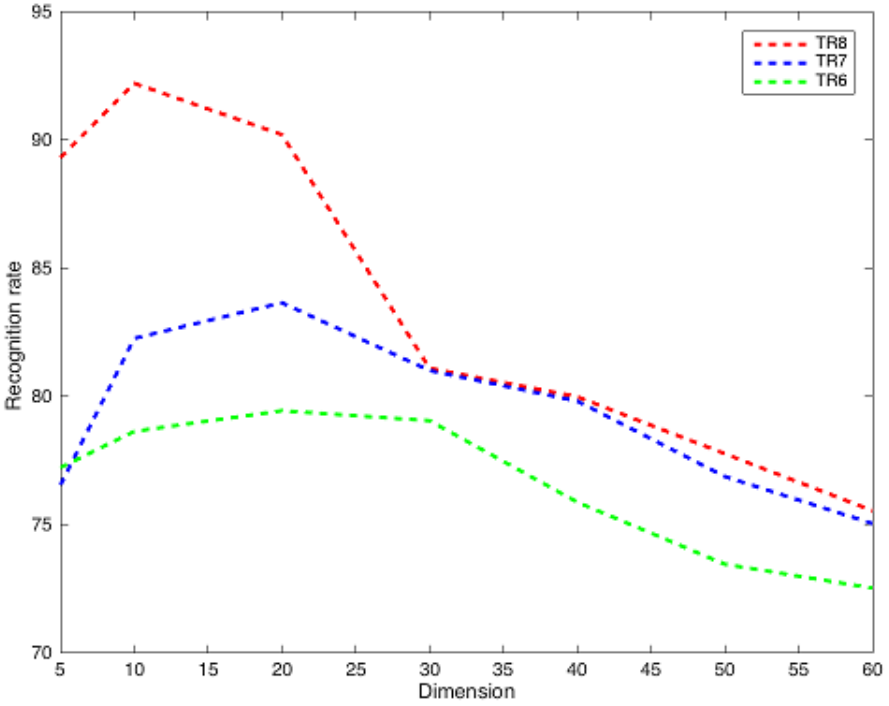
Figure 4.5: (a)-(c) Sample of original and cropped face images from Yale database. (d) Mean face of Yale database

upright, frontal position (with tolerance for some side movement). Some examples of ORL faces with different facial expressions and lighting conditions are given in Fig. 4.7.

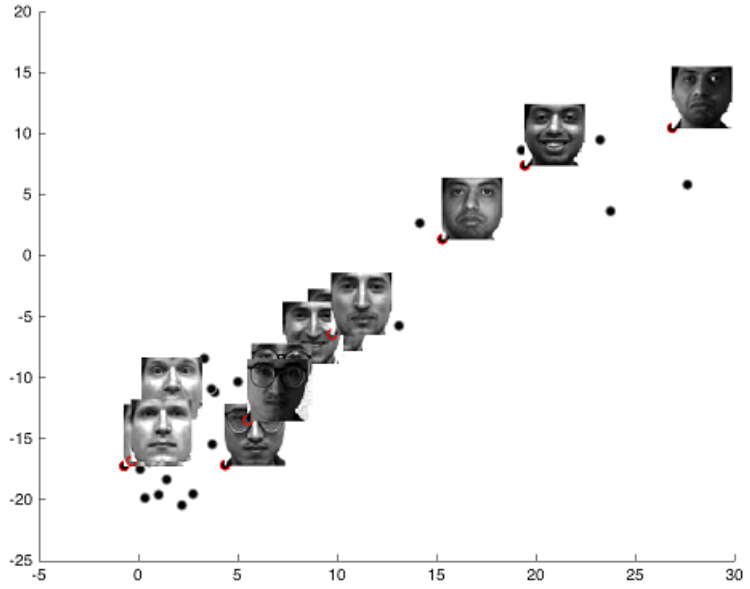
4.5.1 Pre-Processing Step:

In our experiments we used the processed Yale and ORL data obtained from http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html. processed by Cai et al [26]. All the face images are manually aligned and cropped. A sample of cropped images and the mean of all the cropped faces are shown in Fig. 4.5 and Fig. 4.7. The size of each cropped image is 64×64 pixels, with 256 gray levels per pixel. Thus each image is represented as a 4096-dimensional vector. Both the datasets are preprocessed by mean centering and normalized to unit variance.

For each of the data sets a random subset with p $(= 2, 3, 4, 5, 6, 7, 8)$ images per individual was taken with labels to form the training set and the rest of the images were considered to form the testing set. For example, for Yale data set, each of 15 individuals have 11 images. So the training set with $p = 8$ contains a total of $15 \times 8 = 120$ images and the test set contains remaining 3 images of each individual therefore a total of $15 \times 3 = 45$ images. So there is no overlap of images across training and testing samples. For each given p, we considered are 50 randomly splits into training and testing images. For the



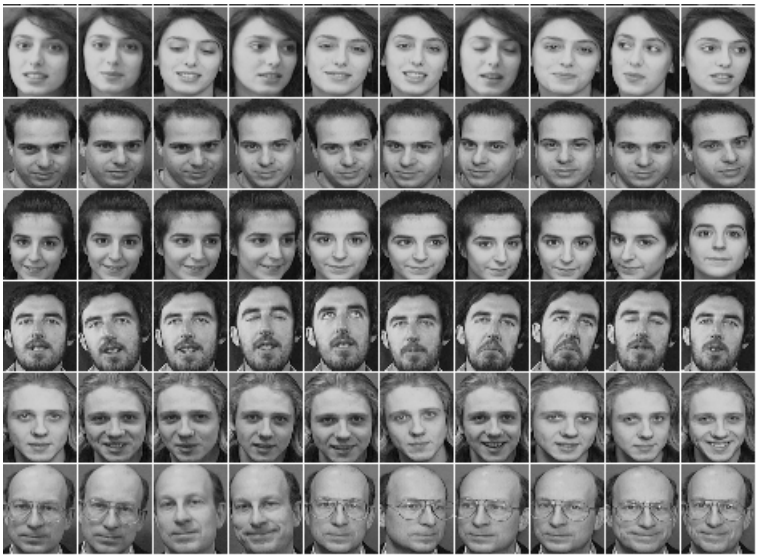
(a) Recognition rate along different dimension



(b) Test images of yale database superimposed on corresponding data point (red circle)

Figure 4.6: (a) Recognition rate of test sample of Yale face image along different dimension. The experiment is carried out by SLS-SDPP for different number of training samples TRp(p indicates the number of different images of each individual). Maximum recognition rate achieves at dimension $D = 9$. (b) 2D projection of Yale test faces and a sample of them superimposed on corresponding data points (red circle). Images of same class are seen to be projected closely.

(a)

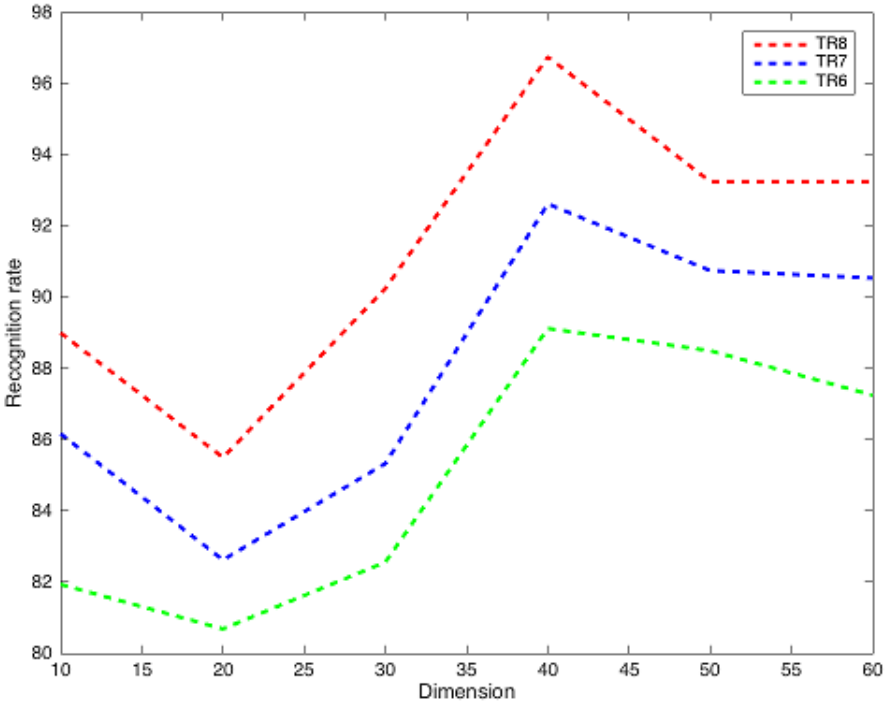


(b)

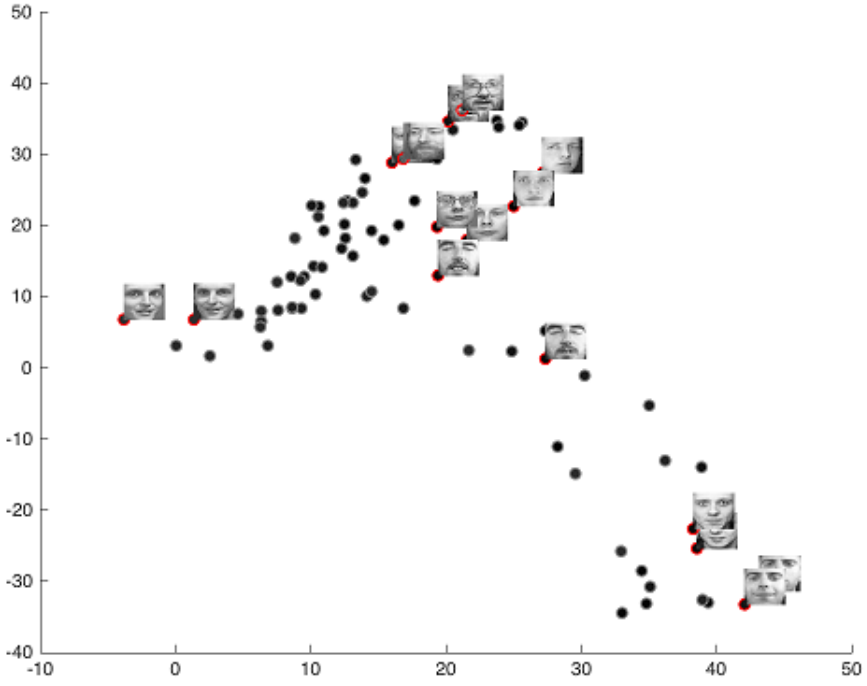
(c) Sample cropped ORL images

(d) Mean ORL face

Figure 4.7: (a)-(b) Illustration of facial expression variation of some individuals from ORL database, (c) Sample of cropped ORL faces. (d) Mean face of ORL database



(a) Recognition rate of prediction of test image along different dimension



(b) Test sample of ORL faces superimposed on corresponding data point (red circle)

Figure 4.8: Figure shows (a) Success rate of SLS-SDPP in predicting of ORL test images along different dimension.The experiment is carried out for different number of training samples. Highest recognition rate achieved at dimension D=41. (b) 2D projection of ORL test faces and a sample of them superimposed on corresponding data points (red circle). Images of same class are seen to be projected closely.

classification task, we used 1-Nearest Neighbor rule. The recognition rate is calculated as the ratio of number of successful recognition and the total number of test samples. In the same manner, the error rate is calculated as the ratio of number of failure in recognition to the total number of test samples.

4.5.2 Experimental results:

The projection quality usually varies with the number of dimensions. So for each of the dataset first we have determined the dimension of the projected space. For Fisheface method the dimension is chosen to be (No. of classes-1)[5]. For Eigenface method, first D eigenvectors are chosen, where (D=Number of training samples)[5]. For SLS-SDPP, we choose the dimension of projected space by observing the performance of the method at different dimension. Fig. 4.6(a) represents the recognition rate in identifying the test faces of Yale dataset with different training samples (TR_6, TR_7, TR_8) along different dimension which suggested us to project the Yale data set in 9 dimensional space. For ORL dataset we choose 41 relevant features to predict the class of test samples as SLS-SDPP obtains best result at $D = 41$ for ORL which can be verified from Fig. 4.8(a). The parameter (neighborhood) k is chosen to be between 2-6 using cross validation in the training samples of each of the data set.

A 2D projection of the Yale data base and ORL database obtained by SLS-SDPP are depicted in figure Fig. 4.6(b) and Fig. 4.8(b). Sample of the test images of both datasets are superimposed on the respective 2D plots which shows that images of same individuals are clustered and therefore proves the preservation of local structure of the data.

The average recognition accuracy of all the three algorithms along different number of training images of Yale and ORL databases are presented in Fig. 4.9 which is also reported on the Table 4.1 and 4.2 respectively. For each TR_p, where p is the number of training image, we took average of the results over 50 random splits and reported the mean as well as the standard deviation.

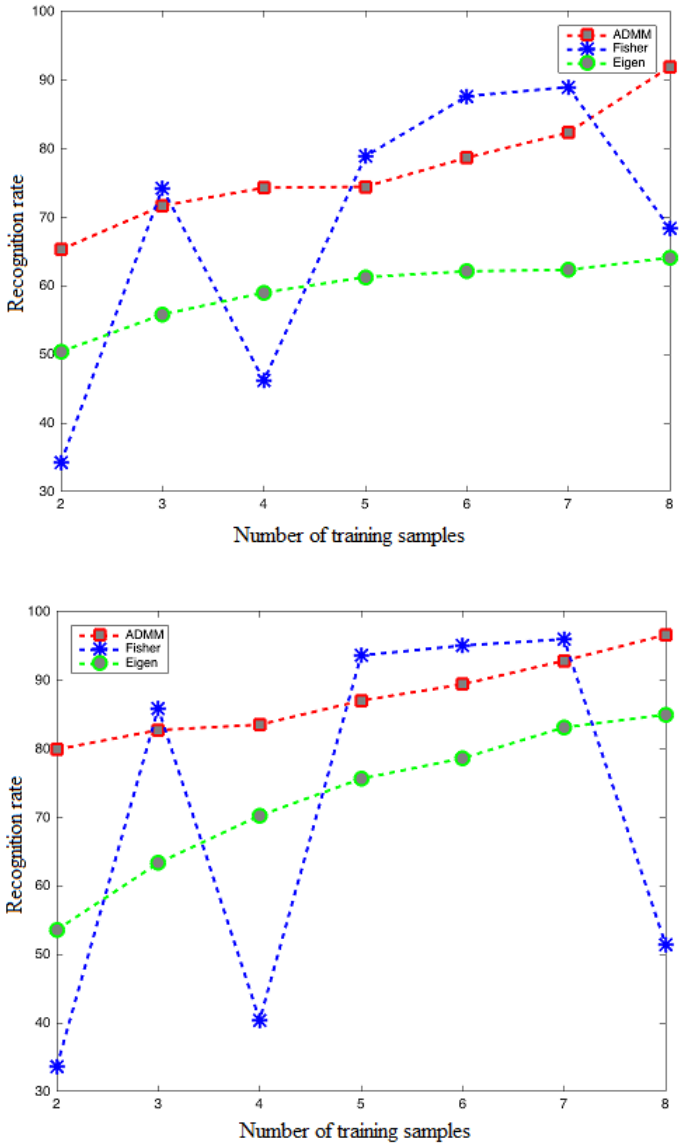


Figure 4.9: Average recognition rate of faces along different number of training samples (a) Yale dataset (b) ORL dataset. Though Fisherface gives better recognition rate than our method in some cases , its performance is much unstable whereas SLS-SDPP shows a consistent performance throughout the experiment.

Table 4.1 shows that for SLS-SDPP, the average recognition rate increases from 66% to 92% with the increase in number of training images from 2 to 8. Eigenface method also follows the same pattern but obtains much lower recognition rate in comparison to SLS-SDPP. Moreover the small standard deviation indicates the stability of our algorithm as well as Eigenface regarding the random splitting. However Fisherface method shows a different pattern. It gives best performance for training samples with 7 images of each individual. Fig. 4.9(a) illustrates that the recognition rate of Fisherface drops

drastically when the number of training images per class is 8 and 4. Also its performance is very much unstable in this cases which can be observed from the large values of std, 23% for $p = 4$ and 29% for $p = 8$) recognition rate in comparison to other two methods which can also be observed from Fig. 4.9(a) .

Similar to Yale face database, for ORL data set our method outperforms Eigenface method and Fisher face method (in some case). The improvement of recognition rate of the test images in our algorithm for ORL data set is from 79% to 96% with the increase of number of training samples. Though for both Yale and ORL dataset Fisherafce gives better recognition rate than our method for some values of p, its performance is unstable whereas our method shows a consistent performance throughout the experiment which is beneficial for practical applications with any training sample size.

Table 4.1: Average recognition rate of Yale test sample achieved by SLS-SDPP, Fisherface and Eigenface methods along different number of training points.

	TR_p	SLS-SDPP	Fisherface	Eigenface
Recognition Rate (mean±std)	TR_2	0.6529±0.0561	0.3421± 0.0382	0.5040±0.0238
	TR_3	0.7167±0.0460	0.7422±0.0362	0.5582±0.0426
	TR_4	0.7429±0.0492	0.4617±0.2254	0.5901±0.0307
	TR_5	0.7444±0.0311	0.7889±0.0213	0.6127±0.0346
	TR_6	0.7867±0.0401	0.8763±0.0353	0.6213±0.0410
	TR_7	0.8233±0.0324	0.8897±0.0408	0.6233±0.0426
	TR_8	0.9189±0.0353	0.6841±0.2856	0.6409±0.0659

Table 4.2: Average recognition rate of ORL test sample achieved by SLS-SDPP, Fisherface and Eigenface methods along different number of training points.

	TR_p	SLS-SDPP	Fisherface	Eigenface
Recognition Rate (mean±std)	TR_2	0.7989±0.0261	0.3362±0.0236	53.568±0.0257
	TR_3	0.8271±0.0340	0.8588±0.0342	0.6331±0.0223
	TR_4	0.8350±0.0281	0.4041±0.1871	0.7024±0.0267
	TR_5	0.8700±0.0305	0.9359±0.0121	0.7561±0.0216
	TR_6	0.8938±0.0262	0.9504±0.0206	0.7860±0.0413
	TR_7	0.9283±0.0309	0.9597±0.0192	0.8310±0.0374
	TR_8	0.9659±0.0353	0.5145±0.2031	0.8495±0.0319

4.6 Recognition from Blurred image:

In practice an image suffers from blur effect due to various reasons. For example if camera or the subject moves while the shutter is open or the camera is out-of-focus or the analyzed image is a small fragment of a large image etc. In such conditions,

Figure 4.10: Example of images artificially blurred with standard deviation (σ=1(origin),2,3,4,5 respectively) of Gaussian filter. (a) Yale face (b) ORL face.

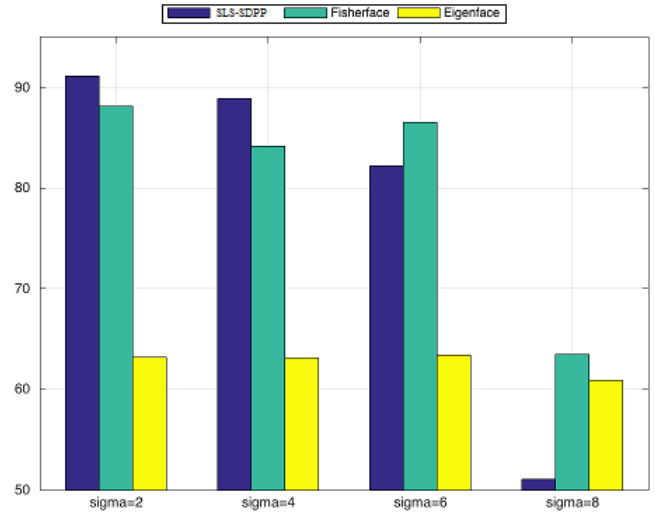
performance of most face identification algorithms drop drastically. Various methods have been proposed to deal with the recognition of blurry images [30, 94, 1]. In this section we have conducted numerical experiments on various level of artificially blurred images using the three methods SLS-SDPP, Eigenface and Fisheface to demonstrate their behavior in recognizing blur faces.

4.6.1 Pre-processing step:

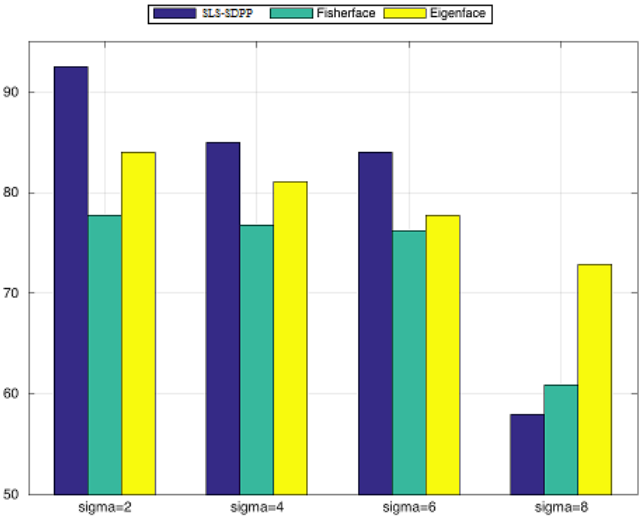
To test the performance of the algorithms in uncontrolled scenery, we have artificially generated set of blurred images from Yale and ORL datasets. For the degradation step, Gaussian filter available in MATLAB library is applied on the original images. The set of blurred images shown in Fig. 4.10 corresponds to the standard deviations $\sigma = 2, 4, 6$ and 8 respectively with $masksize = 1.5 * \sigma$. The training set is degraded to the same blur level of the test set to do the matching.

4.6.2 Experimental results:

The recognition rate of test samples for Yale and ORL data set along different level of blur images are depicted in Fig. 4.11. The bar diagrams illustrate that performance of



(a) Recognition rate of Yale blur faces.



(b) Recognition rate of ORL blur faces.

Figure 4.11: Bar diagrams represent performance of three methods SLS-SDPP, Fisherface and Eigenface in recognizing face images of Yale and ORL database along different blur level. For both data sets, Fisherface and Eigenface methods obtain much lower recognition rate in comparison to SLS-SDPP with the variation of standard deviation from 2 to 6 and therefore SLS-SDPP outperforms two other methods.

SLS-SDPP is consistent along various blur level. For Yale dataset the recognition rate varies approximately from 92% to 88% and for ORL data it varies from 93% to 84% as the standard deviation varies from 2 to 6. Note that for each of the datasets, the prediction rate drops suddenly for SLS-SDPP and Fisherface when standard deviation $sigma = 8$. For Yale face dataset, though the recognition rate of Eigenface method remained much lower than other two methods through out the experiment, it shows almost the same performance for any level of blur images which is beneficial for face recognition problems with much blur effect. Same behaviour of Eigenface method can also be observed for ORL database. However, for both of the data set the SLS-SDPP outperforms other two methods in most of the cases.

4.7 Summary

Face recognition problem, though solved quite satisfactorily by some existing methods in constrained scenarios, the general task of face recognition still poses a number of challenges. In the past few years, a number of face recognition techniques have been proposed. The new techniques includes recognizing faces across changes in illumination and pose, recognition from three dimensional scan, recognition from still images, recognition from video clips, handling poor resolution images, expression recognition from face etc. Most of the techniques use high dimensional data which leads to high computational and storage demands. Dimensionality reduction has thus become a necessity for preprocessing data for representation and classification.

In this chapter we have addressed application of our proposed distance preserving dimension reduction method SLS-SDPP on gallery images and blurred images of various level. Numerical experiments on both gallery and probe images demonstrate that the performance of our algorithm is promising in comparison to two leading approach Eigenface and Fisherface. Eigenface method obtains much lower recognition rate in comparison to SLS-SDPP. Though Fisherface gives better recognition rate than our method in some cases , its performance is much unstable whereas our method shows a consistent performance throughout the experiment which is beneficial for practical applications with any training sample size.

For testing the blur images we assumed that we have the blur degree of the testing image which we used to degrade the training images. In real problems with test images of unknown blur level, several methods [30, 74, 75] exist to infer the blur degree of the images which can be used to degrade the training set or deblur the testing set. A detailed descriptive work on this area is beyond the scope of this thesis and is planned to be considered in future. Also further research will focus on improving the transformation learning matrix to apply on images with other uncontrolled scenario mentioned above as well as other image recognition problems such as finger print, digit, signature etc.

Chapter 5

Conclusion and Future Work

Conclusion

In this thesis we have worked on a class of matrix optimization problems. A matrix optimization problems (MOP) involves optimizing the sum of a linear function and a proper closed simple convex function subject to affine constraints in the matrix space. MOP has many important applications such as data mining, network localization, etc arising from a wide range of fields such as engineering, finance and so on. This thesis is focused on the application of MOPs in data mining specially on data visualization, regression and classification. Data mining is the computational process of discovering useful and interesting patterns and knowledge from large amount of data. Due to the huge size and being collected from multiple, heterogeneous sources, real world data are mostly noisy and with missing and inconsistent data. So pre-processing of data significantly improves the quality of the data and, consequently, of the mining results. Among several data pre-processing techniques such as Data cleaning (smoothing noise, filling in missing values), Data reduction by eliminating redundant features, Data transformations etc, dimension reduction technique involves to choose a suitable mathematical technique which can extract most relevant features from data efficiently. Among a great number of methods, used to project high dimensional data into low dimensional space, the classical Multi-Dimensional Scaling (cMDS) is very important and efficient. Non-linear variants of cMDS have been developed to improve its performance. At the first

part of this research we have addressed a key problem in selecting the effective centers for a multidimensional scaling method, which involves radial basis functions. We took a novel approach that casts the problem as a multi-task learning problem. This approach has led to introduce the $(2,1)$-norm as a regularization term to the stress function used by Webb [107]. We then developed two reformulations, namely the diagonal and the spectral, that aim to ease the difficulties in solving the $(2,1)$-norm minimization problem. An iterative block majorization algorithm is developed to solve our models. The two reformulation models were compared to the original model in [107] on three well-known data sets. Discriminant analysis of the methods are also discussed. Numerical experiments on three benchmarking data set illustrate significant improvement of our models over the original one in terms of projection quality and CPU time. We would like to emphasize that the spectral model is more robust than the diagonal model, but with higher computational complexity. Both the models work very well for small data set but for large data set they are a little time consuming.

So at the second part of the research our goal was to develop a method that can project large data set efficiently. Therefore we have introduced SLS-SDPP which is a modification of recently proposed supervised distance preserving projection (SDPP) [117]. The SDPP is a supervised learning method whose basic formulation aims to preserve distances locally between data point in the projected space (reduced feature space) and the output space. The SDPP learns a linear mapping from the input space to the reduced feature space that leads to an efficient regression design. But for classification problems the preservation of local structure approach forces data of different classes to project very close to one another in the projected space which ends up with low classification rate.

To avoid the crowdedness of SDPP approach we have proposed a modification of SDPP which deals both regression and classification problem and significantly improves the performance of SDPP. In our research, we have incorporated the total variance of the projected co-variates to the SDPP problem which prevents data of different classes to stay close therefore preserves the global structure. Thus the purpose of our model is to keep the distance relation with neighbors (local structure) and at the same time to

preserve the global structure by maximizing the total variance. This approach not only facilitates efficient regression like SDPP but also successfully classifies data into different classes.

We have reformulated modified SDPP as a semidefinite least square (SLS-SDPP) model. A two block alternating direction method of multipliers (ADMM) have been developed to solve the SLS-SDPP problem. Several synthetic and real world data sets are considered to demonstrate the performance of our model in reducing the dimension of data by comparing the results with SDPP and some other state-of-art approaches Supervised Principal Component Analysis (SPCA), Partial Least Square (PLS), Kernel Dimension Reduction (KDR) and Fishers Discriminant Analysis (FDA). Experimental evaluation shows that, in most of the cases our method successfully projects the data into lower dimensional space that preserves the most effective features. Also in regression task, the measurement of root mean square error (RMSE) and mean absolute error (MAE) of some real world dataset illustrate an equivalent or superior performance of SLS-SDPP compared to all other methods. In classification problems SLS-SDPP improves SDPP significantly and classification error rates is much lower than all other methods in most of the cases.

Finally we have applied our proposed dimension reduction method SLS-SDPP on some well known face recognition datasets. Numerical experiments carried out to demonstrate the performance of our model compared to two leading face recognition techniques Eigenface and Fisherface which show that our model is $92 - 96\%$ successful to handle out of sample images. The method showed consistently better performance compared to other two methods. Also the performance of SLS-SDPP in recognizing various level of blur images is satisfactory which signify the applicability of our approach to a wide range of image recognition problems.

Future Work

A number of tasks are planned to be completed in near future.

- **Fine-tuning parameters:** The weight matrix α_{ij} in the stress term (2.2) can be specified as suggested by Webb in [108]. There are many possibilities which include:
  - $\alpha_{ij} = 1.0$ for all $i, j = 1, 2, \ldots, N$. In our case of minimization (2.7), we set all the weghts equal to 1.0.
  - To preserve local structure α_{ij} may take the form

$$\alpha_{ij} = \begin{cases} 1.0 & \text{if } d_{ij} < \ a\ threshold\ t_i \\ 0.0 & \text{otherwise.} \end{cases}$$

    In this case points with interpoint distances greater than t_i do not contribute to the stress. This will preserve the local structure of the data.
  - Another smoothed form of α_{ij} that may preserve the local structure is $\alpha_{ij} = \exp(-d_{ij}^2/t_i^2)$.
  - To put emphasis on greater distances α_{ij} can be chosen as $\alpha_{ij} = 1.0 - \exp(-d_{ij}^2/t_i^2)$, where less emphasis is put on smaller distances.

  The threshold t_i can be chosen using k-nearest neighbor, ie. for each point x_i, t_i can be set equal to the distance of its k nearest neighbor. It would be interesting to work with any of the above weights to observe the change of the projection of data and extract some more information from the data set.

- Our formulation of SLS-SDPP can be extended to its kernalized version KSLS-SDPP to handle nonlinearly distributed data which has already been done by Zhu et al. [117] and others. The idea is, at first to map the dataset $x_1, x_2, \ldots, x_N$ to another higher (even infinite) dimensional feature space $\mathcal{F}$ via $\phi : \mathcal{X} \to \mathcal{F}$ and then perform the linear projection $z = \mathbf{W}^T\phi(x)$ in this new feature space. Define the inner product $\langle \phi(x_i), \phi(x_j) \rangle = k(x_i, x_j)$. Here $k(x_i, x_j)$ is the element of the

kernel matrix. Let $\mathbf{\Phi} = [\phi(x_1), \phi(x_2), \ldots, \phi(x_N)]$. The transformation matrix W may have the form $\mathbf{W} = \mathbf{\Phi\Omega}$, where the unknown parameter is $\mathbf{\Omega}$.

The points in $\mathcal{F}$ are $\{\phi(x_1), \phi(x_2), \ldots, \phi(x_N)\}$. Therefore the squared distance in the reduced feature space has the form

$$\begin{aligned} d_{ij}^2(\mathbf{\Omega}) &= \|\boldsymbol{W}^T(\phi(x_i) - \phi(x_j))\|^2 \\ &= \|\mathbf{\Omega}^T\Phi^T(\phi(x_i) - \phi(x_j))\|^2 \\ &= \big((\phi(x_i) - \phi(x_j))^T\mathbf{\Phi\Omega\Omega}^T\mathbf{\Phi}^T(\phi(x_i) - \phi(x_j))\big) \\ &= (\mathbf{\Phi}^T\phi(x_i) - \mathbf{\Phi}^T\phi(x_j))^T\mathbf{\Omega\Omega}^T(\mathbf{\Phi}^T\phi(x_i) - \mathbf{\Phi}^T\phi(x_j)) \\ &= (\boldsymbol{K}_i - \boldsymbol{K}_j)^T\mathbf{\Xi}(\boldsymbol{K}_i - \boldsymbol{K}_j), \end{aligned}$$

where $K = (k(x_i, x_j))$ and K_i denotes the i^{th} column of $\boldsymbol{K}$, $\mathbf{\Xi} = \mathbf{\Omega\Omega}^T$ is the PSD matrix. Now the squared distances of the responses in the feature space can be expressed by $\delta_{ij}^2 = \boldsymbol{K}_{ii}^y + \boldsymbol{K}_{jj}^y - 2\boldsymbol{K}_{ij}^y$, where $\boldsymbol{K}^y$ is the kernel matrix for the response space. Therefore we only need to know the kernel $k(.,.)$ (No need to know the function ϕ). Optimization models can be obtained by using $d_{ij}^2(\Omega)$ and δ_{ij}^2. The kernel can be of any form, for example it can be any of the radial basis functions given in (2.1). In application areas like Image processing such as face recognition, we are hoping that KSLS-SDPP will reduce the computational complexity for learning the transformation matrix. So in near future our plan is to work on this kernalized version of SLS-SDPP.

- Much research has been devoted to image processing and recognition in recent years. Biswas et al. [8, 9] recently worked on matching low resolution images with high resolution gallery images. They have applied iterative majorization algorithm with the transformation function $\phi(x) = x$, which showed outstanding performance. This indicates the applicability of our first two models RMDS-S and RMDS-D based on iterative mejorization algorithm with Gaussian kernel as the transformation function.

We have applied our algorithm SLS-SDPP to face recognition problem which showed very good performance in recognizing gallery images. Also the method performed well in recognizing various level of blur images. Further research will be focused on improving the transformation matrix to apply on images with uncontrolled scenario as well as other image recognition problems such as expression recognition from faces, finger print recognition, digit recognition, signature recognition, etc.

- Optimization of correlation matrix is another area of research that is closely related to our work. Previous works on this area includes [79, 80, 39, 53]. For a set of data points $x_1, x_2, ..., x_N$, $C_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ defines a correlation matrix. Using principle component analysis (PCA) on this correlation matrix, a low rank matrix can be obtained that can be used to project data in a lower dimensional space. Sometimes instead of co-ordinates of the data points, we have the information of interpoint distances d_{ij} which may have noise. Also the distances may not be Euclidean distances or there may be some missing distances. In any of these cases the matrix obtained by $C_{ij} = \exp(-d_{ij}^2)$ is not a correlation matrix. Therefore It will be interesting to explore some methods to determine the nearest correlation matrix.

Chapter 6

Appendix

The convergence of ADMM has been explored in several references, such as [12, 37, 27]. Many of these give more sophisticated results. Among them the structure of theorems mentioned by S. Boyd et al. in [12] is familiar to our problem. So in this section we include a proof similar to that given in [12].

Proof of convergence of ADMM:

Based on assumptions 3.1 and 3.3 we will prove the residual convergence, objective convergence and the dual residual convergence. That is

We will show that if θ_1 and θ_2 are closed, proper and convex and the Lagrangian L_0 has a saddle point, then we have primal residual convergence $r^k \to 0$, objective convergence $pk \to \hat{p}$ and dual residual converges $s^k \to 0$ where

- $L_0(z_1, z_2; x) := \theta_1(z_1) + \theta_2(z_2) + \langle x, \beta_1^\star(z_1) + \beta_2^\star(z_2) - b\rangle$,
- $r^k = \beta_1^\star(z_1^k) + \beta_2^\star(z_2^k) - b$,
- $p^k = \theta_1 z_1^k + \theta_2 z_2^k$ and
- $s^k = \sigma\beta_1\beta_2^\star(z_2^k - z_2^{k-1})$.

Let $(\hat{z_1}, \hat{z_2}, \hat{x})$ be a saddle point for L_0 and define

$$V^k = \frac{1}{\sigma}\|x^k - \hat{x}\|_2^2 + \sigma\|\beta_2^\star(z_2^k - \hat{z})\|_2^2$$

V^k is unknown while the algorithm runs, since it depends on the unknown values $\hat{z}_2$ and $\hat{x}$. We will see that V^k is a Lyapunov function for the algorithm, i.e. a nonnegative quantity that decreases in each iteration. The proof relies on three key inequalities,

- The first inequality is

$$V^{k+1} \leq V_k - \sigma\|r^{k+1}\|_2^2 - \|\beta_2^\star(z_2^{k+1} - z_2^k)\|_2^2 \tag{6.1}$$

- The second key inequality is

$$p^{k+1} - \hat{p} \leq -(x^{k+1})^T r^{k+1} - \sigma(\beta_2^\star(z_2^{k+1} - z_2^k))^T(-r^{k+1} + \beta_2^\star(z_2^{k+1} - \hat{z})), \tag{6.2}$$

- The third inequality is

$$\hat{p} - p^{k+1} \leq \hat{x}^T r^{k+1}. \tag{6.3}$$

The first inequality implies the convergence of primal and dual residula r^k and s^k.
For, the first inequality states that V^k decreases by an amount that with the change in norm of the residual and in z_2 in each iteration. Because $V^k \leq V^0$, it follows that x^k and $\beta_2^\star z_2^k$ are bounded. Iterating the inequality above gives that

$$\sigma \sum_{k=0}^{\infty} \|r^{k+1}\|_2^2 + \|\beta_2^\star(z_2^{k+1} - z_2^k)\|_2^2 \leq V^0$$

which implies that $r^k \to 0$ and $\beta_2^\star(z_2^{k+1} - z_k) \to 0$ as $k \to 0$. Multiplying the second expression by $\sigma\beta_1$ shows that the dual residual $s^k = \sigma\beta_1\beta_2^\star(z_2^{k+1} - z_2^k)$ converges to zero. This shows that the stopping criterion 3.16 , which requires the primal and dual residuals to be small, will eventually hold.

The righthand side in 6.2 goes to zero as $k \to \infty$, since $\beta_2^\star(z_2^{k+1} - \hat{z}_2)$ is bounded and both r^{k+1} and $\beta_2^\star(z_2^{k+1} - z_2^k)$ tends to zero . Also The righthand side in 6.3 goes to zero as $k \to \infty$, since r^k goes to zero. Therefore we have the objective convergence $lim_{k\to\infty} p^k = \hat{p}$.

So we have to show that the three inequality holds. Before giving the proofs of the three key inequalities, the inequality 3.15 mentioned in stopping criterion is derived from the inequality 6.2. Observe that

$$-r^{k+1} + \beta_2^\star(z_2^{k+1} - z_2^k) = -\beta_1^\star(z_1^{k+1} - \hat{z_1});$$

substituting this into 6.2 yields 3.15, $p^{k+1} - \hat{p} \leq -(x^{k+1})^T r^{k+1} + (z_1^{k+1} - \hat{z_1})^T s^{k+1}$.

Now we will prove the three inequalities in reverse order.

Proof of inequality 6.3:

Since $(\hat{z_1}, \hat{z_2}, \hat{x})$ is a saddle point for L_0, we have

$$L_0(\hat{z_1}, \hat{z_2}, \hat{x}) \leq L_0(z_1^{k+1}, z_2^{k+1}, \hat{x})$$

Using $\beta_1^\star \hat{z_1} + \beta_2^\star \hat{z_2} = b$, the left hand side of the inequality becomes

$$\begin{aligned} L_0(\hat{z_1}, \hat{z_2}, \hat{x}) &= \theta_1(\hat{z_1}) + \theta_2(\hat{z_2}) + \langle \hat{x}, \beta_1^\star(\hat{z_1}) + \beta_2^\star(\hat{z_2}) - b \rangle \\ &= \theta_1(\hat{z_1}) + \theta_2(\hat{z_2}) \\ &= \hat{p} \end{aligned}$$

Now with $p^{k+1} = \theta_1 z_1^{k+1} + \theta_2 z_2^{k+1}$, and $r^{k+1} = \beta_1^\star(z_1^{k+1}) + \beta_2^\star(z_2^{k+1}) - b$ the right hand side of the inequality implies

$$\begin{aligned} L_0(z_1^{k+1}, z_2^{k+1}, \hat{x}) &= \theta_1(z_1^{k+1}) + \theta_2(z_2^{k+1}) + \langle \hat{x}, \beta_1^\star(z_1^{k+1}) + \beta_2^\star(z_2^{k+1}) - b \rangle \\ &= p^{k+1} + \langle \hat{x}, r^{k+1} \rangle \\ &= p^{k+1} + \hat{x}^T r^{k+1} \end{aligned}$$

Therefore we have $\hat{p} \leq p^{k+1} + \hat{x}^T r^{k+1}$, which proves 6.3.

Proof of inequality 6.2:

By definition we have z_1^{k+1} minimizes $L_\sigma(z_1, z_2^k, x^k)$. Since θ_1 is closed, proper, and

convex it is subdifferentiable, and so is L_σ. From section 3.5.2 The necessary and sufficient optimality condition is

$$0 \in \partial L_\sigma(z_1^{k+1}, z_2^k, x^k) = \partial\theta_1 z_1^{k+1} + \beta_1 x^k + \sigma\beta_1(\beta_1^\star z_1^{k+1} + \beta_2^\star z_2^k - b).$$

Which can be written using basic properties of subdifferential [85, sec. 23].) Using $x^k = x^{k+1} - \sigma r^{k+1}$ and $r^{k+1} = \beta_1^\star(z_1^{k+1}) + \beta_2^\star(z_2^{k+1}) - b$ we have

$$\begin{aligned} 0 &\in \partial\theta_1 z_1^{k+1} + \beta_1(x^{k+1} - \sigma r^{k+1}) + \sigma\beta_1(\beta_1^\star z_1^{k+1} + \beta_2^\star z_2^k - b) \\ &= \partial\theta_1 z_1^{k+1}\beta_1(x^{k+1} - \sigma(\beta_1^\star(z_1^{k+1}) + \beta_2^\star(z_2^{k+1}) - b - \beta_1^\star z_1^{k+1} - \beta_2^\star z_2^k + b)) \\ &= \partial\theta_1 z_1^{k+1}\beta_1(x^{k+1} - \sigma(\beta_2^\star(z_2^{k+1} - z_2^k))) \end{aligned}$$

This implies that z_1^{k+1} minimizes $\theta_1 z_1 + (x^{k+1} - \sigma\beta_2^\star(z_2^{k+1} - z_2^k))^T\beta_1^\star z_1$. Similarly it can be shown that z_2^{k+1} minimizes $\theta_2 z_2 + x^{(k+1)^T}\beta_2^\star z_2$. It follows that

$$\theta_1 z_1^{k+1} + (x^{k+1} - \sigma\beta_2^\star(z_2^{k+1} - z_2^k))^T\beta_1^\star z_1^{k+1} \leq \theta_1\hat{z_1} + (x^{k+1} - \sigma\beta_2^\star(z_2^{k+1} - z_2^k))^T\beta_1^\star\hat{z_1}$$

and that $\theta_2 z_2^{k+1} + x^{(k+1)T}\beta_2^\star z_2^{k+1} \leq \theta_2\hat{z_2} + x^{(k+1)T}\beta_2^\star\hat{z_2}$. Adding the two inequalities above and using $\beta_1^\star\hat{z_1} + \beta_2^\star\hat{z_2} = b$ and rearranging, we obtain 6.2.

Proof of inequality 6.1:

Adding 6.2 and 6.3, regrouping terms and multiplying through by 2 gives

$$2(x^{k+1}-\hat{x})^T r^{k+1} - 2\sigma(\beta_2^\star(z_2^{k+1}-z_2^k))^T r^{k+1} + 2\sigma(\beta_2^\star(z^{k+1}-z^k))^T(\beta_2^\star(z_2^{k+1}-\hat{z})) \leq 0. \quad (6.4)$$

Substituting $x^{k+1} = x^k + \sigma r^{k+1}$ in the first term of 6.4 gives

$$2(x^k + \sigma r^{k+1} - \hat{x})^T r^{k+1} = 2(x^k - \hat{x})^T r^{k+1} + \sigma\|r^{k+1}\|_2^2 + \sigma\|r^{k+1}\|_2^2$$

and substituting $r^{k+1} = \frac{1}{\sigma}(x^{k+1} - x^k)$ in the first two terms gives

$$\frac{2}{\sigma}(x^k - \hat{x})^T(x^{k+1} - x^k) + \frac{1}{\sigma}\|x^{k+1} - x^k\|_2^2 + \sigma\|r^{k+1}\|_2^2.$$

Since $x^{k+1} - x^k = (x^{k+1} - \hat{x}) - (x^k - \hat{x})$, the above equation can be written as

$$\frac{1}{\sigma}(\|x^{k+1} - \hat{x}\|_2^2 - \|x^k - \hat{x}\|_2^2) + \sigma\|r^{k+1}\|_2^2 \tag{6.5}$$

Therefore equation 6.4 takes the form

$$\frac{1}{\sigma}(\|x^{k+1}-\hat{x}\|_2^2-\|x^k-\hat{x}\|_2^2)+\sigma\|r^{k+1}\|_2^2-2\sigma(\beta_2^\star(z_2^{k+1}-z_2^k))^T r^{k+1}+2\sigma(\beta_2^\star(z_2^{k+1}-z_2^k))^T(\beta_2^\star(z_2^{k+1}-\hat{z_2})) \leq 0 \tag{6.6}$$

Now we consider last three terms of the left side of above inequality

$$\sigma\|r^{k+1}\|_2^2 - 2\sigma(\beta_2^\star(z_2^{k+1} - z_2^k))^T r^{k+1} + 2\sigma(\beta_2^\star(z_2^{k+1} - z_2^k))^T(\beta_2^\star(z_2^{k+1} - \hat{z_2}))$$

Substituting

$$z_2^{k+1} - \hat{z_2} = (z_2^{k+1} - z_2^k) + (z_2^k - \hat{z_2})$$

in the last term gives

$$\sigma\|r^{k+1}\|_2^2-2\sigma(\beta_2^\star(z_2^{k+1}-z_2^k))^T r^{k+1}+2\sigma\|\beta_2^\star(z_2^{k+1}-z_2^k)\|_2^2+2\sigma(\beta_2^\star(z_2^{k+1}-z_2^k))^T(\beta_2^\star(z_2^k-\hat{z}))$$

Now substituting

$$z_2^{k+1} - z_2^k = (z_2^{k+1} - \hat{z}) - (z_2^k - \hat{z})$$

in the last two terms, we get

$$\sigma\|r^{k+1} - \beta_2^\star(z_2^{k+1} - z_2^k)\|_2^2 + \sigma(\|\beta_2^\star(z_2^{k+1} - \hat{z_2})\|_2^2 - \|\beta_2^\star(z_2^k - \hat{z_2})\|_2^2).$$

With the previous steps, 6.6 can be written as

$$\frac{1}{\sigma}(\|x^{k+1}-\hat{x}\|_2^2-\|x^k-\hat{x}\|_2^2)+\sigma\|r^{k+1}-\beta_2^\star(z_2^{k+1}-z_2^k)\|_2^2+\sigma(\|\beta_2^\star(z_2^{k+1}-\hat{z_2})\|_2^2-\|\beta_2^\star(z_2^k-\hat{z_2})\|_2^2) \leq 0$$

Using $V^k = \frac{1}{\sigma}\|x^k - \hat{x}\|_2^2 + \sigma\|\beta_2^\star(z_2^k - \hat{z})\|_2^2$ and after some manipulation we have

$$V^k - V^{k+1} \geq \sigma\|r^{k+1} - \beta_2^\star(z_2^{k+1} - z_2^k)\|_2^2 \tag{6.7}$$

$$\begin{aligned} V^{k+1} &\leq V^k - \sigma\|r^{k+1} - \beta_2^\star(z_2^{k+1} - z_2^k)\|_2^2 \\ &= V^k - \sigma\|r^{k+1}\|_2^2 - \sigma\|\beta_2^\star(z_2^{k+1} - z_2^k)\|_2^2 + 2\sigma r^{(k+1)T}\beta_2^\star(z_2^{k+1} - z_2^k) \end{aligned}$$

To show 6.1, it now suffices to show that the last term $2\sigma r^{(k+1)T}\beta_2^\star(z_2^{k+1} - z_2^k) \leq 0$. To see this, recall that z_2^{k+1} minimizes

$\theta_2 z_2 + x^{(k+1)T}\beta_2^\star z_2$ and z_2^k minimizes $\theta_2 z_2 + x^{kT}\beta_2^\star z_2$, so we can add

$\theta_2 z_2^{k+1} + x^{(k+1)T}\beta_2^\star z_2^{k+1} \leq \theta_2 z_2^k + x^{(k+1)T}\beta_2^\star z_2^k$ and $\theta_2 z_2^k + x^{kT}\beta_2^\star z_2^k \leq \theta_2 z_2^{k+1} + x^{kT}\beta_2^\star z_2^{k+1}$

to get that $(x^{k+1} - x^k)^T\beta_2^\star(z_2^{k+1} - z^k) \leq 0$. Substituting $x^{k+1} - x^k = \sigma r^{k+1}$ gives the result, since $\sigma > 0$.

List of datasets and their sources:

We have already mentioned the original dimension and the sources of all the data sets used in this thesis in respective chapters. But here we have listed all of them for easy reference :

Table 6.1: List of datasets used in this thesis and their sources :

	Dataset	Dim	Class	no. of ins.	Source
Classification	Iris	4	3	150	UCI Repository
	Cancer	9	2	683	UCI Repository
	Seeds	7	3	210	UCI Repository
	TaiChi	5	2	2000	Zhu et al. [117]
	Seismic bump	19	2	2584	UCI Repository
	Cardiotocography	21	3	2126	UCI Repository
	Diabetic Retinopathy	19	2	1115	UCI Repository
	Mushroom	22	2	8124	UCI Repository
	Yale	64×64	15	165	UCSD computer vision
	ORL	64×64	40	400	AT&T laboratories Cambridge
Regression	Smooth Parity	5	-	1000	Zhu et al. [117]
	Swissroll	3	-	1000	Zhu et al. [117]
	Parkinson's Telemonitoring	16	-	5875	UCI Repository
	Concrete Compressive Strength	8	-	1030	UCI Repository
	Human face	64×64	-	698	http://isomap.stanford.edu

References

[1] Ahonen, T., Rahtu, E., Ojansivu, V. and Heikkil, J. (2008), 'Recognition of blurred faces using local phase quantization', *Pattern Recognition International Conference on Pattern Recognition[ICPR]*, pp. 1–4.

[2] Argyriou, A., Evgeniou, T. and Pontil, M. (2007), *Multi-task Feature Learning,* In B. Schoelkopf, J. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems, MIT Press.

[3] Argyriou, A., Evgeniou, T. and Pontil, M. (2008), 'Convex Multi-task Feature Learning', *Machine Learning, Special Issue on Inductive Transfer Learning*, vol. 73, pp. 243–272.

[4] Barshan, E., Ghodsi, A., Azimifar, Z. and Jahromi, M. Z. (2010), 'Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds', *Pattern Recognition*, vol. 44, pp. 1357–1371.

[5] Belhumeur, P., Hespanha,P., and Kriegman, D. (1997), 'Eigenfaces vs fisherfaces: recognition using class specific linear projection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vo. 19, no. 7, pp. 711–720.

[6] Belkin, M. and Niyogi, P. (2001), 'Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering', *Advances in Neural Information Processing Systems*, vol. 14, pp. 585–591.

[7] BéNasséNi, J. (1994), 'Partial additive constant', *Journal of Statistical Computation and Simulation,* vol. 49, pp. 179–193.

[8] Biswas, S., Boyer, K. W. and Flynn, P. J. (2012), 'Multidimensional Scaling for Matching Low-resolution Face Images', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 2019–30.

[9] Biswas, S., Aggarwal, G. and Flynn, P. J. (2011), 'Pose-robust recognition of low-resolution face images', *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 601–608, doi: 10.1109/CVPR.2011.5995443.

[10] Borg, I. and Groenen, P. J. F. (2005), 'Modern Multidimensional Scaling. Theory and Applications' 2^{nd} edition, *Springer Series in Statistics, Springer.*

[11] Borwein, J. and Lewis, A.S. (2006), *Convex Analysis and Non Linear Optimization : theory and examples,* Springer, vol. 3.

[12] Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2010), 'Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers', *Machine Learning*, vol. 3, no. 1, pp 1–122.

[13] Boyd, S. and Vandenberghe, L. (2004), *Convex Optimization,* Cambridge University Press.

[14] Brito, M.R., Chvez, E.L., Quiroz, A.J. and Yukich, J.E. (1997), 'Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection', *Statistics and Probability Letters*, vol. 35, no. 1, pp. 33–42.

[15] Buhmann, M.D. (2003), *Radial basis functions: theory and implementations* Cambridge University Press, Cambridge. ISBN 978-0-521-10133-2.

[16] Cailliez, F. (1983), 'The analytical solution of the additive constant problem', *Psychometrika*, vol. 48, pp. 305–308.

[17] Chen, G. and Teboulle, M. (1994), 'A proximal-based decomposition method for convex minimization problems', *Mathematical Programming*, vol. 64, pp. 81–101, 1994.

[18] Chen, H-T., Chang, H-W. and Liu, T-L. (2005), 'Local discriminant embedding and its variants', *In Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

[19] Chellapa, R., Wilson, C. and Sirohey, S. (1995), 'Human and machine recognition of faces: a survey', *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–741.

[20] Cheng Yeh, I. (1998), ' Modeling of strength of high performance concrete using artificial neural networks', *Cement and Concrete Research*, vol. 28, no. 12, pp. 1797–1808.

[21] Cooper, L.G. (1972), 'A new solution to the additive constant problem in metric and multidimensional scaling', *Psychometrika*, vol. 37, pp. 311–321.

[22] Coronaa, F., Zhu, Z., Souza Jr, M. Mulasd, A. H. d., Muruf, E., Sassuf, L., Barretob, G. and Baratti, R. (2015), 'Supervised Distance Preserving Projections: Applications in the quantitative analysis of diesel fuels and light cycle oils from NIR spectra', *Journal of Process Control*, vol. 30 , pp. 10–21.

[23] Cox, T.F. and Cox, M.A. (2002), *Multidimensional Scaling*, 2^{nd} edition Chapman and Hall/CRC.

[24] Cox, T. F. and Ferry,G. (1993), 'Discriminant analysis using nonmetric multidimensional scaling', *Pattern Recognition*, vol. 26, no. 1, pp. 145–153.

[25] Decencière et al. (2014) 'Feedback on a publicly distributed database: the Messidor database', *Image Analysis & Stereology*, vol. 33, no. 3, pp. 231–234, doi. 10.5566/ias.1155.

[26] Cai, D., He, X., Hu, Y., Han, J. and Huang, T. (2007), 'Learning a Spatially Smooth Subspace for Face Recognition', *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition Machine Learning.*

[27] Eckstein, J. and Bertsekas, D.P. (1992), 'On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators', *Mathematical Programming*, vol. 55, pp. 293–318.

[28] Eckstein, J and Fukushima, M. (1993), 'Some reformulations and applications of the alternating direction method of multipliers', *Large Scale Optimization: State of the Art*, pp. 119–138.

[29] Eckstein, J. and Yao, W. (2014), 'Understanding the convergence of Alternating Direction Method of Multipliers', *Theoritical and Computational Perspectives, RUTCOR Research Report.*

[30] Fichea, C. , Ladreta, P. and Vua, N-S. (2010), 'Blurred Face Recognition Algorithm Guided by a No-Reference Blur Metric', *Image Processing: Machine Vision Applications III, France*, doi : 10.1117/12.840245.

[31] Fisher, R.A. (1936), 'The use of multiple measurements in taxonomic problems', Annals of Eugenics, vol. 7, pp. 179–188.

[32] Fleming, M. and Cottrell, G. (1990), 'Categorization of faces using unsupervised feature extraction', *In Proceeding of EEE IJCNN International Joint Conference on Neural Networks*, pp. 65–70.

[33] Fortin, M. and Glowinski, R. (1983), *Augmented Lagrangian methods:Applications to the Numerical Solution of Boundary-Value Problems,* North-Holland Publishing Co., Amsterdam.

[34] Fortin, M. and Glowinski, R. (1983), 'On decomposition-coordination methods using an augmented Lagrangian', *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*, North-Holland: Amsterdam.

[35] Fukumizu, K., Bach, FR. and Jordan, M. (2009), 'Kernel dimension reduction in regression', *Annals of Statistics*, vol. 37, no. 4, pp. 1871–1905, doi. 10.1214/08-AOS637.

[36] Fukushima, M. (1992), 'Application of the alternating direction method of multipliers to separable convex programming problems', *Computational Optimization and Applications*, vol. 1, pp. 93–111.

[37] Gabay, D. (1983), *Applications of the method of multipliers to variational inequalities, in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems,* M. Fortin and R. Glowinski, eds.of studies in Mathematics, vol. 15, pp. 299–331.

[38] Gabay, D. and Mercier, B. (1976), ' A dual algorithm for the solution of nonlinear-variational problems via Finite element approximation', *Computers and Mathematics with Applications*, vol. 2, pp. 17–40..

[39] Gao, Y. and Sun, D. (2009), 'Calibrating least squares semidefinite programming with equality and inequality constraints', *SIAM Journal of Matrix Analysis and Application*, vol. 31, pp. 1432–1457.

[40] Glowinski, R. (1980), *Lectures on Numerical methods for Nonlinear variational problem,* Tata Institute of Fundamental Research Lectures on Mathematics and Physics, Tata Institute of Fundamental Research, Bombay, Notes by M. G. Vijayasundaram and M. Adimurthi.

[41] Glowinski, R. and Marrocco, A. (1975), 'Sur l'approximation, par *éléments* finis d'ordre un, et la *ré*solution, par *pé*nalisation-dualit*é*, d'une classe de problèmes de dirichlet non lin*é*ares', *Revue Francaise d'Automatique, Informatique et Recherche Opérationelle*, vol. 9, pp. 41–76.

[42] Glowinski, R. and Tallec, P.L. (1987), 'Augmented Lagrangian methods for the solution of variational problems', *Technical Report*, 2965, University of Wisconsin-Madison.

[43] Glunt, W., Hayden, T.L., Hong, S. and Wells, J. (1990), 'An alternating projection algorithm for computing the nearest Euclidean distance matrix', *SIAM Journal on Matrix Analysis and Application*, vol. 11, pp. 589–600.

[44] Glunt, W. Hayden, T.L. and Raydan, R. (1993), 'Molecular conformations from distance matrices', *Jornal of Computational Chemistry*, vol. 14, pp. 114–1203.

[45] Gower, J.C. (1966), 'Some distance properties of latent rootand vector methods in multivariate analysis', *Biometrika*, vol. 53, pp. 315–328.

[46] Han, J., Kamber, M. and Pei, J. (2011), *Data Mining: Concepts and Techniques,* Elsevier, ISBN: 978-0-12-381479-1 .

[47] He, X., Cai, D., Yan, S. and Zhang, H-J. (2005), ' Neighborhood preserving embedding', *In Proceeding of IEEE International Conference on Computer Vision.*

[48] He, B., Yang, H. and Wang, S. (2000), 'Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities', *Journal of Optimization theory and applications*, vol. 106, pp. 337–356.

[49] He, X. and Niyogi, P. (2003), 'Locality preserving projections', *Advances in Neural Information Processing Systems 16*.

[50] Heisele, B., Ho, P. and Poggio, T. (2001), 'Face Recognition with Support Vector Machines: Global versus Component-based Approach', *In the proceeding of Eighth IEEE International Conference on Computer Vision*, vol. 2, doi: 10.1109/ICCV.2001.937693.

[51] Hestenes, M. R. and Stiefel, E. (1952), 'Methods of Conjugate Gradients for Solving Linear Systems', *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409-436.

[52] Hestenes, M.R. (1969), 'Multiplier and Gradient Methods', *Journal of Optimization Theory and Application*, vol. 4, pp. 303–320.

[53] Higham, N. J. (2002), 'Computing the nearest correlation matrixa problem from finance', *IMA Journal of Numerical Analysis*, vol.22 pp. 329–343.

[54] Huang, W. and Yin, H. (2009), 'Linear and Nonlinear Dimensionality Reduction for Face Recognition' *16th IEEE International Conference on Image Processing (ICIP)*, pp. 3337–3340, doi. 10.1109/ICIP.2009.5413898.

[55] Huang, W. and Yin, H. (2012), 'On Nonlinear Dimensionality Reduction for Face Recognition', *Image and Vision Computing*, vol. 30, no. 45, pp. 355–366, doi: 10.1016/j.imavis.2012.03.004.

[56] Jiang, K., Sun, D. and Toh, K.-C. (2014), 'Solving nuclear norm regularized and semidefinite matrix least square problems with linear equality constraints', *Discrete Geometry and Optimization, Publisher: Springer*, pp. 133–162.

[57] Jonsson, K., Matas, J., Kittler, J. and Li, Y. (2000), 'Learning support vectors for face verification and recognition', *In Proceeding of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 208–213.

[58] Koontz, W. L. G. and Fukunaga, K. (1972), 'A nonlinear feature extraction algorithm using distance information', *IEEE Transactions on Computers*, vol.21, no. 1, pp. 56–63.

[59] Kruskal, J. (1964), 'Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis', *Psychometrika*, vol. 29, pp. 1–27.

[60] Kukharev, G and Forczmańnski, P. (2004), 'Data dimensionality reduction for face recognition', *Machine Graphics & Vision*, vol. 13, pp 99–121.

[61] Lanitis, A., Taylor, C. and Cootes, T. (1997), 'Automatic interpretation and coding of face images using flexible models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 743–756.

[62] Lee, J. and Verleysen, M. (2007), *Nonlinear dimensionality reduction,* Springer, New York.

[63] Leeuw, J. de. (1977), 'Applications of convex analysis to multidimensional scaling', *J. Barra, F. Brodeau, G. Romier, and B. van Cutsen, eds, 'Recent Developments in Statistics'*, North Holland Publishing Company, Amsterdem, The Netherlands, pp. 133–145.

[64] Leeuw, J. de, (1994), 'Block relaxation algorithms in statistics'. *In: Bock, H.H. et al. (eds) Information Systems and Data Analysis*, Springer, Berlin, pp. 308–325.

[65] Li, K. (1991), 'Sliced inverse regression for dimension reduction', *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 316–327.

[66] Li, X., Sun, D. and Toh, K.C. (2014), 'A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions', *Mathematical Programming Series*, vol. 155, no. 1, pp. 333–373, doi. 10.1007/s10107-014-0850-5.

[67] Li, J., Zhang, C.,Hu, J. and Deng, W. (2014), 'Blur-Robust Face Recognition via Transformation Learning', *Computer Vision - ACCV 2014 Workshops, Lecture Notes in Computer Science (series)*,vol. 9010 , pp. 15–29.

[68] Lowe, D. (1993), 'Novel topographic nonlinear feature extraction using radial basis functions for concentration coding in the artificial nose', *IEEE International Conference on Artificial Neural Networks*, pp. 95–99.

[69] Maaten, L.J.P., Postma, E. and Herik, H. V. D. (2009), 'Dimensionality reduction: a comparative review', Technical Report TiCC-TR 2009005, Tilburg University Technical, Tilburg.

[70] Mardia, K.V., Kent, J.T. and Bibby, J.M. (1995), *Multivariate Analysis* Academic Press.

[71] Messick, S.J. and Abelson, R.P. (1956), 'The additive constant problem in multidimensional scaling', *Psychometrika*, vol. 21 , pp. 1–15.

[72] Mika, S., Ratsch, G., Weston, J., Scholkopf, B. and Mullers, K. (1999), 'Fisher discriminant analysis with kernels', *Neural networks for signal processing IX, Proceedings of the IEEE signal processing society workshop, Piscataway*, pp. 41–48.

[73] Murase, H. and Nayar, S.K. (1995), 'Visual Learning and Recognition of 3-D Objects from Appearance', *International Journal of Computer Vision*, vol. 14, 5–24.

[74] Nishiyama, M., Takeshima, H., Shotton,J., Kozakaya, T. and Yamaguchi, O. (2009), 'Facial deblur inference to improve recognition of blurred faces', *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1115–1122, doi: 10.1109/CVPR.2009.5206750.

[75] Nishiyama, M., Hadid, A., Takeshima,H., Shotton, J., Kozakaya, T. and Yamaguchi, O. (2011), 'Facial Deblur Inference Using Subspace Analysis for Recognition of Blurred Faces', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 838–845, doi: 10.1109/TPAMI.2010.203.

[76] Pękalaska and R.P.W.Duin, E. (2005), 'The Dissimilarity Representation for Pattern Recognition: Foundations and Application', *Series in Machine Perception Artificial Intelligence 64*. World Scientific.

[77] Powell, M. (1969), 'A Method for Non-linear Constraints in Minimization Problem', *Optimization*, R. Fletcher, ed., Academic press, pp. 283–298.

[78] Powell, M. (1981), *Approximation Theory and Methods*, Cambridge University Press, Cambridge.

[79] Qi, H.-D. and Sun, D. (2006), 'A quadratically convergent Newton method for computing the nearest correlation matrix', *SIAM Journal of Matrix Analysis and Application*, vol. 28, no.2, pp. 360–385.

[80] Qi, H.-D. and Sun, D. (2011), 'An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem', *IMA Journal of Numerical Analysis*, vol. 31, pp. 491–51.

[81] Qi, H.-D. and Xiu, N. (2012), À convex quadratic semidefinite programming approach to the partial additive constant problem in multidimensional scaling', *Journal of Statistical Computation and Simulation*, vol. 82, pp. 1317–1336.

[82] Qi, H.-D. (2013), 'A semismooth Newton method for the nearest Euclidean distance matrix problem', *SIAM Journal of Matrix Analysis and Applications*, vol. 34, pp. 67–93.

[83] Qi, H.-D., Xiu, N.H. and Yuan, X.M. (2013), 'A Lagrangian dual approach to the single source localization problem', *IEEE Transactions on Signal Processing*, vol. 61 , pp. 3815–3826.

[84] Qi, H.-D. and Yuan, X.M. (2014), 'Computing the nearest Euclidean distance matrix with low embedding dimensions', *Mathematical Programming*, vol. 147, no. 1, pp 351389, doi 10.1007/s10107-013-0726-0.

[85] Rockafellar, R. T. (1970), *Convex Analysis* Princeton University Press.

[86] Rockafellar, R. T. (1976), 'Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming', *Mathematics of Operations Research*, vol. 1, pp. 97–116.

[87] Rockafellar, R. T. (1976), 'Monotone operators and the proximal point algorithm', *SIAM Journal on Control and Optimization*, vol.14, no. 5, pp. 877–898.

[88] Rosipal,R. and Krmer, N. (2006), 'Overview and recent advances in partial least squares', *In: Subspace, latent structure and feature selection Techniques, Lecture Notes in Computer Science*, pp. 34–51.

[89] Rosipal, R. and Trejo, L. (2002), 'Kernel partial least squares regression in reproducing kernel Hilbert space', *Journal of Machine Learning Research*, vol. 2, pp. 97–123.

[90] Roweis, S. and Saul, L. (2000), 'Nonlinear dimensionality reduction by locally linear embedding', *Science*, vol. 290, no. 5500, pp. 2323–2326.

[91] Schoenberg, I.J. (1935), 'Remarks to Maurice Fréchet's article "Sur la définition axiomatque d'une classe d'espaces vectoriels distanciés applicables vectoriellement sur l'espace de Hilbet', *Annals of Mathematics*, vol. 36, pp. 724–732.

[92] Schlkopf, B., Smola, A. and Mller, K. (1998), 'Nonlinear component analysis as a kernel eigenvalue problem', *Neural computation*, vol. 10, no. 5, pp. 1299–1319.

[93] Sirovitch, L. and Kirby, M. (1987), 'Low-dimensional procedure for the characterization of human faces', *Journal of the Optical Society of America A*, vol. 2, pp. 519–524.

[94] Stainvas, I. and Intrator, N. (2000), 'Blurred face recognition via a hybrid network architecture', *Proceedings of 15th International Conference on Pattern Recognition*, vol. 2, pp. 805–808.

[95] Sun, D., Toh, K.-C. and Yang, L. (2014), 'A Convergent 3-Block Semi-Proximal Alternating Direction Method of Multipliers for Conic Programming with 4-Type of Constraints', *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 882–915.

[96] Tenenbaum, J., Silva, V. and Langford, J. (2000), 'A global geometric framework for nonlinear dimensionality reduction', *Science*, vol.290, no. 5500, pp. 2319–2323.

[97] Theodoridis, S. and Koutroumbas, K. (2010), *An Introduction to Pattern Recognition, A MATLAB approach,* Elsevier Inc.

[98] Theodoridis, S. and Koutroumbas, K. (2009), *Pattern Recognition,* Elsevier Inc.

[99] Tseng, P. (1991), 'Applications of a splitting algorithm to decomposition in convex programming and variational inequalities', *SIAM Journal on Control and Optimization*, vol. 29, pp. 119–138.

[100] Torgerson, W.S. (1952), 'Multidimensional Scaling: Theory and Methods', *Psychometrica*, vol. 17, pp. 401–419 (The first major MDS breakthrough).

[101] Torgerson, W.S. (1958), *Theory and Methods for Scaling*, Wiley, New York.

[102] Tsanas, A., Little, M. A., McSharry, P. E. and Ramig, L. O. (2009), 'Accurate telemonitoring of Parkinson.s disease progression by non-invasive speech tests', *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 884–893, doi. 10.1109/TBME.2009.2036000.

[103] Turk, M. and Pentland, A. (1991), 'Face recognition using eigenfaces', *In Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591.

[104] Venna, J. and Kaski, S. (2007), 'Comparison of visualization methods for an atlas of gene expression data sets', *Information Visualization*, vol. 6, pp. 139–154.

[105] Vapnik V.N. and Chervonenkis, A.Ya. (1964), 'On a class of perceptrons', *Automation and Remote Control*, vol. 25, no. 1, pp. 103–109 (original article was in Russian language submitted in 1963).

[106] Wang, S. and Liao, L. (2001), 'Decomposition method with a variable parameter for a class of monotone variational inequality problems', *Journal od optimization theory and applications*, vol. 109, pp. 415–429.

[107] Webb, A.R. (1995), 'Multidimensional Scaling by iterative majorization using radial basis functions' *Pattern Recognition*, vol. 28, pp. 753–759.

[108] Webb, A.R. (1996), 'Nonlinear feature extraction with radial basis functions using a weighted multidimensional scaling stress measure' *Pattern Recognition*, IEEE Conference Publications, vol. 4, pp. 635–639.

[109] Webb, A.R. (1996), 'An approach to nonlinear principal component analysis using radially-symmetric kernel functions', *Statistics and Computing*, vol. 6 , pp. 159-168.

[110] Weinberger, K., Sha, F. and L. Saul, L. (2004), 'Learning a kernel matrix for nonlinear dimensionality reduction', *Proceedings of the 21st international conference on, machine learning*, Banff.

[111] Wold, H. (1975), 'Soft modeling by latent variables: the nonlinear iterative partial least squares approach', *Perspectives in probability and statistics, papers in honour of MS Bartlett*, pp. 520–540.

[112] Wold, H. (2006), Partial Least Squares, *Encyclopedia of Statistical Sciences*. vol. 9, DOI: 10.1002/0471667196.ess1914.pub2.

[113] Yeh, Y., Huang, S., Lee, Y. (2009), 'Nonlinear dimension reduction with kernel sliced inverse regression', *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1590–1603.

[114] Young, F. W. and Hamer, R. M. (1994), *Theory and Application of Multidimensional Scaling.* Eribaum Associates. Hillsdale, NJ.

[115] Young, G. and Householder, A.S. (1938), 'Discussion of a set of points in terms of their mutual distances', *Psychometrika*, vol. 3, pp. 19–22.

[116] Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A. (2003), 'Face Recognition: A Literature Survey', *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458.

[117] Zhu, Z, Similä, T. and Corona, F. (2013), 'Supervised Distance Preserving Projection', *Neural Processing Letters*, vol. 38, no. 3, pp. 445–463.

[118] Zhuang, X-S. and Dai, D-Q. (2005), 'Inverse Fisher discriminate criteria for small sample size problem and its application to face recognition', *Pattern Recognition*, vol. 38, pp 2192–2194.