# Envelope-Based Support Vector Machines Classification

Alya Alzahrani

CARDIFF
UNIVERSITY

PRIFYSGOL
CAERDYDD

School of Mathematics

Cardiff University

Submitted in partial fulfillment of
the requirements for the degree of

*Doctor of Philosophy*

2022

# Acknowledgements

First and foremost I would like to thank my supervisor Dr. Andreas Artemiou for his encouragements, valuable advice and guidance throughout my PhD study, without his support this work could not be completed. I also would like to thank my co-supervisor Dr. Bertrand Gauthier for his helpful comments and advice.

I extend my gratitude to my parents for their support and prayers that make my journey possible. Most important I am thankful to my amazing kids Rafi and Aseel for inspiring me and giving me the encouragement I needed.

Lastly I am thankful to Taif University for the financial support.

# Abstract

Envelope methodology is a promising dimension reduction approach. It was introduced in the regression framework. In this work, we extended envelope application and focused on the reduce-and-classify approach in supervised learning. The first contribution is that we extended this method to classification and developed a new projection-based approach based on a Support Vector Machine (SVM) classifier. Our proposed classifier ESVM (Envelope-based Support Vector Machines) is obtained by combining the envelope method and SVM to achieve a better and more efficient classification. Using the idea of the envelope to extract a lower-dimensional subspace projected the data on has advanced the classification performance. The empirical results show a low misclassification rate based on ESVM

Furthermore, we extended the ESVM classifier to sparse data. In that, the reducing subspace reduces the dimension and selects significant variables simultaneously. We employ an adaptive group lasso penalty to impose the sparsity in the reducing subspace. The classifier is evaluated based on simulation and real data.

# Talks

- **Projection-based classification (Talk)**. *The* 34*th* *Panhellenic Statistics Conference of the Greek Statistical Institute.*, Graak (online), May 2022.

- **Envelope-based Support Vector Machines (Talk)**.*The* 17*th* *conference of the International Federation of Classification Societies*, University of Porto, Portugal, July 2022.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

The collection of high-dimensional data across many disciplines has increased exponentially in the last two decades. The advances in technologies and computation facilities with less cost allow big data collection in many fields. Examples include microarray analysis, image analysis, document classification, astronomy, and atmospheric science (Johnstone and Titterington, 2009). If we denote the number of collected samples by $n$ and the number of features by $p$, one common characteristic among this type of data is the growth of the number of features compared to the number of samples, that is $p > n$. One possible reason for the increase in the number of features is the lack of knowledge about the informative features (Verleysen et al., 2003). Despite the field in which the data were collected or the study's objective, classification, clustering, or regression, the challenge remains the same. That is, many statistical results were designed for well-behaved data where the ratio $n/p \geq 5$ is satisfied (Johnstone and Titterington, 2009).

The objective in many of these datasets is to explain the relationship between the outcome (response(s)) and a set of explanatory variables. This relationship is represented by:

$$\boldsymbol{Y} = f(\boldsymbol{x}) + \mathcal{E}, \tag{1.1}$$

where $\boldsymbol{Y}$ is the univariate or multivariate response(s), $f(\boldsymbol{x})$ is the function which facilitates the nature of the relationship between the response(s) and the predictor variables (also known as the link function), and $\mathcal{E}$ is the normally distributed errors. The relationship represented by $f(\boldsymbol{x})$ could be linear or nonlinear; however, the nonlinear models are beyond the scope of this work; our focus is on linear models with univariate responses.

The existence of high dimensionality in a dataset arises undesirable consequences. One difficulty caused by large $p$ is that most data points are concentrated at the decision boundary. That is, having $p$ dimensional data points divided into $K$ groups, as $p \to \infty$ that makes finding the decision boundary between the groups challenging. This phenomenon makes the prediction of the training sample points that are near the edges difficult. The other issue is that the required data samples grow with the dimension. That is, the sampling density is proportional to $(n^{1/p})$, for example if we have $n = 50$ for $p = 1$, hence we need $50^{10}$ when we increase the dimension to $p = 10$ to gain the same information (Hastie et al., 2009).

The other issue that one may encounter is the multicollinearity among predictor variables. Multicollinearity indicates a dependency between the features, which affects the estimation of the parameters. Indeed, the problem of multicollinearity causes the variance of the coefficients estimators for the dependent variables to be large. Consequently, other effects might include; the estimates of the coefficients may be unacceptably large, and the sign of the estimates might differ from what is theoretically expected. That leads to unreliable statistical tests conclusion (Mansfield and Helms, 1982). It is worth noting that multicollinearity may exist even when $n > p$. A proper multicollinearity detection test is an important initial step in the latter case.

Hence, one aim is to eliminate the effect of the non-informative predictors and select the informative variables among the large group of predictors. Regression-based techniques are considered the most commonly used statistical tools to explain the relationship between response and predictor variables. However, some techniques have limitations in that they do not handle multicollinearity, ordinary least squares is an example. In statistical literature, the dimensionally and multicollinearity problems are tackled via dimension reduction in two ways: feature extraction and feature selection.

Formally, dimension reduction is a procedure applied prior to model formulation to extract a linear combinations of the original variables (known as feature extraction), or to select a small subset of the original variables (denoted by feature selection) (Li, 2007). Feature extraction can be defined as a function $\mathcal{D}(\boldsymbol{X})$ that maps $\boldsymbol{X}$ into a $d$-dimensional subspace, where $p > d$. Precisely, suppose $\mathcal{D}(\boldsymbol{X}) = \boldsymbol{\eta}^T\boldsymbol{X}$, $\boldsymbol{\eta} \in \mathbb{R}^{p\times d}$ (Weng and Young, 2017). To date, a variety of feature extraction procedures have been proposed in statistical literature. Among others, Principal Components Analysis (PCA) (Jolliffe, 1986), Supervised Principal Components (Bair et al., 2006), Sliced Average Variance Estimation (SAVE) (Cook and Weisberg, 1991), Principal Support Vector Machines (Li et al., 2011), Lasso Principal Support Vector Machines (Pircalabelu and Artemiou, 2022), Principal Distance-Weighted Discrimination (Randall et al., 2021), Graph Sliced Inverse Regression (Pircalabelu and Artemiou, 2021).

The other method to handle high dimensionality is feature selection. Feature selection involves selecting the best subset of the predictor variables then fit the model based on the selected variables. This approach can be performed via classical methods such as best subset selection and forward/backward stepwise selection or via regularization (James et al., 2013). The regularization method relies on the sparsity assumption, that is, among a large number of predictor variables, only a few have non-zero coefficients and, hence informative. The regularization technique works by imposing a pre-determined penalty to the minimization/maximization criterion such that it shrinks the coefficients of the non-informative predictor variables to zero. Thus, the final model includes only the predictor variables whose coefficients estimates are non-zero. Least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996), elastic net penalty (Zou and Hastie, 2005), ridge regression (Hoerl and Kennard, 1970), and the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001) are examples of commonly used penalties in statistical literature.

On the other hand, classification is the process of allocating an individual to the original group. The high dimensionality, however, has an impact on classification accuracy. The presence of the noise predictor variables increases the misclassification rate. Further, some classifiers may break down when the number of features is large; linear discriminate analysis is an example. In the cases, if the fitting is possible, the classification may as bad as random guessing due to accumulation noise (Fan and Fan, 2008).

In this thesis, we focus on classification based on reduced data. Motivated by the efficiency gain that the envelope method (Cook et al., 2010) achieved in regression we apply this method to extract a lower-dimensional subspace that captures all information related to the classification. Support vector machines is considered a robust classifier. Hence, we combine the two techniques with an aim to improve classification accuracy.

That is, we use the envelope technique as an initial step to reduce the dimension of the data then we classify the data based on classic SVM.

## 1.2 Notation

In this section we define the notations that are used throughout this thesis. The scalar is denoted by $(a, b, c)$. For vectors we use lower case bold symbols $(\boldsymbol{x}, \boldsymbol{y})$. The vectors are a column vectors with number of elements denoted by $\boldsymbol{x} \in \mathbb{R}^d$. However, to distinguish column vector from row vector we emphasize that via the subscript in that $\boldsymbol{x}_{\cdot j}$ is a column vector while $\boldsymbol{x}_{i\cdot}$ is row vector. Matrices are denoted by upper case bold letters $(\boldsymbol{A}, \boldsymbol{B})$. The dimension of a matrix is denoted by $a \times b$ and is written as $(\boldsymbol{X} \in \mathbb{R}^{a \times b})$ or as a subscript $\boldsymbol{X}_{a \times b}$. The identity matrix is denoted by $\boldsymbol{I}_d$ where the subscript $d$ referred to the dimension. The notation $\mathbf{1}_d$ referred to a vector of ones of length $d$.

The superscript $\boldsymbol{X}^T$ indicates the transpose of a matrix $\boldsymbol{X}$ and $\boldsymbol{X}^{-1}$ is the inverse of a matrix.

The $\ell_1$ norm of a vector $\boldsymbol{x} \in \mathbb{R}^p$ is given by:

$$|\boldsymbol{x}| = \sum_{i=1}^{p} |x_i|,$$

where $|x_i|$ is the absolute value of $x_i$. The $\ell_2$ norm of a vector $\boldsymbol{x} \in \mathbb{R}^p$ is:

$$||\boldsymbol{x}|| = \sqrt{\sum_{i=1}^{p} x_i^2}.$$

## 1.3 Thesis structure

The thesis is organized as follows: In Chapter 2, we discuss topics including a review of regression and classification techniques. We review the linear regression for uni and multivariate response variables. We explore the limitations encountered by researchers and discover the remedies for these limitations. We review the projection-based methods to deal with the dimensionality issue, as well as the regularization technique. Further, we define the classification problem and introduce some well-known classifiers. In Chapter 3, we rigorously define envelope method. We discuss the response envelope model, which aims to reduce the dimension of response variables. Similarly, we review the predictor envelope, which targets reducing the dimension of the predictor variables. In both models, we show the theoretical method to estimate the envelope subspace. Further, we demonstrate the algorithm for extracting the reducing subspace. Chapter 4 we review the support vector machines classifier. We illustrate the classification rule in the linear dataset in both cases; separable and non-separable data. Further, we introduce kernel support vector machine or equivalently nonlinear SVM. Chapter 5 we introduce our proposed classifier, the Envelope-based support vector machines (ESVM). We demonstrate how to estimate the reduced basis. The performance of our classifier has been tested on simulated and real data. In Chapter 6 we extended our proposed classifier ESVM to sparse data. We have introduced the adaptive group lasso as a penalization to impose the sparsity in the estimated subspace.

# Chapter 2

# Preliminaries

## 2.1 Linear regression model

The general concept of regression in modelling the data is to study the relationship between a set of variables. Simple linear regression facilitates the linear relationship between dependent variable and an explanatory variable(s). Generally speaking, by fitting a linear model to a scientific data we want to detect an evidence of association between predictor variables and the outcome. The strength of the relationship impact the prediction accuracy.

### 2.1.1 Model formulation

Suppose we have dependent variable $\boldsymbol{y}$ with $n$ observations, $\boldsymbol{y} = (y_1, ..., y_i, ..., y_n)^T$, and a set of explanatory variables $\boldsymbol{X} \in \mathbb{R}^{p \times n}$. Hence, if we want to predict observation $i$ as a linear combination of ($p$) independent explanatory variables, we can do so by assuming the following model:

$$y_i = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i + e_i, \quad i = 1, ..., n \tag{2.1}$$

where $\beta_0$ is the intercept, $\boldsymbol{\beta} \in \mathbb{R}^p$ is the model coefficients vector and $e_i$ is the error terms and assumed to be independent normally distributed with equal variance, $e_i \sim N(0, \sigma^2)$.

In model (2.1), $y_i$ is assumed to be continues normally distributed with mean $\mu_y = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}_i$ and variance $\sigma^2$. If the normality assumption for the response data is violated, we may model our data with a generalisation of model which is known as the generalized linear model (GLM). A GLM consists of three components (Dobson and Barnett, 2018):

- Dependent variable which follows a member of the exponential family such as: Poisson, Gamma, Normal or Binomial distribution,
- Explanatory variable(s) and set of coefficients; $\boldsymbol{\beta}^T\boldsymbol{X}$ and,
- Link function $\boldsymbol{\eta}$ explains the relation between the expected value of the response ($E(\boldsymbol{y}) = \boldsymbol{\mu}$) and $\boldsymbol{\beta}^T\boldsymbol{X}$ such that $\boldsymbol{\eta} = g(\boldsymbol{\mu}) = \beta_0 + \boldsymbol{\beta}^T\boldsymbol{X}$. The link function can be specified based on the selected model. The estimation of the model parameters ($\boldsymbol{\beta}$) will be discussed in the next section.

### 2.1.2 Estimation

Having a specified model, the estimation of its parameters is required. Consider model (2.1), the estimation of the model coefficients vector ($\beta_0, \boldsymbol{\beta}$) has been studied extensively in statistical literature. The parameters can be estimated either via maximum likelihood estimation or the least squares fit.

To demonstrate the least squares method, define the residual as:

$$e_i = y_i - \beta_0 - \boldsymbol{\beta}^T\boldsymbol{x}_i. \tag{2.2}$$

Equivalently, (2.2) can be written in matrix form:

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{\beta}, \tag{2.3}$$

where $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}) \in \mathbb{R}^{p+1}$ and $\boldsymbol{X} \in \mathbb{R}^{(p+1)\times n}$.

The least squares estimation of $\boldsymbol{\beta}$ is the parameters that minimize the sum of squares residuals (McCullagh and Nelder, 1989). That is

$$\boldsymbol{e}^T\boldsymbol{e} = (\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}^T\boldsymbol{\beta})$$

That yields:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{y}. \tag{2.4}$$

On the other hand, the maximum likelihood estimation is obtained based on the assumption that $y_i \sim N(\boldsymbol{\beta}^T\boldsymbol{x}_i, \sigma^2)$. Thus the maximum likelihood estimation of $\boldsymbol{\beta}$ is given by:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{y}. \tag{2.5}$$

## 2.2 Multivariate linear regression

### 2.2.1 Model formulation

Multivariate linear regression (MLR) describes the linear relationship between a set of $r$ response variables and $p$ predictors (Chatfield and Collins, 1981). This relationship is described via statistical model. The model coefficients facilitate the effect of the predictors on the response variables. That is, consider the multivariate linear regression model:

$$\boldsymbol{Y} = \boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{X} + \boldsymbol{E}, \tag{2.6}$$

where $\boldsymbol{\alpha}$ is a vector of $r$ elements (intercepts), $\boldsymbol{Y}$ is $(r \times n)$ matrix of $r$ responses, $\boldsymbol{\beta}$ is an $r \times p$ regression coefficients matrix, $\boldsymbol{X}$ is $p \times n$ the predictor matrix and $\boldsymbol{E}$ $(r \times n)$ is the normally distributed errors such that $\boldsymbol{e}_{.j} \sim N(0, \sigma_j^2 \boldsymbol{I}_n), j = 1, ..., r$, and $\boldsymbol{e}_{i.} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}), \quad i = 1, ..., n$, where the diagonal elements of $\boldsymbol{\Sigma}$ are $\text{var}(y_j) = \sigma_j^2$ and the off diagonal elements are $cov(y_j, y_k) = 0, j \neq k$.

### 2.2.2 Estimation

MLR is used for prediction and estimation, where the main interest is to estimate the regression coefficients ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$). If the data $\boldsymbol{X}$ is of full rank the estimated coefficients matrix $\hat{\boldsymbol{\beta}}$ is achieved via ordinary least squares(OLS) or equivalently the method of maximum likelihood estimation can be used for each response variable simultaneously. That is, suppose $\boldsymbol{B} \in \mathbb{R}^{r \times (p+r)}$ is the parameters of interest matrix, and $\boldsymbol{X} \in \mathbb{R}^{(p+r) \times n}$ is the design matrix. Hence, the OLS estimates is given by:

$$\hat{\boldsymbol{B}}_{\text{ols}} = \boldsymbol{Y}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1} \tag{2.7}$$

In (2.7) the dependency among the response variables is ignored, however it should be taking into consideration when the aim is to estimate the coefficients jointly.

In the world of high dimensional data when the number of features is high it is common to encounter the multicollinearity in univariate and MLR. Thus, the method of OLS breaks down due to rank deficiency i.e $(\boldsymbol{X}\boldsymbol{X}^T)^{-1}$ does not exit.

To solve this problem it is assumed that only a small subset of the features is relevant to the analysis. A class of statical techniques restrict the analysis on a set of linear combinations less than the original dimension without losing the information. In section 2.3, we discuss the dimension reduction via feature extraction methods while in section 2.4 we review the variable selection methods.

## 2.3 Dimension reduction

### 2.3.1 Problem statement

Dimension reduction is a very active area in statistics and machine learning due the exponential growth of scientific data in size and complexity. This complexity has made the determination of the relation between response variable(s) and predictor variables challenging. Statistics literature tackled this problem via two approaches: *variable selection* and *feature extraction*. Variable selection is the technique where out of large group of predictor variables only a few are significantly related to the outcome. While feature extraction methods task is to come up with a set of linear combinations of the original covariates that is related to the response. That is, suppose $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ is the predictors matrix, feature extraction aims to find $\boldsymbol{\eta} \in \mathbb{R}^{p \times d}, d < p$ that satisfies the following condition:

$$Y \mid \boldsymbol{X} \sim Y \mid \boldsymbol{\eta}^T \boldsymbol{X}, \tag{2.8}$$

This condition indicates that the conditional distributions of $Y \mid \boldsymbol{X}$ and $Y \mid \boldsymbol{\eta}^T \boldsymbol{X}$ are the same. That is, $\boldsymbol{\eta}^T \boldsymbol{X}$ can replace the original data without loss of information. $\boldsymbol{\eta}$ is denoted by projection matrix and the estimation of $\boldsymbol{\eta}$, hence, is of special interest. In the following section we discuss the construction of the projection matrix $\boldsymbol{\eta}$.

### 2.3.2 Sufficient dimension reduction

The objective of sufficient dimension reduction is to estimate a lower dimensional subspace that contains all the information in the data. Suppose $\mathcal{S}$ is a subspace with the following property:

$$Y \perp \boldsymbol{X} | \boldsymbol{P}_{\mathcal{S}} \boldsymbol{X}, \tag{2.9}$$

where $\boldsymbol{P}_{\mathcal{S}}$ is the projection matrix. If $\mathcal{S}$ is a reducing subspace, then $\boldsymbol{P}_{\mathcal{S}} \boldsymbol{X}$ contains all the information that $\boldsymbol{X}$ has about $Y$. Further, assume $\boldsymbol{\eta} \in \mathbb{R}^{p \times d}, \quad d < p$ is a basis for $\mathcal{S}$, then $\boldsymbol{\eta}^T \boldsymbol{X}$ is used for regression (Cook et al., 2010), (Li, 2018). Any subspace which satisfies (2.9) is a dimension reduction subspace; however, the intersection of all subspaces if itself a dimension reduction subspace is known as a *central subspace* (CS) and denoted by $\mathcal{S}_{Y|\boldsymbol{X}}$, (Cook, 1994), (Cook, 1998), (Cook et al., 2010). The central subspace does not always exist; however, it has been shown that if it exists it is unique under mild regularity conditions (Cook, 1998), (Yin et al., 2008). Sliced inverse regression (SIR) (Li, 1991) and sliced average variance estimation (SAVE) (Cook and Weisberg, 1991) are the pioneers work to estimate CS.

## 2.4 Penalized likelihood methods

In this section we discuss the dimension reduction via variable selection techniques. In particular, we study the variable selection via penalization. The penalized log-likelihood is simply a log-likelihood accompanied with a penalty that will shrink the final likelihood estimates (Cole et al., 2014). The ultimate goal of penalized likelihood methods is to improve the model efficiency by producing a set of coefficients that are exactly zero. These method can be viewed as reducing the variability of the model coefficients estimates and raising some degree of bias. In this section we demonstrate some of the most widely used penalties.

### 2.4.1 Ridge regression

Ridge regression or equivalently $\ell_2$ norm is one of the widely used penalty. It shrinks the parameters by imposing a quadratic constraint to the objective function. In the regression content; the ridge coefficients are the values that minimizes the penalized sum of squares:

$$\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \tag{2.10}$$

where $\lambda$ is the imposed penalty and $\lambda \sum_{j=1}^{p} \beta_j^2$ denoted by the shrinkage penalty. The shrinkage term works to shrink the coefficient values towards zero, not to set them to zero. The value of $\lambda$ affects the amount of the shrinkage, as a larger value indicates more shrinkage (Hastie et al., 2009). In other words, the model fitted by ridge regression will include all predictor variables, which causes interpretation challenges, especially when $p$ is large.

### 2.4.2 LASSO

The least absolute shrinkage and selection operator (LASSO) or $\ell_1$-norm (Tibshirani, 1996) is probably the most commonly used penalties. For linear regression model, LASSO is defined as:.

$$\begin{aligned}\hat{\beta}_{\text{lasso}} =& \arg\min_{\beta} \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \\ &\text{subject to} \sum_{j}^{p} |\beta_j| \leq t,\end{aligned} \tag{2.11}$$

where $|.|$ is $\ell_1$-norm, however, (2.11) can be written with different parametrization as follows:

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{2.12}$$

where $\lambda$ is the tuning parameter. In contrast to ridge regression, LASSO produces sparse estimates as it sets the coefficients that have small OLS estimates to zero. In other words, making $t$ sufficiently small in (2.11) leads to some coefficients to be exactly zero (Hastie et al., 2009). However, LASSO ignores the relative importance of each variable and applies an equal amount of shrinkage for each coefficient which lacks model selection consistency. Hence, adaptive LASSO was proposed as a remedy to this problem.

#### Adaptive LASSO

Adaptive LASSO (aLASSO) (Zou, 2006), is a modified version of LASSO that was proposed as a remedy for the inconsistency in variable selection with LASSO. The key difference is that aLASSO uses adaptive weight for penalizing different coefficients instead of penalizing the coefficients equally. That is, the aLASSO is defined as follows:

$$\hat{\boldsymbol{\beta}}_{\text{alasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ ||\boldsymbol{y} - \sum_{j=1}^{p} \beta_j \boldsymbol{x}_j||^2 + \lambda \sum_{j=1}^{p} \hat{w}_j |\beta_j| \right\}, \tag{2.13}$$

where $\hat{w}_j$ is the adapted weight and defined as: $\boldsymbol{w} = 1/|\hat{\boldsymbol{\beta}}|^{\tau}, \quad \tau > 0$, and $\hat{\boldsymbol{\beta}}$ is a $\sqrt{n}$ consistence estimates of $\boldsymbol{\beta}$, for instance it can be the OLS estimates. This modification allows aLASSO to enjoy the oracle property as indicated in the following proposition that is given in (Zou, 2006).

**Proposition 2.4.1.** *Let $\mathcal{A}^* = \{j : \hat{\boldsymbol{\beta}}_{alasso} \neq 0\}$. Suppose $\lambda/\sqrt{n} \to 0$ and $\lambda n^{(\tau-1)/2} \to \infty$. Then the adaptive lasso estimates must satisfy the following:*

1. *Consistenc in variable selection:* $\lim_n P(\mathcal{A}^* = \mathcal{A}) = 1$

2. *Asymptotic normality:* $\sqrt{n}(\hat{\boldsymbol{\beta}}_{alasso,\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}) \rightarrow^d N(\mathbf{0}, \sigma^2 \times \boldsymbol{C}_{11}^{-1})$.

where $\mathcal{A} = \{j : \hat{\boldsymbol{\beta}} \neq 0\}$ and $|\mathcal{A}| = p_0 < p$. $\boldsymbol{C}_{11}$ is a $p_0 \times p_0$ is a positive definite matrix.

#### Group lasso and adaptive group lasso

Group LASSO (gLASSO) (Yuan and Lin, 2006), is the natural extension of LASSO that selects variables in a group manner. The motivation behind it is that if we aim to exclude a group of predictor variables instead of selecting predictor variables individually. The group lasso solves the following:

$$\hat{\beta} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \frac{1}{2} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} ||\beta_j|| \right\}, \tag{2.14}$$

where $||.||$ is the $\ell_2$-norm.

(Wang and Leng, 2008) have argued that similar to LASSO, gLASSO suffers from estimates inefficiency as well as variable selection inconsistency. Hence, they proposed adaptive group LASSO (agLASSO) as a solution:

$$\hat{\beta} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \frac{1}{2} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} w_j ||\beta_j|| \right\}, \tag{2.15}$$

where the weight $w_j = ||\hat{\beta}_j||^{-\tau}$, $\hat{\beta}_j$ can be OLS.

### 2.4.3 Elastic net

Elastic net (Zou and Hastie, 2005) combines $\ell_1$-norm (LASSO) and $\ell_2$-norm (ridge) to improve the performance of LASSO. In scenario where $p > n$ LASSO selects at most $n$ variables. Further, in the classical settings where $n > p$, if the predictor variables are highly correlated LASSO tends to select only one variable. Hence, elastic net was introduce to overcome the limitations by LASSO such that it encourages group selection and removes the limitations on the number of selected variables. Elastic net is defined as:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + (1-\lambda) \sum_{j=1}^{p} |\beta_j| + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}. \tag{2.16}$$

The function $\lambda \sum_{j=1}^{p} |\beta_j| + (1-\lambda) \sum_{j=1}^{p} \beta_j^2$ referred to as *elastic net*. When the tuning parameter $\lambda = 1$ the elastic net reduced to ridge regression. In this setting, $\ell_1$ part of the penalty introduce the sparse model while the quadratic part encourages the grouping effect as well as removes the limitations on the number of selected variables.

### 2.4.4 Smoothly clipped absolute deviation penalty

The Smoothly Clipped Absolute Deviation (SCAD) penalty (Fan and Li, 2001) was proposed to improve the biasedness in LASSO estimates. (Fan and Li, 2001) argued that a good panelized estimator should have properties: unbiasedness, sparsity and continuity. They showed that in case of LASSO penalty the sparsity and continuity hold where SCAD satisfied all three properties. The SCAD penalty is defined as:

$$p_\lambda(\theta) = \lambda\left\{I(\theta \leq \lambda) + \frac{(a\lambda - \theta)}{(a-1)\lambda} + I(\theta > \lambda)\right\}, \quad \text{for some } a > 2 \text{ and } \theta > 0. \tag{2.17}$$

## 2.5 Classification

In the previous sections, we have discussed the regression and dimension reduction. In this section, we will discuss the other major topic in this thesis; that is classification.

### 2.5.1 Problem statement

In many scientific datasets the response is a qualitative variable, such as the type of treatment the participant is taken (treatment A or treatment B), the level of education, or any similar problem in which the outcome describes a category or a status. In such data, the observations are assumed to be grouped into $Z, (Z \geq 2)$ distinct groups, which are also known as *classes*. This type of data is referred to as *labeled data* where the response is a qualitative or categorical variable and represents the *label* associated with each object. For a new observation, the task is to predict a discrete value ( *label*) that represents the class to which it belongs. The procedure of assigning the new object into a class is known as *classification*. The classification process acquires two elements: first is a classification rule or *decision function* which is a mathematical formula that creates decision boundaries between the distinct classes. The other element is a *classifier*, which is a developed algorithm that employs the classification rule to allocate the observations into the appropriate class with reasonable accuracy (Abe (2005), Hastie et al. (2009), James et al. (2013)).

Suppose we have $n$ independent observations: $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$ where $\boldsymbol{x}_i \in \mathbb{R}^p$ is known as the *features*, and a prespecified $y_i \in \{1, ..., Z\}, Z \geq 2$, the class label. Hence, the classifier works to allocate objects into the distinguish groups based on the decision rule. In statistics and machine learning literature various classifiers have been studied such as: linear discriminant analysis (Fisher, 1936), logistic regression (Efron, 1975), decision trees, random forests, neural network (Bishop et al., 1995) and support vector machines(Cortes and Vapnik, 1995). The performance of theses classifiers is measured via misclassification rates. That is, suppose $\tau(\boldsymbol{x})$ is the estimated class for observation $\boldsymbol{x}$, hence the misclassification rate is $\Pr(y \neq \tau(\boldsymbol{x}))$ where $y$ is the true class. A good classifier gives low misclassification rate.

In the next section we explore some of the most commonly used classifiers:

### 2.5.2 Linear discriminate analysis

Linear discriminate analysis (LDA) is one of the oldest and most widely used classifiers. If we have $\{y_i, \boldsymbol{x}_i\}_{i=1}^n$, are $n$ independent observations such that $y_i$ represents the class label and $\boldsymbol{x}_i$ is a $p$-dimensional normally distributed predictor variable; however, LDA assumes that each class has different mean and shared variance such that:

$$f_z(\boldsymbol{x}) = \frac{1}{(2/\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_z)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_z)\}.$$

In case $Z > 2$ the Bayes rule is used to determine the classification decision (Mai, 2013):

$$\delta_z(\boldsymbol{x}) = \arg\max_z\{\log \pi_z + \boldsymbol{\mu}_z^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_z)\}.$$

In practice the parameters $(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}, \pi_z)$ are unknown. Hence, LDA tends to estimate them from the training sample:

$$\hat{\pi}_z = \frac{n_z}{n}, \quad \boldsymbol{\mu}_z = \sum_{y_i = z} \boldsymbol{x}_i / n_z, \quad \hat{\boldsymbol{\Sigma}} = \sum_{z=1}^{Z} \sum_{y_i = z} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_z)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_z)^T / n - z.$$

Another property of LDA is that it reduces the data dimension while perceiving the class separation. Given a $p$-dimensional dataset that has $Z$ classes; the $Z$ centroids lie in a subspace of dimension $\leq Z$ (Hastie et al., 2009). The reduction is performed by projecting the data into a subspace spans by the centroids. The subspace can be extracted as follows:

- compute $\boldsymbol{M} \in \mathbb{R}^{Z \times p}$ whose entries are the class centroids and the within class covariance $\boldsymbol{W}$;
- compute $\boldsymbol{M}^* = \boldsymbol{M}\boldsymbol{W}^{-1/2}$ using the eigen-decomposition of $\boldsymbol{W}$;
- compute the covariance matrix of $\boldsymbol{M}^*$, $\boldsymbol{B}^*$ and its eigen-decomposition $\boldsymbol{B}^* = \boldsymbol{V}^*\boldsymbol{D}_B\boldsymbol{V}^{*T}$. The coordinate of the desired subspace are defined by the columns of $\boldsymbol{V}^*$, $v_z^*$.

Performing the steps explained above, the $z^{th}$ discriminant variable is $A_z = v_z^T\boldsymbol{X}$ where $v_z = \boldsymbol{W}^{-1/2}v_z^*$.

### 2.5.3 Logistic regression

Logistic regression model comes from the desire to model the posterior probabilities of the $Z$ classes through linear functions in $\boldsymbol{x}$. That is, logistic regression models the probability that $y_i$ belongs to a specific category instead of modelling the response $y_i$ directly. In other words, the linear regression to produce the probabilities via:

$$p(\boldsymbol{X}) = \boldsymbol{\beta}^T\boldsymbol{X}. \tag{2.18}$$

Using (2.18) to compute the probabilities is inappropriate because it gives negative probabilities and probabilities larger than one. To avoid this problem logistic regression model is detailed in terms of $Z-1$ logit-odds transformations assuring the constraint that the probabilities sum to one. That is, the conditional probabilities are given by:

$$\log\frac{Pr(Y=z|\boldsymbol{X}=\boldsymbol{x})}{Pr(Y=Z|\boldsymbol{X}=\boldsymbol{x})} = \beta_{z0} + \boldsymbol{\beta}_z^T\boldsymbol{x}, \quad z = 1, ..., Z-1. \tag{2.19}$$

(2.19) simplified as follows:

$$\begin{aligned} Pr(Y=z|\boldsymbol{X}=\boldsymbol{x}) &= \frac{\exp(\beta_{z0}+\boldsymbol{\beta}_z^T\boldsymbol{x})}{1+\sum_{j=1}^{Z-1}\exp(\beta_{j0}+\boldsymbol{\beta}_j^T\boldsymbol{x})}, \quad z = 1, ..., Z-1, \\ Pr(Y=Z|\boldsymbol{X}=\boldsymbol{x}) &= \frac{1}{1+\sum_{j=1}^{Z-1}\exp(\beta_{j0}+\boldsymbol{\beta}_j^T\boldsymbol{x})}. \end{aligned} \tag{2.20}$$

Since the $Pr(Y|X)$ is completely specified, the parameters of the model are estimated via the maximum likelihood method (Hastie et al., 2009), (James et al., 2013). For binary class data the problem is straightforward via binomial distribution and the multinomial is suitable in the case of multi-classes. Once the parameters are estimated, for a given $\boldsymbol{x}$ the probability is calculated as given in (2.20). The decision is determined by a specifying a threshold probability $a \in (0, 1)$. For binary class problem, that is $y \in \{0, 1\}$ then the prediction of $y$ is given by:

$$y = 1 \text{ if } \hat{p}(\boldsymbol{x}) \geq a;$$
$$y = 0 \text{ if } \hat{p}(\boldsymbol{x}) < a.$$

For multi-class problems we use the techniques explained in Chapter 4.

### 2.5.4 Classification evaluation

In developing a classifier, the classification evaluation is crucial. It shows the generalization ability of such classifier. A number of ways are used to evaluate the classification process. Suppose we generate data from $Z$ classes one way to evaluate the classifier on the test data is to create a *confusion matrix* $(A)$ whose entries $a_{ij}$ is the number of data points from class $i$ that classified in class $j$, Table 2.1 (Abe, 2005).

| | Assigned.positive | Assigned.negative |
|---|---|---|
| Actual positive | TP | FN |
| Actual negative | FP | TN |

Table 2.1: The confusion matrix for diagnosis data.

The other way is the *recognition rate* $(R)$ or accuracy which is given by:

$$R = \frac{\sum_{i=1}^{Z} a_{ii}}{\sum_{i,j=1}^{Z} a_{ij}} \times 100.$$

Alternatively, the *error rate* is used to measure the overall performance of a classifier and given by:

$$E = \frac{\sum_{i \neq j, i,j=1}^{Z} a_{ij}}{\sum_{i,j=1}^{Z} a_{ij}} \times 100.$$

Under the assumption that there is no unclassified data, $R + E = 100\%$. To the purpose of comparing classifiers one may generate several dataset that divided into training and test data. For each data the error rate is obtained then investigate if there is a statistical difference in the mean error rate and their standard deviations of the classifiers.

In some situations where one class is dominant; that is, the data in hand has imbalanced classes. For instance, data for diagnosis problem with negative (normal) and positive (abnormal) outcome. The imbalanced caused by the difficulty in obtaining samples for positive class while data samples for negative class are easy to obtain. The misclassification of positive sample into negative class is more risky compared to the misclassification of negative sample into positive class (Abe, 2005). In this case, the commonly used measures are precision, recall and the receiver operator characteristic (ROC). The precision given by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}.$$

while the Recall is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}},$$

where

- True Positive (TP)=number of subjects that correctly classified as positive.
- True Negative (TN)=number of subjects that correctly classified as negative.
- False Positive (FP)=number of subjects that falsely classified as positive.
- False Negative (FN)=number of subjects that falsely classified as negative.

The other measure is the ROC that is plotted using the calculated values of the true positive rate on the $y$-axis and the false positive rate on the $x$-axis. The true positive rate is given by:

$$\text{True-positive rate} = \frac{\text{TP}}{\text{TP+FN}} \tag{2.21}$$

while the false-positive rate is:

$$\text{False-positive rate} = \frac{\text{FP}}{\text{FP+TN.}} \tag{2.22}$$

The ROC is used to determine a threshold based on the classification is performed. For instance, let $t \in \{0.1, 1\}$ be the threshold, Figure 2.1 shows a ROC for classifying binary two dimensional data. The numbers on the curve indicate different values of $t$. Once the optimal value of $t$ is selected the classification is performed. That is, suppose the class membership associated with individual $i$ is $\pi_i$, then $\boldsymbol{x}_i$ allocated in class 1 if $\pi_i > t$ and allocated in class 2 otherwise.

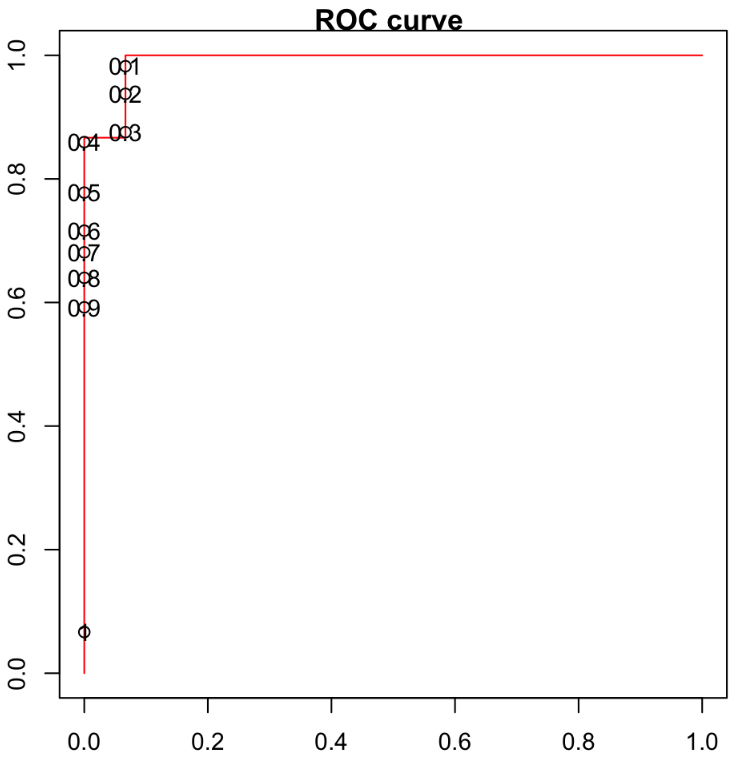


Figure 2.1: The ROC based on classifying two dimensional binary data using logistic regression.

# Chapter 3

# Review of Envelope Method for Linear Dimension Reduction

## 3.1 Introduction

In this chapter we review the envelope model, the technique for dimensionality reduction. It was introduced by Cook et al. (2010) and has been developed and expanded since then by several authors. For an overview of the envelope methodology see Lee and Su (2019) and Cook (2019).

The argument is that in regression setting, especially when the number of features is large, we believe not all of the features are informative. Hence, it is important to be able to distinguish the informative features from the non-informative ones. Consider the model given in (2.6), this technique can be applied to reduce the dimension of the responses, predictors, or responses and predictors simultaneously. Envelope was introduced in regression framework and aims to increase the efficiency in parameter estimation ($\hat{\boldsymbol{\beta}}$). The gain in efficiency is achieved by excluding the non important variation which might be present in the data. In other words, the response (or predictor) variables can be divided into two parts: material part and immaterial part. The first part (*material* part) contains the information related to the goal of the study. The other group (*immaterial*) has no impact on estimating $\boldsymbol{\beta}$. Envelope procedure relies on reducing the dimensionality of the data by extracting a projection matrix out of the material part only then project the data on. Thus, it improves the efficiency via based the estimation only on the material part.

This chapter is organised as follows: in Section 3.2 we introduce the model notation and demonstrate the method for response reduction. Similarly, in Section 3.3 we explore the envelope model for reducing the predictor variables. In Section 3.4 we discuss the algorithm for estimating the envelope basis. In Section 3.5 we review some of the developed envelope based methods. We emphasise that the content of this chapter depends heavily on the material from Cook (2019).

## 3.2 Response envelope

Suppose we have the model

$$\boldsymbol{y} = \boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{x} + \boldsymbol{e}, \tag{3.1}$$

where

- $\boldsymbol{y}$ is the response variables vector of $r > 1$ responses,
- $\boldsymbol{\alpha}$ is a vector of $r$ elements (intercepts),
- $\boldsymbol{\beta}$ is an $r \times p$ regression coefficients matrix,
- $\boldsymbol{x} \in \mathbb{R}^p$ the predictor variables, and
- $\boldsymbol{e}$ is the normally distributed errors such that $\boldsymbol{e} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$.

Having $r > 1$ response variables, the classical method of estimating $\boldsymbol{\beta}$ is to perform an independent univariate regression of $y$ on $\boldsymbol{x}$ for each response. This technique ignores the relationship among the response variables. The motivation behind the response envelope is that there exists a linear combination of the response variables that is irrelevant to the analysis. Keeping this group will affect the estimation of $\boldsymbol{\beta}$. In other words, the response envelope aims to reduce the dimension of the multivariate response and base the analysis on the relevant response variables, which will improve the estimation efficiency. Obtaining the estimation of the model coefficients based on the relevant part only, as the envelope suggested, will reduce the variance of the estimates.

To construct the reducing basis consider model (3.1), furthermore, suppose $\mathcal{M}$ is a $u$-dimensional subspace ($u < r$). Let $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0)$ be an $r \times r$ semi-orthogonal matrix, where $\boldsymbol{\Gamma} \in \mathbb{R}^{r \times u}$ is orthogonal basis of $\mathcal{M}$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{r \times (r-u)}$ be an orthogonal basis of $\mathcal{M}^{\perp}$; the orthogonal complement of $\mathcal{M}$, and $u$ is the envelope dimension such that $u < r$. When the goal is to reduce the response variables, $\boldsymbol{\Gamma}^T\boldsymbol{y}$ forms the reduction in $\boldsymbol{y}$. $\boldsymbol{\Gamma}$ known as the envelope subspace and has to satisfy the following conditions:

i. The marginal distribution of $\boldsymbol{\Gamma}_0^T\boldsymbol{y}$ does not depend on $\boldsymbol{x}$ and is not affected by the change in $\boldsymbol{x}$; that is, $\boldsymbol{\Gamma}_0^T\boldsymbol{y} \sim \boldsymbol{\Gamma}_0^T\boldsymbol{y} \mid \boldsymbol{x}$, where $\sim$ means has the same distribution, and

ii. $\boldsymbol{\Gamma}^T\boldsymbol{y}$ and $\boldsymbol{\Gamma}_0^T\boldsymbol{y}$ are uncorrelated; that is, $\boldsymbol{\Gamma}^T\boldsymbol{y} \perp (\boldsymbol{\Gamma}_0^T\boldsymbol{y} \mid \boldsymbol{x})$.

As stated in Cook et al. (2010), the above conditions hold if and only if:

i. $\text{span}(\boldsymbol{\beta}) \subseteq \text{span}(\boldsymbol{\Gamma})$, and

ii. $\text{cov}(\boldsymbol{\Gamma}^T\boldsymbol{y}, \boldsymbol{\Gamma}_0^T\boldsymbol{y} \mid \boldsymbol{x}) = 0$.

This approach establishes a parametric link between the covariance matrix $\boldsymbol{\Sigma}$ and the coefficients $\boldsymbol{\beta}$. This link is defined via the envelope subspace ($\boldsymbol{\Gamma}$) that satisfies the conditions i, and ii. However, this subspace is not unique but its span, $\text{span}(\boldsymbol{\Gamma})$, is identifiable. Hence envelope extracts the smallest subspace contains $\boldsymbol{\beta}$ that is called $\boldsymbol{\Sigma}$-*envelope* of $\text{span}(\boldsymbol{\beta})$, and denoted by $\boldsymbol{\mathcal{E}}_\Sigma(B)$, where $B$=$\text{span}(\boldsymbol{\beta})$. However, based on the classification of the response variables into material and immaterial parts, the covariance matrix decomposes into two semi-orthogonal matrices; $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T$ where $\boldsymbol{\Omega}, \boldsymbol{\Omega}_0$ carried the coordinate of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}_0$ respectively.

The following definition facilitates the decomposition of a matrix by a subspace; for a subspace $\mathcal{R} \in \mathbb{R}^r$ and $\boldsymbol{M} \in \mathbb{R}^{r\times r}$.

**Definition 3.2.1.** $\mathcal{R}$ reduces $\boldsymbol{M} \in \mathbb{R}^{r\times r}$ if and only if $\boldsymbol{M}$ can be written in the form $\boldsymbol{M} = \boldsymbol{P}_\mathcal{R}^T\boldsymbol{M}\boldsymbol{P}_\mathcal{R} + \boldsymbol{Q}_\mathcal{R}^T\boldsymbol{M}\boldsymbol{Q}_\mathcal{R}$,

where $\boldsymbol{P}_{(A)}$ is the projection onto the column space of $\boldsymbol{A}$ and $\boldsymbol{Q}_{(A)}$ is the orthogonal complement of $\boldsymbol{P}_{(A)}$ such that $Q_{(A)} = \boldsymbol{I}_r - \boldsymbol{P}_{(A)}$; $\boldsymbol{I}_r$ is the $r \times r$ identity matrix.

The response envelope model is obtained via the re-parametrisation of model (2.6) be become:

$$\boldsymbol{y} = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x} + \boldsymbol{e}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T. \tag{3.2}$$

To demonstrate the gain achieved by envelope in the estimated parameters we gave example in Section 3.2.1 then in Section 3.2.2 we show the the parameters estimation for model (3.2).

### 3.2.1 Illustrative example

To give an illustration of the envelope methodology on dimension reduction. Consider the case for response dimension reduction (response envelope), with Berkeley dataset (Tuddenham, 1954). This data has the height measurements for 93 individuals (39 boys, 54 girls) were born between 1928 and 1929 in Berkeley, CA. The dataset has 31 response variables and a univariate predictor, for illustration purpose, consider ($\boldsymbol{Y}_{2\times 93}$) is bivariate response that contains the height measurement for age 13 and 14, while the predictor $\boldsymbol{x}$ is the gender indicator binary variable (0 indicates boys, 1 for girls). That is, we have the following linear model:

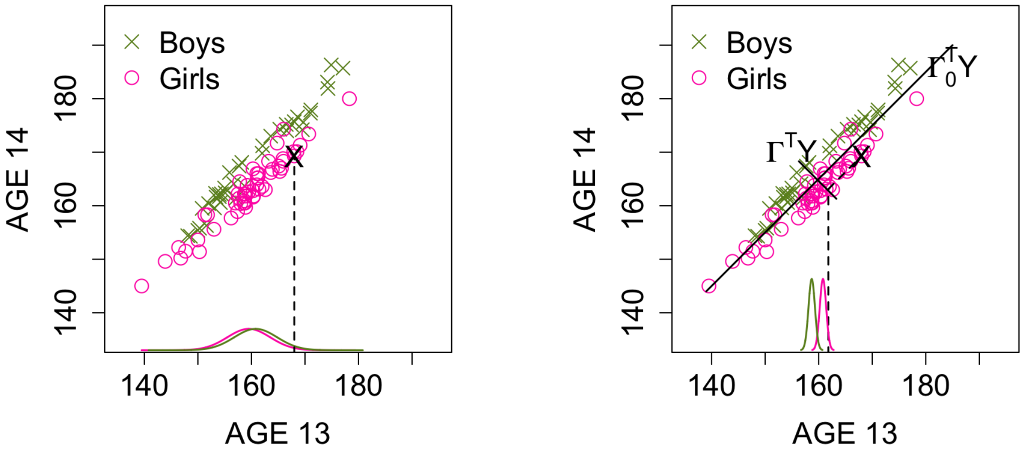
$$\boldsymbol{y}_i = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} + \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} x_i + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix},$$

where

$$\alpha_1 = E(\boldsymbol{y}_1|\boldsymbol{x}=0), \beta_1 = E(\boldsymbol{y}_1|\boldsymbol{x}=1) - E(\boldsymbol{y}_1|\boldsymbol{x}=0),$$
$$\alpha_2 = E(\boldsymbol{y}_2|\boldsymbol{x}=0), \beta_2 = E(\boldsymbol{y}_2|\boldsymbol{x}=1) - E(\boldsymbol{y}_2|\boldsymbol{x}=0).$$

The main interest here is to estimate $\boldsymbol{\beta} = (\beta_1, \beta_2)^T$. First, we estimated the coefficient via OLS i.e the standard method by regressing each of the response variables on the binary predictor then estimate the parameters simultaneously. Figure 3.1 (a) shows the standard inference on $\beta_1$, the dotted line shows the projection path for randomly chosen point $x$. While the two curves represent the projection distribution of the two groups onto $Y_1$. It can be seen clearly that the two curves are not distinguishable. Envelope suppose to improve the inference by eliminating the immaterial variance and make clear separation. Figure 3.1 (b) shows the envelope subspace ($\boldsymbol{\Gamma}^T\boldsymbol{Y}$) and its orthogonal complement ($\boldsymbol{\Gamma}_0^T\boldsymbol{Y}$), and clearly we can see that the variation among the envelope subspace is less than on its complement. Hence, projecting the data points onto the envelope subspace eliminates the immaterial variation. In fact this is reflected on the distribution curves of the two groups that are now distinguishable. That is because envelope based the inference on the material part only by projecting the data points onto $\boldsymbol{\Gamma}^T\boldsymbol{Y}$ first.

In the following section we discuss the estimation method of the envelope basis $\mathbf{\Gamma}$ and the parameters of response envelope model.



(a)

(b)

Figure 3.1: Height measurement for age 13 and age 14. (a) shows inference under linear model; (b) inference under envelope model.

### 3.2.2 Estimation

In this section we discuss the estimation of the parameters of the response envelope model (3.2). The parameters to be estimated for model (3.2) are $(\mathbf{\Gamma}, \mathbf{\Gamma}_0, \boldsymbol{\eta}, \mathbf{\Omega}, \mathbf{\Omega}_0)$ and estimated via maximum likelihood estimation.

Let $\boldsymbol{y} \in \mathbb{R}^r$ be the vector of response variables, and $\boldsymbol{x} \in \mathbb{R}^p$ be the vector of the predictor variables. Further, let multivariate linear regression is given by:

$$\boldsymbol{y}_i = \boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{x}_i + \boldsymbol{e}_i, \tag{3.3}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^r$ is the intercept and $\boldsymbol{\beta} \in \mathbb{R}^{r \times p}$ is the regression coefficients matrix. The conditional distribution of $\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}$ is:

$$\boldsymbol{Y}|\boldsymbol{x} \sim N(\boldsymbol{\mu}_y, \mathbf{\Sigma}_{Y|X}), \tag{3.4}$$

where $\boldsymbol{\mu}_y = \boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{x}_i$. Hence, the likelihood function is given by:

$$L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-nr/2}|\boldsymbol{\Sigma}|^{-n/2} \exp\{\sum_{i=1}^{n} \frac{-1}{2}(\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{\beta}\boldsymbol{x}_i)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{\beta}\boldsymbol{x}_i)\} \tag{3.5}$$

The log likelihood function is:

$$\ell = \frac{-nr}{2}\log(2\pi) - \frac{n}{2}\log|\boldsymbol{\Sigma}| - \sum_{i=1}^{n}\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{\beta}\boldsymbol{x}_i)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{\beta}\boldsymbol{x}_i) \tag{3.6}$$

The previous equation can be re-parametrised based on envelope model. That is, $\boldsymbol{\beta} = \boldsymbol{\Gamma}\boldsymbol{\eta}$, $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T$, where $\boldsymbol{\Gamma} \in \mathbb{R}^{r\times u}$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{r\times(r-u)}$, $u$ =dimension of envelope basis, and $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{r\times r}$ is an orthogonal matrix, $\boldsymbol{\eta} \in \mathbb{R}^{u\times p}$ carries the coordinates of $\boldsymbol{\beta}$ relative to the basis matrix $\boldsymbol{\Gamma}$, $\boldsymbol{\Omega} \in \mathbb{R}^{u\times u}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(r-u)\times(r-u)}$ are positive definite matrices. Assume the envelope dimension $u$ is fixed, and substituting theses quantities in (3.6):

$$\ell = \frac{-nr}{2}\log(2\pi) - \frac{n}{2}\log|\boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T| - \sum_{i=1}^{n}\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x}_i)^T(\boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T)^{-1}(\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x}_i) \tag{3.7}$$

Now, to simplify, (Cook, 2018) introduced the following corollary.

**Corollary 3.2.1.** *Let $\mathcal{R}$ reduce $\boldsymbol{M} \in \mathbb{R}^{r\times r}$, let $\boldsymbol{A} \in \mathbb{R}^{r\times u}$ be a semi-orthogonal basis matrix for $\mathcal{R}$, and let $\boldsymbol{A}_0$ be a semi-orthogonal basis matrix for $\mathcal{R}^{\perp}$. Then*

1. *$\boldsymbol{M}$ and $\boldsymbol{P}_{\mathcal{R}}$, and $\boldsymbol{M}$ and $\boldsymbol{Q}$ commute.*
2. *$\mathcal{R} \subseteq$ span($\boldsymbol{M}$) if and only if $\boldsymbol{A}^T\boldsymbol{M}\boldsymbol{A}$ is full rank.*
3. *$|\boldsymbol{M}| = |\boldsymbol{A}^T\boldsymbol{M}\boldsymbol{A}| \times |\boldsymbol{A}_0^T\boldsymbol{M}\boldsymbol{A}_0|$.*
4. *If $\boldsymbol{M}$ is full rank, then $\boldsymbol{M}^{-1} = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{M}\boldsymbol{A})^{-1}\boldsymbol{A}^T + \boldsymbol{A}_0(\boldsymbol{A}_0^T\boldsymbol{M}\boldsymbol{A}_0)^{-1}\boldsymbol{A}_0^T = \boldsymbol{P}_{\mathcal{R}}\boldsymbol{M}^{-1}\boldsymbol{P}_{\mathcal{R}} + \boldsymbol{Q}\boldsymbol{M}^{-1}\boldsymbol{Q}$.*
5. *If $\mathcal{R} \subseteq$ span($\boldsymbol{M}$), then: $\boldsymbol{M} = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{M}\boldsymbol{A})^{-1}\boldsymbol{A}^T + \boldsymbol{A}_0(\boldsymbol{A}_0^T\boldsymbol{M}\boldsymbol{A}_0)\boldsymbol{A}_0^T$.*

Using (3) from corollary 3.2.1,thus (3.7) becomes: :

$$L_u = -\frac{nr}{2}\log(2\pi) - \frac{n}{2}\log|\boldsymbol{\Omega}| - \frac{n}{2}\log|\boldsymbol{\Omega}_0| - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x}_i)^T(\boldsymbol{\Gamma}\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0^{-1}\boldsymbol{\Gamma}_0^T)(\boldsymbol{y}_i - \boldsymbol{\alpha} - \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x}_i) \tag{3.8}$$

substituting $\hat{\boldsymbol{\alpha}} = \bar{\boldsymbol{y}}$;

$$\begin{aligned} L_u = &-\frac{nr}{2}\log(2\pi) - \frac{n}{2}\log|\boldsymbol{\Omega}| - \frac{n}{2}\log|\boldsymbol{\Omega}_0| \\ &- \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{y}_i - \bar{\boldsymbol{y}} - \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x}_i)^T(\boldsymbol{\Gamma}\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0^{-1}\boldsymbol{\Gamma}_0^T)(\boldsymbol{y}_i - \bar{\boldsymbol{y}} - \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x}_i) \end{aligned} \tag{3.9}$$

Now, decompose $(\boldsymbol{y}_i - \bar{\boldsymbol{y}})$ as $(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) = \boldsymbol{P}_\Gamma(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) + \boldsymbol{Q}_\Gamma(\boldsymbol{y}_i - \bar{\boldsymbol{y}})$, where $\boldsymbol{P}_\Gamma = \boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T, \boldsymbol{Q}_\Gamma = \boldsymbol{I} - \boldsymbol{P}_\Gamma$. Then (3.9) becomes:

$$\begin{aligned} L_u = &-\frac{nr}{2}\log(2\pi) - \frac{n}{2}\log|\boldsymbol{\Omega}| - \frac{n}{2}\log|\boldsymbol{\Omega}_0| \\ &- \frac{1}{2}\sum_{i=1}^{n}\{(\boldsymbol{P}_\Gamma(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) + \boldsymbol{Q}_\Gamma(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) - \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x}_i)^T(\boldsymbol{\Gamma}\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0^{-1}\boldsymbol{\Gamma}_0^T)(\boldsymbol{P}_\Gamma(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) \\ &+ \boldsymbol{Q}_\Gamma(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) - \boldsymbol{\Gamma}\boldsymbol{\eta}\boldsymbol{x}_i)\} \end{aligned} \tag{3.10}$$

after simplification:

$$L_u = -\frac{nr}{2}\log(2\pi) - L_u^{(11)} - L_u^{(12)}, \tag{3.11}$$

where $L_u^{(11)} = \frac{n}{2}\log|\boldsymbol{\Omega}| - \frac{1}{2}\sum_{i=1}^{n}\{\boldsymbol{\Gamma}^T(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) - \boldsymbol{\eta}\boldsymbol{x}_i\}^T\boldsymbol{\Omega}^{-1}\{\boldsymbol{\Gamma}^T(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) - \boldsymbol{\eta}\boldsymbol{x}_i\}$, $L_u^{(12)} = \frac{n}{2}\log|\boldsymbol{\Omega}_0| - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T\boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0^{-1}\boldsymbol{\Gamma}_0(\boldsymbol{y}_i - \bar{\boldsymbol{y}})$.

$L_u^{(11)}$ can be considered as the log-likelihood for multivariate regression of $\boldsymbol{\Gamma}^T(\boldsymbol{y}_i - \bar{\boldsymbol{y}})$ on $\boldsymbol{x}_i$. Hence, $L_u^{(11)}$ is maximised over $\boldsymbol{\eta}$ at $\hat{\boldsymbol{\eta}} = \boldsymbol{\Gamma}^T\boldsymbol{\beta}_{ols}$. Substituting this in $L_u^{(11)}$

$$\begin{aligned} L_u^{(11)} &= -\frac{n}{2}\log|\boldsymbol{\Omega}| - \frac{1}{2}\sum_{i=1}^{n}\{\boldsymbol{\Gamma}^T(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) - \boldsymbol{\Gamma}^T\boldsymbol{\beta}_{ols}\boldsymbol{x}_i\}^T\boldsymbol{\Omega}^{-1}\{\boldsymbol{\Gamma}^T(\boldsymbol{y}_i - \bar{\boldsymbol{y}}) - \boldsymbol{\Gamma}^T\boldsymbol{\beta}_{ols}\boldsymbol{x}_i\} \\ &= -\frac{n}{2}\log|\Omega| - \frac{n}{2}\sum_{i=1}^{n}(\boldsymbol{\Gamma}^T\boldsymbol{r}_i)^T\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{r}_i, \end{aligned} \tag{3.12}$$

where $\boldsymbol{r}_i$ is the $r^{th}$ residual. Now, maximise $L_u^{(11)}$ over $\boldsymbol{\Omega}$, $\hat{\boldsymbol{\Omega}} = \boldsymbol{\Gamma}^T\boldsymbol{S}_{Y|X}\boldsymbol{\Gamma}$. Putting everything together, $L_u^{(11)}$ becomes:

$$L_u^{(11)} = \frac{n}{2}\log|\boldsymbol{\Gamma}^T\boldsymbol{S}_{Y|X}\boldsymbol{\Gamma}| - \frac{nu}{2} \tag{3.13}$$

Similarly, $L_u^{12}$ is maximised over $\boldsymbol{\Omega}_0$ at the value $\hat{\boldsymbol{\Omega}}_0 = \boldsymbol{\Gamma}_0^T \boldsymbol{S}_Y \boldsymbol{\Gamma}_0$, Hence

$$L_u^{(12)} = -\frac{n}{2} \log |\boldsymbol{\Gamma}_0^T \boldsymbol{S}_Y \boldsymbol{\Gamma}_0| - \frac{n(r-u)}{2} \tag{3.14}$$

Since $\log |\boldsymbol{\Gamma}_0^T \boldsymbol{S}_Y \boldsymbol{\Gamma}_0| = \log |\boldsymbol{S}|_Y + \log |\boldsymbol{\Gamma}^T \boldsymbol{S}_Y^{-1} \boldsymbol{\Gamma}|$. Now, substituting in $L_u$ yields:

$$L_u = -\frac{nr}{2}\log(2\pi) - \frac{nr}{2} - \frac{n}{2}\log|\boldsymbol{S}_Y| - \frac{n}{2}\log|\boldsymbol{\Gamma}^T \boldsymbol{S}_{Y|X}\boldsymbol{\Gamma}| - \frac{n}{2}\log|\boldsymbol{\Gamma}^T \boldsymbol{S}_Y^{-1}\boldsymbol{\Gamma}| \tag{3.15}$$

Hence, the maximum likelihood estimate of envelope subspace is obtained by optimising the following objective function over a Grassman manifold:

$$L_u(\boldsymbol{\Gamma}) = \text{span}\left\{ \arg\min_{\Gamma} \left( \ln |\boldsymbol{\Gamma}^T \boldsymbol{S}_{\boldsymbol{Y}|\boldsymbol{X}} \boldsymbol{\Gamma}| + \ln |\boldsymbol{\Gamma}^T (\boldsymbol{S}_{\boldsymbol{Y}})^{-1} \boldsymbol{\Gamma}| \right) \right\}, \tag{3.16}$$

To summarize, the maximum likelihood estimates for parameters given in the response envelope model are given by:

$$\begin{aligned}
\hat{\boldsymbol{\eta}} &= \boldsymbol{\Gamma}^T \boldsymbol{\beta}_{ols} \\
\hat{\boldsymbol{\beta}}_{env} &= \boldsymbol{\Gamma}\boldsymbol{\eta} \\
\hat{\boldsymbol{\Omega}} &= \hat{\boldsymbol{\Gamma}}^T \boldsymbol{S}_{Y|X} \hat{\boldsymbol{\Gamma}} \\
\hat{\boldsymbol{\Omega}}_0 &= \hat{\boldsymbol{\Gamma}}_0^T \boldsymbol{S}_Y \hat{\boldsymbol{\Gamma}}_0 \\
\hat{\boldsymbol{\Sigma}} &= \hat{\boldsymbol{\Gamma}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\Gamma}}^T + \hat{\boldsymbol{\Gamma}}_0\hat{\boldsymbol{\Omega}}_0\hat{\boldsymbol{\Gamma}}_0^T.
\end{aligned}$$

## 3.3 Predictor envelope

In Section 3.2 we discuss the concept of response envelope model; similarly, in this section we discuss the development of predictor envelope model. Predictor envelope was developed by Cook et al. (2013) and aims to increase the efficiency of the model coefficients estimation. To explore the predictor model, consider the regression model:

$$y = \alpha + \boldsymbol{\beta}^T \boldsymbol{x} + e, \tag{3.17}$$

where $y \in \mathbb{R}$ could be a univariate or multivariate response variable; in this context we assume a univariate response, $\boldsymbol{x} \in \mathbb{R}^p$ the predictor variables that is normally distributed with mean $\boldsymbol{\mu}_X$ and variance $\boldsymbol{\Sigma}_X$, $\alpha \in \mathbb{R}$ is the intercept and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the model coefficients. Further, suppose $\mathcal{S}$ is a subspace in $\mathbb{R}^p$ such that, $\boldsymbol{P}_{\mathcal{S}}$ is the projection onto $\mathcal{S}$ and $\boldsymbol{Q}_{\mathcal{S}} = \boldsymbol{I}_p - \boldsymbol{P}_{\mathcal{S}}$ is the projection complement. The predictor envelope decomposes the predictor variables $\boldsymbol{x}$ into material and immaterial, such that $\boldsymbol{P}_{\mathcal{S}}\boldsymbol{x}$ and $\boldsymbol{Q}_{\mathcal{S}}\boldsymbol{x}$ form the material and immaterial parts, respectively. The choice of $\boldsymbol{P}_{\mathcal{S}}$ and $\boldsymbol{Q}_{\mathcal{S}}$ has to satisfy the following conditions:

i. $y$ and $\boldsymbol{Q}_{\mathcal{S}}\boldsymbol{x}$ are uncorrelated, i.e $\mathrm{Cov}(y, \boldsymbol{Q}_{\mathcal{S}}\boldsymbol{x}|\boldsymbol{P}_{\mathcal{S}}\boldsymbol{x}) = 0$, and

ii. $\boldsymbol{P}_{\mathcal{S}}\boldsymbol{x}$ and $\boldsymbol{Q}_{\mathcal{S}}^T\boldsymbol{x}$ are uncorrelated; that is, $\mathrm{Cov}(\boldsymbol{P}_{\mathcal{S}}\boldsymbol{x}, \boldsymbol{Q}_{\mathcal{S}}\boldsymbol{x}) = 0$.

$\mathcal{S}$ that satisfies properties in i and ii is referred to as the reducing subspace of $\boldsymbol{\Sigma}_X$ that contains $span(\boldsymbol{\beta})$, and denoted by $\mathcal{S} = \mathcal{E}_{\Sigma_X}(span(\boldsymbol{\beta}))$, (Cook et al., 2013). Let $u$ be known envelope dimension, and $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{p\times p}$ is an orthogonal matrix such that $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times u}$ is a semi orthogonal basis of $\mathcal{E}_{\Sigma_X}(span(\boldsymbol{\beta}))$, and $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-u)}$ is a semi orthogonal basis of $\mathcal{E}^{\perp}_{\Sigma_X}(span(\boldsymbol{\beta}))$. Thus, model (3.17) is re-parametrised to form the envelope predictor model:

$$y = \alpha + (\boldsymbol{\Gamma}\boldsymbol{\eta})^T\boldsymbol{x} + e \quad \boldsymbol{\Sigma}_X = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T, \tag{3.18}$$

where $\boldsymbol{\beta} = \boldsymbol{\Gamma}\boldsymbol{\eta}$ , $\boldsymbol{\eta} \in \mathbb{R}^u$ carries out the coordinate of $\boldsymbol{\beta}$ with respect to $\boldsymbol{\Gamma}$, the matrices $\boldsymbol{\Omega} \in \mathbb{R}^{u\times u}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(p-u)\times(p-u)}$ are positive definite.

Given the model described earlier, in the following section, we discuss the estimation procedure of the model parameters.

### 3.3.1 Estimation

In this section we will discus the estimation of the parameters involved in model (3.18). The parameters to be estimated are $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0, \boldsymbol{\Omega}, \boldsymbol{\Omega}_0, \boldsymbol{\eta})$, where we assume that the envelope dimension $u$, is known. The estimation of the parameters is based on the maximum likelihood is of the joint distribution of $(\boldsymbol{X}, Y)$ as a product of the conditional distribution $(f(Y|\boldsymbol{x}))$ and the marginal distribution of $\boldsymbol{x}$, that is,

$$f(\boldsymbol{x}, y) = f(Y|\boldsymbol{x})f(\boldsymbol{x}), \tag{3.19}$$

where $f(\boldsymbol{x}) = (2\pi)^{-p/2}|\boldsymbol{\Sigma}_{(\boldsymbol{x})}|^{-1/2}\exp\{\frac{-1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_x)^T\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_x\})$
and $f(y|\boldsymbol{x}) = (2\pi)^{-r/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\{-\frac{1}{2}(y-\boldsymbol{\mu}_{Y|\boldsymbol{x}})^T\boldsymbol{\Sigma}^{-1}(y-\boldsymbol{\mu}_{Y|x})\}$

Thus, the likelihood function is given by:

$$L = (2\pi)^{\frac{n}{2}(r+p)}|\boldsymbol{\Sigma}_X|^{-n/2}|\boldsymbol{\Sigma}_{Y|x}|^{-n/2}\exp\{-\frac{1}{2}\sum_{i=1}^{n}(y_i-\boldsymbol{\beta}^T\boldsymbol{x}_i)^T\boldsymbol{\Sigma}_{Y|x}^{-1}(y_i-\boldsymbol{\beta}^T\boldsymbol{x}_i)\}$$
$$\times\exp\{\frac{-1}{2}\sum_{i=1}^{n}(\boldsymbol{x}-\boldsymbol{\mu}_x)^T\boldsymbol{\Sigma}_x^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_x\}$$

The log-likelihood is given by:

$$\ell = c - \frac{n}{2}\log|\boldsymbol{\Sigma}_X| - \frac{n}{2}\log|\boldsymbol{\Sigma}_{Y|x}| - \frac{1}{2}tr(\boldsymbol{\Sigma}_{Y|x}^{-1}\sum_{i=1}^{n}(y_i-\boldsymbol{\beta}^T\boldsymbol{x}_i)^T(y_i-\boldsymbol{\beta}^T\boldsymbol{x}_i))$$
$$-\frac{1}{2}tr(\boldsymbol{\Sigma}_x^{-1}\sum_{i=1}^{n}(\boldsymbol{x}-\boldsymbol{\mu}_x)^T(\boldsymbol{x}-\boldsymbol{\mu}_x)$$
$$\ell = c - \frac{n}{2}\log|\boldsymbol{\Gamma}^T\boldsymbol{S}_X\boldsymbol{\Gamma}| - \frac{n}{2}\log|\boldsymbol{\Gamma}_0^T\boldsymbol{S}_X\boldsymbol{\Gamma}_0| - \frac{n}{2}\log|\boldsymbol{\Sigma}_{Y|x}| \tag{3.20}$$

where $c$ is a constant. Note:

$$\begin{aligned}\boldsymbol{S}_{Y|X} &= \boldsymbol{S}_Y - \boldsymbol{S}_{XY}^T\boldsymbol{S}_X^{-1}\boldsymbol{S}_{XY}\\ &= \boldsymbol{S}_Y - \boldsymbol{S}_{XY}\boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{S}_X^{-1}\boldsymbol{\Gamma})\boldsymbol{\Gamma}^T\boldsymbol{S}_{XY} \qquad (3.21)\\ \boldsymbol{\Gamma}_0^T\boldsymbol{S}_X^{-1}\boldsymbol{\Gamma}_0 &= |\boldsymbol{S}_X||\boldsymbol{\Gamma}^T\boldsymbol{S}_X^{-1}\boldsymbol{\Gamma}|\\ (\boldsymbol{S}_Y - \boldsymbol{S}_{XY}\boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{S}_X^{-1}\boldsymbol{\Gamma})\boldsymbol{\Gamma}^T\boldsymbol{S}_{XY})||\boldsymbol{\Gamma}^T\boldsymbol{S}_X^{-1}\boldsymbol{\Gamma}| &= \boldsymbol{S}_Y|\boldsymbol{\Gamma}^T(\boldsymbol{S}_X - \boldsymbol{S}_Y^{-1}\boldsymbol{S}_{XY}\boldsymbol{S}_{XY}^T)\boldsymbol{\Gamma}|\end{aligned}$$

Thus, substituting (3.21) in (3.20), gives the following:

$$L_u(\boldsymbol{\Gamma}) = \log|\boldsymbol{\Gamma}^T\boldsymbol{S}_{X|Y}\boldsymbol{\Gamma}| + \log|\boldsymbol{\Gamma}^T\boldsymbol{S}_X^{-1}\boldsymbol{\Gamma}|. \tag{3.22}$$

Thus, the maximum likelihood estimate of the predictor envelope subspace is obtain by optimizing the following objective function over Grassmann manifold:

$$\hat{\boldsymbol{\Gamma}} = \text{span}\Big\{\arg\min_{\Gamma} L_u(\boldsymbol{\Gamma})\Big\}, \tag{3.23}$$

where $L_u(\boldsymbol{\Gamma})$ is given in (3.22). The model parameters are given by:

$$\begin{aligned}\hat{\boldsymbol{\eta}} &= \boldsymbol{\Gamma}^T\boldsymbol{\beta}_{ols}\\ \hat{\boldsymbol{\beta}}_{env} &= \boldsymbol{\Gamma}\boldsymbol{\eta}\\ \hat{\boldsymbol{\Omega}} &= \hat{\boldsymbol{\Gamma}}^T\boldsymbol{S}_X\hat{\boldsymbol{\Gamma}}\\ \hat{\boldsymbol{\Omega}}_0 &= \hat{\boldsymbol{\Gamma}}_0^T\boldsymbol{S}_X\hat{\boldsymbol{\Gamma}}_0\\ \hat{\boldsymbol{\Sigma}}_X &= \hat{\boldsymbol{\Gamma}}\hat{\boldsymbol{\Omega}}\hat{\boldsymbol{\Gamma}}^T + \hat{\boldsymbol{\Gamma}}_0\hat{\boldsymbol{\Omega}}_0\hat{\boldsymbol{\Gamma}}_0^T.\end{aligned}$$

## 3.4 Envelope subspace estimation

In this section, we discuss the algorithm for estimating envelope subspace. The goal of the envelope method is to estimate the basis that employs as reducing subspace. For a given dimension $u$, the envelope subspace can be constructed via optimising a non-convex likelihood-based objective function over Grassmann Manifold, denoted by $\mathcal{G}(d,u)$ where $d$ can be the number of response or predictor variables (Cook (2018), and Cook and Zhang (2016)). The usual practice is to construct the envelope basis via direct optimization over suitable Grassmannian. This technique requires a carefully chosen initial value for $\boldsymbol{\Gamma}$. Further, the optimization is computationally expensive, especially in the case where the required number to specify an element in $\mathcal{G}(d,u)$; $u(d-u)$ is large.

Cook and Zhang (2016) proposed an algorithm that breaks down the Grassmann optimization into a series of one dimensional (1D) optimization. This way is proven to be faster, and the starting value is no longer an issue. Given the dimension of the envelope basis $u$, as well as the positive definite matrices $\boldsymbol{M} > 0$ and $\boldsymbol{U} > 0$, the algorithm estimates one dimension at a time until the desired dimension is obtained. Please note that in sample version of $\boldsymbol{M}$ and $\boldsymbol{U}$ are substituted by $\hat{\boldsymbol{M}}$ and $\hat{\boldsymbol{U}}$. In the case of response envelope, $\hat{\boldsymbol{M}}$ denotes the covariance matrix of the residuals from the ordinary least squares ($\boldsymbol{S}_{Y|X}$), while $\hat{\boldsymbol{M}} + \hat{\boldsymbol{U}}$ is $\boldsymbol{S}_Y$, the marginal sample covariance of $\boldsymbol{Y}$. While for predictor envelope $\hat{\boldsymbol{M}} = \boldsymbol{S}_{X|Y}, \hat{\boldsymbol{M}} + \hat{\boldsymbol{U}} = \boldsymbol{S}_X$. The following proposition explains the concept of the sequential optimization:

**Proposition 3.4.1.** *Assume* $(\boldsymbol{G}, \boldsymbol{G}_0)$ *are an orthogonal basis of* $\mathbb{R}^d$, *such that* $\boldsymbol{G} \in \mathbb{R}^{d\times q}, \boldsymbol{G}_0 \in \mathbb{R}^{d\times(d-q)}$, *and span*$(\boldsymbol{G}) \subseteq \mathcal{E}_M(\mathcal{B})$. *Then* $\boldsymbol{v} \in \mathcal{E}_{G_0^T M G_0}(\boldsymbol{G}_0^T\mathcal{B})$ *implies that* $\boldsymbol{G}_0\boldsymbol{v} \in \mathcal{E}_M(\mathcal{B})$.

In the light of proposition 3.4.1, if we assume that $\boldsymbol{G}$ is a known semi-orthogonal basis for envelope subspace $\mathcal{E}_M(\mathcal{B})$, we can obtain the rest of $\mathcal{E}_M(\mathcal{B})$ via considering $\mathcal{E}_{G_0^T M G_0}(\mathcal{B})$. Further, suppose span$(\boldsymbol{\gamma}_1, ..., \boldsymbol{\gamma}_u) = \mathcal{E}_M(\mathcal{B})$, the algorithm for constructing $\boldsymbol{\gamma}_i, i = 1, ..., u$ is summurized in algorithm 1.

**Algorithm 1** one dimensional optemization

```
1. initiate $\gamma_0 = \boldsymbol{\Gamma} = 0$.
2. For $i = 1, 2, ..., u - 1$
(a) $\boldsymbol{\Gamma}_i = (\boldsymbol{\gamma}_1, ..., \boldsymbol{\gamma}_i)$ if $i \geq 1$, and let $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0)$ be an orthogonal basis for $\mathbb{R}^d$.
(b) Define a stepwise objective function.
        $D_i(\boldsymbol{g}) = \log(\boldsymbol{g}^T \boldsymbol{M}_i \boldsymbol{g}) + \log\{\log(\boldsymbol{g}^T (\boldsymbol{M}_i + \boldsymbol{U}_i)^{-1} \boldsymbol{g})\}, \quad (3.24)$
where $\boldsymbol{M}_i = \boldsymbol{\Gamma}_{0i}^T \boldsymbol{M} \boldsymbol{\Gamma}_{0i}, \boldsymbol{U}_i = \boldsymbol{\Gamma}_{0i}^T \boldsymbol{U} \boldsymbol{\Gamma}_{0i}$ and $\boldsymbol{g} \in \mathbb{R}^{d-i}$.
(c) Solve $\boldsymbol{g}_{i+1} = \arg\min_g D_i(\boldsymbol{g})$ such that $\boldsymbol{g}^T \boldsymbol{g} = 1$.
(d) Define $\boldsymbol{\gamma}_{i+1} = \boldsymbol{G}_{0i} \boldsymbol{g}_{i+1}$ to be the unit length $(i + 1)$ stepwise direction.
```

However, Cook et al. (2016) proposed a new non-Grassmann algorithm that improves the optimization process to estimate the envelope basis. The new algorithm proposed here relies on choosing a starting value effectively and a re-parametrization of $\boldsymbol{\Gamma}$. That is, the commonly used objective function for envelope estimation is

$$L_u(\boldsymbol{\Gamma}) = \ln |\boldsymbol{\Gamma}^T \boldsymbol{M} \boldsymbol{\Gamma}| + \ln |\boldsymbol{\Gamma}^T (\boldsymbol{M} + \boldsymbol{U})^{-1} \boldsymbol{\Gamma}|. \tag{3.25}$$

Under normality assumption, the maximum likelihood estimation of the envelope is given by:

$$\hat{\mathcal{E}} = \text{Span}\{\text{arg min}_\Gamma (L_u(\boldsymbol{\Gamma}))\} \tag{3.26}$$

The objective function given in (3.25) is a non-convex. Therefore, finding a solution that is a global minimum might be challenging. Trying various starting values is an inefficient way and time-consuming. Hence, the choice of starting values is crucial.

Cook et al. (2016) proposed an iterative non-Grassmann method to find the arg min of $L_u(\boldsymbol{\Gamma})$. Their approach relies on selecting the starting value that makes the optimization possible. The authors show how to select a $u$ columns from the eigenvectors of $\hat{\boldsymbol{M}}$ or $\hat{\boldsymbol{M}} + \hat{\boldsymbol{U}}$. The key to selecting which matrix the starting value is chosen out of its eigenvectors can be explained as follows, knowing that :

$$\begin{aligned} \boldsymbol{U} &= \boldsymbol{\Gamma V \Gamma}^T \\ \boldsymbol{M} &= \boldsymbol{\Gamma \Omega \Gamma}^T + \boldsymbol{\Gamma}_0 \boldsymbol{\Omega}_0 \boldsymbol{\Gamma}_0^T \\ (\boldsymbol{M}+\boldsymbol{U})^{-1} &= \boldsymbol{\Gamma}(\boldsymbol{\Omega}+\boldsymbol{V})^{-1}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0 \boldsymbol{\Omega}_0^{-1} \boldsymbol{\Gamma}_0^T \end{aligned} \tag{3.27}$$

where $\boldsymbol{\Gamma} = \arg\min_{\Gamma}(L_u(\boldsymbol{\Gamma}))$, $\boldsymbol{V} \in R^{u\times u}$, is a positive semi-definite matrix. For selecting the starting value from the eigenvectors of $\boldsymbol{M}$, the eigenvalues of $\boldsymbol{\Omega}$ has to be distinguishable from the eigenvalues of $\boldsymbol{\Omega}_0$. If this condition is violated, i.e some eigenvalues of $\boldsymbol{\Omega}$ are closed to the eigenvalues of $\boldsymbol{\Omega}_0$ attempting the minimization of (3.25) will be miss-leading. That is, a vector near span$(\boldsymbol{\Gamma}_0)$ will be chosen instead of picking a vector near span$(\boldsymbol{\Gamma}) = \hat{\mathcal{E}}$, and hence choosing the starting value from $\boldsymbol{M}+\boldsymbol{U}$ is more efficient. Similarly, to choose the starting value from the eigenvectors of $\boldsymbol{M}+\boldsymbol{U}$ requires the eigenvalues of $\boldsymbol{\Omega}+\boldsymbol{V}$ to be well distinguished from the eigenvalues of $\boldsymbol{\Omega}_0$. The starting value may be chosen from the scaled $\hat{\boldsymbol{M}}$ or $\hat{\boldsymbol{M}}+\hat{\boldsymbol{U}}$, as well. In Cook et al. (2016), the authors have shown 4 ways to choose the starting values; from the scaled or unscaled $\hat{\boldsymbol{M}}$ or $\hat{\boldsymbol{M}}+\hat{\boldsymbol{U}}$. The starting value that minimises $L_u(\boldsymbol{\Gamma})$ is used. Once the starting value is selected, the optimisation is carried out based on re-parametrized version of $L_u(\boldsymbol{\Gamma})$ that does not required optimisation over a Grassmannian. That is, (3.25) becomes:

$$L_u(\boldsymbol{A}) = -2\ln|\boldsymbol{C}_A^T\boldsymbol{C}_A| + \ln|\boldsymbol{C}_A^T\hat{\boldsymbol{M}}\boldsymbol{C}_A| + \ln|\boldsymbol{C}_A^T(\hat{\boldsymbol{M}}+\hat{\boldsymbol{U}})^{-1}\boldsymbol{C}_A|, \tag{3.28}$$

this new objective function depends on partitioned the starting value ($G_{\text{start}} \in R^{r\times u}$) as follows:

$$\boldsymbol{G} = \begin{bmatrix} (\boldsymbol{G}_1)_{u\times u} \\ (\boldsymbol{G}_2)_{(r-u)\times u} \end{bmatrix} = \begin{bmatrix} I_u \\ \boldsymbol{A} \end{bmatrix} \boldsymbol{G}_1 = \boldsymbol{C}_A \boldsymbol{G}_1, \tag{3.29}$$

where $\boldsymbol{G}_1$ is non-singular, $\boldsymbol{A} = \boldsymbol{G}_2\boldsymbol{G}_1^{-1} \in R^{(r-u)\times u}$, and $\boldsymbol{C}_A = (\boldsymbol{I}_u, \boldsymbol{A}^T)^T$. This algorithm estimates $\boldsymbol{\Gamma}$ row by row. When $u$ is large, the authors claimed that it shows superiority over the 1D algorithm.

## 3.5 Review of envelope-based methods

In this section, we discuss some of the developed envelope-based methods. Several authors have employed the efficiency of the envelope basis $\boldsymbol{\Gamma}$, to improve the outcome of existing statistical methods.

### 3.5.1 Envelope-based sparse partial least squares

In principle, there is a link connection between the envelopes and the Partial Least Squares (PLS), which are shown in Cook et al. (2013). In their work, the authors showed a connection between PLS and envelope; however, envelope outperforms PLS in prediction and estimation. Zhu and Su (2019) extended this work and proposed a developed version of sparse PLS via the link between PLS and envelope. The connection comes from dividing the predictors into *active*, and *inactive* variables, where active variables refer to the predictor variables whose coefficients are not zero, and inactive variables refer to the predictor variables that have zero coefficients. This classification is reflected in $\mathbf{\Gamma}$ by seeing zero and non-zero rows, where the zero rows correspond to the inactive predictors. Thus, the predictor envelope subspace has the following structure:

$$\Gamma = \begin{bmatrix} \mathbf{\Gamma}_{\mathcal{A}} \\ \mathbf{0} \end{bmatrix} \tag{3.30}$$

where $\mathbf{\Gamma}_{\mathcal{A}}$ is the active predictor variables. The regression coefficient $\boldsymbol{\beta}$ is estimated based on the active predictor and denoted by $\boldsymbol{\beta}_{\mathcal{A}} = \mathbf{\Gamma}_{\mathcal{A}}\boldsymbol{\eta}$. Recall (3.29), the parameterisation of $\boldsymbol{A}$ serves the sparse structure of $\mathbf{\Gamma}$. That is, $\mathbf{\Gamma}$ has zero row if and only if the corresponding row in $\boldsymbol{A}$ is zero. Hence, inactive predictors are identified via sparsity structure of $\boldsymbol{A}$. To make the envelope-based sparse PLS estimator $\boldsymbol{\beta}$ a sparse estimator, the author induce the sparsity in $\boldsymbol{A}$ by adding an adaptive group lasso penalty to the objective function in (3.28) to be as follows:

$$L_u(\boldsymbol{A}) = -2\ln|\boldsymbol{C}_A^T\boldsymbol{C}_A| + \ln|\boldsymbol{C}_A^T\hat{\boldsymbol{M}}\boldsymbol{C}_A| + \ln|\boldsymbol{C}_A^T(\hat{\boldsymbol{M}}+\hat{\boldsymbol{U}})^{-1}\boldsymbol{C}_A| + \lambda\sum_{i=1}^{p-u}\boldsymbol{w}_i||\boldsymbol{a}_i||, \tag{3.31}$$

where $||.||$ is the $\ell_2$ norm of a vector, $\lambda$ is the tuning parameter and $\boldsymbol{w}_i$ is the adaptive weights vector. On the other hand, in high dimensional data $\boldsymbol{S}_{x|y}$ is replace by the sparse permutation invariant covariance (SPICE), $\boldsymbol{S}_{x|y,spice}$. Then the optimisation is carried out over (3.29) and (3.31) to find $\hat{\boldsymbol{A}}$. Once $\hat{\boldsymbol{A}}$ is determined thus the estimate of the regression parameter vector $\hat{\boldsymbol{\beta}}$ under the envelope based sparse PLS is given by:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \hat{\mathbf{\Gamma}}(\hat{\mathbf{\Gamma}}^T\boldsymbol{S}_X\hat{\mathbf{\Gamma}})^{-1}\hat{\mathbf{\Gamma}}\boldsymbol{S}_{XY}, \\ &= \boldsymbol{P}_{\hat{\mathbf{\Gamma}}(\boldsymbol{S}_X)}\hat{\boldsymbol{\beta}_{\text{ols}}}.\end{aligned} \tag{3.32}$$

### 3.5.2 Envelope quantile regression

Ding et al. (2019) adapted envelope methodology to Quantile Regression (QR) and proposed Envelope Quantile Regression (EQR) to improve the efficiency of the standard QR. This method builds the estimation inference on the material part only via estimating the envelope subspace. However, this work varies from the previous work in the point that the inference and the estimation were derived based on the Generalised Method of Moment (GMM). In contrast, previous work was developed based on the maximum likelihood principle. To illustrate the technique in estimating the envelope subspace in EQR, suppose:

$$Q_Y(\tau|\boldsymbol{X}=\boldsymbol{x}) = \mu_\tau + \boldsymbol{\beta}_\tau^T\boldsymbol{x}, \tag{3.33}$$

is the linear quantile regression which describes the relation between the $\tau^{\text{th}}$- conditional quantile of the univariate response $y$ and the predictor vector $\boldsymbol{x} \in \mathbb{R}^p$; where $Q_Y(\tau|\boldsymbol{X} = \boldsymbol{x}) = \inf\{y : F_Y(y|\boldsymbol{X} = \boldsymbol{x}) \geq \tau\}, \quad 0 < \tau < 1,\ F_Y(y|\boldsymbol{X} = \boldsymbol{x}) = P(Y \leq y|\boldsymbol{X} = \boldsymbol{x})$ is the cumulative distribution function of $Y$, $\mu_\tau$ is the intercept and $\boldsymbol{\beta}_\tau \in \mathbb{R}^p$ is the coefficients vector. Under model (3.33) the estimates of the model parameters $(\hat{\mu}_\tau, \hat{\boldsymbol{\beta}}_\tau)$ is

$$(\hat{\mu}_\tau, \hat{\boldsymbol{\beta}}_\tau) = \arg \min_{\mu_\tau\in\mathbb{R}, \boldsymbol{\beta}_\tau\in\mathbb{R}^p} \sum_{i=1}^{n} \rho_\tau(y_i - \mu_i - \boldsymbol{\beta}_\tau^T\boldsymbol{x}_i), \quad i = 1, ..., n, \tag{3.34}$$

where $\rho_\tau(z) = z[\tau - I(z < 0)]$. Now, under the EQR the parameters to be estimated are $(\mu_\tau, \mathcal{E}_{\Sigma_X}(\boldsymbol{\beta}_\tau), \mathrm{vech}(\Omega_\tau^T), \boldsymbol{\eta}_\tau^T, \mathrm{vech}(\Omega_{0\tau}^T))$. Ding et al. (2019) introduced a distribution free objective function that depends on GMM:

$$\hat{\boldsymbol{\theta}} = \mathrm{argmin}_{\boldsymbol{\theta}} h(\boldsymbol{\theta})^T\hat{\boldsymbol{\Delta}}h(\boldsymbol{\theta}), \tag{3.35}$$

where $\hat{\boldsymbol{\theta}} = (\hat{\mu}_\tau, \hat{\mathcal{E}}_{\Sigma_X}(\boldsymbol{\beta}_\tau), \mathrm{vech}(\hat{\boldsymbol{\Omega}}_\tau^T), \hat{\boldsymbol{\eta}}_\tau^T, \mathrm{vech}(\hat{\boldsymbol{\Omega}}_{0\tau}^T))$ is the parameters vector, $\hat{\boldsymbol{\Delta}}$ is chosen to be $\sqrt{n}$-consistent estimator of $[\mathbb{E}(hh^T)]^{-1}$, where $h$ is defined as follows:

$$h = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{n}\sum_{i=1}^{n}(1, \boldsymbol{x}_i^T)^T[I(Y_i < \mu_\tau + \boldsymbol{\eta}_\tau^T\boldsymbol{\Gamma}_\tau\boldsymbol{x}_i) - \tau] \\ \mathrm{vech}(\boldsymbol{\Gamma}_\tau\boldsymbol{\Omega}_\tau\boldsymbol{\Gamma}_\tau^T + \boldsymbol{\Gamma}_{0\tau}\boldsymbol{\Omega}_{0\tau}\boldsymbol{\Gamma}_{0\tau}^T) - \mathrm{vech}(\boldsymbol{S}_X) \\ \boldsymbol{\mu}_X - \bar{\boldsymbol{X}} \end{bmatrix},$$

where vech denotes the vector-half operator that stretches the lower triangle of a symmetric matrix into a vector. The optimisation of (3.35) is done using to Nelder-mead method over the parameters vector $\boldsymbol{\theta}$. Once the envelope subspace $\mathcal{E}_{\Sigma_X}(\boldsymbol{\beta})_\tau$ is estimated, then $\boldsymbol{\Gamma}_\tau = \mathrm{span}(\mathcal{E}_{\Sigma_X}(\boldsymbol{\beta}_\tau))$. Hence, the EQR parameters $\hat{\boldsymbol{\beta}}_\tau = \boldsymbol{\Gamma}_\tau^T\boldsymbol{\eta}_\tau$. The authors argued that asymptotically the estimates found via EQR are at least as efficient as the standard QR estimators.

### 3.5.3 Non-linear envelope

Zhang et al. (2020) proposed methods to tackle the response reduction when there is a nonlinearity relationship between $\boldsymbol{Y}$ and $\boldsymbol{X}$, and heterogeneity in the conditional covariance. The authors proposed a Central Mean Envelope (CME), which detects the heterogeneity on the conditional variance ($\boldsymbol{\Sigma}_{Y|X}$), and Martingale Difference Divergence Envelope (MDDE), which captures the nonlinearity in the conditional mean $E(\boldsymbol{Y}|\boldsymbol{X})$. The CME method differs from the standard envelope in that it focuses on the conditional mean. In contrast, the standard envelope focuses on the characteristics of the conditional distribution of $\boldsymbol{Y}|\boldsymbol{X}$ when performing the reduction. The martingale difference divergence matrix is applied to measure the dependency in the mean.

In CME the interest is in the conditional mean function $E(\boldsymbol{Y}|\boldsymbol{X})$, that is, the goal is to find a subspace $\mathcal{S}$ such that $E(\boldsymbol{Y}|\boldsymbol{X}) = E(\boldsymbol{P}_{\mathcal{S}}\boldsymbol{Y}|\boldsymbol{X}) + E(\boldsymbol{Q}_{\mathcal{S}}Y|X) = E(P_{\mathcal{S}}\boldsymbol{Y}|\boldsymbol{X}) + E(\boldsymbol{Q}_{\mathcal{S}}\boldsymbol{Y})$ and denoted by $\mathcal{E}_{E(\boldsymbol{Y}|\boldsymbol{X})}$. The following proposition facilities the definition of CME.

**Proposition 3.5.1.** *The CME of* $\boldsymbol{Y}$ *on* $\boldsymbol{X}$ *reduces* $\boldsymbol{\Sigma}_X$. *Moreover,* $\mathcal{E}_{E(Y|X)} = \sum_x(\boldsymbol{M}_{Y|X})$. *Where* $(\boldsymbol{M}_{Y|X})$ *is the martingale difference divergence matrix (MDDM) and defined as:*

$$\boldsymbol{M}_{Y|X} = MDDM(\boldsymbol{Y}|\boldsymbol{X}) = -E\left[\{\boldsymbol{Y} - E(\boldsymbol{Y})\}\{\boldsymbol{Y'} - E(\boldsymbol{Y'})\}^T||\boldsymbol{X} - \boldsymbol{X'}||\right], \tag{3.36}$$

where $(\boldsymbol{Y'}, \boldsymbol{X'})$ is an independent copy of $(\boldsymbol{X}, \boldsymbol{Y})$. Direct estimation of CME as the sum of subspace $\sum_x(\boldsymbol{M}_{Y|X})$ is difficult. Hence, MDDE is introduced to facilitate the CME estimation. MDDE is a portion of CME which defined based on the expectation of the conditional covariance $\boldsymbol{\Sigma} = E\{\text{cov}(\boldsymbol{Y}|\boldsymbol{X})\}$ as follows:

**Definition 3.5.1.** The martingale difference divergence envelope of $\boldsymbol{Y} \in \mathbb{R}^r$ on $\boldsymbol{X} \in \mathbb{R}^p$, denoted as $\mathcal{E}_{\Sigma}(\boldsymbol{M}_{Y|X})$, is the intersection of all the reducing subspaces of $\boldsymbol{\Sigma} = E\{\text{cov}(\boldsymbol{Y}|\boldsymbol{X})\}$ that contain $\text{span}(\boldsymbol{M}_{Y|X}) = \text{span}\,[\text{cov}\{E(\boldsymbol{Y}|\boldsymbol{X})\}]$.

Now, since it is challenging to find $\mathbf{\Sigma} = E\{\text{cov}(\boldsymbol{Y}|\boldsymbol{X})\}$ in the nonlinear regression and because $\mathbf{\Sigma}_Y = \text{cov}(\boldsymbol{Y}) = \text{cov}\{E(\boldsymbol{Y}|\boldsymbol{X})\} + E\{\text{cov}(\boldsymbol{Y}|\boldsymbol{X})\} = \text{cov}\{E(\boldsymbol{Y}|\boldsymbol{X})\} + \mathbf{\Sigma}$ the following proposition shows that the marginal covariance $\mathbf{\Sigma}_Y$ instead of $\mathbf{\Sigma}$ in the MDDE.

**Proposition 3.5.2.** *The martingale difference divergence envelope* $\mathcal{E}_{\Sigma}(\boldsymbol{M}_{Y|X}) = \mathcal{E}_{\Sigma_Y}(\boldsymbol{M}_{Y|X})$ *and is the intersection of all* $\mathcal{S} \subseteq \mathbb{R}^r$ *such that:*
*(i)*$E(\boldsymbol{QY}|\boldsymbol{X}) = E(\boldsymbol{QY})$*, and*
*(ii)*$cov(\boldsymbol{QY}, \boldsymbol{PY}) = 0$.

Given the envelope dimension $u_1 = dim\left\{\mathcal{E}_{\Sigma_Y}(\boldsymbol{M}_{Y|X})\right\}$, on one hand, the MDDE=span($\hat{\boldsymbol{G}}$) can be estimated by optimising the following objective function:

$$\hat{\boldsymbol{G}} = \arg \min_{\boldsymbol{G}^T\boldsymbol{G}=\boldsymbol{I}_{u_1}} \log|\boldsymbol{G}^T(\hat{\mathbf{\Sigma}}_Y + \hat{\boldsymbol{M}}_{Y|X})^{-1}\boldsymbol{G}| + \log|\hat{\boldsymbol{G}}\hat{\mathbf{\Sigma}}_Y\boldsymbol{G}|, \tag{3.37}$$

where $\hat{\boldsymbol{M}}_{Y|X}$ is the sample MDDM and $\hat{\mathbf{\Sigma}}_Y$ is the sample covariance of $\boldsymbol{Y}$.

On the other hand, since the CME is defined as $\mathcal{E}_{E(Y|X)} = \sum_x \mathcal{E}_{\mathbf{\Sigma}_X}(M_{Y|X})$, the estimation process is performed by slicing $\boldsymbol{X}$ into $H$ slices and approximate $\mathbf{\Sigma}_X$ by a finite number of covariance matrices $\mathbf{\Sigma}_h$, $h = 1, ...H, H \geq 2$. Each of the covariance matrices $\mathbf{\Sigma}_h$ represents the conditional covariance matrix for each slice or cluster $\mathbf{\Sigma}_h = \text{cov}(Y|X \in \mathcal{R}_h)$, where $\mathcal{R}_1, ..., \mathcal{R}_H$ is the partition of the support of $\boldsymbol{X}$. For a univariate $\boldsymbol{X}$ it is partitioned into $H$ non-overlapping slices similar to sliced inverse regression. For multivariate $\boldsymbol{X}$, $H$ clusters are constructed similar to the idea of K-mean inverse regression. If normality is assumed for each slice $\mathcal{R}_h$, that is, $(\boldsymbol{Y}|\boldsymbol{X} \in \mathcal{R}_h) \sim N(\boldsymbol{\mu}_h, \mathbf{\Sigma}_h)$. Then CME= span($\hat{\mathbf{\Gamma}}$) becomes the smallest subspace reduces all $\mathbf{\Sigma}_h$ and contains the mean subspace span($\boldsymbol{\mu}_1 - E(\boldsymbol{X}), ..., \boldsymbol{\mu}_H - E(\boldsymbol{X})$). Hence, is estimated via optimising the likelihood-based objective function:

$$\hat{\mathbf{\Gamma}} = \arg \min_{\Gamma^T\Gamma=I_u} \log|\mathbf{\Gamma}^T\hat{\mathbf{\Sigma}}_Y^{-1}\mathbf{\Gamma}| + \sum_{h=1}^{H} \frac{n_h}{n} \log|\mathbf{\Gamma}^T\hat{\mathbf{\Sigma}}_h\mathbf{\Gamma}|. \tag{3.38}$$

# Chapter 4

# Support Vector Machine

## 4.1 Introduction

In Chapter 3, we define the envelope-based techniques in the regression framework, where the response is a continuous variable (a scalar or a vector). In contrast, consider classification and discriminate analysis setting where the response $y$ is a categorical variable. Each element (category) $y_i$ indicates the class where the observation belongs. Hence, the classification task is to predict to which class each object belongs with a low misclassification rate. The successful process of such data requires a classifier of high classification accuracy. Support Vector Machines (SVM) is one of the widely used classifiers. Since its first proposal by Cortes and Vapnik (1995), wide range of SVM-based researches developed in machine/statistical learning. Numerous researchers have studied SVM in linear and non-linear cases. Among others, (Suykens and Vandewalle (1999), John Lu (2010), James et al. (2013), Li et al. (2011)). In this chapter, we discuss in more detail the classic SVM classifier. The cornerstone in performing SVM is constructing the hyperplane. That is, the process of constructing a linear decision boundary that separates the data under consideration into distinguished groups as possible. We review the hyperplane construction as well as other SVM related terminologies.

In this chapter, we discuss support vector machines in classification. In Section 4.2 we give the geometrical illustration of the concept of the separating hyperplane. Section 4.3 we investigate the linear SVM for separable and non-separable data. In Section 4.4 we discuss the classification of non-linear data, in which we define the kernel functions and demonstrate some of the commonly used kernels. Finally in Section 4.5 we discuss the multi-class classification. Please note that, the content of this chapter was based on the content of the following Gareth et al. (2013), Abe (2005), Hastie et al. (2009), and Christmann and Steinwart (2008).

## 4.2 Hyperplane

Generally, the hyperplane in a $p$-dimensional space is a $(p-1)$ flat subspace. For instance, the hyperplane is a line in two-dimensional space, while in three-dimensional space, the hyperplane is a flat plane (two-dimensional subspace). From a mathematical point of view, the hyperplane in a $p$-dimensional space is given by:

$$b + w_1 x_1 + ... + w_p x_p = c, \tag{4.1}$$

where $c$ is a constant, $b$ is the offset of the decision boundary from the origin, and $w_1, ..., w_p$ are the weights. The value of $c$ in (4.1) determines where $\boldsymbol{x}$ lies. That is:

$$\begin{array}{lll} c = 0 & \boldsymbol{x} & \text{lies on the hyperplane,} \\ c > 0 & \boldsymbol{x} & \text{lies on one side of the hyperplane,} \\ c < 0 & \boldsymbol{x} & \text{lies on the other side of the hyperplane.} \end{array}$$

It can be seen that the location of $\boldsymbol{x}$ with respect to the hyperplane is affected by the value of $c$. According to the value of $c$ associated with each data point, some points lie on the hyperplane, some lie above, and others lie below. Hence, the hyperplane can be viewed as a separating decision boundary. For instance, if we have two groups of data points in two-dimensional space, the separating hyperplane is defined as a line drawn to create the widest space possible that separates one group of data (say class one) from the other. The optimal separating hyperplane is the one that maximizes the distance between the nearest point from each class and the hyperplane. This distance is known as the *margin*. Maximizing the margin decreases the chance of misclassification. To illustrate the idea of finding the optimal hyperplane suppose we have $n$ data points $(\boldsymbol{x}_i, y_i)$, $i = 1, ..., n$. Further, suppose the data of interest come from two classes that are linearly separable datasets; where $\boldsymbol{x}_i$ is a $p$-dimensional input and $y_i$ is defined as follows

$$y_i = \begin{cases} 1, & \text{if} \quad \boldsymbol{x}_i \in \text{class } 1 \\ -1, & \text{if} \quad \boldsymbol{x}_i \in \text{class } 2 \end{cases} \tag{4.2}$$

There are infinitely many separating hyperplanes that can be found between the two classes. However, the optimal hyperplane is the one that has the widest margin, see Figure 4.1. Let the distance from the closest point from each class to the hyperplane is $M$. That is, the margin on each side is equal, and $M$ units are far away from the hyperplane. Hence, the optimal hyperplane seeks to solve the following optimization:

$$\begin{aligned} L = &\max_{b,\boldsymbol{w},||\boldsymbol{w}||} M \\ \text{such that} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}+b) \geq M, \end{aligned} \tag{4.3}$$

where $||.||$ is the $\ell_2$ norm. The constraint in (4.3) is to assure that the closest data point is at least $M$ units far away from the decision boundary. Hence, the larger $M$ is, the better. The alternative way of writing (4.3) is as follows:

$$\frac{1}{||\boldsymbol{w}||} y_i(\boldsymbol{w}^T\boldsymbol{x}+b) \geq M, \tag{4.4}$$

or

$$y_i(\boldsymbol{w}^T\boldsymbol{x}+b) \geq ||\boldsymbol{w}||M. \tag{4.5}$$

We will use (4.5). For any $(\boldsymbol{w}, b)$ satisfying these inequalities, any positively scalar multiplication satisfies them too. Set $||\boldsymbol{w}|| = \frac{1}{M}$, i.e minimizing $||\boldsymbol{w}||$ indicates maximizing $M$ thus (4.5) becomes:

$$\begin{aligned} L = &\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2 \\ \text{such that} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}+b) \geq 1, \quad i = 1, ..., n. \end{aligned} \tag{4.6}$$

The constraint in (4.6) define a margin around the decision boundary of width $\frac{1}{||\boldsymbol{w}||}$. Hence, when solving this optimization, one seeks $(b, \boldsymbol{w})$ that maximizes the width. (4.6) is a convex (quadratic) optimization problem with linear constraints. A *Lagrangian* method is used to find the minimum of the objective function. Thus, we introduce a Lagrangian multiplier $(\alpha)$ to the objective function $L$ in (4.6). The modified objective function is given by:

$$L = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{n} \alpha_i[y_i(\boldsymbol{w}^T\boldsymbol{x}+b) - 1]. \tag{4.7}$$

The hyperplane is determined by the values of $(\boldsymbol{w}, b)$; however, the solution has to satisfy Karush Kuhn Tucker (KKT) conditions (Fletcher, 1987) (given in the appendix). That is,

$$\frac{\partial L(\boldsymbol{w}, b, \alpha)}{\partial w} = 0,$$
$$\frac{\partial L(\boldsymbol{w}, b, \alpha)}{\partial b} = 0.$$

Thus, finding the derivatives and equating it to zero yields:

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i, \tag{4.8}$$
$$0 = \sum y_i \alpha_i, \tag{4.9}$$

This gives the solution to the primal form of optimization. Hence, we need to solve the dual part of the original optimization, i.e., to solve for $\alpha_i$. Substituting (4.8) and (4.9) in (4.7) we get Wolfe dual that is given by:

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j, \tag{4.10}$$
$$\text{subject to} \quad 0 \leq \alpha \leq c, \quad \sum y_i \alpha_i = 0. \tag{4.11}$$

To find the values of $\boldsymbol{\alpha}$, we maximize (4.10).

For new test data $\boldsymbol{x}^*$, the decision function is given by:

$$D(\boldsymbol{x}^*) = sign(\boldsymbol{w}^T \boldsymbol{x}^* + b), \tag{4.12}$$

hence, $\boldsymbol{x}^*$ is located in class 1 or class 2 based on the sign of $D(\boldsymbol{x}^*)$. If $D(\boldsymbol{x}^*) > 0$ it belongs to class 1 and if $D(\boldsymbol{x}^*) < 0$ it belongs to class 2.

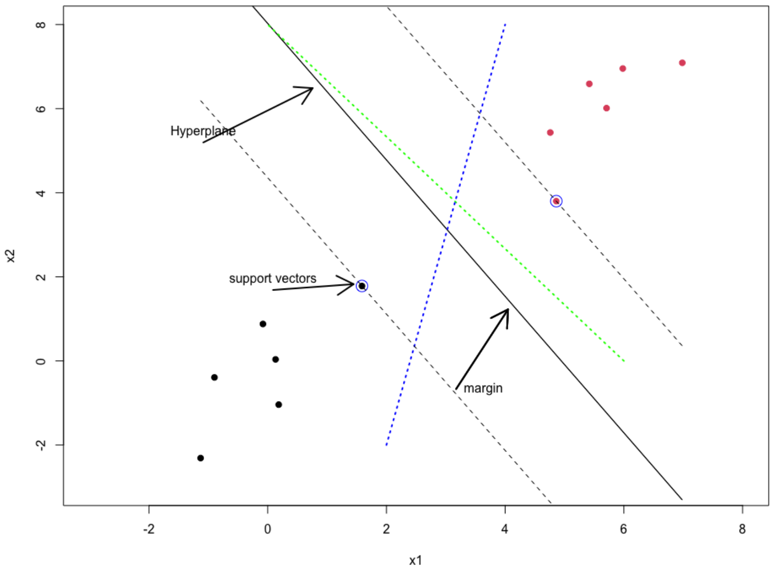


Figure 4.1: The optimal separating hyperplane (the solid black line) for separable data. The dotted green and blue lines are possible hyperplane.

## 4.3 Linear support vector machines

Suppose we have a sample of $n = 12$ observations divided into two classes as in Figure 4.1. In this example, the two classes are well separated and represented by two colors. A linearly separating *decision boundary* is drawned to produce homogeneity between the classes. In fact, there are countless possible hyperplanes; however, the one drawn with a view to maximize the distance that separates the classes is the optimal one. In SVM language, the nearest points to the decision boundary are known as *support vectors*. Through the support vectors, one seeks two parallel lines that maximize the perpendicular distance from the decision boundary (margin) (Berk et al., 2008).

Considering Figure 4.2, it is clear that some of the data points from both classes are misclassified. Practically, such a dataset is more realistic, and it is common to have points cross their margin (misclassification). When constructing the optimal hyperplane, one needs to be more flexible in allowing misclassification. In such a case, a small positive number indicates how far the point crosses the margin, denoted by a *slack* variable, is introduced to construct the optimal hyperplane.

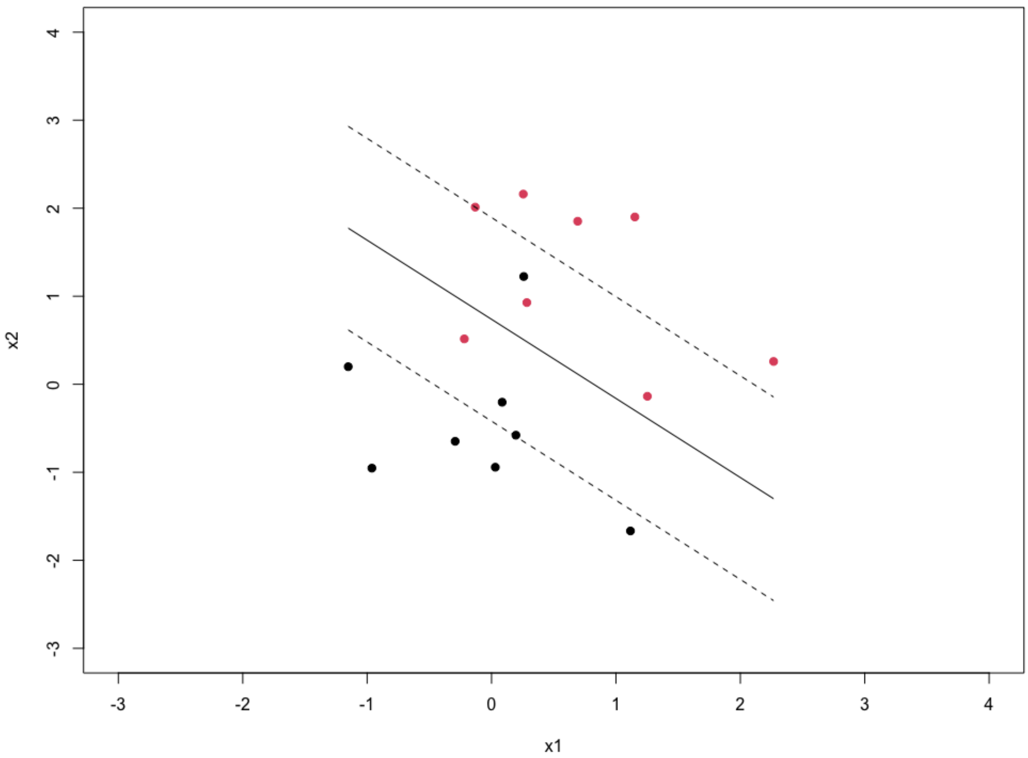


Figure 4.2: The optimal separating hyperplane (the solid black line) for non-separable data.

Figures 4.1, and 4.2 demonstrate the two types of linear SVM. Both are denoted by linearly separable data. However, the theory for constructing the separating hyperplane differs between them. The former case where the data is well separated and no data point is allowed to cross its margin is theoretically known as *hard margin.* In contrast, the later case where this condition is relaxed is referred to as *soft margin.* In the next section, we explain both cases.

### 4.3.1 Hard margin SVM

Suppose we have $n$ data points $(\boldsymbol{x}_i, y_i)$, $i = 1, ..., n$. Further, suppose the data of interest come from two linearly separable classes, where $\boldsymbol{x}_i$ is a $p$-dimensional input, and $y_i$ is defined as in (4.2). The optimal hyperplane is obtained by solving the objective function given by (4.6); however, the process of constructing the optimal separating hyperplane was explained in section 4.2. The dual optimization given by (4.10) is known as *hard* *margin support vector machines.* If the data are linearly separable and the solution exists that implies the optimal solutions for $\boldsymbol{\alpha} \in \mathbb{R}^n$ exists. The solution of dual parameter $\alpha_i, i = 1, ..., n$ either $\alpha_i = 0$ or $\alpha_i \neq 0$. The data points $\boldsymbol{x}_i$ whose dual parameter $\alpha_i \neq 0$ are support vectors.

The decision function is given by:

$$D(\boldsymbol{x}) = sign(\boldsymbol{w}^T \boldsymbol{x} + b). \tag{4.13}$$

If the data point was classified correctly then $y_i D(\boldsymbol{x}_i) > 0$.

### 4.3.2 Soft margin SVM

Practically, it is common to have points that cross their margin (misclassification). This is the case where the data points are non-linearly separable 4.2. In SVM, some of the data points are *allowed* to cross the margin. That means $y_i(\boldsymbol{w^T x}_i + b) \geq 1$ is not satisfied for these points. A *slack* variable is introduced in such a case. The slack variable is a small positive number indicating how far the point crosses its margin. Hence, the only difference between this case and the former is that the optimization function in (4.6) becomes:

$$\begin{aligned} argmin_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T \boldsymbol{w} + \gamma \sum_{i=1}^{n} \xi_i, \\ \text{subject to } y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq & 1 - \xi_i, \\ \sum_{i=1}^{n} \xi_i \geq & 0. \end{aligned} \tag{4.14}$$

where $\gamma \sum_{i=1}^{n} \xi_i$ added to account for the points that violate the margin. Similar to the linearly separable case, we convert this constrained problem to unconstrained and introduce Lagrangian multipliers $(\alpha, \lambda)$ such that:

$$L(\boldsymbol{w}, b, \xi, \alpha, \lambda) = \frac{1}{2}\boldsymbol{w}^T \boldsymbol{w} + \gamma \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n} \lambda_i \xi_i, \tag{4.15}$$

Similarly,

$$\frac{\partial L(\boldsymbol{w}, b, \xi, \alpha, \lambda)}{\partial w} = 0$$
$$\frac{\partial L(\boldsymbol{w}, b, \xi, \alpha, \lambda)}{\partial b} = 0$$
$$\frac{\partial L(\boldsymbol{w}, b, \xi, \alpha, \lambda)}{\partial \xi} = 0$$

That yields:

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i \tag{4.16}$$

subject to

$$\sum y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq \gamma$$

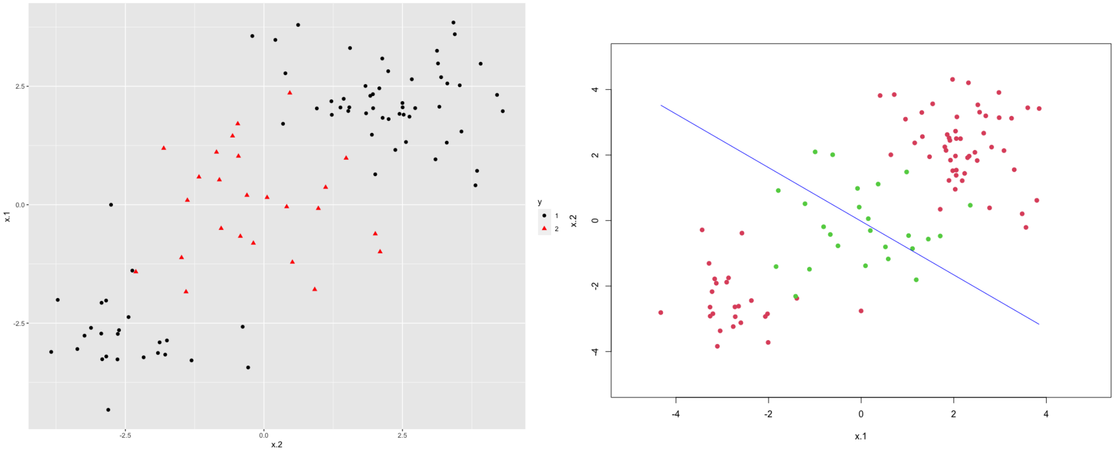
$$\gamma - \alpha_i - \lambda_i = 0$$

The scenario demonstrated here is known as *soft margin* SVM. The decision boundary is given by:

$$D(x) = sign(\boldsymbol{w}^T \boldsymbol{x} + b). \tag{4.17}$$

## 4.4 Non linear support vector machine

In this section, we investigate the case where a strong non-linear relationship is present. For this scenario, the linear statistical methods become not applicable. Hence, there is a need to develop methods that overcome this limitation. These methods should effectively capture the useful information and consider the non-linearity in such a dataset. In Section 4.3 we discussed the different types of linear SVM and demonstrated how to construct the separating hyperplane in each case. Naturally, the SVM classifier is designed to deal with linear data; however, one may encounter non-linear data in practice. For instance, consider the data shown in Figure 4.3 a; the figure shows the scatter plot of two-dimensional data with binary class. It can be clearly seen that the data are non-linearly separable. Hence, applying linear SVM to such data will perform poorly, as shown in Figure 4.3 b. In SVM literature, the technique to handle this problem is to linearize the data via mapping it to a higher dimensional space known as *feature space* (Abe, 2005). If the data under consideration is not linearly separable, one possible solution is to transform the data to a higher dimension where a hyperplane becomes possible to construct. The data is transformed via *kernels* to a subspace referred to as *reproducing kernel Hilbert space* (RKHS), where the inner dot product is possible. In the following section, we introduce the RKHS then we define kernels.



(a)

(b)

Figure 4.3: a: scatter of non-linear data. b decision boundary based on linear SVM.

### 4.4.1 Kernels and Reproducing Kernel Hilbert Space

The idea in kernel-based methods is to embed the data into an RKHS (with a feature map) and then perform the linear techniques on the embedded data. These models are derived from a direct connection between a Reproducing Kernel Hilbert Space (RKHS) and the corresponding feature space representation where the input data are mapped. That is, suppose we have a set of an input data $\{\boldsymbol{x}_i\}_{i=1}^{n} \in \mathbb{R}^p$. Further, suppose $\Phi$ is mapping function such that $\Phi : \mathbb{R}^p \to \mathcal{F}$, where $\Phi(\boldsymbol{x})$ is $d \times n$ a random transformation function in the RKHS that maps $\boldsymbol{X}$ into the d-dimensional feature space, $d > p$.

The following definition given in Steinwart and Christmann (2008) facilitates the meaning of three terminologies that will be used throughout this section. Namely, *kernel, feature space, and feature map.*

**Definition 4.4.1.** Suppose $\mathcal{X}$ is a non-empty set. The the function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called ***kernel*** on $\mathcal{X}$ if there is exists a Hilbert space $\mathcal{F}$ and a map $\Phi : \mathcal{X} \to \mathcal{F}$ such that $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$ we have

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle,$$

where $\Phi$ is referred to as ***feature map*** and $\mathcal{F}$ is a ***feature space***.

The main objective of mapping the input data to a feature space where the dot product is possible. The mapping $\Phi : \boldsymbol{X} \to \mathcal{F}$ allows SVM to construct the decision boundary for the non-linear data. In this process SVM application requirement is to compute the inner products $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{y}) \rangle$ instead of finding $\Phi$ explicitly. The kernels have to be symmetric positive definite, and for any kernel, there exists a unique reproducing kernel Hilbert space (RKHS) that will be established by ***Mercer's theorem***. The following proposition states Mercer's theorem (Steinwart and Christmann, 2008).

**Proposition 4.4.1.** *(**Mercer's theorem**) Let $\mathcal{X}$ be a compact metric space and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a continuous kernel. Further, let $\mu$ be a finite Borel measure; then, for $(e_i)_{i \in I}$ and $(\lambda_i)_{i \in I}$ we have*

$$k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in I} \lambda_i e_i(\boldsymbol{x}) e_i(\boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}.$$

*where the convergence is absolute and uniform.*

As it can be seen, the non-linear SVM relies on reproducing kernel Hilbert space theory. That is, because of the association between the positive definite (or semi-definite) kernels and the RKHS, SVM is able to construct the decision boundary for the non-linear data via kernel trick. The following proposition given in Berg et al. (1984) will define RKHS then we will explain the kernel trick.

**Proposition 4.4.2.** *(**Reproducing kernel Hilbert space** ) Let $\boldsymbol{X}$ be an input data and $k(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X}$ be a positive semidefinite kernel. Further, suppose $\mathcal{F}_0$ is a space spanned by the functions $\{k_{\boldsymbol{x}} | \boldsymbol{x} \in \boldsymbol{X}\}$ where*

$$k_{\boldsymbol{y}} = k(\boldsymbol{x}, \boldsymbol{y}).$$

*Then there exist a Hilbert space $\mathcal{F}$, that is a complete space of $\mathcal{F}_0$, and mapping from $\boldsymbol{X}$ to $\mathcal{F}$ such that*

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle k_{\boldsymbol{x}}, k_{\boldsymbol{y}} \rangle.$$

The determination of the appropriate kernel is a challenging task. The kernel trick is the name of the process of transforming the inseparable data into separable one via special functions known as *kernels* (also known as variance function). In this section, we will demonstrate the commonly used kernels in literature.

From a statistical point of view, kernels can be classified into three classes: stationary, non-stationary, and Locally Stationary Kernels (Genton, 2001). Each of which has its properties and spectral representation. In the following, we will give a summary of the kernel classes:

- **Stationary kernels:** a stationary kernel implies that the value depends on the difference between the two objects, not on the objects themselves; that is, stationary kernels have the following form:

  $$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = K(\boldsymbol{x}_1 - \boldsymbol{x}_2). \tag{4.18}$$

  This class includes a wide range of commonly used kernels, Circular, Spherical, Rational quadratic, Exponential, Gaussian, and Wave kernels.

- **Non-stationary kernels:** In contrast to the stationary class, the predicted value in non-stationary kernels depends on the objects. The polynomial kernel is an example of a non-stationary kernel.

- **Locally stationary kernels:** This class of kernels is of the following form:

  $$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = K_1(\frac{\boldsymbol{x}_1 + \boldsymbol{x}_2}{2})K_2(\boldsymbol{x}_1 - \boldsymbol{x}_2), \tag{4.19}$$

  where $K_1$ and $K_2$ are nonnegative functions and stationary kernels, respectively. Locally stationary kernels include some sceptical cases; from (4.19) and (4.18), we can see that the stationary kernels class is a special case of the local stationary class if $K_1 = 1$. The other case is inherited from the exponentially convex kernels, that is

  $$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = K_1(\boldsymbol{x}_1 + \boldsymbol{x}_2). \tag{4.20}$$

  Hence, since the product of two kernels is a kernel, a vast class of locally stationary kernels can be introduced via the multiplication of stationary kernel and exponentially convex kernel.

In non-linear SVM, a wide range of kernel functions can be used. We will illustrate the most frequent kernels in SVM literature.

- ***Linear kernel*:** This is the simplest type of kernels. It is used when the data is linearly separable and there is no need for mapping to a higher dimensional feature. The linear kernel has the following form:

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^T \boldsymbol{x}_2. \tag{4.21}$$

- ***Radial Basis Function Kernels*:** The radial basis function (RBF) kernel can be written as:

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\gamma ||\boldsymbol{x}_1 - \boldsymbol{x}_2||^2), \tag{4.22}$$

  where $\gamma$ is a positive number controlling the radius. The decision boundary hence becomes

$$D(\boldsymbol{x}) = \sum_{i \in S} \alpha_i y_i \exp(-\gamma ||\boldsymbol{x}_i - \boldsymbol{x}||^2) + b.$$

- ***Polynomial kernels*:** The other possible kernel is the polynomial kernel that takes the form:

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1^T \boldsymbol{x}_2 + 1)^d, \tag{4.23}$$

  where $d$ is the degree of the polynomial.

## 4.5 Multi-classes support vector machine

Suppose we have $\{\boldsymbol{x_i}, y_i\}_{i=1}^n$ an $n$ independent data points, where $y$ represents the class label. Further, suppose the data points come from more than 2-classes such that each group of observations belongs to a different class. Since many algorithms in machine learning, including SVM, are formulated to handle the binary-class problem, applying these classifiers to multi-class data is not straightforward (Rokach, 2010). A modification needs to be applied to the data to apply SVM. A number of approaches were introduced in the literature to handle the multi-classes problem (Abe, 2005). These approaches included and were not limited to:

- one against one (pairwise) classification,
- one against all classification,
- error-correcting output code classification, and

- all at once classification.

The central concept of the approaches listed above is to allow the multi-class problems to be reframed into multiple binary classes and then perform the SVM sequentially. That is, classifying multi-class data is not a one-step task; one needs to perform a sequence of decision boundaries. The number of classes determines the number of decision boundaries one needs to be obtained. We will explain how each method performs the classification:

- ***One against one classification:*** this approach performs classification based on all possible pairs, that is, it constructs $\binom{Z}{2}$ SVM classifiers (James et al., 2013). In other words, for classification, one might compare class $i$ versus class $j$ coded as $+1$ and $-1$, respectively. Hence, all possible $\binom{Z}{2}$ pairwise SVM classifiers are performed for new test data. The observation is assigned to the class most frequently assigned in the pairwise classification.

- ***One against all classification:*** This alternative approach converted the multi-class classification into $Z$ binary classification. Each time we compare class $i$ coded as $+1$ against the remaining $Z-1$ classes coded as $-1$. The new test data is assigned to the class assigned to it the most.

- ***Error correcting output code classification:*** This approach required $r = \lceil \log_2 Z \rceil$ SVMs, where $\lceil . \rceil$ is the ceiling operator. The concept of this approach is to assign a unique r-bits (binary string) known as *code*. That is, for a $Z$-class problem, one needs to construct an $Z \times r$ matrix, say $\mathcal{J}$. Each row of $\mathcal{J}$ represents the code associated with class $z$, and the column represents the classifiers. Then for each column, we fit an independent classifier. For new instance $\boldsymbol{x}$, we find a bit-string, say $b(\boldsymbol{x})$ of length $r$ based on training the classifiers. The decision is taken by finding the minimum Humming distance between $b(\boldsymbol{x})$ and $\mathcal{J}_z$. That is, suppose Humming distance between $b(\boldsymbol{x})$ and the $z^{th}$ class $\mathcal{J}_z$denoted by $h_z$, one seeks to find $\arg\min_z h_z(b(\boldsymbol{x}), \mathcal{J}_z)$. The new input is hence assigned to the class that is closest in the distance.

- ***All at once classification:*** This alternative approach converted the multi-class problem into a binary classification. The conversion is achieved by expanding the input data's dimension from $p$ dimensional into $p \times Z$ and simultaneously performing all the decision boundaries.

# Chapter 5

# Envelope-based support vector machine classifier

## 5.1 Introduction

In this chapter, we employ the efficiency of an envelope method to improve classification accuracy by extending the envelope model to supervised learning. We propose a new classifier, namely an envelope-based support vector machine classifier (ESVM). By introducing ESVM, we aim to enhance classification accuracy via tackling the dimensionality problems in classification. In our approach, we aim to eliminate the effect of redundant features that might affect classification efficiency. We assume the features that are correlated to the outcome form a reducing lower-dimensional subspace in which the data are projected. Hence, one objective is to extract a reducing subspace that contains the informative features only. This reducing subspace is used as a projection matrix to reduce the dimension of the data under consideration. The classification based on classic SVM is then performed on the reduced data.

Consider classification and discriminant analysis settings where the response is a categorical variable. The successful classification process requires a classifier of high classification accuracy. However, when the data is high dimensional, due to its nature, classifying such data might be computationally expensive. Hence, in such data analysis, reducing the data's dimension without loss of information is a primal interest.

In statistical analysis, including classification and discriminant analysis, the problem of dimensionality is addressed in two ways. One approach via performing features selection and classification simultaneously. The other approach is the projection-based techniques and known as reduce and classify. That is, we assume that there exists a reducing subspace that contains the classification-related information. The classification based on the latter approach is a two stages process: in the first stage projecting the data onto a lower-dimensional subspace. Then the second stage is performing the classification of the projected data.

The majority of research has intensively studied and improved the dimension reduction methods in the regression framework to enhance prediction ability. Similarly, the accuracy of classification might be affected by a large number of variables. Hence, reducing the dimensionality improves the generalization and reduces the complexity of the classifier (Aksu et al., 2010). Several authors have considered reducing the dimension of the features as a preliminary step before performing classification. See for example, (Paul et al., 2013), and (Kumar et al., 2007). (Moradibaad and Mashhoud, 2018) have performed singular value decomposition (SVD) as a dimension reduction methods. That is, the data matrix $\boldsymbol{X}$ is decomposed based on SVD and written as $\boldsymbol{X} = USV^T$, then apply SVM to perform the classification. (Bura and Pfeiffer, 2003) used sliced inverse regression and sliced average variance estimation to extract the sufficient dimension reduction and then obtain the graphically based classification. (Shao et al., 2014) proposed classification based on PCA and kernel PCA. (Antoniadis et al., 2003) used minimum average variance estimation to reduce the dimension prior to the classification. (Chen et al., 2018) introduced maximal mean-variance as a dimension reduction method.

Our approach differs in that we employ the effectiveness of the envelope principle to extract the reducing subspace. That is, the reducing subspace estimation algorithm works to include only the informative (material) predictor variables and excludes the redundant (immaterial) predictor variables. Hence, the components of the material predictor variables form the lower-dimensional subspace onto which the dataset is projected. The classification based on the projected data performs better or at least as good as the classification based on full data. However, working with lower dimension data advances the classification process in accuracy as well as in computation costs, as working with lower dimensions is less expensive. The critical point in our approach is the efficiency of estimating the reducing basis.

This chapter is organized as follows: the geometry of the projection-based classification is given in Section 5.2. An overview of envelope methods for labeled data is given in Section 5.3. The formal introduction of our proposed classifier is given in Section 5.4, in which we show the model, the estimation of the ESVM projection basis, and the classification rule. The link between SVM and linear discriminate analysis is discussed in Section 5.5. The criterion to select the dimension of ESVM basis is demonstrated in Section 5.6. The asymptotic properties of the model parameters are given in Section 5.7. Lastly, Section 5.8 contains numerical studies, which include a collection of public data as well as different scenarios of synthetic data.

## 5.2 Projection and margin preservation in support vector machine

In chapter 4, we have demonstrated the SVM algorithm for constructing the separating hyperplane. The key concept in constructing the optimal hyperplane is to maximize the distance between the nearest point from each class and the hyperplane, i.e., maximizing the margin. Hence, it is preferable to keep the margin preservation when transforming the data via projecting it into a projection matrix. The projection matrix can be extracted at random or non-random methods. Gaussian random projection (Shi et al., 2012), for instance, is a widely used technique for random projection. Paul et al. (2013) argued that the Euclidean distance is preserved if the random projection is carefully chosen. They have proved that SVM optimization based on the projected data results in comparable margin and data radius as in the original space. On the other hand, the non-random projection matrix construction techniques include PCA, PLS, SIR, and SAVE.

Furthermore, if we assume that the original data fall within a ball with a random radius; thus, the radius of the minimal ball enclosing the projected data is very close to the minimal ball enclosing the original data. This result is an indication of margin preservation and no classification-related loss of information based on reduced data. On the other hand, it is worth mentioning that classification based on reduced data decreases the computation complexity and memory usage.

## 5.3 Envelope for labeled data

In this section, we discuss constructing envelope subspace for labeled data. As shown in chapter 3, envelope method originaly was developed in regression framework. The concept of envelope-baed reducing basis in discriminant analysis was introduced in Zhang and Mai (2018). The authors proposed envelope based technique for constructing sufficient dimension reduction in discriminate analysis, namely envelope in discriminant subspace (ENDS). Suppose $y_i$ is a categorical response variable that represents the class label such that $y_i \in \{1, ..., Z\}$ where $Z \geq 2, \quad i =, 1, ..., n$. Further, suppose $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ is the data matrix repesenting $n$ observations in $p$ dimensional space. Given a subspace $\mathcal{S} \subseteq \mathbb{R}^p$, let $\Phi_{\mathcal{S}}(\boldsymbol{X}) \equiv \arg\max_{z=1,...,Z} \Pr(y = z|\boldsymbol{P}_{\mathcal{S}}\boldsymbol{X})$. If $\mathcal{S}$ satisfies the condition $\Phi_{\mathcal{S}}(\boldsymbol{X}) = \Phi(\boldsymbol{X})$, where $\Phi(\boldsymbol{X}) \equiv \arg\max_{z=1,...,Z} \Pr(y = z|\boldsymbol{X})$, then $\mathcal{S}$ is known as *discriminant subspace*. Similar to the CS, if the intersection of all discriminate subspace is itself subspace then it is referred to as *central discriminate subspace* (CDS) and denoted by $\mathcal{S}_{D(Y|X)}$.

The envelope discriminate subspace is defined as the smallest subspace that satisfies the following conditions:

- $\Phi_{\mathcal{S}}(\boldsymbol{X}) = \Phi(\boldsymbol{X})$
- $cov(\boldsymbol{P}_{\mathcal{S}}\boldsymbol{X}, \boldsymbol{Q}_{\mathcal{S}}\boldsymbol{X}) = \boldsymbol{0}$

The first condition assures that classification based on the reduced dataset is as efficient as classification based on the complete dataset. While the second condition indicates that the material part $\boldsymbol{P}_{\mathcal{S}}\boldsymbol{X}$ is linearly independent of the immaterial part $\boldsymbol{Q}_{\mathcal{S}}\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{P}_{\mathcal{S}}\boldsymbol{X}$. The second condition is to assure that the immaterial part will not affect the classification directly nor indirectly via its relationship with the material part.

On the other hand, Wang et al. (2020) extended envelopes for unsupervised learning and model-based clustering. They proposed a new mixture model, namely the clustering envelope mixture model (CLEMM), which is based on the commonly used Gaussian mixture model assumptions. The new method is developed based on the belief that there exist two orthogonal subspaces, say $(\mathcal{S}, \mathcal{S}^{\perp})$. That is, projecting the data onto one of these subspaces has no information about the clustering structure and hence immaterial to the clustering algorithm, while projecting the data onto the other subspace contains all relevant information about variation across clusters and hence of interest. To put everything together; let $\boldsymbol{X} \in \mathbb{R}^{p\times n}$ be the observations matrix and assumed to be the Gaussian mixture model distributed, that is:

$$\boldsymbol{X} \sim \sum_{z=1}^{Z} \pi_z N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z), \tag{5.1}$$

where $\pi_z \in \{0, 1\}, \sum_{z=1}^{Z} \pi_z = 1$ is the mixing weight, $\boldsymbol{\mu}_z \in \mathbb{R}^p$ is the mean of cluster $z$, and $\boldsymbol{\Sigma}_z \in \mathbb{R}^{p\times p}$ is the covariance matrix of cluster $z$.

Suppose $\mathcal{S}$ is the subspace of interest, further suppose $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times u}, u < p$ is the basis of $\mathcal{S}$ and $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-u)}$ is the basis of $\mathcal{S}^{\perp}$. Under the CLEMM, $\boldsymbol{X}$ is partitioned into material and immaterial part, such that $\boldsymbol{X}_M = \boldsymbol{\Gamma}^T\boldsymbol{X}$, denotes the material part of $\boldsymbol{X}$ and $\boldsymbol{X}_{IM} = \boldsymbol{\Gamma}_0^T\boldsymbol{X}$ denotes the immaterial part of $\boldsymbol{X}$. The distributions of the material and immaterial parts are given by:

$$\boldsymbol{X}_M = \boldsymbol{\Gamma}^T\boldsymbol{X} \sim \sum_{z=1}^{Z} \pi_z N(\boldsymbol{\alpha}_z, \boldsymbol{\Omega}_z), \quad \boldsymbol{X}_{IM} = \boldsymbol{\Gamma}_0^T\boldsymbol{X} \sim N(\boldsymbol{0}, \boldsymbol{\Omega}_0), \quad \boldsymbol{X}_M \perp \boldsymbol{X}_{IM} \tag{5.2}$$

which states that the distribution of the material part varies across clusters while the distribution of immaterial is unimodal and fixed across clusters.

## 5.4 Definition and notation

In this section, we define the notations of our approach. Suppose $\boldsymbol{y} = (y_1, ..., y_n)$ is a categorical response variable representing the class label, and $\boldsymbol{X} \in \mathbb{R}^{p\times n}$ is the data matrix. Further, assume that there is an orthogonal basis $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0)$, such that $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times u}$ $(u < p)$ is a semi-orthogonal basis and $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-u)}$. Let the $\mathcal{S} \in \mathbb{R}^p$ be the subspace spanned by columns of $\boldsymbol{\Gamma}$, $\mathcal{S} = \text{span}(\boldsymbol{\Gamma})$. Let the complement of $\mathcal{S}$ denoted by $\mathcal{S}^{\perp}$ and constructed with the usual inner product. Given that $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma}$ is positive definite; we define the orthogonal projection of $\boldsymbol{\Gamma}$ onto $\mathcal{S}$ as: $\boldsymbol{P}_{\Gamma} = \boldsymbol{\Gamma}(\boldsymbol{\Gamma}^T\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}^T$, while the projection onto $\mathcal{S}^{\perp}$ is denoted by $\boldsymbol{Q}_{\Gamma} = \boldsymbol{I}_p - \boldsymbol{P}_{\Gamma}$.

For a subspace $\mathcal{S} \in \mathbb{R}^p$ and $\boldsymbol{M} \in \mathbb{R}^{p\times p}$, the following definition facilitates the description of a subspace and an envelope, and equivalent to that given in Cook et al. (2010).

**Definition 5.4.1.** $\mathcal{S}$ reduces $\boldsymbol{M} \in \mathbb{R}^{p\times p}$ if and only if $\boldsymbol{M}$ can be written in the form $\boldsymbol{M} = \boldsymbol{P}_{\mathcal{S}}^T\boldsymbol{M}\boldsymbol{P}_{\mathcal{S}} + \boldsymbol{Q}_{\mathcal{S}}^T\boldsymbol{M}\boldsymbol{Q}_{\mathcal{S}}$.

In our work, $\boldsymbol{M}$ is represented by the covariance matrix $\boldsymbol{\Sigma}$. We assume the that the transformed predictors, $\boldsymbol{\Gamma}^T\boldsymbol{X}$ captures all the information needed for the classification. That is, the classification based on $\boldsymbol{\Gamma}^T\boldsymbol{X}$ is equivalent to the classification based on $\boldsymbol{X}$. Now, estimating the ESVM basis, $\boldsymbol{\Gamma}$ is of interest, and since the only requirement for $\boldsymbol{\Gamma}$ is to be semi-orthogonal, there are a number of ways that one can estimate such a matrix. One is, for example, the use of dimension reduction techniques; the other is the use of likelihood estimation in a similar manner suggested by Cook et al. (2010). The following section facilitates the estimation process.

### 5.4.1 The estimation of envelope basis

In this section, we discuss the estimation of the ESVM basis. It is well known that SVM is a distribution-free classifier; however, for the sake of developing the algorithm to estimate the semi-orthogonal matrix $\boldsymbol{\Gamma}$, suppose $\boldsymbol{X}$ is normally distributed with class mean and shared variance, $\boldsymbol{X} \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma})$. Since we assume the existence of a lower-dimensional semi-orthogonal basis $\boldsymbol{\Gamma}$ that captures all relevant information across classes, suppose $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{p\times p}$ is an orthogonal basis such that $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times u}$ and $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-u)}$. Further, let $\boldsymbol{\Gamma}^T\boldsymbol{X}$ is the material part, the part that contains all the information about classification, and $\boldsymbol{\Gamma}_0^T\boldsymbol{X}$ is the immaterial part. The ESVM basis $\boldsymbol{\Gamma}$ can be estimated, without loss of generality, via a likelihood-based technique. That is, $\boldsymbol{\Gamma}$ is a solution to an optimization over Grassman manifold $\mathcal{G}(p,u)$ under the constraint $\boldsymbol{\Gamma}^T\boldsymbol{\Gamma} = \boldsymbol{I}_u$.

Under the normality assumption of the data, $\boldsymbol{X} \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma})$, that is:

$$f(\boldsymbol{x}) = (2\pi)^{-p/2}|\Sigma|^{-1/2}\exp\left\{\frac{-1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_z)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_z)\right\} \tag{5.3}$$

Now, assume $\boldsymbol{\mu}_z = \boldsymbol{\Gamma}\boldsymbol{\alpha}_z$ is the class mean, where $\boldsymbol{\alpha} \in \mathbb{R}^u$ and $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T$. The likelihood function becomes:

$$L = \prod_{i=1}^{n}(2\pi)^{-p/2}|\boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T+\boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T|^{-1/2}\exp\left\{\frac{-1}{2}(\boldsymbol{x}_{iz}-\boldsymbol{\Gamma}\boldsymbol{\alpha}_z)^T(\boldsymbol{\Gamma}\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^T+\boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0^{-1}\boldsymbol{\Gamma}_0^T)(\boldsymbol{x}_{iz}-\boldsymbol{\Gamma}\boldsymbol{\alpha}_z)\right\} \tag{5.4}$$

The log-likelihood is given by:

$$\ell = \text{constant} - \frac{n}{2}|\boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T+\boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T| - \frac{1}{2}\sum_z\sum_i(\boldsymbol{x}_{iz}-\boldsymbol{\Gamma}\boldsymbol{\alpha}_z)^T(\boldsymbol{\Gamma}\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^T+\boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0^{-1}\boldsymbol{\Gamma}_0^T)(\boldsymbol{x}-\boldsymbol{\Gamma}\boldsymbol{\alpha}_z) \tag{5.5}$$

The parameters to be estimated are $(\boldsymbol{\Gamma}, \boldsymbol{\alpha}_z, \boldsymbol{\Omega}, \boldsymbol{\Omega}_0)$. To find the MLE for the parameters, we find the derivatives with respect to each parameter then equate it to zero. The MLEs for the parameters are given by:
$\hat{\boldsymbol{\alpha}}_z = \boldsymbol{\Gamma}^T \bar{\boldsymbol{x}}_z$,
$\hat{\boldsymbol{\Omega}} = \boldsymbol{\Gamma}^T \boldsymbol{S} \boldsymbol{\Gamma}$,
$\hat{\boldsymbol{\Omega}}_0 = \boldsymbol{\Gamma}_0^T \boldsymbol{S}_x \boldsymbol{\Gamma}_0$,
where $\boldsymbol{S}$ is the shared class variance and $\boldsymbol{S}_x$ is overall variance. Substituting the MLEs in the log-likelihood yields:

$$\hat{\boldsymbol{\Gamma}} = \arg \min_{\Gamma \in \mathcal{G}(u,p)} \left[ \log |\boldsymbol{\Gamma}^T \boldsymbol{S}_x^{-1} \boldsymbol{\Gamma}| + \sum_{z=1}^{Z} n_z \log |\boldsymbol{\Gamma}^T \boldsymbol{S} \boldsymbol{\Gamma}| \right]. \tag{5.6}$$

Thus, $\boldsymbol{\Gamma}$ can be estimated by optimising (5.6) over Grassman manifold. In the following section we explain the algorithm to estimate ESVM basis.

### 5.4.2 Algorithm for ESVM

In this section the algorithm for constructing the ESVM basis $\boldsymbol{\Gamma}$ is presented. The estimated basis, $\hat{\boldsymbol{\Gamma}}$ is a solution to an optimisation over Grassman manifold $\mathcal{G}(u, p)$ under the constraint $\boldsymbol{\Gamma}^T \boldsymbol{\Gamma} = \boldsymbol{I}_u$. The estimation is not straightforward and challenging task. Recall (5.6), is a function that is invariant under right orthogonal transformation; precisely, for any $\boldsymbol{U} \in \mathbb{R}^{u \times u}$ orthogonal matrix, $f(\boldsymbol{\Gamma}\boldsymbol{U}) = f(\boldsymbol{\Gamma})$. Now, suppose $\mathcal{S}_\Gamma$ is the subspace spanned by the columns of $\boldsymbol{\Gamma}$. Then

$$\mathcal{S}_\Gamma = \left\{ \boldsymbol{\Gamma}\boldsymbol{U} | \boldsymbol{U}, \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_u \right\} \in \mathcal{G}(u, p), \tag{5.7}$$

where $\mathcal{G}(u, p)$ is the Grassmann manifold of all $u-$dimensional subspace in $\mathbb{R}^p$ (Adragni et al. (2012), Zhang et al. (2018)). The task now is to find an estimate of $\boldsymbol{\Gamma}$, such that:

$$\hat{\boldsymbol{\Gamma}} = \arg \max_{\boldsymbol{\Gamma} \in \mathcal{G}(u,p)} f(\boldsymbol{\Gamma}). \tag{5.8}$$

The algorithm for estimating $\boldsymbol{\Gamma}$ is an iterative procedure that computes an ascent direction where $f(\boldsymbol{\Gamma})$ increases. Let $\boldsymbol{\Gamma}^* = (\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{p \times p}$ be an orthogonal matrix, where $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p \times (p-u)}$ is the completion of $\boldsymbol{\Gamma}$. Further, let $\boldsymbol{V} = (\nabla f(\boldsymbol{\Gamma}))^T \boldsymbol{\Gamma}_0$ be the rate of change of $f(\boldsymbol{\Gamma})$ in the direction of $\boldsymbol{\Gamma}_0$. Let $\boldsymbol{J} \in \mathbb{R}^{p \times p}$ a skew$-$symmetric matrix depends on the directional derivative $\boldsymbol{V}$, and updated until the stopping criteria is met. $\boldsymbol{J}$ is defined as follows:

$$\boldsymbol{J} = \begin{pmatrix} \boldsymbol{0}_u & \boldsymbol{V} \\ -\boldsymbol{V}^T & \boldsymbol{0}_{p-u} \end{pmatrix}$$

The sufficient condition $\hat{\mathcal{S}}_\Gamma$ be a maximizer of $f$ is that $||\boldsymbol{V}|| < \delta$, where $\delta$ is sufficiently small number. Hence, the algorithm can be summarised as follow:

**Algorithm 2** Grassmann manifold optemization

```
1. initiate $\boldsymbol{\Gamma}_0^*$, the initial value of $\boldsymbol{\Gamma}^*$ at step $m_0$.
2. For $m = 1, 2, ...$ until the convergence reached do the following:
(i) Compute the directional derivative $\boldsymbol{V}$ then form the matrix $\boldsymbol{J}$.
(ii) Update $\boldsymbol{\Gamma}_{m+1}^* = \boldsymbol{\Gamma}_m^* \exp\{\gamma \boldsymbol{J}\}$, where $\gamma \in (0, 1)$.
3. $\hat{\boldsymbol{\Gamma}}$ is the first $u$ columns of $\boldsymbol{\Gamma}^*$ at the last iteration.
```

### 5.4.3 Classification mechanism of Envelope Support Vector Machine classifier

In this section, we discuss the classification mechanism of our proposed classifier, ESVM. Recall that in classic SVM, the key point is to estimate the optimal hyperplane. This hyperplane is determined by the model parameters $(\boldsymbol{w}, b)$. The parameters of interest can be estimated by optimizing the following objective function:

$$\boldsymbol{w}^T\boldsymbol{w} + \lambda\mathbb{E}[1 - (\boldsymbol{w}^T\boldsymbol{x} - b)] \tag{5.9}$$

The material part of $\boldsymbol{X}$ is given by $\boldsymbol{X} = \boldsymbol{\Gamma}^T\boldsymbol{X}$, hence one can substitute $\boldsymbol{x}$ by its reduction. The objective function in the envelope projected space is:

$$\boldsymbol{w}_\Gamma^T\boldsymbol{w}_\Gamma + \lambda\mathbb{E}[1 - (\boldsymbol{w}_\Gamma^T\boldsymbol{\Gamma}^T\boldsymbol{x} - b)], \tag{5.10}$$

where $\boldsymbol{w}_\Gamma \in \mathbb{R}^u$ is the weight and the subscript to distinguish it from the full data based weight. To estimate $\boldsymbol{w}_\Gamma$ one needs to solve the objective given in (5.10).

Similar to the classic SVM, to estimate $\boldsymbol{w}_\Gamma$ consider the following optimisation

$$\begin{aligned} &\arg\min_{\boldsymbol{w}_\Gamma} \frac{1}{2}\boldsymbol{w}_\Gamma^T\boldsymbol{w}_\Gamma + \gamma\sum_{i=1}^n \xi_i \\ &\text{subject to } y_i(\boldsymbol{w}_\Gamma^T\boldsymbol{\Gamma}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i \\ &\sum_{i=1}^n \xi_i \geq 0. \end{aligned} \tag{5.11}$$

we introduce Lagrangian multipliers $(\alpha, \lambda)$ such that:

$$L(\boldsymbol{w}_\Gamma, b, \xi, \alpha, \lambda) = \frac{1}{2}\boldsymbol{w}_\Gamma^T \boldsymbol{w}_\Gamma + \gamma \sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i [y_i(\boldsymbol{w}_\Gamma^T \boldsymbol{\Gamma}^T \boldsymbol{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{n} \lambda_i \xi_i, \quad (5.12)$$

Similarly,

$$\frac{\partial L(\boldsymbol{w}_\Gamma, b, \xi, \alpha, \lambda)}{\partial \boldsymbol{w}_\Gamma} = 0$$
$$\frac{\partial L(\boldsymbol{w}_\Gamma, b, \xi, \alpha, \lambda)}{\partial b} = 0$$
$$\frac{\partial L(\boldsymbol{w}_\Gamma, b, \xi, \alpha, \lambda)}{\partial \xi} = 0$$

That yeilds:

$$\hat{\boldsymbol{w}}_\Gamma = \sum_{i=1}^{n} \boldsymbol{\Gamma}^T \alpha_i y_i \boldsymbol{x}_i = \boldsymbol{\Gamma}^T \boldsymbol{w} \quad (5.13)$$

subject to

$$\sum y_i \alpha_i = 0, \quad 0 \le \alpha_i \le \gamma$$

$$\gamma - \alpha_i - \lambda_i = 0$$

Thus, the classification decision for given data point $\boldsymbol{x}$ is given by:

$$D(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}_\Gamma^T \boldsymbol{x} + b). \quad (5.14)$$

From the above, it is obvious that $\hat{\boldsymbol{w}}_{env} = \hat{\boldsymbol{w}}_\Gamma = \boldsymbol{\Gamma}^T \boldsymbol{w}$, where $\hat{\boldsymbol{w}}_{env}$ is the estimator in the envelope space, $\boldsymbol{w}$ is the SVM solution in the original $p$-dimensional space.

## 5.5 Relation to envelope discriminant analysis

Linear Discriminant Analysis (LDA) and support vector machine in concept, both classifiers compute an optimal hyperplane. In this section, we investigate the relation between SVM and LDA. Further, we will revisit the work proposed by Zhang and Mai (2018) to show that span(Γ) in (5.6) is equivalent to the envelope discriminant subspace.

In the simplest case, if we have two-classed data, let $\boldsymbol{w}_{svm}$ and $\boldsymbol{w}_{LDA}$ be the norms calculated by SVM and LDA, respectively. Generally speaking, since the optimal hyperplane is unique, then we have:

$$||\boldsymbol{w}_{svm}|| \le ||\boldsymbol{w}_{LDA}||,$$

where $\boldsymbol{w}_{svm} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$ and $\boldsymbol{w}_{LDA} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$. If $\boldsymbol{x} \in \mathbb{R}^2$, and tranformed via $\boldsymbol{x} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}$, hence the norm of LDA reduces to the difference between the classes means $\boldsymbol{w}_{LDA} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$, (Gokcen and Peng, 2002). The equality between the norms in the spherical space can be written as:

$$\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i,$$

where $\boldsymbol{\mu}_z$ is the class centroids $z = 1, 2$, $\alpha_i$ is the Lagrange coefficient, and $y_i$ is the class label. Given a data set; if all Lagrange coefficients are equal, then SVM normal is equivalent to LDA normal. That is, SVM is a special case of LDA if the support vectors contain all the data points.

Recently, Zhang and Mai (2018) proposed envelope discriminant subspace (ENDS-LDA), see section 5.3. Let

$$\boldsymbol{X}|y = z \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}), \quad \boldsymbol{\mu}_z = \boldsymbol{\Gamma}\boldsymbol{\alpha}_z, \boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T, \tag{5.15}$$

where $\boldsymbol{\Sigma}$ is the class covariance and assumed to be the same for all classes, i.e. $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = ... = \boldsymbol{\Sigma}_Z = \boldsymbol{\Sigma}$. LDA classifier aims to obtain Bayes rule associated with lowest error rate. Zhang and Mai (2018) introduced envelope discriminant subspace where span($\boldsymbol{\Gamma}$) is the smallest subspace that reduces $\boldsymbol{\Sigma}$ and have shown that the projected data preserves the Bayes rule. Under the normality assumption, ENDS-LDA has a similar parametrization for $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}$ as ESVM as shown in (5.3).

## 5.6 Selecting the dimension of ESVM

Determining the number of components or, equivalently, the dimension of ESVM subspace $u$ is an essential step. This section will discuss the criteria for selecting optimal $u$. To select the optimal $u$, we use the $K$-fold Cross-Validation (CV) method. That is, given a dataset of size $n$, we split it into $K$ groups (folds) such that one group is used for validation, and the method is trained on the remaining $(K-1)$ folds. Then the method accuracy is calculated on the validation or *hold-out* group. The $u$ that produces the best accuracy is then chosen (Cook (2018) and James et al. (2013)). The CV technique for dimension selection can be summarized as follows:

1. Split the data into $K$ almost equal size folds, $G_1, ..., G_K$.

2. For $k = 1, ..., K$

   (i) Train the method on $(\boldsymbol{x}_i, y_i) \notin G_k$ and validate it on the hold-out fold whose elements $(\boldsymbol{x}_i, y_i) \in G_k$.

   (ii) For each value of $u = \{1, 2, .., p-1\}$ estimate the label of validate set and calculate the prediction accuracy as:

   $$Acc_k = \frac{\text{number of correctly classified observations}}{n_{G_k}},$$

   where $n_{G_k}$ is the number of elements in the validation fold.

   (iii) Calculate the $K$-fold CV estimate by averaging the values of $Acc_1$,...,$Acc_K$:

   $$CV_K = \frac{1}{K}\sum_{j=1}^{K} Acc_j$$

3. Repeat steps 1 and 2 $M$ times and find the accuracy average:

   $$CV(u) = \frac{1}{M}\sum_{m=1}^{M} CV_K^{(m)}.$$

4. The optimal $u$; hence, is the one that is associated with the maximum accuracy; that is:

   $$u = \arg\max_u CV(u).$$

## 5.7 Asymptotic properties

In this section, we discuss the asymptotic properties of the coefficients in the solution of ESVM classifier. We mainly derive the asymptotic normality of the coefficients of ESVM, namely $(\boldsymbol{w}_\Gamma, b)$. We apply Bahadur representation to derive the asymptotic properties of the coefficients (Bahadur, 1966) in a similar manner to the work presented by Koo et al. (2008). However, in the former work, the hessian matrix was derived via Radon transformation, while in our work, we employ the result presented in Li et al. (2011).

For simplicity, we developed the results for binary classes. Let $\boldsymbol{\theta} = (\boldsymbol{w}_\Gamma, b)$, $\boldsymbol{\theta^*} = \boldsymbol{w}_\Gamma$, $\boldsymbol{m} = (\boldsymbol{x}^T, y)^T$ and $\widetilde{\boldsymbol{x}} = (\boldsymbol{x}, -1)$. Further, suppose we have linear classification with hyperplane defined by: $g(\boldsymbol{x}, \boldsymbol{\theta}) = b + \boldsymbol{\theta}_\Gamma^{*T}\boldsymbol{x}$. Thus, the support vector machine minimizes the following function:

$$\mathcal{J}(\boldsymbol{\theta}, \boldsymbol{m}) = \boldsymbol{\theta}^{*T}\boldsymbol{\theta}^* + \lambda\mathbb{E}[1 - y_i g(\boldsymbol{x}_i, \boldsymbol{\theta})] \tag{5.16}$$

Let the minimizer of (5.16) denoted by $\hat{\boldsymbol{\theta}}$. Further, let $\boldsymbol{\Omega}_m$ be the support of $\boldsymbol{m}$ and $\boldsymbol{h}$ be a function of $(\boldsymbol{\theta}, \boldsymbol{m})$ such that: $\boldsymbol{h} = \Theta \times \Omega_m \rightarrow \mathbb{R}$. Suppose $\Delta_\theta$ denote the $(u+1)$-dimensional vector of derivatives, $\Delta_\theta = (\partial/\partial\theta_1, ..., \partial/\partial\theta_{u+1})$. The following propositions give the gradient of $\boldsymbol{\theta}$.

**Proposition 5.7.1.** *Suppose the distribution of $\boldsymbol{X}|Y = y_i \quad \forall y_i \in \{-1, 1\}$, is dominated by Lebesgue measure and $\mathbb{E}(||\boldsymbol{x}||^2) < \infty$ then:*

$$G(\boldsymbol{\theta}) = \Delta_\theta \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}, \boldsymbol{m})] = 2\boldsymbol{\theta}^{*T} - \lambda \mathbb{E}[\widetilde{\boldsymbol{x}} y I(1 - \boldsymbol{\theta}^T \widetilde{\boldsymbol{x}} y > 0)]. \tag{5.17}$$

Now, in order to develop the asymptotic covariance of $\hat{\boldsymbol{\theta}}$, the following proposition demonstrates the hessian matrix

**Proposition 5.7.2.** *Suppose $\boldsymbol{x}$ has a convex and open support and the conditional distribution $\boldsymbol{X}|Y = y_i \quad \forall y \in \{-1, 1\}$, is dominated by Lebesgue measure. Further, suppose*

1. *If we have any linearly independent $\boldsymbol{\theta^*, \delta} \in \mathbb{R}^u$, $y_i \in \{-1, 1\}$, and $v \in \mathbb{R}$, the following function is continuous:*

$$d \longmapsto \mathbb{E}(\widetilde{\boldsymbol{x}}|\boldsymbol{\theta}^{*T}\boldsymbol{x} = d, \boldsymbol{\delta}^T\boldsymbol{x} = v, Y = y) f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|\boldsymbol{\delta}^T\boldsymbol{x}, Y}(d|v, y); \tag{5.18}$$

2. *For any $i = 1, ..., u$ and $y \in \{-1, 1\}$, there is a nonnegative function $c_i(v, y)$ with $\mathbb{E}[c_i(V, Y)|y] < \infty$ such that*

$$\mathbb{E}(\boldsymbol{x}|\boldsymbol{\theta}^{*T}\boldsymbol{x} = d, \boldsymbol{\delta}^T\boldsymbol{x} = v, Y = y) f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|\boldsymbol{\delta}^T\boldsymbol{x}, Y}(d|v, y) \leq c_i(v, y); \tag{5.19}$$

3. *There is a nonnegative function $c_0(v, y)$ with $\mathbb{E}[c_0(V, Y)|y] < \infty$ such that $f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|\boldsymbol{\delta}^T\boldsymbol{x}, Y}(d|v, y) \leq c_0(v, y)$. Then, the function $\boldsymbol{\theta} \longmapsto \Delta_\theta \mathbb{E}[\mathcal{J}(\boldsymbol{\theta})]$ is differentiable in all direction with the following derivative matrix:*

$$\boldsymbol{H}(\boldsymbol{\theta}) = 2 + \lambda \sum P(Y = y) f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|Y}(b + y|y) \mathbb{E}(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^T|\boldsymbol{\theta}^{*T}\boldsymbol{x} = b + y). \tag{5.20}$$

The proof of the propositions are given in (Li et al., 2011) and for ease we provide it in the appendix. Lastly, we develop the $\sqrt{n}$ consistency and asymptotic normality of ESVM coefficients estimators.

**Proposition 5.7.3.** *Let $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ be a set of $n$ data points drawn independently from the distribution of $(\boldsymbol{X}, Y)$. Then $\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$ converges to normal distribution with mean $\boldsymbol{0}$ and variance $\boldsymbol{H}(\boldsymbol{\theta})^{-1}\boldsymbol{G}(\boldsymbol{\theta})\boldsymbol{H}(\boldsymbol{\theta})^{-1}$.*

## 5.8 Numerical study

In this section, we conducted several simulation studies to evaluate the performance of the proposed classifier. Public real data have been used for the same purpose as well. The evaluation of our method was set against the wildly known classifiers: support vector machine (SVM), linear discriminant analysis (LDA), and logistic regression. We used classification accuracy as an evaluation measure, which calculated as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}, \tag{5.21}$$

where

- True Positive (TP)=number of subjects that correctly classified as positive.
- True Negative (TN)=number of subjects that correctly classified as negative.
- False Positive (FP)=number of subjects that falsely classified as positive.
- False Negative (FN)=number of subjects that falsely classified as negative.

The built-in `R` functions are used for SVM `(e1071)`, LDA `(lda)`, and for grassmann optimisation we used `(GrassmannOptim)`. The following sections demonstrate the outcome based on real and simulated data.

### 5.8.1 Simulation

In this section, we discuss various simulation settings to compare the classification performance of our method, ESVM classifier, against other popular classifiers. The simulated data have been divided into training and test data as 80% and 20% respectively. We conducted the simulation with different sample sizes $n = 30, 60, 100, 120, 200$, each of which with different choice of parameters $(p, u, Z, \mathbf{\Gamma}, \mathbf{\Gamma}_0, \mathbf{\Omega}, \mathbf{\Omega}_0)$. The summary of the simulation setting is given in Table 5.1. To evaluate the performance of each method, the classifiers were trained on training data and evaluated on test data. For ESVM, the classifier was evaluated at different choices of components, while other classifiers were evaluated based on full dimensioned data. Then the average percentage of classification accuracy was calculated over 100 independent replicates.

The model parameters are generated as follows: the elements of the basis $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times u}$ were randomly generated from uniform distribution $(0, 1)$ then orthogonalised such that $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0)$ forms an orthogonal matrix. The class mean $\boldsymbol{\mu}_z = \boldsymbol{\Gamma}\boldsymbol{\eta}_z$, where $\boldsymbol{\eta}_z \in \mathbb{R}^{u\times 1}$ is generated from standard normal. The symmetric positive definite matrices $\boldsymbol{\Omega} \in \mathbb{R}^{u\times u}, \boldsymbol{\Omega}_0 \in \mathbb{R}^{(p-u)\times(p-u)}$ are diagonal matrices with $\boldsymbol{\Omega} = \tau_1 \boldsymbol{I}_u$ and $\boldsymbol{\Omega}_0 = \tau_2 \boldsymbol{I}_{(p-u)}$. The parameters $\tau_1$ and $\tau_2$ vary to manifest the collinearity among the predictor variables (Cook et al., 2013). The predictor variables are assumed to be multivariate normal vectors with class mean $\boldsymbol{\mu}_z$ and shared covariance $\boldsymbol{\Sigma}$, where $\boldsymbol{\mu}_z = \boldsymbol{\Gamma}\boldsymbol{\eta}_z$ when the subject $\boldsymbol{x}_i \in$ class $z$, the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T$.

The data is assumed to be coming from $Z$ classes such that, the label; $y_i$ is defined as follows:

$$y_i = \begin{cases} z, \quad \text{if} \quad \boldsymbol{x}_i \in \text{class } z \end{cases} \tag{5.22}$$

The simulation has been carried out as follows:

1. Fix the envelope basis dimension $u$, then generate the parameters $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0, \boldsymbol{\Omega}, \boldsymbol{\Omega}_0, \boldsymbol{\eta})$ as explained above.

2. Fix $n, p, Z$ and accordingly generate the data $\boldsymbol{X} \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma})$ and the label $\boldsymbol{y}$.

3. Divide the date into training and test data.

4. Estimate the ESVM basis $\hat{\boldsymbol{\Gamma}}$ from the training data following algorithm 5.4.2.

5. Reduce the full data by projecting it into $\boldsymbol{\Gamma}$, that is $\boldsymbol{X}^* = \boldsymbol{\Gamma}^T\boldsymbol{X}$.

6. Apply SVM algorithm to full dimension and reduced data as well as LDA and logistic regression for full dimension data. The average classification accuracy is reported for each method.

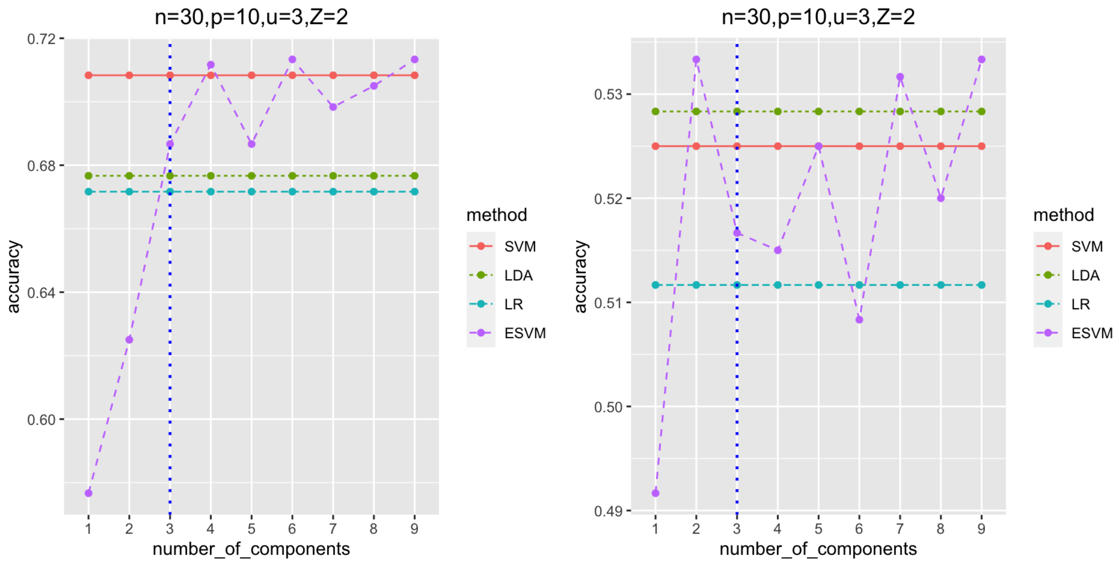
| n | p | u | z | $\tau_1$ | $\tau_2$ |
|---|---|---|---|---|---|
| 30,60,100 | 10 | 3 | 2 | 2 | 0.2 |
| | | | 2 | 20 | 0.4 |
| 60 | 15 | 3 | 3 | 2 | 0.2 |
| | | 6 | 3 | 4 | 0.25 |
| 120 | 15 | | 3 | 2 | 0.2 |
| | | 6 | 3 | 4 | 0.25 |
| 200 | 30 | 8 | 4 | 2 | 0.2 |
| | | | | 4 | 0.25 |

Table 5.1: The different choices of the model parameters for the simulation.

In all simulation settings, the classification accuracy varies. The focus was on ESVM in terms of performance against other classifiers. As shown in table 5.1, we vary the parameters and sample size to observe classifiers' behaviour. The outcome; however, showed that ESVM was uniformly better across all settings.

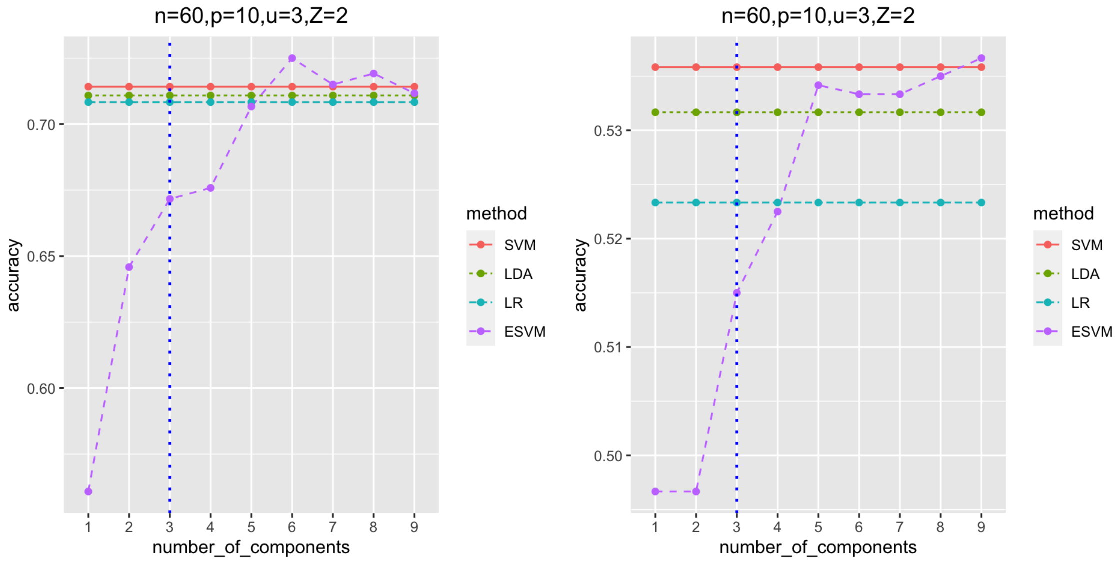
The first study was based on a sample $n = 30$ divided into two equal groups with 10 predictors. For this choice of $n$ we vary the number of components as well as the values of $\tau_1$ and $\tau_2$ to account for multicollinearity. That is, large $\tau_1/\tau_2$ means large correlation among predictors (Cook et al., 2013). The results are shown in Figure 5.1; for the first setting, where the collinearity is moderate, ESVM and SVM performed equally well and achieved the highest classification accuracy. However, ESVM obtained slightly better accuracy performance with fewer components. For the second setting, where the multicollinearity is higher that reflected via the values of $\tau_1$ and $\tau_2$. The overall performance is affected by the multicollinearity; however, we can see that ESVM performs better than the other classifiers.

Figures 5.2 and 5.3 show the outcome of different simulation scenarios when $n = 60$. Figure 5.2 summarizes the binary setting while Figure 5.3 summarizes the multi-classes settings. In both figures, the plot on the left shows the average accuracy when the collinearity is mild, while the plot on the right when the collinearity is higher. Having other factors fixed and varying the number of predictors, the binary class data when $p = 10$ and the size class is $n = 60$ the classifiers performance is better than the multi-classes settings. The ESVM classifier in both settings produces the best accuracy.



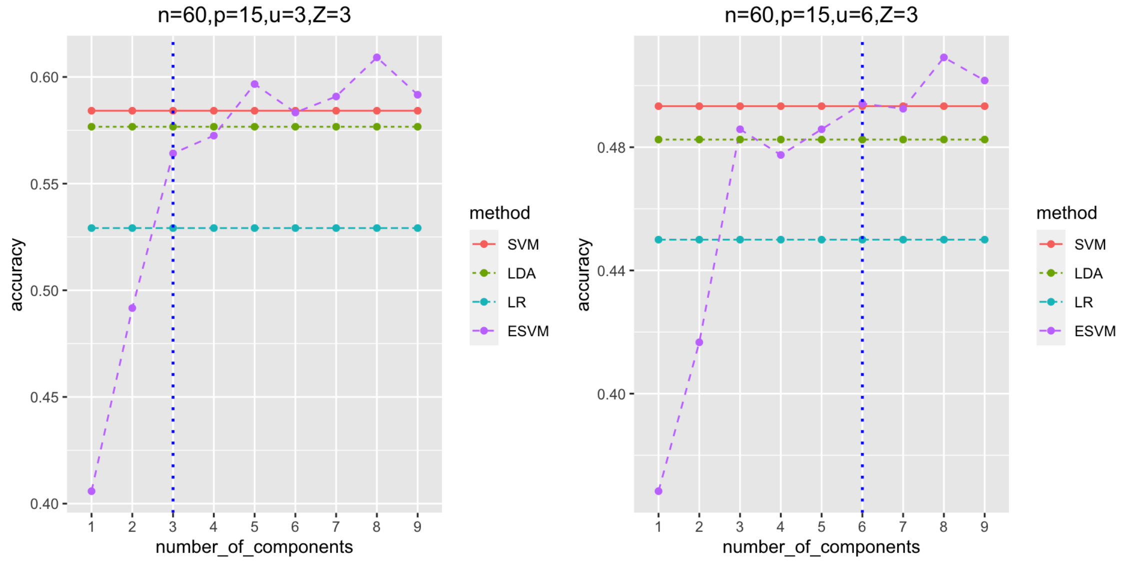
(a) (b)

Figure 5.1: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM. a: sample size $n = 30, p = 10, Z = 2, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 30, p = 10, Z = 2, (\tau_1, \tau_2) = (20, 0.4)$.



(a) (b)

Figure 5.2: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM. a: sample size $n = 60, p = 10, Z = 2, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 60, p = 10, Z = 2, (\tau_1, \tau_2) = (20, 0.4)$. The dashed vertical line is the true number of components.

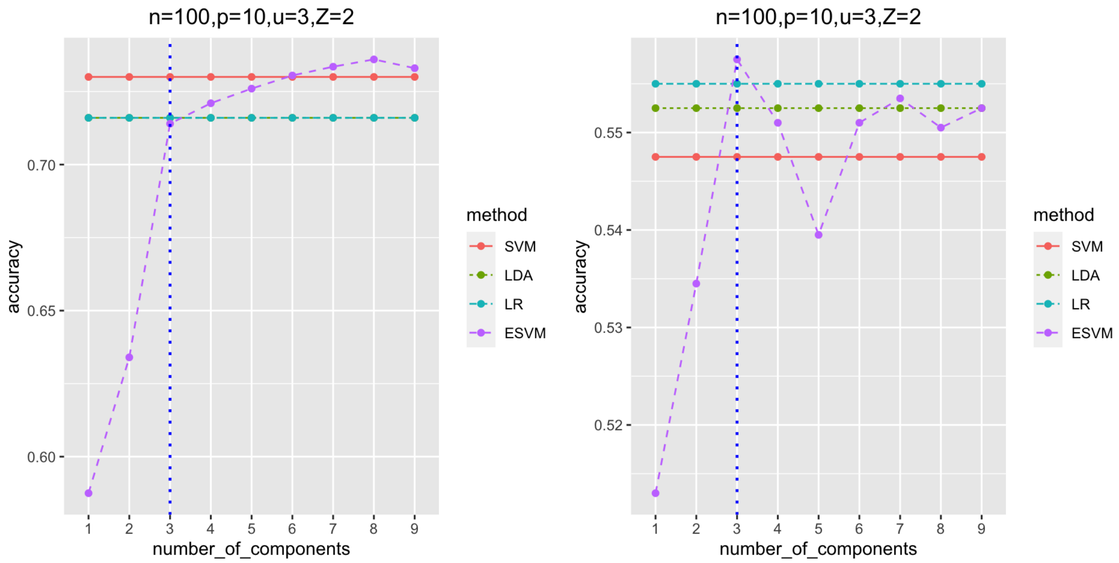


(a)

(b)

Figure 5.3: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM. a: sample size $n = 60, p = 50, Z = 3, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 60, p = 15, Z = 3, (\tau_1, \tau_2) = (4, 0.25)$. The dashed vertical line is the true number of components.

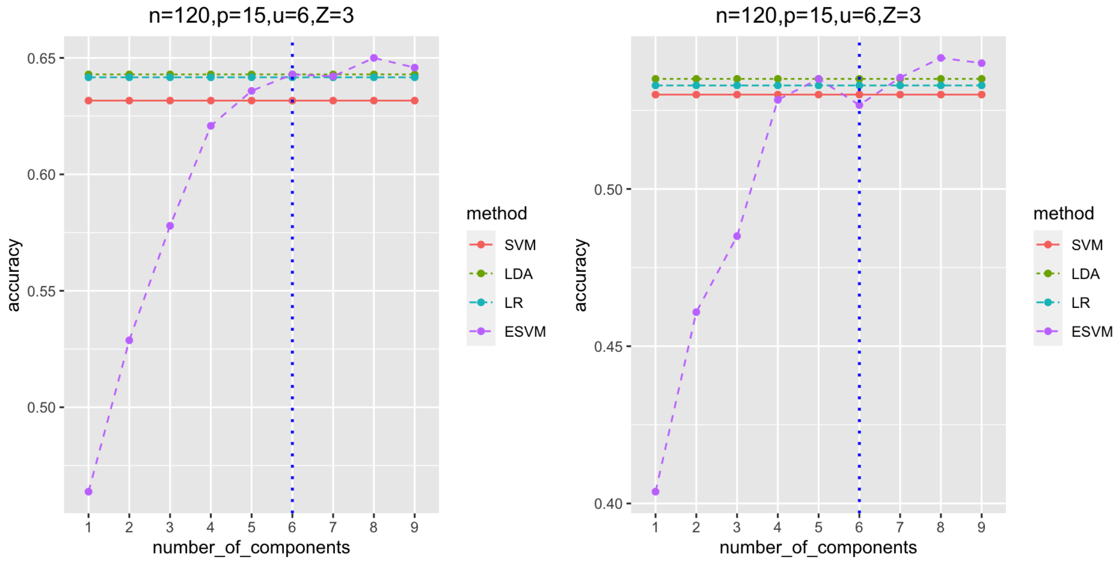
In the following scenario, we increased the sample size to be $n = 100$ and $n = 120$ with three different choices of the number of predictor variables and the classes $p = 10, Z = 2, u = 3$ and $p = 15, Z = 3, u = 6$, respectively. We vary the level of collinearity as well from moderate to high. The intention here is to see the effect of the number of classes as well. From Figures 5.4 and 5.5, the same can be said about the ESVM performance being better than other classifiers.



(a)

(b)

Figure 5.4: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM. a: sample size $n = 100, p = 10, Z = 2, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 100, p = 10, Z = 2, (\tau_1, \tau_2) = (20, 0.4)$. The dashed vertical line is the true number of components.



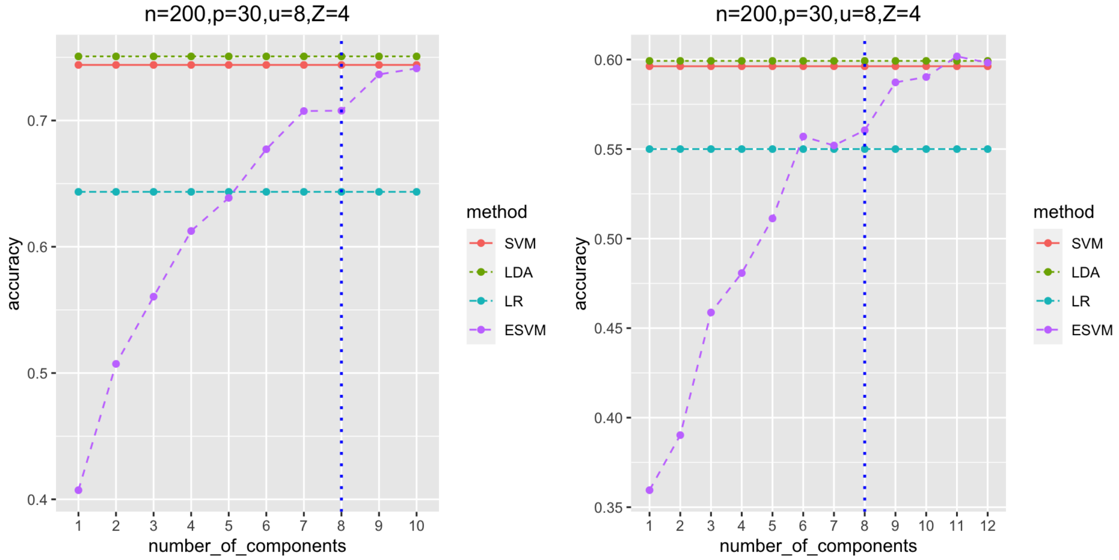
(a)

(b)

Figure 5.5: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM. a: sample size $n = 120, p = 15, Z = 3, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 120, p = 15, Z = 3, (\tau_1, \tau_2) = (4, 0.25)$. The dashed vertical line is the true number of components.

The last study was based on $n = 200$, with two choices of $\tau_1$ and $\tau_2$ was selected to produce moderate and high correlation. As results shown in 5.6, for multiple classes, ESVM performs better than other candidates.

Overall, ESVM performs well in all simulation settings. In some scenarios we noticed that the best accuracy is achieved when the number of components ($u$) is closer to the number of predictors ($p$). However; we noticed as well a number of components less than $p$ produces slightly lower accuracy could be chosen to be the ideal number of components by which the dimension reduction is achieved. For instance Figure 5.1 (b) where $p = 10$, the maximum accuracy is obtained at $u = 9$, however by choosing $u = 5$ we achieve dimension reduction. Further, we noticed that if the number of the predictor variables is closer to the class size, that is; $p \approx n_z$ the classification performance reduces.



(a) (b)

Figure 5.6: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM. a: sample size $n = 200, p = 30, Z = 4, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 200, p = 30, Z = 2, (\tau_1, \tau_2) = (4, 0.25)$. The dashed vertical line is the true number of components.

On the other hand, the 5-fold cross-validation is used on ESVM dimension selection. That is, for each of the simulation scenarios described in this section, we ran a 5-fold cross-validation method based on 30 samples and reported the optimal suggested number of components. For $n = 30$, the simulation was based on $u = 3$. In the first setting, the cross-validated accuracy suggested that $3, 4$ are similar with an accuracy $0.75, 0.77$, respectively. While for the second setting, the accuracy at $u = 5$ and $u = 8$ is similar; 0.53 and 0.54, respectively. For $n = 60$ we have 4 scenarios. The first scenario, as explained earlier, was binary class data with $p = 10, u = 3$ while the other is based on multi-class data with $p = 15, u = 3$ and 6; both with two levels of multicollinearity. The optimal number of components for the two binary class data settings is 4, with accuracy 0.82 and 0.55, respectively. For multi-class settings when $n = 60$, for the first scenario, the cross-validated accuracy suggested that the optimal number of components is 5 and 9 with a very similar accuracy of 0.56 and 0.59 and $u = 6$ for the other scenario. For $n = 100, u = 3$, the optimal $u = 4$ with accuracy 0.81 and 0.56 respectively. The other choice of sample size is $n = 120$, which was simulated with $u = 6$. The suggested number of components based on cross-validated accuracy for both settings for this choice of sample size are 5 and 6 with an accuracy of 0.54 and 0.62, respectively. Lastly, for the setting $n = 200$ where the true number of components is $u = 8$, the cross validated accuracy suggested $u = 12$ with associated accuracy 0.63 for the first scenario and 0.48 for the second scenario.

### 5.8.2 Real data

In this section, we tested our method based on 3 publicly available datasets. Each dataset was divided into training and test data, 80% and 20% respectively. ESVM was compared against SVM, LDA, and logistic regression. In a similar manner to simulated data, the classifiers were trained on training data and then evaluated on test data. The process was repeated 50 times then the average classification accuracy was reported. To determine the optimal number of components, we used the 5 fold cross-validated accuracy.

#### Berkeley dataset

This dataset contains the height of 93 children born in 1928-29 in Berkeley, of which 54 girls and 39 boys. For each child, the height measurement obtained at 31 age points between the age of 1 and 18 (RD, 1954). The children's height measurements were taken at 31 various age points while conducting the study, representing the predictor variables, i.e $p = 31$.The data was classified based on gender into two groups. We evaluated the classification accuracy for ESVM at different choices of $u$, while for SVM, LDA as well as logistic regression based on full dimensioned data. We repeated this process 50 times with a random split of the data. Figure 5.7 shows the average classification accuracy that the classifiers scored. The data has a considerably large number of predictor variables, $p = 31$; however, we can see that ESVM achieved better classification accuracy compared to other classifiers with only 10 components.

To determine the optimal number of components for this dataset, the cross-validated accuracy suggested that $u = 10$ with 0.96 accuracy.

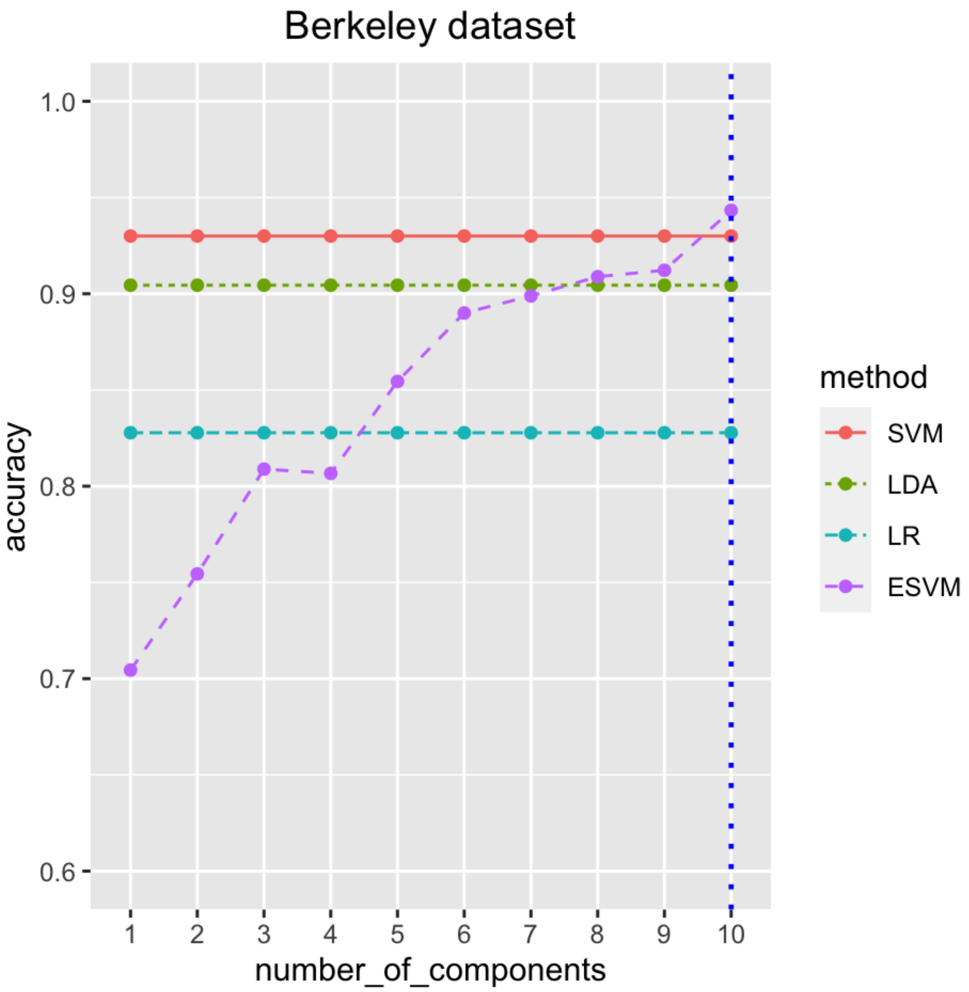


Figure 5.7: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM for Berkeley dataset. The dashed vertical line is the estimated number of components.

### Cattle dataset

The Cattle data (Kenward, 1987) investigated two treatments that were applied to control roundworm in cattle. A sample of 60 cows was divided into two groups equally such that each group had 30 cows. The weights of the cows were measured at the beginning of the study, then every two weeks at time intervals from week 2 till week18; the last measurement was taken after one week (week 19). We classified the data using ESVM, SVM, LDA, and logistic regression, then computed the classification accuracy. The results are shown in Figure 5.8; we can see that the classifiers perform equally well. ESVM achieved classification accuracy similar to SVM only 4 components. That agreed with the optimal number of components suggested by cross-validated accuracy as well .

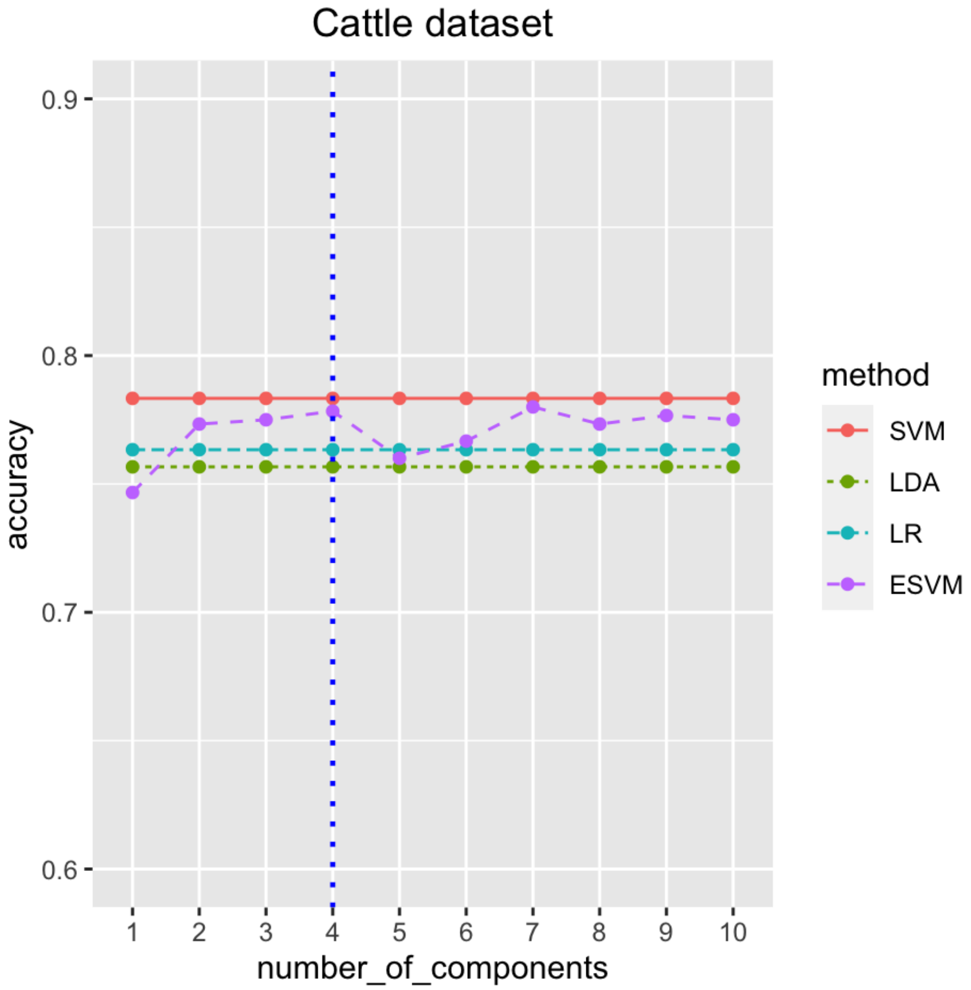


Figure 5.8: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM for Cattle dataset. The dashed vertical line is the estimated number of components.

### Breast cancer dataset

The breast cancer data is public real data variable at UCI Machine Learning Repository Dua and Graff (2017). The data has 569 observations and $p = 30$ predictor variables. The features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass while the response variable indicates the diagnosis status (malignant or benign). We divide the dataset into training and test data. The classifiers are trained on the training data and evaluated on the test data. Figure 5.9 shows the average classification accuracy. ESVM achieved %91 accuracy with only 3 components and %95 with 7 components. In contrast, the other classifiers (SVM, LDA, and LR) performed equally based on full data. In other words, ESVM performed as well as other classifiers with low cost.

The 5-fold cross-validated accuracy showed that $u = 8$ is the optimal choice for the number of components.

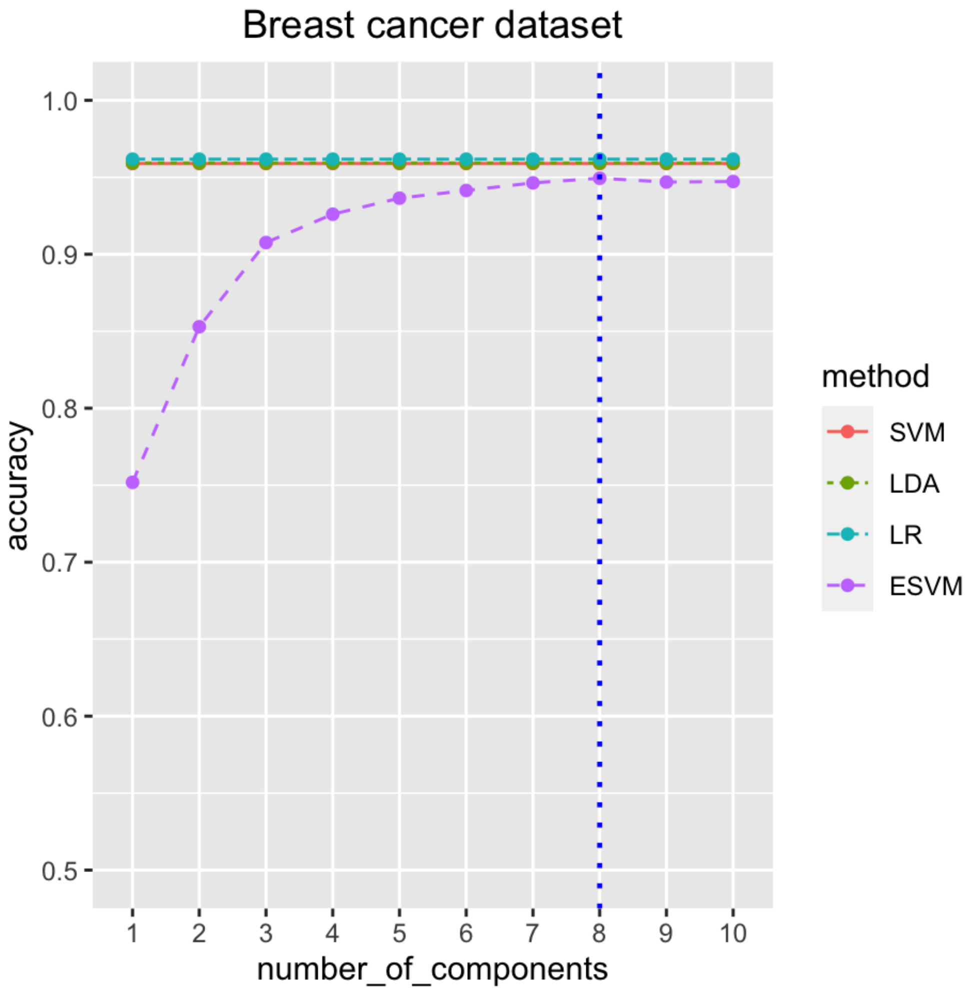


Figure 5.9: The average classification accuracy percentage based on SVM, LDA, logistic regression (LR) and ESVM for breast cancer dataset. The dashed vertical line is the estimated number of components.

## 5.9 Conclusion

In this chapter, we introduce a new projected-baed classifier, Envelope-based Support Vector Machine (ESVM). This work was inspired by the efficiency that envelope-based techniques achieved in regression. Our approach assumes that the misclassification rate increases if the data includes redundant variables. Hence, excluding these variables advances the classifier generalization. We have discussed the algorithm for extracting the reducing subspace that contains the classification-related information. To evaluate the method, we conducted several settings of simulated data. The simulation scenarios addressed issues such as: level of multicollinearity among the predictor variables and the relation between the sample size and the number of predictor variables. Our method has shown promising results when compared to other classifiers: classical SVM, LDA, and logistic regression. We have demonstrated the efficiency of our classifier via different settings of simulated data as well as in real data.

# Chapter 6

# Sparse Envelope Based Support Vector Machine

## 6.1 Introduction

In this chapter, we further extend our ESVM classifier to accommodate the sparsity problem. An important feature of a good classifier is the ability to predict the class of future data with a low misclassification rate. However, due to the large $p$ small $n$ setting, many classifiers may perform well in the training data, but it generalizes poorly with the test data (Bradley and Mangasarian, 1998). A common problem that affects classification performance is sparsity, which implies that only a small portion of the predictor variables contributes to the classification procedure. Keeping the irrelevant variables while performing the classification reduces the accuracy of classifications, (Merchante et al. (2012), Guyon et al. (2002), Tibshirani et al. (2002),Mai et al. (2012),Clemmensen et al. (2011)). In such a situation, one may aim to reduce the data dimensionality by deducting the classification-related subset of the predictor variables and base the classification on this subset. In classification problems, the reduction of the dimension of the features is approached in two ways. The first methodology is to add a penalty to the objective function in such a way the penalty works to estimate sparse coefficients. This is known as regularization or shrinkage, the technique by which the estimated coefficients of non-significant predictor variables shrink towards zero by the added penalty to the objective function. Precisely, the natural approach to tackle the sparsity problem is to regularize the weight associated with the non-significant variables. One may employ one of the various penalties that exist in statistical literature; see Chapter 2. The other way is to process the reduction via the projection technique. In other words, the initial step is preserved to estimate a new subspace with dimension $(d < p)$ that contains all the required information about the classification and then project the data into this subspace. The sparse principal component analysis (Zou et al., 2006) is a common dimension reduction technique in the presence of sparsity.

In Chapter 5, we introduced an envelope-based support vector machine classifier, in which we show the procedure of constructing a projection matrix. This projection matrix is used to reduce the dimensionality of the data prior to the classification. In this chapter, we modified our method to accommodate the sparseness in the data. We added an adaptive group lasso penalty to impose the sparsity in the data. That is, our approach is meant to perform variable selection and feature extraction simultaneously.

This chapter is organized as follows: in Section 6.2 we review the methods that handle sparsity in classification. In Section 6.3, we introduce our classifier and demonstrate the classification algorithm. Section 6.4 we conduct different simulation settings to evaluate our classifier's performance.

## 6.2 Sparsity in classification

Sparsity implies that a small portion of the input predictor variables is related to the classification process. Researchers have proposed various algorithms to address this problem and improve classification accuracy. Among others, Tibshirani et al. (2002) proposed a computational technique for classification and variables selection based on modifying the nearest centroid classifier. They proposed a threshold by which each class centroid is shrunken. Hence, a new sample is trained following the classic nearest centroid algorithm using the shrunken class centroids. This algorithm works as a classifier and variable selection. That is, if a new sample hits zero for all classes, then it is eliminated. Fan and Fan (2008) proposed a new classifier, namely Features Annealed Independence Rules (FAIR). The algorithm employs a component-wise t-test between two classes to select the significant features. That is, for each feature $j$, the following t-test is calculated:

$$T_j = \frac{\bar{x}_{1j} - \bar{x}_{2j}}{\sqrt{S_{1j}^2/n_1 + S_{2j}^2/n_2}}, \tag{6.1}$$

where $\boldsymbol{x}_{kj}$ is the $j^{th}$ feature from class $k$, $n_k$ is the number of observations in class $k$, and $\bar{x}_{kj}$, $S_{kj}$ are the sample mean and variance of the $j^{th}$ feature in class $k$ , $k = 1, 2$, respectively. The absolute values of the calculated t-statistics of the features are sorted in decreasing order based on the importance of the features. The new observation $\boldsymbol{x}$ is classified into class 1 if the decision function $\delta(\boldsymbol{x}) > 0$, where $\delta(\boldsymbol{x})$ is defined as follows:

$$\delta_{\text{FAIR}}(\boldsymbol{x}) = \begin{cases} \sum_{j=1}^{p} \hat{\alpha}_j (x_j - \hat{\mu}_j) I_{\{|\hat{\alpha}|_j > b\}} & \text{if } \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}, \\ \sum_{j=1}^{p} \hat{\alpha}_j (x_j - \hat{\mu}_j) / \hat{\sigma}^2 I_{\{\sqrt{n/(n_1 n_2)|T_j| > b}\}} & \text{otherwise,} \end{cases} \tag{6.2}$$

where $\alpha_j = \hat{\mu}_{1j} - \hat{\mu}_{2j}$, $b$ is a threshold and $T_j$ is the two sample t-statistics. However, the drawback of the above explained algorithms is that these methods do not count for correlation among features.

Clemmensen et al. (2011) developed a sparse discriminant analysis (SDA), a method for performing linear discriminant analysis with sparseness criterion that handles classification and feature selection simultaneously. The proposed method is a sparse version of linear discriminant analysis (LDA) based on LASSO penalty (Tibshirani, 1996). LDA can be seen as originating from Fisher's discriminant analysis, which involves finding discriminative $p$ dimensional vectors $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{K-1}$. The discriminative vectors can be successively maximized over the following objective function:

$$\begin{aligned} &\text{maximize}_{\boldsymbol{\beta}_k} \{\boldsymbol{\beta}_k^T \boldsymbol{\Sigma}_b \boldsymbol{\beta}_k\} \\ &\text{subject to} \quad \boldsymbol{\beta}_k^T \boldsymbol{\Sigma}_w \boldsymbol{\beta}_k = 1, \quad \boldsymbol{\beta}_k^T \boldsymbol{\Sigma}_w \boldsymbol{\beta}_l = 0 \quad \forall l < k, \end{aligned}$$

where $\boldsymbol{\Sigma}_b = \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T$ is the between class covariance matrix, where $\pi_k$ is the prior probability for class $k$ and $\boldsymbol{\Sigma}_w$ is the within class covariance matrix. The SDA makes use of optimal scoring technique (Hastie et al., 1995), and developed a sequential sparse criterion to solve for the $k^{th}$ pair of the parameters $(\boldsymbol{\beta}_k, \boldsymbol{\theta}_k)$. The optimal scoring method reshapes the classification problem in a regression setting by transferring the categorical variable $\boldsymbol{y}$ into a quantitative variable via sequence of scoring. That is, the optimal scoring optimisation is formulated as follows:

$$\begin{aligned} &\text{minimize}_{\boldsymbol{\beta}_k, \boldsymbol{\theta}_k} \{||\boldsymbol{Y}\boldsymbol{\theta}_k - \boldsymbol{X}\boldsymbol{\beta}_k||^2 \\ &\text{subject to} \quad \frac{1}{n} \boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{\theta}_k = 1, \quad \boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{\theta}_l = 0, \quad \forall l < k, \end{aligned}$$

where $\boldsymbol{Y}$ is a $n \times K$ matrix of dummy variables for the $K$ classes such that $y_{ik}$ indicates whether the $i^{th}$ observation belongs to class $k$, $\boldsymbol{\theta}$ is a $K$ dimentional vector of scores. Hence, the $k^{th}$ sparse discriminate solution pair $(\boldsymbol{\beta}_k, \boldsymbol{\theta}_k)$ is found via solving the optimisation:

$$\begin{gathered}\text{minimize}_{\boldsymbol{\beta}_k, \boldsymbol{\theta}_k} \{||\boldsymbol{Y}\boldsymbol{\theta}_k - \boldsymbol{X}\boldsymbol{\beta}_k||^2 + \gamma \boldsymbol{\beta}_k^T \boldsymbol{\Omega} \boldsymbol{\beta}_k + \lambda ||\boldsymbol{\beta}_k||1\} \\ \text{subject to} \frac{1}{n} \boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{\theta}_k = 1, \boldsymbol{\theta}_k^T \boldsymbol{Y}^T \boldsymbol{Y} \boldsymbol{\theta}_l = 0, \forall l < k,\end{gathered}$$

where $\lambda$ and $\gamma$ are tuning parameters.

Mai et al. (2012) have proposed another approach of sparse discriminant analysis for feature selection and classification. The method differs from the former in that it was motivated by the relation between least squares formulation and linear discriminant analysis and used for binary classes problems. The LASSO penalty was added to induce the sparsity. Hence the penalized least squares solutions found as:

$$(\hat{\boldsymbol{\beta}}_{\text{lasso}}, \hat{\beta}_0) = \arg\min_{\beta, \beta_0} \left\{ n^{-1} \sum_{i=1}^{n} (y_i - \beta_0 - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{6.3}$$

The new sample $\boldsymbol{x}$ is placed in class 2 if:

$$\hat{\boldsymbol{\beta}}_{\text{lasso}}^T \boldsymbol{x} + \hat{\beta}_0 > 0. \tag{6.4}$$

On the other hand, sparse support vector machines has been proposed as good candidate to classify high dimensional data. For instance Guyon et al. (2002) proposed a SVM-based technique to feature selection and classification simultaneously. Namely, Recursive Feature Elimination (RFE) support vector machines. The RFE is an instance of backward elimination. This classifier is defined as follows: train the data based on SVM then rank the features based on a chosen ranking criterion. Select a predetermined small number of the features that gave the best ranking. The features were ranked based on sensitivity analysis or correlation to the outcome. The drawback of this classifier is that the recursive feature elimination method is inconsistent with the maximal margin solution(Aksu et al., 2010). SVM is not meant to do variable selection, however, adding appropriate penalty to the objective function makes it possible for SVM to do variable selection indirectly.

Bradley and Mangasarian (1998) have modified classical SVM by adding $\ell_1$ penalty which obtained sparse coefficient in the solution. This algorithm handles binary classes data. Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be the data matrix, further assume $m$ observations allocate in the first class and represented by $\boldsymbol{A}$ while $f$ observations allocate in the second class and represented by $\boldsymbol{B}$, where $m + f = n$. An observation $\boldsymbol{x}$ belongs to $\boldsymbol{A}$ if $\boldsymbol{w}^T\boldsymbol{x} > b$ and belongs to $\boldsymbol{B}$ if $\boldsymbol{w}^T\boldsymbol{x} < b$, where $\boldsymbol{w}$ is the vector of weights while $b$ is the offset. The coefficients are estimated by minimizing the following objective:

$$\min_{\boldsymbol{w},b} g(\boldsymbol{w}, b) = \min_{\boldsymbol{w},b} \frac{1}{m}||(-\boldsymbol{A}\boldsymbol{w} + \boldsymbol{e}b + \boldsymbol{e})_+||_1 + \frac{1}{f}||(\boldsymbol{B}\boldsymbol{w} - \boldsymbol{e}b + \boldsymbol{e})_+||_1,$$

where $\boldsymbol{e}$ is a vector of ones (with arbitrary dimension), $(x)_+ = \max\{0, x\}$, and $||.||_1$ is the $\ell_1$ norm. The sparseness is obtained by introducing a tuning parameter $\lambda \in [0, 1)$ such that the objective function becomes:

$$\begin{aligned}
&\text{minimize}_{\boldsymbol{w},b,y,z}(1-\lambda)\left(\frac{\boldsymbol{e}^T\boldsymbol{y}}{m} + \frac{\boldsymbol{e}^T\boldsymbol{z}}{f}\right) + \lambda\boldsymbol{e}^T|\boldsymbol{w}|_*\\
&\text{subject to} \quad -\boldsymbol{A}\boldsymbol{w} + \boldsymbol{e}b + \boldsymbol{e} < \boldsymbol{y},\\
&\qquad\qquad\quad \boldsymbol{B}\boldsymbol{w} - \boldsymbol{e}b + \boldsymbol{e} < \boldsymbol{z},\\
&\qquad\qquad\quad \boldsymbol{y} \geq \boldsymbol{0}, \quad \boldsymbol{z} \geq \boldsymbol{0},
\end{aligned}$$

where $|\boldsymbol{w}|_* \in \mathbb{R}^p$ has entries equal 1 if the corresponding entries in $\boldsymbol{w}$ is nonzero and zero if the corresponding entries. Generally speaking, the term $(\boldsymbol{e}^T|\boldsymbol{w}|_*)$ counts the number of nonzero features, where the feature is excluded if the entry of $\boldsymbol{w}$ is zero. The fundamental drawback of this method is that it discards the correlation among features (Zhu and Zou, 2007).

Gómez-Verdejo et al. (2011) proposed a modified SVM to perform variable selection and classification simultaneously. Their adjustment based on adding a new slack variable that can be used to variable selection. That is, consider the classical SVM minimisation in (4.14), they assumed the weight is written as $\boldsymbol{w} = \boldsymbol{u} + \boldsymbol{v}$. Further, they introduce a new slack variable $\xi^*$ associate with the features such that the features whose slack variables equal to zero can be illuminated. The determination of the value of $\xi^*$ is based on a small number $\epsilon$ such that if the absolute value of the weight associated with a feature $j$ is greater than $\epsilon$ then the corresponding $\xi_j^*$ is not zero and zero otherwise. The new minimisation becomes:

$$\begin{aligned}
\arg\min \sum_{j=1}^{p}(u_j^2+v_j^2)+\gamma\sum_{i=1}^{n}\xi_i+\gamma^*\sum_{j=1}^{p}\xi_j^* \\
\text{subject to } y_i(\sum_{j=1}^{p}(u_j^2+v_j^2)x_j^{(i)}+b)\geq 1-\xi_i \\
\xi_i\geq 0; \quad \forall i \\
u_j+v_j\leq \epsilon+\xi_j^*; \quad \forall j \\
u_j+v_j\geq 0; \quad \forall j \\
\xi_j^*\geq 0; \quad \forall j.
\end{aligned} \tag{6.5}$$

In the next section we introduce the sparse ESVM.

## 6.3 Sparse Envelope Support Vector Machine (SESVM) classifier

Several implementations in sparse support vector machine have been developed. In this section, we formulate the Sparse Envelope-based Support Vector Machine (SESVM) classifier. The developed classifier is an extension to the ESVM classifier, which was introduced in Chapter 5; however, the main difference is the modification in the basis extraction algorithm to produce a sparse projection matrix. The modification is justified via adding adaptive group LASSO penalty to the objective function to induce the sparsity in the projection matrix. The classification mechanism of SESVM is in a similar fashion to ESVM. That is, once the projection basis is obtained, the first step we project the data onto it. Then after projecting the data onto the lower-dimensional subspace, we perform the classic SVM classifier. Generally speaking, our algorithm performs variable selection and feature extraction simultaneously as an initial step. The next step is to classify the reduced data based on SVM.

To formulate the model, suppose the predictors $\boldsymbol{X}\in\mathbb{R}^{p\times n}$, encompasses of two distinguishable groups of predictor variables: active and inactive predictor variables. A predictor variable is characterised as an active variable if it has non zero coefficient and characterised as inactive otherwise. Further, let $\boldsymbol{X}_{\mathcal{A}}\in\mathbb{R}^{p_{\mathcal{A}}\times n}$ represents the active predictor variables and $\boldsymbol{X}_{\mathcal{I}}\in\mathbb{R}^{p_{\mathcal{I}}\times n}$ represents the inactive predictor variables such that $\boldsymbol{X}=(\boldsymbol{X}_{\mathcal{A}}^T,\boldsymbol{X}_{\mathcal{I}}^T)^T$, where $p_{\mathcal{A}}$ is the number of active predictor variables, $p_{\mathcal{I}}$ is the number of inactive predictor variables such that $p = p_{\mathcal{A}} + p_{\mathcal{I}}$. Now, suppose $(\mathbf{\Gamma}, \mathbf{\Gamma}_0) \in \mathbb{R}^{p \times p}$ is an orthogonal basis where $\mathbf{\Gamma} \in \mathbb{R}^{p \times u}$ and $\mathbf{\Gamma}_0 \in \mathbb{R}^{p \times (p-u)}$ such that:

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{\Gamma}_{\mathcal{A}} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{\Gamma}_0 = \begin{bmatrix} \mathbf{\Gamma}_{\mathcal{A}0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_{p_{\mathcal{I}}} \end{bmatrix}, \tag{6.6}$$

where $\mathbf{\Gamma}_{\mathcal{A}} \in \mathbb{R}^{p_{\mathcal{A}} \times u}$ constitutes the relevant active predictors, the zeros correspond to the inactive variables, $\mathbf{\Gamma}_{\mathcal{A}0} \in \mathbb{R}^{(p_{\mathcal{A}}-u) \times (p_{\mathcal{A}}-u)}$. Further, the coordinate $\mathbf{\Omega}_0$ has the following structure:

$$\mathbf{\Omega}_0 = \begin{bmatrix} \mathbf{\Omega}_{\mathcal{A}0} & \mathbf{\Omega}_{\mathcal{A}\mathcal{I}0} \\ \mathbf{\Omega}_{\mathcal{I}\mathcal{A}0} & \mathbf{\Omega}_{\mathcal{I}0} \end{bmatrix}, \tag{6.7}$$

where $\mathbf{\Omega}_{\mathcal{A}0} \in \mathbb{R}^{(p_{\mathcal{A}}-u) \times (p_{\mathcal{A}}-u)}$. The correlation between the two sources of the immaterial part is preserved in $\mathbf{\Omega}_{\mathcal{A}\mathcal{I}0}$ if $\mathbf{\Omega}_{\mathcal{A}\mathcal{I}0} \neq 0$ (Zhu and Su, 2020).

It is worth noting that the active predictor variables are categorized into material and immaterial based on their contribution to the outcome. The material part is the components of the active variables that are relevant to the outcome. While the immaterial part $\boldsymbol{Q}_{\mathcal{E}} = \boldsymbol{P}_{\Gamma_0}\boldsymbol{X}$ decomposes into: $(\boldsymbol{X}_{\mathcal{A}}^T\boldsymbol{Q}_{\Gamma_{\mathcal{A}}}, \boldsymbol{X}_{\mathcal{I}}^T)^T$. That is, the immaterial part comes from two sources: the immaterial part in the active variables $(\boldsymbol{X}_{\mathcal{A}}\boldsymbol{Q}_{\Gamma_{\mathcal{A}}})$ while the second source is the inactive predictors $(\boldsymbol{X}_{\mathcal{I}})$ (Zhu and Su (2020) and Chun and Keleş (2010)). Hence, we assume a grouping structure; that is, the predictor variables could be divided into disjoint groups, and the same group can predict the class of an individual. The proposed algorithm aims to estimate the SESVM basis $\mathbf{\Gamma}$ such that the basis has to reflect the sparseness in the data. That is, the predictor variable is considered inactive if the corresponding row in the basis is represented by zeros (Chun and Keleş (2010) and Zhu and Su (2020)). This sparsity representation is well known in the literature, and it means that these variables represented by zeros are not informative to the outcome.

In Chapter 5, we have shown the likelihood-based objective function to construct the ESVM basis $\mathbf{\Gamma}$. The SESVM is an extension of ESVM that handles sparse data. Hence, to induce the the sparsity in SESVM basis, we added adaptive group LASSO penalty (Yuan and Lin, 2006) to the objective function in (5.6). That is,

$$L(\Gamma) = \arg \min_{\Gamma \in \mathcal{G}(u,p)} \left\{ \log|\mathbf{\Gamma}^T \boldsymbol{S}_X^{-1} \mathbf{\Gamma}| + \log|\mathbf{\Gamma}_0^T \boldsymbol{S} \mathbf{\Gamma}_0| + \lambda \sum_{i=1}^{p} w_i ||\gamma_i||_2 \right\}, \tag{6.8}$$

where $\lambda$ is tuning parameter, $\boldsymbol{\gamma}_i$ is the $i^{th}$ row of $\boldsymbol{\Gamma}$, and $w_i$ is the weight associated to the $i^{th}$ predictor. The weight is given by: $w_i = 1/||\hat{\gamma}_i||_2^{\tau}$, where $\hat{\gamma}_i$ is an estimate of $\gamma_i$ which can be found by ESVM and $\tau$ is a tuning parameter can be chosen from $\{0.5, 1, 2, 4, 8\}$ (Zou, 2006). The weight plays a role in the regularization process, that is, inactive variables has smaller weight that taking its reciprocal maximizes the penalty and drags it to zero. The adaptive group LASSO penalty differs from LASSO in that it sets group of variables to zero instead of single predictor; hence instead of penalizing one coefficient it penalizes group of them. Furthermore, by adding the weight it solves the estimates inefficiency as well as variable selection inconsistency (Wang and Leng, 2008). This penalty has been used by several researches (Meier et al., 2008).

In a situation when the number of predictor variables exceeds the number of observations, $S_X$ becomes singular. However, $S_X^{-1}$ appears in (6.8) and required in the algorithm to estimate SESVM basis, hence, we substitute this by Sparse Permutation Covariance Estimator (SPICE), (Rothman et al., 2008). The SPICE is used for its simplicity as it does not require sparsity in $S_X$. Thus, (6.8) becomes:

$$L(\Gamma) = \arg \min_{\Gamma \in \mathcal{G}(u,p)} \left\{ \log |\boldsymbol{\Gamma} \boldsymbol{S}_{X_{\text{SPICE}}}^{-1} \boldsymbol{\Gamma}^T| + \log |\boldsymbol{\Gamma}_0 \boldsymbol{S} \boldsymbol{\Gamma}_0^T| + \lambda \sum_{i=1}^{p} w_i ||\boldsymbol{\gamma}_i||_2 \right\}. \tag{6.9}$$

The objective function in (6.9) is optimized over Grassmann Manifold to estimate $\boldsymbol{\Gamma}$. Once the basis $\boldsymbol{\Gamma}$ is found, the data is reduced by projecting it onto $\boldsymbol{\Gamma}$. Similar to what has been explained in Chapter 5, the SVM classifier is performed to the reduced data in the same manner.

## 6.4 Numerical studies

We are interested in the effect of dimension reduction and variable selection in classification accuracy. That is, we would like to investigate the difference that excluding inactive variables and reducing the dimension of the data make in the classification performance. In this section we illustrate the performance of our proposed classifier. Various settings of simulation studies have been conducted. The method have been tested on public real data as well and was compared against other classifiers: sparse LDA (SpLDA), sparse logistic regression (spLR), and SVM.

### 6.4.1 Simulation design

Seven different datasets were generated. For each setting, the observations are assumed to be generated from $Z$ classes, each class has $n_z$ observations such that $\sum_{z=1}^{Z} n_z = n$. The class observations are generated from normal distribution with class mean $\boldsymbol{\mu}_z$ and shared covariance matrix $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = ... = \boldsymbol{\Sigma}_Z = \boldsymbol{\Sigma}$, $\boldsymbol{x}_{iz} \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma})$, $\quad i = 1, ..., n; z = 1, ..., Z$. The class mean $\boldsymbol{\mu}_z = \boldsymbol{\Gamma}\boldsymbol{\eta}_z$ and the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T$. The semi-orthogonal matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times u}$ was generated with same structure as in (6.6). That is, the entries of $\boldsymbol{\Gamma}$ can be classified into two groups: active variables $\boldsymbol{\Gamma}_{\mathcal{A}}$, whose elements represent the active variables and inactive variables $\boldsymbol{\Gamma}_{\mathcal{I}}$ which represents the inactive variables and all zeros. The elements for $\boldsymbol{\Gamma}_{\mathcal{A}} \in \mathbb{R}^{p_{\mathcal{A}}\times u}$ is generated randomly from standard normal distribution. Following orthogonalizing $\boldsymbol{\Gamma}$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-u)}$ was generated. We assume homogenous classes, the shared covariance matrix for each class $\boldsymbol{\Sigma}$ is generated as $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T$. The positive definite matrices $\boldsymbol{\Omega} \in \mathbb{R}^{u\times u}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(p-u)\times(p-u)}$ were generated as diagonal matrices, where $\boldsymbol{\Omega}_0$ follows same structures in (6.7). That is, $\boldsymbol{\Omega} = \tau_1 \boldsymbol{I}_u$ and $\boldsymbol{\Omega}_0 = (\tau_2 \boldsymbol{I}_{p_{\mathcal{A}}}, \boldsymbol{I}_{p_{\mathcal{I}}})$. The mean of each class is calculated as $\boldsymbol{\mu}_z = \boldsymbol{\Gamma}\boldsymbol{\eta}_z$, where $\boldsymbol{\eta}_z$ is a $u$ dimensional vector generated from standard normal distribution. The data generation and the simulation process can be summurized as follows:

1. Fix the values: sample size ($n$), number of predictors ($p$), envelope subspace dimension ($u$), number of active variables ($p_{\mathcal{A}}$), and the number of classes ($Z$).

2. Initiate the parameters $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0, \boldsymbol{\Omega}, \boldsymbol{\Omega}_0, \boldsymbol{\eta})$ such that $\boldsymbol{\Gamma}$ has the following sparse structure:
$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{\Gamma}_{\mathcal{A}} \\ \boldsymbol{0} \end{bmatrix}$$

3. Generate the label $Y$ and the data $\boldsymbol{X} \sim N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma})$.

4. Divide the date into 80% training and 20% testing data.

5. Estimate the sparse ESVM basis $\hat{\boldsymbol{\Gamma}}$ from the training data over Grassmann manifold via optimizing equation (6.8) over a range of hyper parameter $\lambda$. The Gamma associated with the optimal $\lambda$ was chosen based on cross validation.

6. Reduce the full data by projecting it onto $\boldsymbol{\Gamma}$, that is $\boldsymbol{X}^* = \boldsymbol{\Gamma}^T \boldsymbol{X}$.

7. Apply SVM algorithm to full and reduced data, as well as spLDA and spLR then report the accuracy of classification.

Table 6.1 shows the different simulation settings.

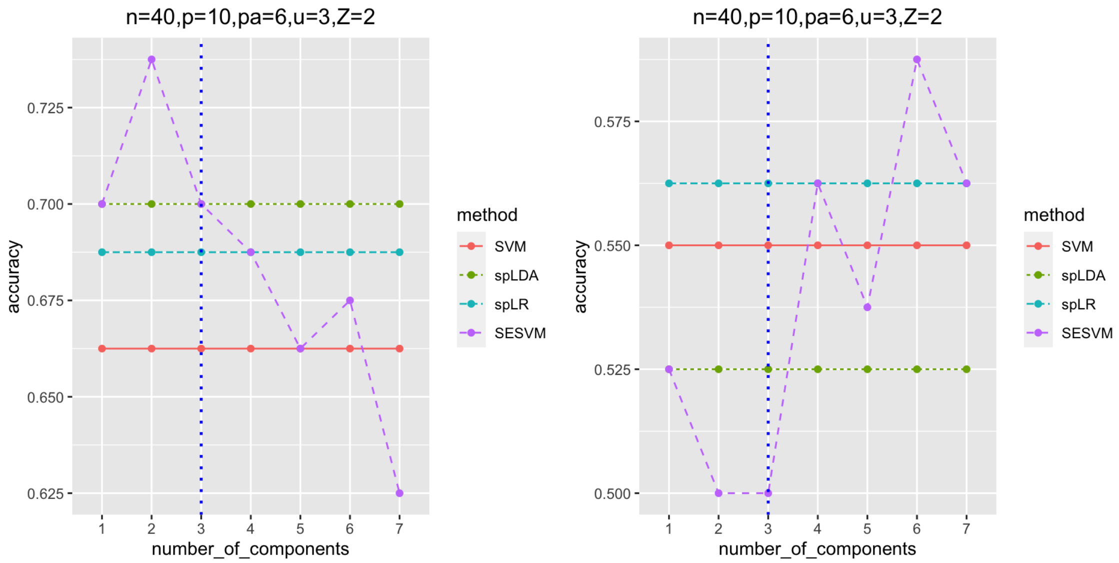
| n | p | $p_{\mathcal{A}}$ | u | z | $\tau_1$ | $\tau_2$ |
|---|---|---|---|---|---|---|
| 40 | 10 | 6 | 3 | 2 | 2 | 0.2 |
| | | | | 2 | 20 | 0.5 |
| 40 | 45 | 10 | 4 | 2 | 2 | 0.2 |
| | | | | | 10 | 0.5 |
| 120 | 12 | 6 | 4 | 3 | 10 | 0.5 |
| | | | | | 5 | 0.4 |
| 300 | 30 | 8 | 3 | 4 | 4 | 0.3 |

Table 6.1: The different choices of the model parameters for the simulation.

### 6.4.2 The classification performance

In this section, we illustrated the outcome of the simulations. As shown in Table 6.1, we considered different scenarios when $p < n$ and one scenario for $p > n$ because the later is computationally expensive. For each setting, 10 samples were generated. For each replicate, our method, was compared against existing classifiers: SVM, SpLDA, and spLR. The classification performance was measured via classification accuracy. For SESVM, the classification accuracy was calculated at various number of components, while for other classifiers, the evaluation is based on full dimensioned data. The average classification accuracy for each classifier is reported.

In the first setting, we generated $n = 40$ observations allocated randomly into two equal classes. The number of predictor variables is assumed to be ($p = 10$), out of which only 6 variables are active. The number of components is $u = 3$ components. This setting was generated with two choices of $(\tau_1, \tau_2)$, that is, the choice of $(\tau_1, \tau_2)$ manifested the collinearity among the predictor variables (Cook et al., 2013). Figure 6.1 shows the average classification accuracy for the above-mentioned classifiers. The figure shows that the methods perform similarly. However, in 6.1a, where the collinearity is moderate, while figure 6.1b, where the collinearity is high. In both scenarios, we see that SESVM is slightly better compared to other classifiers.



(a) (b)

Figure 6.1: Classification accuracy for simulated datasets $n = 40, Z = 2, u = 3, p = 10, p_{\mathcal{A}} = 6$, a $(\tau_1, \tau_2) = 2, 0.2$, b $(\tau_1, \tau_2) = 20, 0.5$. The dashed vertical line is the true number of components..

The second setting of the simulation study explores the case $p > n$, we generated $n = 40$ observations allocated randomly into two equal classes. In this setting, the number of the predictor variables was chosen to be greater than the class size as well as the sample size, $p = 45$. The number of the active variables is 10 variables. Figure 6.2 shows the average classification accuracy. In the first scenario, it can be seen that SESVM achieved less miss-classification than other classifiers. However, for the other scenario, it performed competitively with sparse LDA.

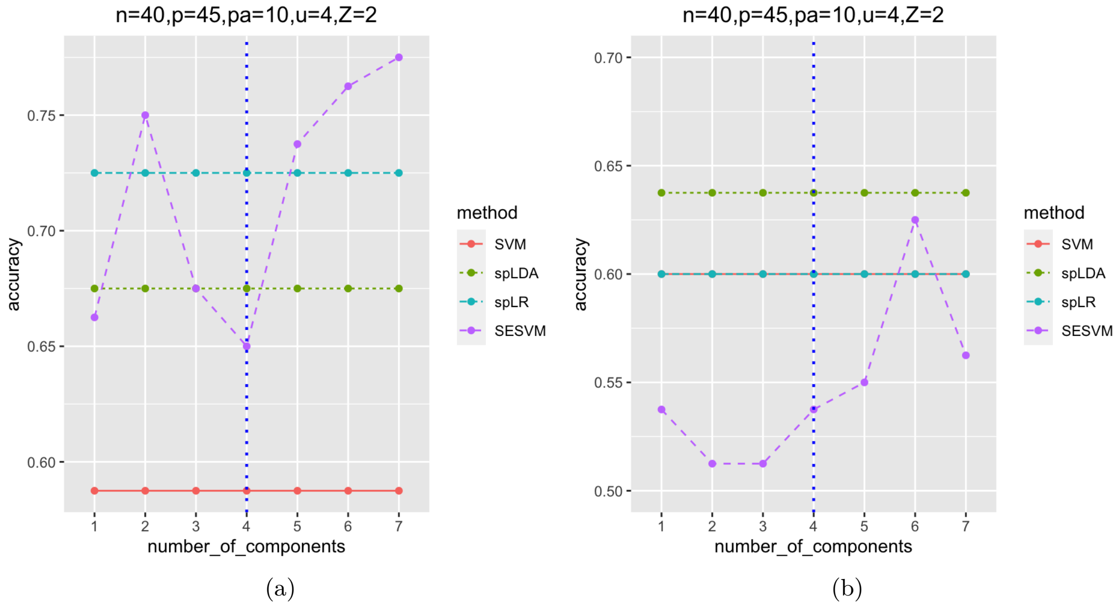
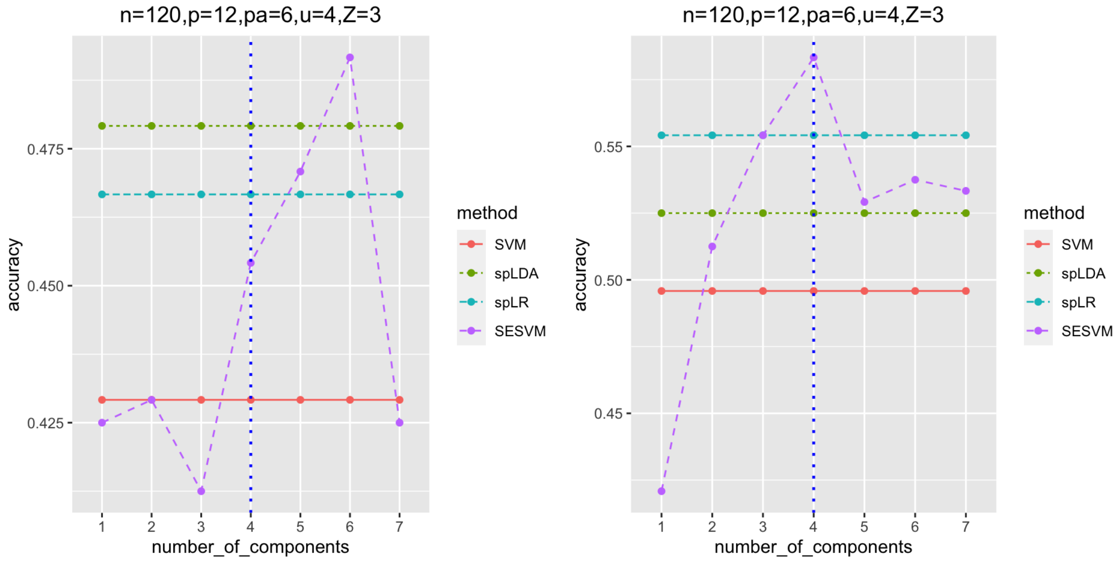


Figure 6.2: Classification accuracy for simulated datasets $n = 40, Z = 2, u = 4, p = 45, p_{\mathcal{A}} = 10$, a $(\tau_1, \tau_2) = 2, 0.2$, b $(\tau_1, \tau_2) = 10, 0.5$. The dashed vertical line is the true number of components.

Figures 6.3 and 6.4 show the average classification accuracy for the last two settings in Table 6.1. Figure 6.3 summarizes the performance when $n = 120$. In this setting, we have two scenarios of collinearity among predictor variables. While Figure 6.4 shows the last model in our simulation when $n = 300$. In both figures, the performance of SESVM is better than other classifiers. The performance of SESVM fluctuated as the number of components changed; however, it achieved its best performance at or near the true number of components.



(a)

(b)

Figure 6.3: Classification accuracy for simulated datasets $n = 120, Z = 3, u = 4, p = 12, p_{\mathcal{A}} = 6$, a $(\tau_1, \tau_2) = 10, 0.5$, b $(\tau_1, \tau_2) = 5, 0.4$. The dashed vertical line is the true number of components.

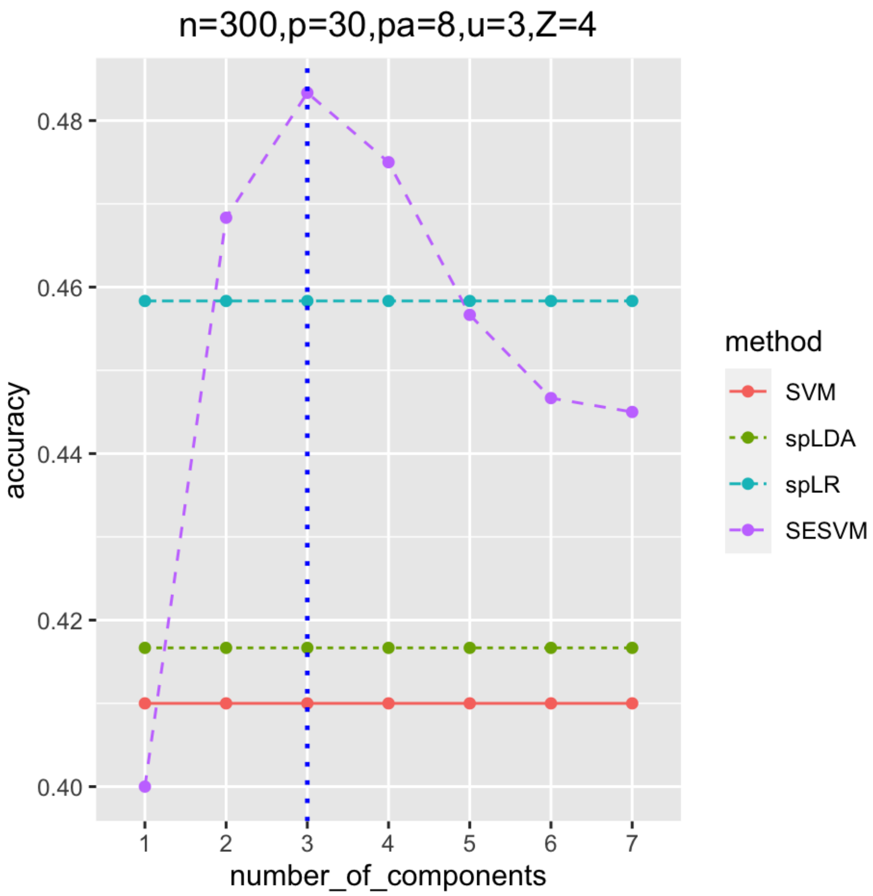


Figure 6.4: Classification accuracy for simulated datasets $n = 300, Z = 4, u = 3, p = 30, p_{\mathcal{A}} = 8$ $(\tau_1, \tau_2) = 4, 0.3$. The dashed vertical line is the true number of components.

### 6.4.3 Real data

In this section we applied the our proposed classifier to public real data. Similar to the simulation studies, the focus was on the performance of the SESVM. We set a comparison against SVM, spLDA, and spLR and calculate the classification accuracy as performance measure.

**Parkinson data:**

The Parkinson data (Naranjo et al., 2016) contains the voice recording for 80 individuals (48 male and 32 female) 40 of them classified as healthy and 40 were classified as Parkinson diseased (PD), as shown in Table 6.2. The data has $p = 45$ predictor variables used to predict the class of each participate (healthy, PD). The data was divided into 80%

training data and 20% testing data then classified based on SESVM and calculate the classification accuracy for the values of $u = (1, ..., 8)$. The data were classified based on SpLDA and sparse logistic regression as well. Figure 6.5 shows the classification accuracy for each classifier. It is can be seen that SESVM slightly better than other classifiers with $u = 6$ components.

| status | Male | Female |
|---|---|---|
| Healthy | 22 | 18 |
| PD | 26 | 14 |

Table 6.2: Summary of Parkinson dataset.

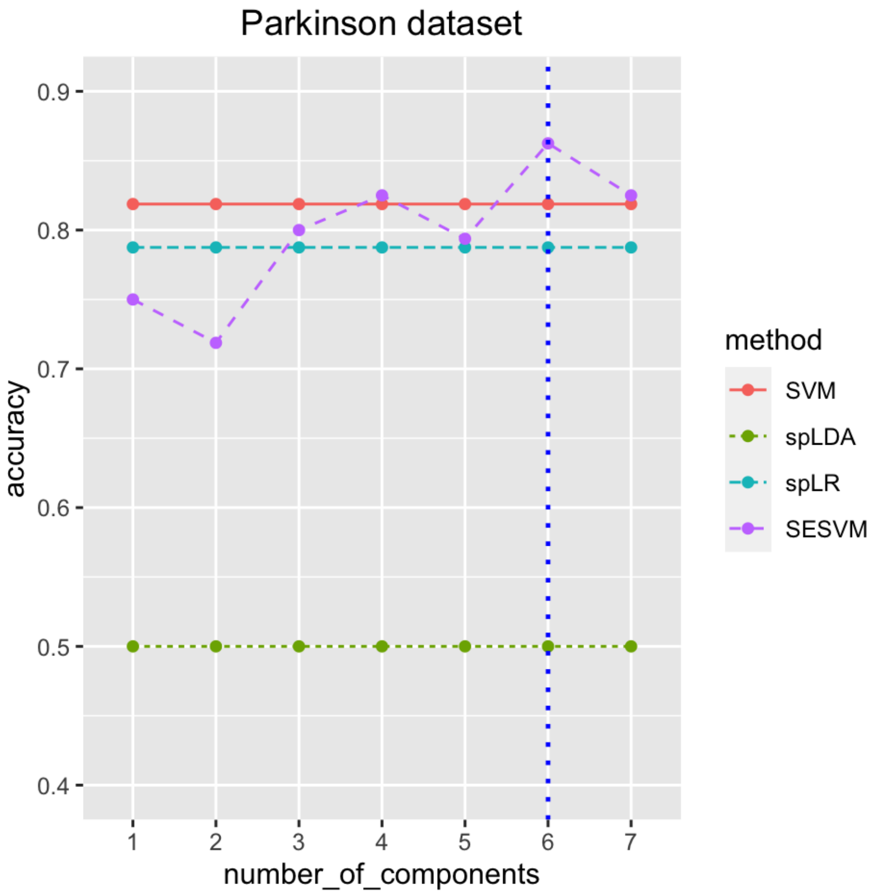


Figure 6.5: The classification accuracy for Parkinson data. The dashed vertical line is the estimated number of components.

## 6.5 ESVM and SESVM

In this section we compare the performance of our proposed classifiers, namely envelope-based support vector machines and sparse envelope-based support vector machines .That is, we aim to evaluate the classification efficiency of the proposed classifiers in present and absent of sparsity. We selected two scenarios for each setting as shown in Tables 6.3 and 6.4. The data for the former setting was generated as explained in Section 5.8. Similarly, the data for the later setting was generated as explained in 6.4. For each setting, 50 samples were generated then we compared the performance of the classifiers based on the classification accuracy.

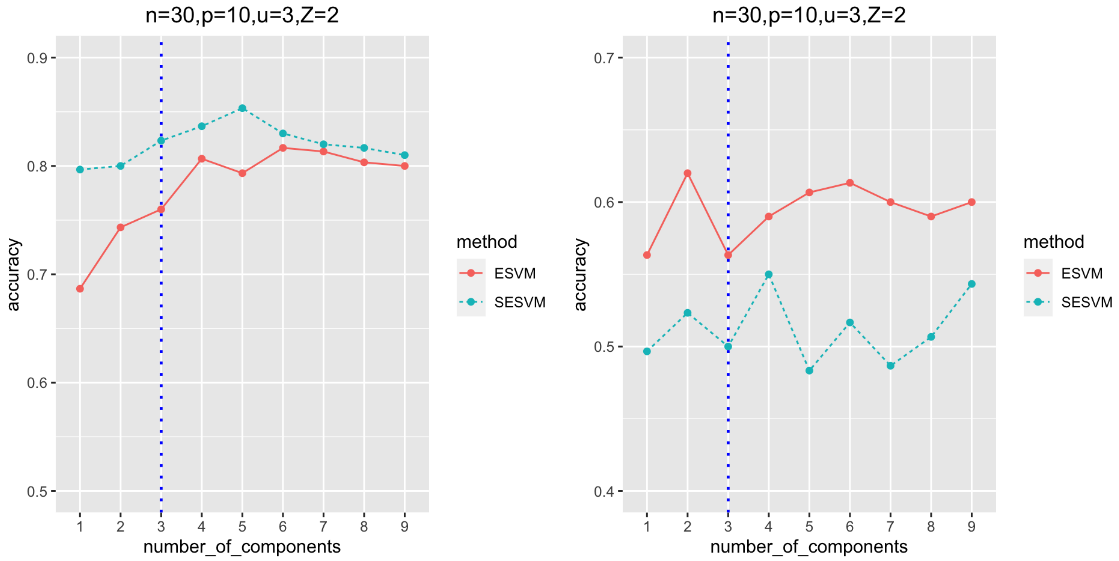
Figures 6.6 and 6.7 summarized the performance of the classifiers in the absent of sparsity, while Figures 6.8 and 6.9 summarized the performance in present of sparsity. Both classifiers perform similarly. However, the sparse envelope-based support vector machines classifier can handle the case were $p > n$ while envelope-based support vector machines breaks down.

| n | p | u | z | $\tau_1$ | $\tau_2$ |
|---|---|---|---|---|---|
| 30 | 10 | 3 | 2 | 2 | 0.2 |
| | | | 2 | 20 | 0.4 |
| 120 | 15 | | 3 | 2 | 0.2 |
| | | 6 | 3 | 4 | 0.25 |

Table 6.3: The simulation design for non sparse data.

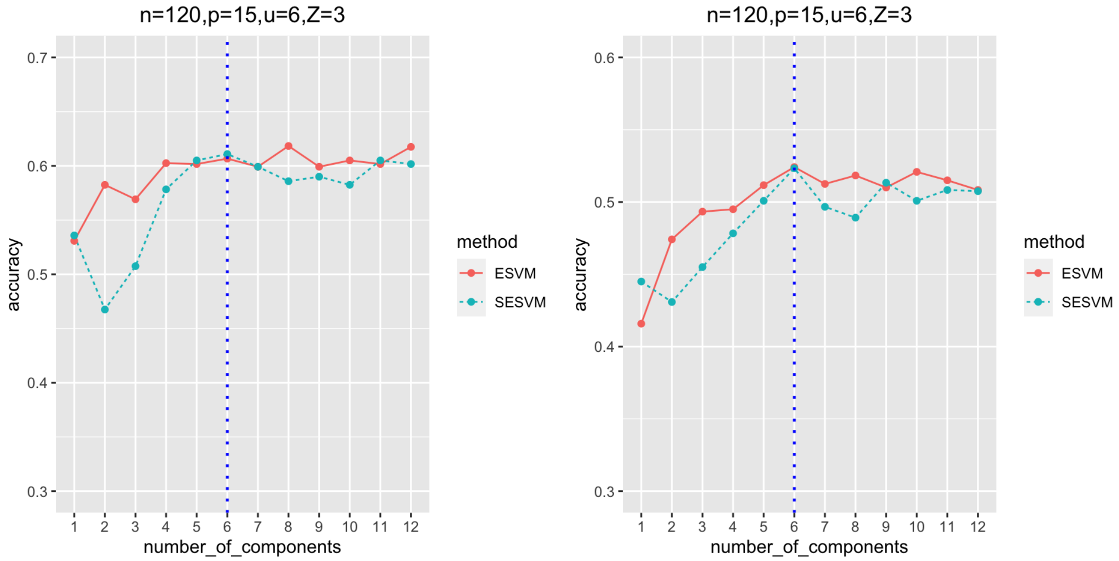
| n | p | $p_{\mathcal{A}}$ | u | z | $\tau_1$ | $\tau_2$ |
|---|---|---|---|---|---|---|
| 40 | 10 | 6 | 3 | 2 | 2 | 0.2 |
| | | | | 2 | 20 | 0.5 |
| 120 | 12 | 6 | 4 | 3 | 10 | 0.5 |
| | | | | | 5 | 0.4 |

Table 6.4: The simulation design for sparse data.



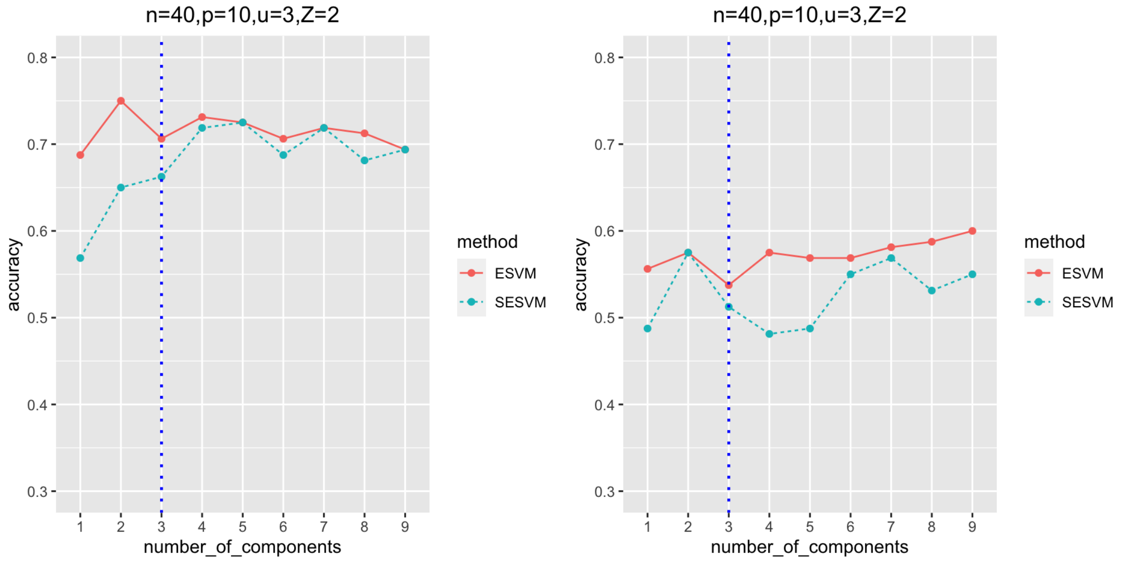
(a) (b)

Figure 6.6: The average classification accuracy percentage based on ESVM and SESVM. a: sample size $n = 30, p = 10, Z = 2, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 30, p = 10, Z = 2, (\tau_1, \tau_2) = (20, 0.4)$.



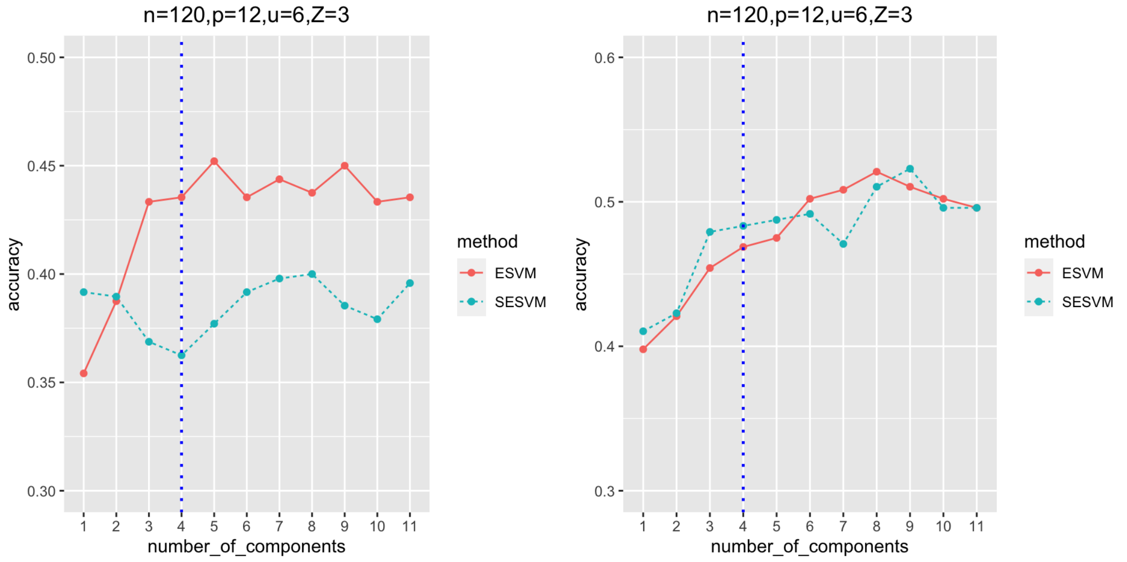
(a) (b)

Figure 6.7: The average classification accuracy percentage based on ESVM and SESVM. a: sample size $n = 120, p = 15, Z = 3, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 120, p = 15, Z = 3, (\tau_1, \tau_2) = (4, 0.25)$. The dashed vertical line is the true number of components.



(a) (b)

Figure 6.8: The average classification accuracy percentage based on ESVM and SESVM. a: sample size $n = 40, p = 10, Z = 2, (\tau_1, \tau_2) = (2, 0.2)$. b sample size $n = 40, p = 10, Z = 2, (\tau_1, \tau_2) = (10, 0.5)$.



(a) (b)

Figure 6.9: The average classification accuracy percentage based on ESVM and SESVM. a: sample size $n = 120, p = 15, p_a = 6, Z = 3, (\tau_1, \tau_2) = (10, 0.5)$. b sample size $n = 120, p = 15, p_a = 6, Z = 3, (\tau_1, \tau_2) = (5, 0.4)$. The dashed vertical line is the true number of components.

## 6.6 Conclusion

In this chapter, we consider the classification of data with sparsity structure. We proposed a sparse envelope support vector machine classifier as an extension to the ESVM classifier that was introduced in Chapter 5. The newly developed classifier is assumed to perform variable selection and dimension reduction simultaneously prior to the data classification. Our method takes that only a few components of the active variables are sufficient to process the classification. We have compared it against other existing classifiers: SVM, sparse LDA, and sparse logistic regression. Furthermore, it has shown competitive performance based on real and simulated data.

# Chapter 7

# Conclusion

In this chapter, we summarize the main conclusions of our work and discuss possible extension. Our work contributes to reduce-and-classify approach in supervised learning and highlights the efficiency gain that dimension reduction adds to classification performance. We consider extending the envelope method for dimension reduction to supervised learning.

In Chapter 5, we developed an Envelope-based Support Vector Machines (ESVM) classifier as an extension to the well-known Support Vector Machines classifier. The condition that this classifier developed based on is that a few components of the original predictor variables are sufficient to perform the classification with acceptable accuracy. Motivated by the envelope method, we developed an algorithm to construct a lower-dimensional subspace. The algorithm to construct the projection matrix is based on optimizing a likelihood-based objective function over the grassmann manifold. Hence, the constructed subspace is used as a projection matrix to reduce the dimension of the data. Our approach is useful in the presence of multicollinearity in that it concentrates the classification-related information in a few components. We generated synthetic data with different scenarios. We calculate the classification accuracy as a performance measure based on out-of-sample data. The proposed technique was tested on real data as well. The synthetic and real data show promising classification performance of the proposed classifier.

In Chapter 6, we extended the ESVM classifier to consider the sparse data. That is, we developed Sparse Envelope-based Support Vector Machines (SESVM) classifier that addresses the variable selection as well as dimension reduction. The variable selection procedure assumes that only a small set out of the predictor variables is related to the analysis (denoted by active variables) while the remaining predictor variables are irrelevant (denoted by inactive variables). Thus, the restrictions, in this case, are $(i)$ a few linear combinations of the predictor variables are sufficient for the classification procedure, and $(ii)$ among the predictor variables, only a few are significant (has non-zero coefficient). The modification includes adjusting the algorithm to construct the projection matrix. That is, we imposed an adaptive group lasso penalty to induce sparsity structure in the estimated subspace. The penalty works to force the non-significant predictor variables to have zero weight. Hence, the projection matrix is a linear combination of the classification-related features (active features) and inactive variables. We conducted several simulation scenarios to investigate the proposed method's performance. The method was tested on real data as well. The outcome of the numerical studies shows the improvement in classification accuracy based on SESVM over other classifiers. The computational aspect of this work was performed using $R$ software.

The future work, one aspect is to improve the computation performance. We noticed that the computational performance related to Chapter 6 is slow, especially when $p > n$. Furthermore, the concept of envelope basis was introduced in the regression framework and has shown its efficiency; therefore, this method may extend to support vector regression. The presented work is developed for linear data; hence, another possibility is to extend this work to nonlinear classification. That requires developing an algorithm to construct the projection matrix for nonlinear data. The early work in this context is proposed in a multivariate regression framework by Zhang et al. (2020). We want to mention that we have studied constructing an envelope basis for the nonlinear data; however, due to incompletion, we omit it.

The other development is to extend the work from the support vector machines to the support tensor machine. Early envelope-based work in tensor data was developed in a regression framework. (Zhang and Li, 2017) extended the work developed by Cook et al. (2013) from vector to tensor data and proposed envelope model for tensor predictor. On the other hand, Li and Zhang (2017) extended the work by Cook et al. (2010) to develop tensor response. As shown in Chapter 5 Zhang and Mai (2018) has proposed the construction of envelope basis in discriminant analysis. In a similar manner, the envelope model may extend from vector data to tensor data in the classification framework.

# Appendix A

## A.1 Karush-Kuhn-Tucker conditions

Given the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}^T\boldsymbol{x}, && \text{(A.1)} \\ \text{subject to} \quad & g_i(\boldsymbol{x}) = \boldsymbol{r}_i^T\boldsymbol{x} + b_i \geq 0, && \text{(A.2)} \\ & h_i(\boldsymbol{x}) = \boldsymbol{z}_i^T\boldsymbol{x} + e_i = 0, && \text{(A.3)} \end{aligned}$$

**Proposition A.1.1.** *The optimal solution* $(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$*, exists if and only if the following conditions re satisfied:*

$$\frac{\partial L(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)}{\partial \boldsymbol{x}} = \mathbf{0}, \tag{A.4}$$

$$\alpha_i^* g_i(\boldsymbol{x}^*) = 0 \tag{A.5}$$

$$\alpha_i^* \geq 0, \tag{A.6}$$

$$h_i(\boldsymbol{x}^*) = 0. \tag{A.7}$$

The conditions given in (A.4) to (A.7) referred to as *Karush-Kuhn-Tucker conditions* (KKT). However, the condition given by (A.5) known as *Karush Kuhn Tucker complementarity conditions*, which means if $\alpha_i^* > 0, g_i(\boldsymbol{x}^*) = 0$; and $\alpha_i^* = 0, g_i(\boldsymbol{x}^*) \geq 0$.

## A.2 Proofs

### A.2.1 Proof of proposition 5.7.1

This proof is given in Li et al. (2011), and it is to show the gradient of the SVM objective function. Let $\Delta_{\boldsymbol{\theta}}^2$ be the operator $\Delta_{\boldsymbol{\theta}}\Delta_{\boldsymbol{\theta}}^T$. Hence, $\Delta_{\boldsymbol{\theta}}^2\mathcal{J}(\boldsymbol{\theta}, \boldsymbol{m})$ is a $(p+1) \times (p+1)$ matrix whose $(i,j)$ element is $\partial^2 \mathcal{J}(\boldsymbol{\theta},\boldsymbol{m})/\partial\theta_i\partial\theta_j$. Moreover, for each $\boldsymbol{\theta} \in \Theta$, let $N_{\boldsymbol{\theta}}(\mathcal{J})$ be the set of $\boldsymbol{m}$ for which a function $\mathcal{J}(\boldsymbol{m},.)$ is not differentiable at $\boldsymbol{\theta}$. That is,

$$N_{\boldsymbol{\theta}}(\mathcal{J}) = \{\boldsymbol{m} : \mathcal{J}(\boldsymbol{m},.)\text{is not differentiable at}\boldsymbol{\theta}\}.$$

The following lemma is used for proving proposition 5.7.1

**Lemma A.2.1.** *Suppose that $\mathcal{J} : \Theta \times \Omega_{\boldsymbol{m}} \to \mathbb{R}$ satisfies the following conditions:*

1. *(almost surely differentiable) $Pr[\boldsymbol{m} \in N_{\boldsymbol{\theta}}(\mathcal{J})], \forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$.*
2. *(Lipschitz condition) there is an integrable function $a(\boldsymbol{m})$, independent of $\boldsymbol{\theta}$, such that for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$,*

$$|\mathcal{J}(\boldsymbol{\theta}_2,\boldsymbol{m}) - \mathcal{J}(\boldsymbol{\theta}_1,\boldsymbol{m})| \leq a(\boldsymbol{m})||\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1||.$$

*Then $\Delta_{\boldsymbol{\theta}}[\mathcal{J}(\boldsymbol{\theta}.\boldsymbol{m})]$ is integrable, $\mathbb{E}[\mathcal{J}(\boldsymbol{\theta},\boldsymbol{m})]$ is differentiable, and*

$$\Delta_{\boldsymbol{\theta}}\mathbb{E}[\mathcal{J}(\boldsymbol{\theta},\boldsymbol{m})] = \mathbb{E}[\Delta_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta},\boldsymbol{m})]. \tag{A.8}$$

**Proof of proposition 5.7.1:** Suppose $H(\boldsymbol{w},b)$ represents the hyperplane $\{\boldsymbol{x} : \boldsymbol{w}^T\boldsymbol{x} = b\}$. We first satisfy the conditions indicated in Lemma A.2.1.

$$Pr[(\boldsymbol{x},y) \in N_{\boldsymbol{\theta}}(\mathcal{J})] = \sum_{y\in\{-1,1\}} Pr(Y=y)Pr[\boldsymbol{x} \in H(\boldsymbol{w},b+y)|Y=y].$$

Since for $y \in \{-1,1\}$ the Lebesgue measure of $H(\boldsymbol{w},b+y)$ is 0, the above probability is 0 by condition 1. That is, condition 1 of Lemma A.2.1 is satisfied. Now, let $\mathcal{J}_1(\boldsymbol{\theta},\boldsymbol{m}) = \boldsymbol{w}^T\boldsymbol{w}$ and $\mathcal{J}_2(\boldsymbol{\theta},\boldsymbol{m}) = [1 - y(\boldsymbol{w}^T\boldsymbol{x} - b)]^+$. Hence $\mathcal{J}(\boldsymbol{\theta},\boldsymbol{m})$ can be written as $\mathcal{J}_1(\boldsymbol{\theta},\boldsymbol{m}) + \lambda\mathcal{J}_2(\boldsymbol{\theta},\boldsymbol{m})$. Since $\mathcal{J}_1$ is nonrandom and differentiable, it satisfies $\Delta_{\boldsymbol{\theta}}\mathbb{E}[\mathcal{J}_1(\boldsymbol{\theta},\boldsymbol{m})] = \mathbb{E}[\Delta_{\boldsymbol{\theta}}\mathcal{J}_1(\boldsymbol{\theta},\boldsymbol{m})]$. To verify the second condition of Lemma A.2.1, that is $\mathcal{J}_2$ is Lipschitz, assume $(\boldsymbol{w}_1,b_1),(\boldsymbol{w}_2,b_2) \in \mathbb{R}^{p+1}$. Then

$$\mathcal{J}_2(\boldsymbol{\theta}_2,\boldsymbol{x},y) - \mathcal{J}_2(\boldsymbol{\theta}_1,\boldsymbol{x},y) = [1 - y(\boldsymbol{w}_2^T\boldsymbol{x} - b_2)]^+ - [1 - y(\boldsymbol{w}_1^T\boldsymbol{x} - b_1)]^+. \tag{A.9}$$

Given that for any two real numbers $a$ and $b$,

$$|b^+ - a^+| \leq |b - a|$$

Then

$$\begin{aligned}\mathcal{J}_2(\boldsymbol{\theta}_2, \boldsymbol{x}, y) - \mathcal{J}_2(\boldsymbol{\theta}_1, \boldsymbol{x}, y) &\leq |\boldsymbol{w}_1^T \boldsymbol{x} - \boldsymbol{w}_2^T \boldsymbol{x} + b_2 - b_1| \\ &\leq (1 + ||\boldsymbol{x}||^2)^{1/2} ||\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1||.\end{aligned}$$

Since $\mathbb{E}(||\boldsymbol{x}||^2) < \infty$,

$$\mathbb{E}(1 + ||\boldsymbol{x}||^2)^{1/2} \leq [1 + \mathbb{E}(||\boldsymbol{x}||^2)]^{1/2} < \infty.$$

Thus, condition 2 of lemma A.2.1 is verified.

Lastly, for $\boldsymbol{m} \notin N_{\boldsymbol{\theta}}(\mathcal{J})$,

$$\begin{aligned}\Delta_w[\mathcal{J}(\boldsymbol{\theta}, \boldsymbol{m})] &= 2\boldsymbol{w} - \lambda \boldsymbol{x} y \boldsymbol{I}[1 - y(\boldsymbol{w}^T \boldsymbol{x} - b) > 0], \\ \Delta_t[\mathcal{J}(\boldsymbol{\theta}, \boldsymbol{m})] &= \lambda y \boldsymbol{I}[1 - y(\boldsymbol{w}^T \boldsymbol{x} - t > 0].\end{aligned}$$

Then

$$\Delta_{\boldsymbol{\theta}}[\mathcal{J}(\boldsymbol{\theta}, \boldsymbol{m})] = 2\boldsymbol{w}^T - \lambda \boldsymbol{x}^* y \boldsymbol{I}(1 - \boldsymbol{\theta}^T \boldsymbol{x}^* y > 0).$$

### A.2.2 Proof of proposition 5.7.2

This proof is given in Li et al. (2011), for simplicity we give it here. We want to show that (5.17) is jointly differentiable with respect to $(\boldsymbol{\theta}^*, b)$. We will provide the lemmas introduced in Li et al. (2011) that aid facilitating the proof. They They define the derivative of an expectation of a non-Lipschitz function. Suppose $D_{\kappa=0}$ is the operation of taking the derivative with respect to $\kappa$ then evaluating the derivative at $\kappa = 0$.

**Lemma A.2.2.** *Let $U$ and $V$ be a random variables and $\boldsymbol{g}(u, v) \in \mathbb{R}^q$ be a measurable function. Further, suppose*

1. *the joint distribution of $(U, V)$ is dominated by the Lebesgue measure;*
2. *the function $u \to g(u, v) f_{U|V}(u|v)$ is continuous for each $v$, where $f_{U|V}$ denotes the conditional probability density function of $U|V$;*
3. *there is a function $c_i(v) \geq 0$ for each component $g_i(u, v)$ of $\boldsymbol{g}(u, v)$, such that*

$$|g_i(u, v)| f_{U|V}(u|v) \leq c_i(v), \quad E[c_i(V)] < \infty. \tag{A.10}$$

*Then, for any constant $t$, the function $\kappa \to E[\boldsymbol{g}(U,V)I(U+\kappa V < t+\kappa\tau)]$ is differentiable at $\kappa = 0$ with derivative*

$$D_{\kappa=0}E[\boldsymbol{g}(U,V)|(U+\kappa V < t+\kappa\tau)] = f_U(t)E[(\tau - V)\boldsymbol{g}(U,V)|U=t]. \qquad \text{(A.11)}$$

**Lemma A.2.3.** *Let $U$ and $V$ be linearly dependent random variables and $\boldsymbol{g}(u,v) \in \mathbb{R}^q$ be a measurable function. Further, suppose*

1. *the distribution of $U$ is dominated by the Lebesgue measure;*
2. *$\boldsymbol{g}(u)f_U(u)$ is continuous.*

*Hence, for any constant $t$, the function $\kappa \to E[\boldsymbol{g}(U)|(U+\kappa V < t+\kappa\tau)]$ is differentiable at $\kappa = 0$ with derivative given by (A.11).*

**Proof of proposition 5.7.2:** We want to show that (5.17), that is given by:

$$G(\boldsymbol{\theta}) = \Delta_\theta \mathbb{E}[\mathcal{J}(\boldsymbol{\theta})] = 2\boldsymbol{\theta}^{*T} - \lambda\mathbb{E}[\widetilde{\boldsymbol{x}}yI(1-\boldsymbol{\theta}^T\widetilde{\boldsymbol{x}}y > 0)].$$

is jointly differentiable with respect to $(\boldsymbol{\theta}^*, b)$, where $\boldsymbol{\theta} = (\boldsymbol{w}_\Gamma, b)$, $\boldsymbol{\theta}^* = \boldsymbol{w}_{\Gamma},$. The first term is straightforward with derivative (2). Thus, we need to show the differentiability of the second term $\mathbb{E}[\widetilde{\boldsymbol{x}}yI(1-\boldsymbol{\theta}^T\widetilde{\boldsymbol{x}}y > 0)]$, that is given by:

$$\sum_{y=-1,1} P(Y=y)f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|Y}(b+y|y)\mathbb{E}(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^T|\boldsymbol{\theta}^{*T}\boldsymbol{x} = b+y).$$

We need to verfy the directional differentiability of the function $(\boldsymbol{\theta}^*, b) \to \mathbb{E}[\widetilde{\boldsymbol{x}}I(\boldsymbol{\theta}^{*T}\boldsymbol{x} < b+1)|y]$ at $y=1$ any $y=-1$.

First, consider the case $y=1$. Suppose $\boldsymbol{\theta}^*, \boldsymbol{\delta} \in \mathbb{R}^q$ are linearly independent vectors. The directional derivative along $(\boldsymbol{\delta}^T, \tau)^T$, where $\tau \in \mathbb{R}$, is given by the derivative of the following function with respect to $\kappa$ at $\kappa = 0$:

$$\mathbb{E}[\widetilde{\boldsymbol{x}}I(\boldsymbol{\theta}^{*T}\boldsymbol{x}+\kappa\boldsymbol{\delta}^T\boldsymbol{x} < b+1+\kappa\tau)|y=1] = \mathbb{E}[\mathbb{E}(\widetilde{\boldsymbol{x}}|\boldsymbol{\theta}^{*T}\boldsymbol{x}, \boldsymbol{\delta}^T\boldsymbol{x}, y=1)I(\boldsymbol{\theta}^{*T}\boldsymbol{x}+\boldsymbol{\delta}^T\boldsymbol{x} < b+1+\kappa\tau)|y=1].$$

Now, let $U = \boldsymbol{\theta}^{*T}\boldsymbol{x}, V = \boldsymbol{\delta}^T\boldsymbol{x}, \boldsymbol{g}(U,V) = \mathbb{E}(\widetilde{\boldsymbol{x}}|U,V)$. Thus, by Lemma A.2.2, the above derivative is:

$$f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|y}(b+1|y=1)\mathbb{E}[(\tau-V)\mathbb{E}(\widetilde{\boldsymbol{x}}|U,V)|U=b+1] = f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|y}(b+1|y=1)\mathbb{E}[((\tau-V)\widetilde{\boldsymbol{x}}|U,V)|U=b+1].$$

Since this is hold for all $(\boldsymbol{\delta}^T, \tau)^T$, then the function

$$(\boldsymbol{\theta}^*, b) \to \mathbb{E}[\widetilde{\boldsymbol{x}} I(\boldsymbol{\theta}^{*T}\boldsymbol{x} < b+1)|y=1]$$

is directionally differentiable with the following derivative matrix:

$$- f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|y}(b+1|y=1)\mathbb{E}(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^T|\boldsymbol{\theta}^{*T}\boldsymbol{x} = t+1, y-1). \tag{A.12}$$

If $\boldsymbol{\theta}^*, \boldsymbol{\delta} \in \mathbb{R}^q$ are independent vectors, then $\boldsymbol{\theta}^{*T}\boldsymbol{X}$ and $\boldsymbol{\delta}^T\boldsymbol{X}$ are linearly independent random vectors. lemma A.2.3 is applied in similar manner to arrive at the same directional derivative (A.12).

On the other hand, the case where $y = -1$ can be proved in a similar manner. Thus, the directional derivative of $\Delta_\theta \mathbb{E}[\mathcal{J}(\boldsymbol{\theta})]$ is given in proposition 5.7.2. Moreover, if $f_{\boldsymbol{\theta}^{*T}\boldsymbol{x}|y}(b+y|y)\mathbb{E}(\widetilde{\boldsymbol{x}}\widetilde{\boldsymbol{x}}^T|\boldsymbol{\theta}^{*T}\boldsymbol{x} = t+1)$ is continuous, then the directional derivative is continuous. Accordingly, $\Delta_{\boldsymbol{\theta}}\mathbb{E}[\mathcal{J}(\boldsymbol{\theta}]$ is jointly differentiable.

# References

Abe, S. (2005), *Support vector machines for pattern classification*, Vol. 2, Springer. 13, 16, 17, 37, 43, 47

Adragni, K. P., Cook, R. D., Wu, S. et al. (2012), 'Grassmannoptim: An r package for grassmann manifold optimization', *Journal of Statistical Software* **50**(5), 1–18. 55

Aksu, Y., Miller, D. J., Kesidis, G. and Yang, Q. X. (2010), 'Margin-maximizing feature elimination methods for linear and nonlinear kernel-based discriminant functions', *IEEE Transactions on Neural Networks* **21**(5), 701–717. 50, 77

Antoniadis, A., Lambert-Lacroix, S. and Leblanc, F. (2003), 'Effective dimension reduction methods for tumor classification using gene expression data', *Bioinformatics* **19**(5), 563–570. 50

Bahadur, R. R. (1966), 'A note on quantiles in large samples', *The Annals of Mathematical Statistics* **37**(3), 577–580. 59

Bair, E., Hastie, T., Paul, D. and Tibshirani, R. (2006), 'Prediction by supervised principal components', *Journal of the American Statistical Association* **101**(473), 119–137. 3

Berg, C., Christensen, J. P. R. and Ressel, P. (1984), *Harmonic analysis on semigroups: theory of positive definite and related functions*, Vol. 100, Springer. 45

Berk, R. A. et al. (2008), *Statistical learning from a regression perspective*, Vol. 14, Springer. 40

Bishop, C. M. et al. (1995), *Neural networks for pattern recognition*, Oxford university press. 14

Bradley, P. S. and Mangasarian, O. L. (1998), Feature selection via concave minimization and support vector machines., *in* 'ICML', Vol. 98, Citeseer, pp. 82–90. 74, 77

Bura, E. and Pfeiffer, R. M. (2003), 'Graphical methods for class prediction using dimension reduction techniques on dna microarray data', *Bioinformatics* **19**(10), 1252–1258. 50

Chatfield, C. and Collins, A. (1981), *Introduction to multivariate analysis*, Vol. 1, CRC Press. 8

Chen, X., Wu, J., Yao, Z. and Zhang, J. (2018), 'Sufficient dimension reduction for classification', *arXiv preprint arXiv:1812.03775* . 50

Christmann, A. and Steinwart, I. (2008), 'Support vector machines'. 37

Chun, H. and Keleş, S. (2010), 'Sparse partial least squares regression for simultaneous dimension reduction and variable selection', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**(1), 3–25. 80

Clemmensen, L., Hastie, T., Witten, D. and Ersbøll, B. (2011), 'Sparse discriminant analysis', *Technometrics* **53**(4), 406–413. 74, 76

Cole, S. R., Chu, H. and Greenland, S. (2014), 'Maximum likelihood, profile likelihood, and penalized likelihood: a primer', *American journal of epidemiology* **179**(2), 252–260. 10

Cook, R. (1998), 'Tutorial: Regression graphics', *COMPUTING SCIENCE AND STATISTICS* pp. 59–66. 10

Cook, R. D. (1994), Using dimension-reduction subspaces to identify important inputs in models of physical systems, *in* 'Proceedings of the section on Physical and Engineering Sciences', pp. 18–25. 10

Cook, R. D. (2018), *An introduction to envelopes: dimension reduction for efficient estimation in multivariate statistics*, Vol. 401, John Wiley & Sons. 24, 29, 58

Cook, R. D. (2019), 'Envelope methods', *Wiley Interdisciplinary Reviews: Computational Statistics* p. e1484. 19, 20

Cook, R. D., Forzani, L. and Su, Z. (2016), 'A note on fast envelope estimation', *Journal of Multivariate Analysis* **150**, 42–54. 30, 31

Cook, R. D., Helland, I. and Su, Z. (2013), 'Envelopes and partial least squares regression', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **75**(5), 851–877. 26, 27, 32, 62, 63, 83, 94

Cook, R. D., Li, B. and Chiaromonte, F. (2010), 'Envelope models for parsimonious and efficient multivariate linear regression', *Statistica Sinica* pp. 927–960. 3, 9, 10, 19, 21, 53, 54, 94

Cook, R. D. and Weisberg, S. (1991), 'Sliced inverse regression for dimension reduction: Comment', *Journal of the American Statistical Association* **86**(414), 328–332. 3, 10

Cook, R. D. and Zhang, X. (2016), 'Algorithms for envelope estimation', *Journal of Computational and Graphical Statistics* **25**(1), 284–300. 29

Cortes, C. and Vapnik, V. (1995), 'Support-vector networks', *Machine learning* **20**(3), 273–297. 14, 36

Ding, S., Su, Z., Zhu, G. and Wang, L. (2019), 'Envelope quantile regression', *Statistica Sinica* . 33

Dobson, A. J. and Barnett, A. G. (2018), *An introduction to generalized linear models*, Chapman and Hall/CRC. 6

Dua, D. and Graff, C. (2017), 'UCI machine learning repository'.
**URL:** *http://archive.ics.uci.edu/ml* 71

Efron, B. (1975), 'The efficiency of logistic regression compared to normal discriminant analysis', *Journal of the American Statistical Association* **70**(352), 892–898. 14

Fan, J. and Fan, Y. (2008), 'High dimensional classification using features annealed independence rules', *Annals of statistics* **36**(6), 2605. 3, 75

Fan, J. and Li, R. (2001), 'Variable selection via nonconcave penalized likelihood and its oracle properties', *Journal of the American statistical Association* **96**(456), 1348–1360. 3, 13

Fisher, R. A. (1936), 'The use of multiple measurements in taxonomic problems', *Annals of eugenics* **7**(2), 179–188. 14

Fletcher, R. (1987), 'Practical methods of optimization john wiley & sons', *New York* **80**, 4. 38

Gareth, J., Daniela, W., Trevor, H. and Robert, T. (2013), *An introduction to statistical learning: with applications in R*, Spinger. 37

Genton, M. G. (2001), 'Classes of kernels for machine learning: a statistics perspective', *Journal of machine learning research* **2**(Dec), 299–312. 46

Gokcen, I. and Peng, J. (2002), Comparing linear discriminant analysis and support vector machines, *in* 'International Conference on Advances in Information Systems', Springer, pp. 104–113. 58

Gómez-Verdejo, V., Martínez-Ramón, M., Arenas-García, J., Lázaro-Gredilla, M. and Molina-Bulla, H. (2011), 'Support vector machines with constraints for sparsity in the primal parameters', *IEEE transactions on neural networks* **22**(8), 1269–1283. 78

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002), 'Gene selection for cancer classification using support vector machines', *Machine learning* **46**(1), 389–422. 74, 77

Hastie, T., Buja, A. and Tibshirani, R. (1995), 'Penalized discriminant analysis', *The Annals of Statistics* **23**(1), 73–102. 76

Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media. 2, 10, 11, 13, 14, 15, 37

Hoerl, A. E. and Kennard, R. W. (1970), 'Ridge regression: Biased estimation for nonorthogonal problems', *Technometrics* **12**(1), 55–67. 3

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013), *An introduction to statistical learning*, Vol. 112, Springer. 3, 13, 15, 36, 48, 58

John Lu, Z. (2010), 'The elements of statistical learning: data mining, inference, and prediction', *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **173**(3), 693–694. 36

Johnstone, I. M. and Titterington, D. M. (2009), 'Statistical challenges of high-dimensional data'. 1

Jolliffe, I. (1986), Generalizations and adaptations of principal component analysis, *in* 'Principal Component Analysis', Springer, pp. 223–234. 3

Kenward, M. G. (1987), 'A method for comparing profiles of repeated measurements', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **36**(3), 296–308. 69

Koo, J.-Y., Lee, Y., Kim, Y. and Park, C. (2008), 'A bahadur representation of the linear support vector machine', *The Journal of Machine Learning Research* **9**, 1343–1368. 59

Kumar, K., Bhattacharya, C. and Hariharan, R. (2007), 'A randomized algorithm for large scale support vector learning'. 50

Lee, M. and Su, Z. (2019), 'A review of envelope models'. 19

Li, B. (2018), *Sufficient dimension reduction: Methods and applications with R*, CRC Press. 9

Li, B., Artemiou, A., Li, L. et al. (2011), 'Principal support vector machines for linear and nonlinear sufficient dimension reduction', *The Annals of Statistics* **39**(6), 3182–3210. 3, 36, 59, 60, 95, 97

Li, K.-C. (1991), 'Sliced inverse regression for dimension reduction', *Journal of the American Statistical Association* **86**(414), 316–327. 10

Li, L. (2007), 'Sparse sufficient dimension reduction', *Biometrika* **94**(3), 603–613. 2

Li, L. and Zhang, X. (2017), 'Parsimonious tensor response regression', *Journal of the American Statistical Association* **112**(519), 1131–1146. 94

Mai, Q. (2013), 'A review of discriminant analysis in high dimensions', *Wiley Interdisciplinary Reviews: Computational Statistics* **5**(3), 190–197. 14

Mai, Q., Zou, H. and Yuan, M. (2012), 'A direct approach to sparse discriminant analysis in ultra-high dimensions', *Biometrika* **99**(1), 29–42. 74, 77

Mansfield, E. R. and Helms, B. P. (1982), 'Detecting multicollinearity', *The American Statistician* **36**(3a), 158–160. 2

McCullagh, P. and Nelder, J. (1989), 'Generalized linear models new york chapman & hall', *McCullagh2Generalized Linear Models1989* . 7

Meier, L., Van De Geer, S. and Bühlmann, P. (2008), 'The group lasso for logistic regression', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(1), 53–71. 81

Merchante, L. F. S., Grandvalet, Y. and Govaert, G. (2012), 'An efficient approach to sparse linear discriminant analysis', *arXiv preprint arXiv:1206.6472* . 74

Moradibaad, A. and Mashhoud, R. J. (2018), 'Use dimensionality reduction and svm methods to increase the penetration rate of computer networks', *arXiv preprint arXiv:1812.03173* . 50

Naranjo, L., Perez, C. J., Campos-Roca, Y. and Martin, J. (2016), 'Addressing voice recording replications for parkinson?s disease detection', *Expert Systems with Applications* **46**, 286–292. 87

Paul, S., Boutsidis, C., Magdon-Ismail, M. and Drineas, P. (2013), Random projections for support vector machines, *in* 'Artificial intelligence and statistics', PMLR, pp. 498–506. 50, 51

Pircalabelu, E. and Artemiou, A. (2021), 'Graph informed sliced inverse regression', *Computational Statistics & Data Analysis* **164**, 107302. 3

Pircalabelu, E. and Artemiou, A. (2022), 'High-dimensional sufficient dimension reduction through principal projections', *Electronic Journal of Statistics* **16**(1), 1804–1830. 3

Randall, H., Artemiou, A. and Qiao, X. (2021), 'Sufficient dimension reduction based on distance-weighted discrimination', *Scandinavian Journal of Statistics* **48**(4), 1186–1211. 3

RD, T. (1954), 'Physical growth of california boys and girls from birth to eighteen years.', *Publications in Child development. University of California, Berkeley* **1**(2), 183–364. 68

Rokach, L. (2010), *Pattern classification using ensemble methods*, Vol. 75, World Scientific. 47

Rothman, A. J., Bickel, P. J., Levina, E. and Zhu, J. (2008), 'Sparse permutation invariant covariance estimation', *Electronic Journal of Statistics* **2**, 494–515. 81

Shao, R., Hu, W., Wang, Y. and Qi, X. (2014), 'The fault feature extraction and classification of gear using principal component analysis and kernel principal component analysis based on the wavelet packet transform', *Measurement* **54**, 118–132. 50

Shi, Q., Shen, C., Hill, R. and Hengel, A. v. d. (2012), 'Is margin preserved after random projection?', *arXiv preprint arXiv:1206.4651* . 51

Steinwart, I. and Christmann, A. (2008), *Support vector machines*, Springer Science & Business Media. 44, 45

Suykens, J. A. and Vandewalle, J. (1999), 'Least squares support vector machine classifiers', *Neural processing letters* **9**(3), 293–300. 36

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288. 3, 11, 76

Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, G. (2002), 'Diagnosis of multiple cancer types by shrunken centroids of gene expression', *Proceedings of the National Academy of Sciences* **99**(10), 6567–6572. 74, 75

Tuddenham, R. D. (1954), 'Physical growth of california boys and girls from birth to eighteen years', *University of California publications in child development* **1**, 183–364. 22

Verleysen, M. et al. (2003), 'Learning high-dimensional data', *Nato Science Series Sub Series III Computer And Systems Sciences* **186**, 141–162. 1

Wang, H. and Leng, C. (2008), 'A note on adaptive group lasso', *Computational statistics & data analysis* **52**(12), 5277–5286. 12, 81

Wang, W., Zhang, X., Mai, Q. et al. (2020), 'Model-based clustering with envelopes', *Electronic Journal of Statistics* **14**(1), 82–109. 52

Weng, J. and Young, D. S. (2017), 'Some dimension reduction strategies for the analysis of survey data', *Journal of Big Data* **4**(1), 1–19. 2

Yin, X., Li, B. and Cook, R. D. (2008), 'Successive direction extraction for estimating the central subspace in a multiple-index regression', *Journal of Multivariate Analysis* **99**(8), 1733–1757. 10

Yuan, M. and Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 49–67. 12, 80

Zhang, J., Zhu, G., Heath Jr, R. W. and Huang, K. (2018), 'Grassmannian learning: Embedding geometry awareness in shallow and deep learning', *arXiv preprint arXiv:1808.02229* . 55

Zhang, X., Lee, C. and Shao, X. (2020), 'Envelopes in multivariate regression models with nonlinearity and heteroscedasticity', *Biometrika* **107**(4), 965–981. 34, 94

Zhang, X. and Li, L. (2017), 'Tensor envelope partial least-squares regression', *Technometrics* **59**(4), 426–436. 94

Zhang, X. and Mai, Q. (2018), 'Efficient integration of sufficient dimension reduction and prediction in discriminant analysis', *Technometrics* . 52, 57, 58, 94

Zhu, G. and Su, Z. (2019), 'Envelope-based sparse partial least squares', *Annals of Statistics. To appear* . 32

Zhu, G. and Su, Z. (2020), 'Envelope-based sparse partial least squares', *The Annals of Statistics* **48**(1), 161–182. 80

Zhu, J. and Zou, H. (2007), 'Variable selection for the linear support vector machine', *Trends in Neural Computation* pp. 35–59. 78

Zou, H. (2006), 'The adaptive lasso and its oracle properties', *Journal of the American statistical association* **101**(476), 1418–1429. 11, 81

Zou, H. and Hastie, T. (2005), 'Regularization and variable selection via the elastic net', *Journal of the royal statistical society: series B (statistical methodology)* **67**(2), 301–320. 3, 12

Zou, H., Hastie, T. and Tibshirani, R. (2006), 'Sparse principal component analysis', *Journal of computational and graphical statistics* **15**(2), 265–286. 75