STATISTICAL LEARNING FOR BIOMEDICAL DATA UNDER VARIOUS FORMS OF HETEROGENEITY

Guanhua Chen

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Biostatistics in the Gillings School of Global Public Health.

Chapel Hill
2014

Approved by:

Dr. Michael R. Kosorok

Dr. Yufeng Liu

Dr. Patrick F. Sullivan

Dr. Wei Sun

Dr. Donglin Zeng

ABSTRACT

GUANHUA CHEN: Statistical Learning for Biomedical Data under Various Forms of Heterogeneity
(Under the direction of Dr. Michael R. Kosorok)

In modern biomedical research, an emerging challenge is data heterogeneity. Ignoring such heterogeneity can lead to poor modeling results.

In cancer research, clustering methods are applied to find subgroups of homogeneous individuals based on genetic profiles together with heuristic clinical analysis. A notable drawback of existing clustering methods is that they ignore the possibility that the variance of gene expression profile measurements can be heterogeneous across subgroups, leading to inaccurate subgroup prediction. In Chapter 2, we present a statistical approach that can capture both mean and variance structure in gene expression data. We demonstrate the strength of our method in both synthetic data and two cancer data sets.

For a binary classification problem, there can be potential subclasses within the two classes of interest. These subclasses are latent and usually heterogeneous. We propose the Composite Large Margin Classifier (CLM) to address the issue of classification with latent subclasses in Chapter 3. The CLM aims to find three linear functions simultaneously: one linear function to split the data into two parts, with each part being classified by a different linear classifier. Our method has comparable prediction accuracy to a general nonlinear kernel classifier without overfitting the training data while at the same time maintaining the interpretability of traditional linear classifiers.

There is a growing recognition of the importance of considering individual level heterogeneity when searching for optimal treatment doses. Such optimal individualized treatment rules (ITRs) for dosing should maximize the expected clinical benefit. In Chapter 4, we consider a randomized trial design where the candidate dose levels are continuous. To find the optimal ITR under such a design, we propose an outcome weighted learning method which directly maximizes the expected beneficial clinical outcome. This method converts the individualized dose selection problem into a nonstandard weighted regression problem. A difference of convex functions (DC) algorithm is adopted to efficiently solve the associated non-convex optimization problem. The consistency and convergence rates for the estimated ITR are derived and small-sample performance is evaluated via simulation studies. We illustrate the method using data from a clinical trial for Warfarin dosing.

I dedicate this dissertation work to my parents,
Yixin Chen and Yanping Xiong,
and to my beloved wife and son,
Zhengzheng Tang and Patrick Chen.

ACKNOWLEDGMENTS

I feel very lucky to have been admitted into a world class Biostatistics program, and to have opportunities to learn from so many talented people. The six years at UNC is one of the most enjoyable times in my life.

I want express my deepest gratitude to my advisor, Dr. Michael R. Kosorok. His support and guidance of me for both research and non-research issues have been very important to my stay at UNC. I am very fortunate to have him as my advisor.

I would like to give sincere thanks to other committee members: Dr. Patrick Sullivan, Dr. Yufeng Liu, Dr. Donglin Zeng, and Dr. Wei Sun. My special thanks goes to Dr. Patrick Sullivan for his generous financial support and mentoring during my masters and PhD years. Without him, I could not have continued my studies at UNC. I am deeply grateful to Dr. Yufeng Liu for research advice and allowing me to sit in on his group meeting as a regular member of his group. His knowledge in statistical learning has greatly enriched my understanding in the area. I gratefully thank Dr. Donglin Zeng for his tremendous help in my dissertation and job search. His patience and knowledge are key for the research I conducted in last two years and I wish I could have started learning from him on the first day of me pursuing my PhD. I would also like to thank my committee members Dr. Wei Sun, who directed my masters paper and several other projects, together with Dr. Sullivan, as well as provided many thoughtful suggestions on career development.

I am thankful to Dr. Hongtu Zhu, Dr. Chirayath Suchindran, Dr. Fei Zou, Dr. Michael Wu, and all other faulty and staff members in the Department. I am also thankful to former and current members of Michael's and Yufeng's groups for their insightful discussions on various research topics. I really appreciate Dr. Lisa LaVange and the CDER Office of Biostatistics at the FDA for providing financial support for my dissertation research.

Lastly, I want to express my appreciation to my family for supporting my career.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER1: INTRODUCTION

With the revolutions in technology and science, we are now entering the era of Big Data. The enrichment of data promises achievement of many biological, medical, and public health goals. However, reaching these goals is challenging due to the increasing magnitude and complexity of the data. An important aspect of the complexity is data heterogeneity. Ignoring data heterogeneity can cause bias in making decisions. This is a common problem in modern biological and biomedical research, e.g. the one-size-fit-all treatment strategy can fail due to heterogeneous response to drugs among patients. To deal with data heterogeneity, I have developed statistical learning and data mining methods under various heterogeneity settings. These methods are applicable for a wide range of problems—potentially high dimensional—with direct interest to clinicians and biomedical investigators.

In this dissertation, we investigate three data heterogeneity problems: (1) Biclustering a data matrix when no label information is given. Such problems are also referred to as unsupervised learning. We are interested in the problem where the variances for data entries are not homogeneous (see Chapter 2). (2) When there exists label information or there are dependent variables, the problem is supervised learning. We focus on studying binary classification and further we assume that within the class of interest, there exists heterogenous subclasses (see Chapter 3). (3) Most effective Dosage identification is a common task in modern clinical trials. The traditional dose finding trial only attempts to identify a single best dose, which may not be enough for a number of situations, including, for example, insulin dosing. Our goal is to identify a personalized rule using training data where patients are potentially treated with sub-optimal doses. Information such as patient characteristics and outcome after receiving the dose are used as tailoring variables. This kind of problem can be viewed as a semi-supervised learning problem. Unlike the traditional semi-supervised learning problem where some observations have accurate labels and others have no labels, all observations are provided with the label information (observed dose). However, such label information is not precise since the observed dose is generally suboptimal. Our proposed approach attempts to directly identify the dose rule which takes patient heterogeneity of response to treatment into account (see Chapter 4). All of our methods connect to existing statistical learning methods (Hastie et al. 2009) and are problem oriented.

In each of Chapters 2 to 4, I will review the background of the problems, the existing methods and their limitations. Our proposed methods will then be described in detail, followed by theoretical proof or basis for methodological intuition. Simulations and real data will be used to demonstrate the use of our methods. Lastly, I will describe future research in Chapter 5.

CHAPTER2: BICLUSTERING WITH HETEROGENEOUS VARIANCE

In cancer research, as in all of medicine, it is important to classify patients into etiologically and therapeutically relevant subtypes to improve diagnosis and treatment. One way to do this is to use clustering methods to find subgroups of homogeneous individuals based on genetic profiles together with heuristic clinical analysis. A notable drawback of existing clustering methods is that they ignore the possibility that the variance of gene expression profile measurements can be heterogeneous across subgroups, and methods that do not consider heterogeneity of variance can lead to inaccurate subgroup prediction. Research has shown that hypervariability is a common feature among cancer subtypes. In this chapter, we present a statistical approach that can capture both mean and variance structure in genetic data. We demonstrate the strength of our method in both synthetic data and in two cancer data sets. In particular, our method confirms the hypervariability of methylation level in cancer patients, and it detects clearer subgroup patterns in lung cancer data (see Chen et al. (2013)).

2.1 Introduction

Clustering is an important type of unsupervised learning algorithm for data exploration. Successful examples include K-mean clustering and hierarchical clustering, both of which are widely used in biological research to find cancer subtypes and to stratify patients. These and other traditional clustering algorithms depend on the distances calculated using all of the features. For example, individuals can be clustered into homogeneous groups by minimizing the summation of within clusters sum of squares (the Euclidean distances) of their gene expression profiles. Unfortunately, this strategy is ineffective when only a subset of features are informative. This phenomenon can be demonstrated by K-means clustering (Hastie et al. 2009) results for a toy example using only the variables which determine the underlying true cluster compared to using all variables (which includes many uninformative variables). As can be seen in Figure 2.1, clustering performance is poor when all variables are used in the clustering algorithm (Witten and Tibshirani 2010).

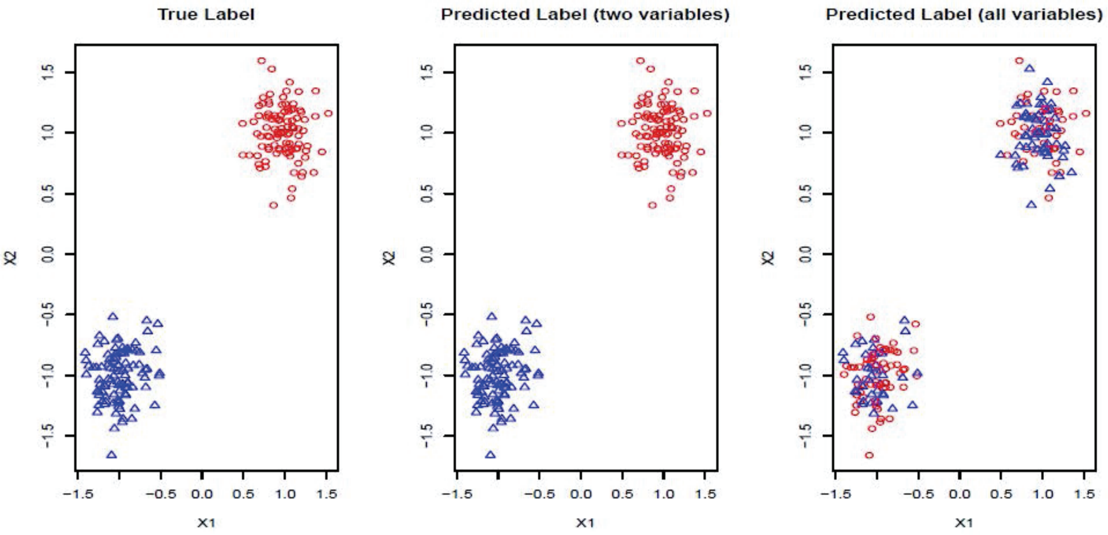

Figure 2.1: The data set contains two clusters determined by two variables X_1 and X_2 such that points around $(1, 1)$ and $(-1, -1)$ naturally form clusters. There are 200 observations (100 for each cluster) and 1002 variables (X_1, X_2 and 1000 random noise variables). We plot the data in the 2D space of X_1 and X_2. The graphs with true cluster labels and predicted cluster labels obtained by clustering using only X_1 and X_2 and clustering by using all variables are laid from left to right. The predicted labels are the same as the true labels only when X_1 and X_2 are used for clustering; however, the performance is much worse when all variables are used.

To solve this problem, sparse clustering methods have been proposed to allow clustering decisions to depend on only a subset of feature variables (the property of sparsity). Prominent sparse clustering methods include Sparse PCA (Ma 2013, Shen and

Huang 2008, Zou et al. 2006) and Sparse K-means (Witten and Tibshirani 2010), among others (Kriegel et al. 2009). However, sparse clustering still fails if the true sparsity is a local rather than a global phenomenon (Kriegel et al. 2009). More specifically, different subsets of features can be informative for some samples but not all samples, or, in other words, sparsity exists in both features and samples jointly. Biclustering methods are a potential solution to this problem, and further generalize the sparsity principle by considering samples and features as exchangeable concepts to handle local sparsity (Cheng and Church 2000, Kriegel et al. 2009). For example, gene expression data can be represented as a matrix with genes as columns, and subjects as rows (with various and possibly unknown diseases or tissue types). Traditional methods will either cluster the rows—as done, for example, in microarray research, where researchers want to find subpopulation structure among subjects to identify possible common disease status—or cluster the columns, as done, for example, in gene clustering research, where genes are of interest and the goal is to predict the biological function of novel genes from the function of other well studied genes within the same clusters. In contrast, biclustering involves clustering rows and columns simultaneously in order to account for the interaction of row and column sparsity. This local sparsity perspective provides an intuition for using sparse singular value decomposition algorithms (SSVD) for biclustering Busygin et al. (2002), Lee et al. (2010), Yang et al. (2014), Witten et al. (2009). SSVD assumes that the signal in the data matrix can be represented by a low rank matrix $\mathbf{X} \approx \mathbf{UDV^T} = \sum_{i=1}^{r} d_i \mathbf{u}_i \mathbf{v}_i^T$ with $\mathbf{X} \in \Re^{n \times p}$. $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r] \in \Re^{n \times r}$ and $\mathbf{V} = [\mathbf{v}_1,\ \mathbf{v}_2, \ldots, \mathbf{v}_r] \in \Re^{r \times p}$ contain left and right sparse singular vectors and are orthonormal with only a few non-zero elements (corresponding to local sparsity). $\mathbf{D} \in \Re^{r \times r}$ is diagonal (with diagonal elements $d_1, d_2, \ldots, d_r$) with $r << rank(\mathbf{X})$. The outer product of each pair of sparse singular vectors ($\mathbf{u}_i \mathbf{v}_i^T$, $i = 1, 2, \ldots, r$) will designate two biclusters corresponding to positive and negative elements respectively.

A common assumption of existing SSVD biclustering methods is that the observed data can be decomposed into a signal matrix plus a fully exchangeable random noise matrix:

$$\mathbf{X} = \boldsymbol{\Xi} + \boldsymbol{\Phi}, \tag{2.1}$$

where $\mathbf{X}$ is the observed data, $\boldsymbol{\Xi} = (\xi_{ij})$ is an $n \times p$ matrix representing the signal, and $\boldsymbol{\Phi} = (\phi_{ij})$ is an $n \times p$ random noise/residual matrix with i.i.d. entries (Yang et al. 2014, Hoff 2010, Johnstone and Lu 2009). A method based on model (2.1) is proposed in Lee et al. (2010) which minimizes the sum of the Frobenius norm of $\mathbf{X} - \hat{\boldsymbol{\Xi}}$ and a penalty function with variable selection, such as the ℓ_1-norm (Tibshirani 1994) or $SCAD$ (Fan and Li 2001). A similar Loss plus Penalty minimization approach can be seen in Witten et al. (2009). A different method for SSVD employs iterative thresholding QR decomposition to estimate $\hat{\boldsymbol{\Xi}}$ in Yang et al. (2014). We refer to Lee et al. (2010) as LSHM and Yang et al. (2014) as FIT-SSVD, and compare these approaches to our method. An alternative approach, which is more direct, is based on a mixture model (Lazzeroni and Owen 2002, Shabalin et al. 2009). For example, Shabalin et al. (2009) defines the bicluster as a submatrix with a large positive or negative mean. Although these approaches have proven successful in some settings, they are limited by their focus on only the mean signal approximation. In addition, the explicit homogeneous residual variance assumption is too restrictive in many applications.

To our knowledge, the only extension of the traditional model given in (2.1) is the generalized PCA approach (Allen et al. 2014) which assumes that if the random noise matrix were stacked into a vector, vec($\boldsymbol{\Phi}$), it would have mean $\mathbf{0}$ and variance $\mathbf{R}^{-1} \otimes \mathbf{Q}^{-1}$, where $\mathbf{R}^{-1}$ is the common covariance structure of the random variables within the same column, and $\mathbf{Q}^{-1}$ is the common covariance structure of the random variables within the same row. This approach is especially suited to denoising nuclear magnetic resonance data for which there is a natural covariance structure of the form given above (Allen et al. 2014). Drawbacks of the generalized PCA method, however, are that it remains focused on mean signal approximation and the structure of $\mathbf{R}^{-1}$ and $\mathbf{Q}^{-1}$ must be explicitly known in advance.

In this chapter, we present a new biclustering framework based on sparse singular value decomposition called heterogeneous sparse singular value decomposition (HSSVD). This method can detect both mean biclusters and variance biclusters in the presence of unknown heterogeneous residual variance. To our knowledge, both the heterogeneous residual variance and variance only bicluster detection aspects are completely novel. We also apply our method, as well as competing approaches, to two cancer data sets, one with methylation data and the other with gene expression data. Our method delivers better pattern detection and is able to confirm the biological findings originally made for each of the data sets. We also apply our method to synthetic data to demonstrate its superior performance over competing approaches quantitatively. We demonstrate that our proposed method is robust, location and scale invariant, and computationally feasible.

2.2 Model Assumptions for HSSVD

We define biclusters as subsets of the data matrix which have the same mean and variance. We assume that there exists a dominate null cluster in which all elements have a common mean and variance and that all other biclusters are restricted to rectangular structures which have either a distinct mean or variance compared to the null cluster. We can also express our model in the framework of a random effect model wherein

$$\mathbf{X} = \mathbf{\Xi} + \rho^2 \mathbf{\Sigma} \times \mathbf{\Phi} + b\mathbf{J}, \tag{2.2}$$

where $\mathbf{X}$ and Ξ are the same structures given in the traditional model (2.1), and where we require $\mathbf{\Phi}$, an n×p matrix, to have i.i.d. random components with mean 0 and variance 1. Moreover, the × in (2.2) is defined element wisely: see the next section for details. New components in the model include $\mathbf{\Sigma} = (\sigma_{ij})$, an $n \times p$ matrix representing the heterogeneous variance signal; $\mathbf{J}_{n \times p}$, an $n \times p$ matrix with all values equal to 1; ρ, a finite positive number serving as a common scale factor; and b, a finite number serving as a common location factor. We also make the sparsity assumption that the majority of (ξ_{ij}) values are 0 and the majority of (σ_{ij}) values are 1. Further, just as we assumed for the mean structure Ξ, we also assume that the variance structure $\mathbf{\Phi}$ is low rank.

From the definitions, the traditional model (2.1) is a special case of our model (2.2), with $b = 0$, $\mathbf{\Sigma} = \mathbf{J}$, and $\rho = 1$. The presence of b and ρ in the model allows the new method to be scale invariant, while the presence of $\mathbf{\Sigma}$ enables the new method to incorporate heterogeneous variance signals.

2.3 HSSVD method

We propose HSSVD based on the model (2.2) with a hierarchical structure for signal recovery. First, we properly scale the matrix elements to minimize false detection of pseudo mean biclusters which can arise as artifacts of high-variance clusters. This motivates us to add the quadratic rescaling step in the procedure. Then we can detect mean biclusters based on the scaled data and later detect variance biclusters based on the logarithm of the squared residual data after subtracting out the mean biclusters. The quadratic rescaling step works well in practice, as shown in the simulation studies and data analysis. The pseudo code for the algorithm is provided as follows:

1. Input Step: Input the raw data matrix $\mathbf{X}_{origin}$. Standardize $\mathbf{X}_{origin}$ (treat each cell as i.i.d.) to have mean 0 and variance 1. Denote the overall mean of $\mathbf{X}_{origin}$ as $\hat{\mu}$ and the overall standard deviation as $\hat{\sigma}$, and let the standardized matrix be

defined as $\mathbf{X} = (\mathbf{X}_{origin} - \hat{\mu}\mathbf{J})/\hat{\sigma}$.

2. Quadratic Rescaling: Apply SSVD on $\mathbf{X}^2 - \mathbf{J}$ to obtain the approximation matrix $\mathbf{U}$.

3. Mean Search: Let $\mathbf{Y} = \mathbf{X}/\sqrt{\mathbf{U} + \mathbf{J} - c\mathbf{J}}$, where c is a small nonpositive constant to ensure that $\sqrt{\mathbf{U} + \mathbf{J} - c\mathbf{J}}$ exists. Then apply SSVD on $\mathbf{Y}$ to obtain the approximation matrix $\tilde{\mathbf{Y}}$.

4. Variance Search: Let $\mathbf{Z}_{origin} = \log(\mathbf{X} - \tilde{\mathbf{Y}} \times \sqrt{\mathbf{U} + \mathbf{J} - c\mathbf{J}})^2$, center $\mathbf{Z}_{origin}$ to have mean 0, and denote the centered version as $\mathbf{Z}$. Perform SSVD on $\mathbf{Z}$ to obtain the approximation matrix $\tilde{\mathbf{Z}}$.

5. Background Estimation: Let $\mathbf{P} = \{p_{ij}\}$ denote the $n \times p$ matrix of indicators of whether the corresponding cells belong to the background cluster, with $p_{ij} = 1$ if both $\tilde{\mathbf{Y}}_{ij} = 0$ and $\tilde{\mathbf{Z}}_{ij} = 0$, and $p_{ij} = 0$ otherwise. Based on the assumption that most elements in the matrix should be in the null cluster, we can estimate $\hat{b}$ with $\frac{\mathbf{1}'(\mathbf{X}_{origin} \times \mathbf{P})\mathbf{1}}{\mathbf{1}'\mathbf{P}\mathbf{1}}$ and $\hat{\rho}$ with $\frac{\mathbf{1}'(\mathbf{X}_{origin} \times \mathbf{P} - \hat{b}\mathbf{P})^2\mathbf{1}}{\mathbf{1}'\mathbf{P}\mathbf{1} - 1}$, where $\mathbf{1}$ is a vector with all elements equal to one.

6. Scale Back: Define $\mathbf{P}_1 = \{p_{ij}\}$, with $p_{ij} = 1$ if $\tilde{\mathbf{Y}}_{ij} = 0$, $p_{ij} = 0$ otherwise. Similarly, define $\mathbf{P}_2 = \{p_{ij}\}$, with $p_{ij} = 1$ if $\tilde{\mathbf{Z}}_{ij} = 0$, $p_{ij} = 0$ otherwise. The mean $(\mathbf{\Xi} + b\mathbf{J})$ approximation is computed with $\hat{\sigma}(\tilde{\mathbf{Y}} \times \sqrt{\mathbf{U} + \mathbf{J} - c\mathbf{J}}) + \hat{\mu}(\mathbf{J} - \mathbf{P}_1) + \hat{b}\,\mathbf{P}_1$, and the variance $(\rho^2\mathbf{\Phi})$ approximation is computed with $[\hat{\rho}^2\,\mathbf{P}_2 + \hat{\sigma}^2(\mathbf{J} - \mathbf{P}_2)] \times \exp(\tilde{\mathbf{Z}})$.

The operators “×”,“/”, “exp()”,“log()”, “exp()”, “min()” and “$\sqrt{()}$” used above are defined element wisely when they are applied to the matrix, e.g. $\mathbf{U}_{n\times p} \times \mathbf{V}_{n \times p} = (u_{ij}v_{ij})$. In all steps involving sparse singular value decomposition, we implement the FIT-SSVD method Yang et al. (2014). We use FIT-SSVD since it is computational fast and has similar or superior performance compared to other competing methods under the homogeneous variance assumption Yang et al. (2014). The matrix $\sqrt{\mathbf{U}+\mathbf{J}-c\mathbf{J}}$ provides a working variance level estimate of the data and makes our method more robust. Note that the reason for working on the log scale for the variance detection is two fold. First, working on the log scale makes the detection of the deflated variance (less than 1) bicluster possible. Intuitively, as variance measures deviance from the mean, we can work on the squared residuals to find the variance structure. For the deflated variance bicluster setting, if the mean structure is estimated correctly, the residuals within the bicluster are close to zero. The SSVD based methods shrink the small non-zero elements to zero to achieve sparsity. As a result, if we work on the squared residuals directly, the SSVD based methods will fail to detect the low variance structure. Second, to use the well-established SSVD method in the variance detection steps we need to work on the log scale. To see this, we can rewrite the equation in (2.2) as $\log(\mathbf{X}-\mathbf{\Xi}-b\mathbf{J})^2 = \log(\mathbf{\Sigma}^2)+\log(\rho^2\mathbf{\Phi}^2)$, which is similar to the model in (2.1). Consequently, we can apply any methods which are applicable to (2.1) in our variance detection step if we work on the log scale and $\mathbf{\Phi}$ is low rank. We also want to point out that results obtained directly from FIT-SSVD are relative to the location and scale of the background cluster. In addition, we have scaled the data in the "Input Step". To provide a correct mean and variance approximation of the original data, we need the "Scale Back" step. Assuming that the detection of null clusters is close to the truth, then the pooled mean and variance estimates based on elements exclusively from the identified null cluster ($\hat{b}$ and $\hat{\rho}$) are more accurate than estimates based on all elements of the matrix ($\hat{\mu}$ and $\hat{\sigma}$). As a result, we need to use the comprehensive formula proposed in the "Scale Back" step.

The FIT-SSVD method, as well as any other SVD based method, requires an approximation of the rank of the matrix (which is essentially the number of true biclusters) as input. We adapt the bi-cross validation method (BCV) by Owen and Perry (2009) for rank estimation, and we notice that in some cases the rank is underestimated. For this reason, we introduce additional steps following a BCV rank estimation of rank k: First, we approximate the data with a sparse matrix $\hat{\mathbf{X}}_{k+1}$ (rank $= k+1$), where $\hat{\mathbf{X}}_{k+1} = \sum_{j=1}^{k+1} \hat{d}_j \hat{\mathbf{u}}_j \hat{\mathbf{v}}_j^T$. Define the proportion of variance explained by the top i rank sparse matrix as $R_i = \sum_{j=1}^{i} \hat{d}_j^2 / \sum_{j=1}^{k+1} \hat{d}_j^2$ (Allen and Maletic-Savatic 2011). R_i is between 0 and 1 and is increasing with i, and we believe that the redundant components of the sparse matrix should not contribute much to the total variance. The final rank estimation for HSSVD is the smallest integer r which satisfies $R_r > 0.95$, and $1 \leq r \leq k+1$. Note that FIT-SSVD (Yang et al. 2014) used the modified BCV method for rank estimation, however, the authors require that most rows (the whole row) and most columns (the whole column) are sparse, which appears to be too restrictive. In practice, this assumption is violated if the data is block diagonal or has certain other commonly assumed data structures. For this reason, we use the original BCV method as our starting point.

2.4 Application to cancer data

2.4.1 Hypervariability of methylation in cancer

We demonstrate the capability of variance bicluster detection with methylation data in cancer versus normal patients (Hansen et al. 2011). The experiments were conducted by a custom nucleotide-specific Illumina bead array to increase the precision of DNA methylation measurements on previously identified cancer-specific differentially methylated regions (cDMRs) in colon cancer (Irizarry et al. 2009). The data set (GEO accession: GSE29505) consists of 290 samples including cancer samples (colon, breast, lung, thyroid and Wilms' tumor cancers) and matched normal samples. Each sample had 384 methylation probes which covered 151 cDMRs. The authors of the primary report concluded that cancer samples had hypervariability in these cDMRs across all cancer types (Hansen et al. 2011).

First, we wish to verify that HSSVD can provide a good mean signal approximation of methylation. In this data set, all the probes measuring the methylation are placed in the cDMRs identified in colon cancer patients. As a result, we would expect that mean methylation levels differ between colon cancer samples and the matched normal samples. Under this assumption, we require the biclustering methods to capture this mean structure before investigating the information gained from variance structure estimation. Note that the numerical range of methylation level is between 0 and 1. Hence we applied the logit transformation on the original data for further biclustering analysis. We compare three methods, HSSVD, FIT-SSVD and LSHM; all based on SVD. Only colon cancer samples and their matched normal samples are used for this particular analysis. In Figure 2.2, we can see from the hierarchical clustering analysis that the majority of colon cancer samples (labeled blue in the side bar) are grouped together and most of the cDMRs are differentially expressed in colon tumor samples compared to normal samples. The conclusion is the same for all three methods compared, including our proposed HSSVD method.

Second, our proposed HSSVD method confirms the most important finding in Hansen et al. (2011) that cancer samples tended to have hypervariability in methylation level regardless of tumor subtype. We compared the mean approximation and variance approximation results of HSSVD. All samples were used in this analysis. The variance approximation of HSSVD (see Figure 2.3(a)) shows that nearly all normal samples have low variance compared to cancer samples, and this pattern is consistent across all cDMRs. Notably, our method provides additional information beyond the

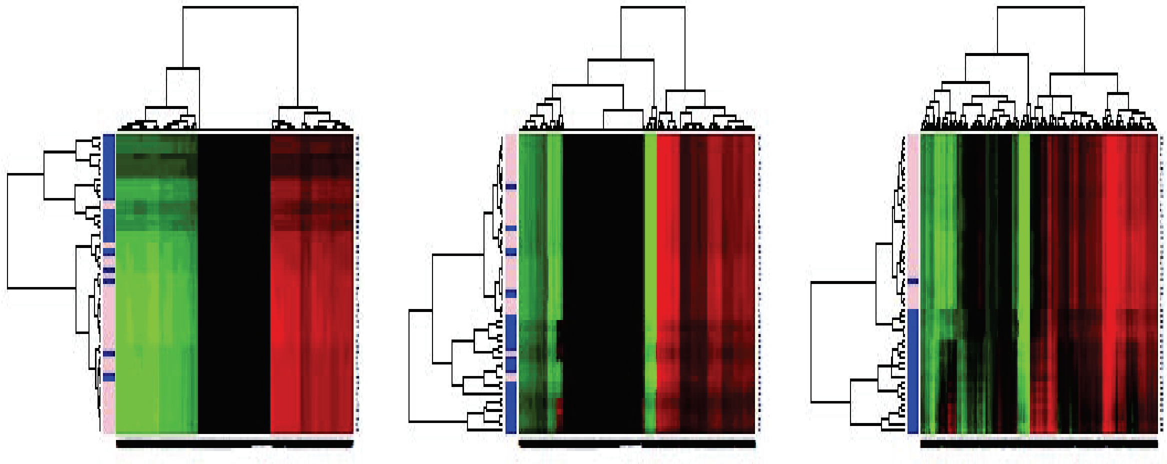

Figure 2.2: Mean Approximation of colon cancer and the normal matched samples. From left to right the methods are HSSVD, FIT-SSVD and LSHM. The colon cancer samples are labeled in blue, and the normal matched samples are labeled in pink in the sidebar. The genes and samples are ordered by hierarchical clustering. The colon cancer patients are clustered together which indicates the mean approximations for these three methods achieves the expected signal structure.

conclusion from Hansen et al. (2011). Specifically, our variance approximation suggests that some cancer samples are not characterized by hypervariability in methylation level for certain cDMRs. More precisely, some cDMRs for a few cancer samples (surrounded by normal samples) are predicted to have low variance (see lower left part of Figure 2.3(a)). Our method also highlights cDMRs with the greatest contrast variance between cancer and normal samples. This is new information and the corresponding cDMRs with high contrast variance (especially some of the first and middle columns of Figure 2.3(a)) warrant further study for biological and clinical relevance. We also want to emphasize that the analysis in Hansen et al. (2011) relies on the disease status information, while for HSSVD, the disease status is only used for result interpretation. Note that most cancer patients cluster together by hierarchical clustering of the variance approximation from HSSVD. In contrast, clustering the mean approximation from HSSVD in Figure 2.3(b) fails to reveal such a pattern. This indicates that most cancer samples may have hypervariability of methylation as a common feature while their mean level methylation varies from sample to sample. Hence, identifying variance biclusters can provide potential new insight for cancer epigenesis.

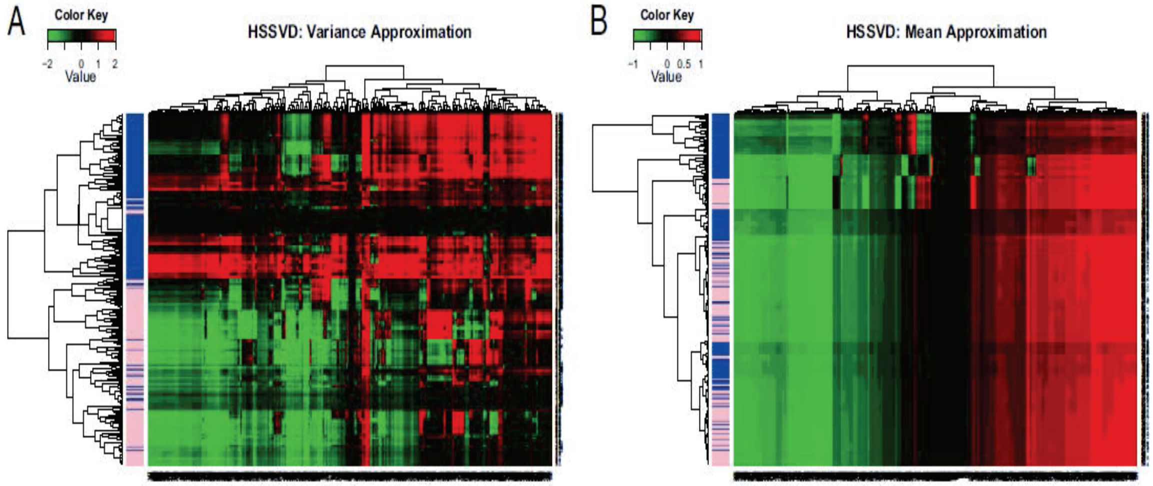

Figure 2.3: HSSVD approximation result for all samples. The variance approximation is in panel (a) and the mean approximation is in panel (b). Blue represents cancer samples, and pink represents normal samples in the sidebar. The genes and samples are ordered by hierarchical clustering. Red color represents large values, and green color represents small values. Only the variance approximation can discriminate between cancer and normal samples. More importantly, within the same gene, the heatmap for the variance approximation indicates that cancer patients have larger variance than normal individuals. This result matches the conclusion in Hansen et al. (2011). In addition, the cDMRs with the greatest contrast variance across cancer and normal samples are highlighted by the variance approximation, while the original paper does not provide such information.

2.4.2 Gene expression in lung cancer

Some biological settings, in contrast to the methylation example above, do not express variance heterogeneity. Usually, the presence or absence of such heterogeneity is not known in advance for a given research data set. Thus it is important to verify that the proposed approach remains effective in either case for discovering mean-only biclusters. We now demonstrate that even in settings without variance heterogeneity, HSSVD can outperform other methods, including FIT-SSVD (Yang et al. 2014), LSHM (Lee et al. 2010) and traditional SVD. We utilize a lung cancer data set which has been studied in the statistics literature (Lee et al. 2010, Shabalin et al. 2009, Yang et al. 2014). The samples are a subset of patients (Liu et al. 2008) having lung cancer with gene expression measured by the Affymetrix $95av2$ GeneChip (Bhattacharjee et al. 2001). The data set contains the expression levels of $12,625$ genes for 56 patients, each having one of four disease subtypes: normal lung (20 samples), pulmonary carcinoid tumors (13 samples), colon metastases (17 samples), and small cell carcinoma (6 samples).

The performance of different methods is evaluated based on the pattern difference of subtypes based on the mean approximations. For all methods, we set the rank of the mean signal matrix equal to 3 to maintain consistency with the ranks used in FIT-SSVD (Yang et al. 2014) and LSHM (Lee et al. 2010). Further, we use the measurement "support" to evaluate the sparsity of the estimated gene signal (Yang et al. 2014). "Support" is the cardinality of the non-zero elements in the right and left singular vectors across the three layers (i.e., "support" is an integer that cannot exceed the data dimension). Smaller "support" values suggest a sparser model. Table 2.1 shows that HSSVD, FIT-SSVD and LSHM yield similar levels of sparsity in the gene signal, while SVD is not sparse, as expected. Figure 2.4 shows checkerboard plots of rank-three approximations by the four methods. Patients are placed on the vertical axis, and the patient order is the same for all images. Patients within the same subtype are stacked together and different subtypes are separated by white lines. Within each image, genes are laid on the horizontal axis and are ordered by the value of $\hat{\mathbf{v}}_2$ (Yang et al. 2014). We can see a clear block structure in both the FIT-SSVD and HSSVD methods, indicating biclustering. The block structure suggests we can discriminate the four cancer subtypes using either the FIT-SSVD or HSSVD methods, while LSHM and SVD are unable to achieve such separation among subtypes.

Table 2.1: Lung cancer: summary of cardinality of union support of the first three singular vectors for different methods

	HSSVD	FIT-SSVD	LSHM	SVD
$\cup_{i=1}^{3}\|\mathbf{u}_i\|_0$	4689	4686	4655	12625
$\cup_{i=1}^{3}\|\mathbf{v}_i\|_0$	56	56	56	56

2.5 Simulation study

To evaluate the performance of HSSVD quantitatively, we conducted a simulation study. We compared HSSVD with the most relevant existing biclustering methods, FIT-SSVD and LSHM Yang et al. (2014), Lee et al. (2010). HSSVD includes a rank estimation component, while the other methods do not automatically include this. For this reason, we will use a fixed oracle rank (at the true value) for the non-HSSVD methods. For comparison, we also evaluate HSSVD with fixed oracle rank (HSSVD-O).

The performance of these methods on simulated data was evaluated on four criteria. The first criterion is "sparsity of estimation", defined as the ratio between the size of the correctly identified background cluster and the size of the true background cluster. The second criterion is "biclustering detection rate", defined as the ratio of the intersection of the estimated bicluster and the true bicluster over their union (also known as the Jaccard index). For the first two criteria, larger values indicate better performance. The third and fourth criteria are "overall matrix approximation errors" for mean and variance biclusters, consisting of the scaled recovery error for the low-rank mean signal matrix $\tilde{\boldsymbol{\Xi}} = \boldsymbol{\Xi} + b\mathbf{J}$, computed via

$$L_{mean}(\tilde{\boldsymbol{\Xi}}, \hat{\boldsymbol{\Xi}}) = \|\hat{\boldsymbol{\Xi}} - \tilde{\boldsymbol{\Xi}}\|_F^2 / \| \boldsymbol{\Xi} \|_F^2;$$

and the scaled recovery error for the low-rank variance signal matrix $\log(\tilde{\boldsymbol{\Sigma}}) = \log(\boldsymbol{\Sigma}) +$

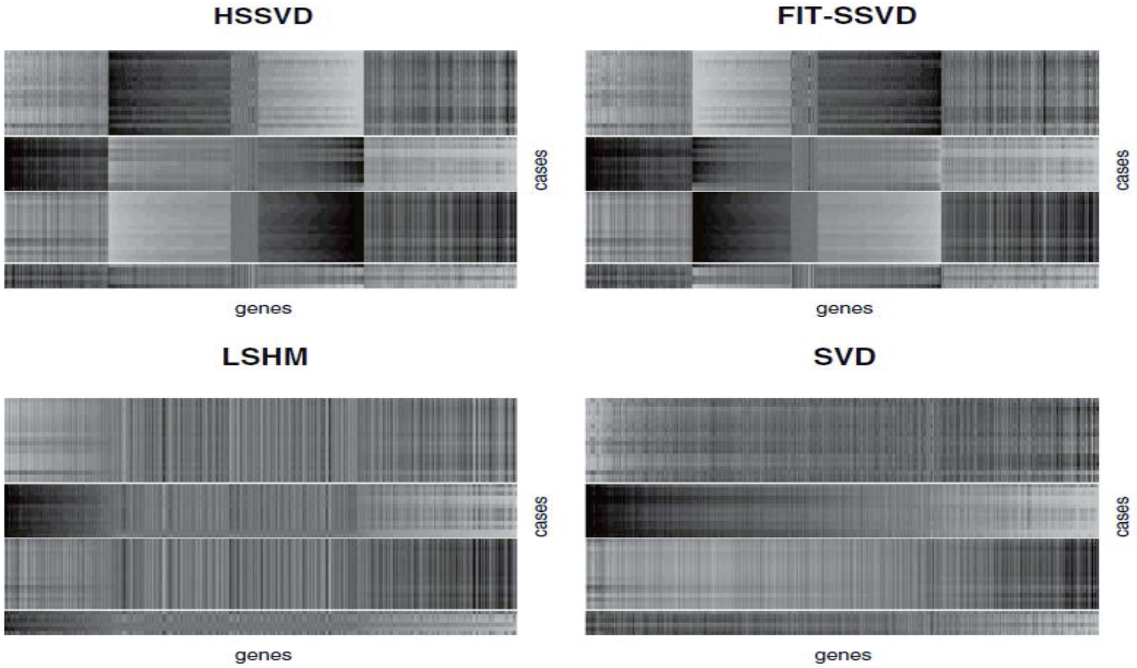

Figure 2.4: Checkerboard plots for four methods. We plot the rank-three approximation for each method. Within each image, samples are laid in rows, and genes are in columns. We order the samples by subtype for all images (top to bottom: Carcinoid, Colon, Normal, and Smallcell), and different subtypes are separated by white lines. Genes are sorted by the estimated second right singular vector ($\hat{u}_2$), and we only included genes that are in the support (defined in Table 2.1). Across all methods, the HSSVD and FIT-SSVD methods provide the clearest block structure reflecting biclusters.

$\log(\rho^2 \mathbf{J})$, computed via

$$L_{var}(\log(\tilde{\mathbf{\Sigma}}), \log(\hat{\mathbf{\Sigma}})) = \| \log(\hat{\mathbf{\Sigma}}^{1/2}) - \log(\tilde{\mathbf{\Sigma}}^{1/2}) \|_F^2 / \| \log(\mathbf{\Sigma}^{1/2}) \|_F^2,$$

with $\|.\|_F$ being the Frobenius norm.

The simulated data comprise a 1000×100 matrix with independent entries. The background entries follow a normal distribution with mean 1 and standard deviation 2. We denote the distribution as $N(1, 2^2)$, where $N(a, b^2)$ represents a normal random variable with mean a and standard deviation b. There are five non-overlapping rectangularly shaped biclusters: bicluster 1, bicluster 2, and bicluster 5 are mean clusters, bicluster 3 is a mean and small variance cluster, and bicluster 4 is a large variance cluster. More precisely, bicluster 1 (size 100 × 20) is generated from $N(7, 2^2)$, bicluster 2 (size 100 × 10) is generated from $N(-5, 2^2)$, bicluster 3 (size 100 × 10) is generated from $N(7, 0.4^2)$, bicluster 4 (size 100 × 20) is generated from $N(1, 8^2)$, and bicluster 5 (size 100 × 20) is generated from $N(6.8, 2^2)$. The biclustering results are shown in Table 2.2: HSSVD and HSSVD-O can detect both mean and variance biclusters, while FIT-SSVD-O and LSHM-O can only detect mean biclusters. For mean bicluster detection, all methods performed well since the "biclustering detection rates" are all greater than 0.7. For variance bicluster detection, HSSVD and HSSVD-O deliver a similar "biclustering detection rate". On average, the computation time of LSHM-O is about 30 times that of HSSVD and 60 times that of FIT-SSVD-O.

Both FIT-SSVD and LSHM are provided with the oracle rank as input. We also evaluated an automated rank version for these methods, but determined the performance was worse than the corresponding oracle rank version (results not shown). Note that the input data are standardized to mean 0 and standard deviation 1 element-wise for FIT-SSVD-O and LSHM-O. Although this step is not mentioned in the original papers Lee et al. (2010), Yang et al. (2014), this simple procedure is critical for accurate mean bicluster detection. From Table 2.2, we can see that HSSVD-O provides the best overall performance, while HSSVD is close to the best; however, in practice, the oracle rank is unknown. For this reason, HSSVD is clearly the best method and is the only fully automated approach which delivers robust mean and variance detection in the present of unknown heterogeneous residual variance among those considered.

Table 2.2: Comparison of four methods in the simulation study. The L_{mean} and L_{var} is measuring the difference between the approximated signal and the true signal, and so smaller is better. For the other measures of accuracy of bicluster detection, the larger the better. The rows “BLK1” to “BLK5” represent the “biclustering detection rate” for each bicluster.“-O” indicates that the oracle rank is provided.

	HSSVD		HSSVD-O		FITSSVD-O		LSHM-O	
L_{mean}	0.013	(0.01)	0.013	(0.01)	0.081	(0.01)	0.019	(0.01)
L_{var}	0.157	(0.03)	0.156	(0.03)	NA		NA	
Sparsity	0.950	(0.04)	0.950	(0.03)	0.988	(0.02)	0.997	(0.01)
BLK1 (mean)	0.861	(0.10)	0.862	(0.10)	0.818	(0.08)	0.872	(0.08)
BLK2 (mean)	0.934	(0.18)	0.936	(0.17)	0.939	(0.18)	0.976	(0.01)
BLK3 (mean)	0.972	(0.10)	0.974	(0.10)	0.971	(0.11)	0.987	(0.01)
BLK5 (mean)	0.977	(0.11)	0.948	(0.11)	0.977	(0.11)	0.996	(0.01)
BLK3 (var)	0.977	(0.02)	0.977	(0.02)	NA		NA	
BLK4 (var)	0.628	(0.25)	0.633	(0.24)	NA		NA	

2.6 Properties of HSSVD

2.6.1 HSSVD as a denoising procedure

We study overlapped mean bicluster detection through a simulated example. The plaid model is usually preferred to SVD based methods for overlapped mean bicluster detection, although the plaid model (Lazzeroni and Owen 2002, Turner et al. 2005) can be sensitive to variance heterogeneity. We want to show that our method as an SVD based method is still useful for overlapped mean bicluster detection as a denoising step. First, we represent the data in a 200×100 matrix. The elements in the null cluster follow $N(0, 1^2)$. At the same time, we have two overlapping biclusters with their sizes both equal to 20×20. The elements in the two biclusters follow $N(7, 2^2)$ and $N(-5, 3^2)$, respectively, and the overlapped block size is 10×10. Hence, under the additive assumption, the elements in the overlapped block follow $N(2, \sqrt{13}^2)$. Here, we only focus on mean bicluster detection since this is the traditional purpose of biclustering methods. For comparison, we could directly apply the plaid model or the HSSVD method on the raw data to detect mean biclusters. Alternatively, we could first apply HSSVD to obtain a mean approximation of the raw data and then apply the plaid model to the mean approximation to detect biclusters. We utilize the "BCPlaid" function in the R package "biclust" (http://CRAN.R-project.org/package=biclust) as the implementation of the plaid model (Lazzeroni and Owen 2002, Turner et al. 2005). The graphical results are presented in Figure 2.5. The detected biclusters are highlighted by a black frame. From Figure 2.5 (b), we see that although sparse singular value decomposition is good at mean signal approximation for non-overlapping bi-clusters, it cannot recover the true overlapped bicluster structure. Meanwhile, we can see that after applying HSSVD, the plaid model (Figure 2.5 (c)) successfully picks out the underlying true structure, while applying the plaid model alone (Figure 2.5 (a)) was not successful. This result implies that for overlapped mean bicluster detection, the plaid model is generally better, but when there is variance heterogeneity present, the HSSVD can be quite helpful as a denoising process.

2.6.2 The necessity of the variance detection step in HSSVD

As we assume a low rank structure in both mean signal and variance signal, a natural question to ask is whether such a structure can be approximated well by a higher-rank matrix for the mean structure only. In other words, can we represent the variance biclusters by using pseudo-mean biclusters. Our conclusion is that it is improper to use mean biclusters for variance bicluster detection. Pseudo-mean biclusters cannot recover the small variance biclusters at all, due to the natural shrinkage inherent in sparse singular value decomposition methods. Further, we will show that pseudo-mean biclusters can reveal some structure for the large variance biclusters, however the approximation is rough. Consider the simulation data in Example 1 (see Figure 2.6 for graphical display).

Here, we can compare the bicluster detection results of FIT-SSVD (in Figure 2.7)

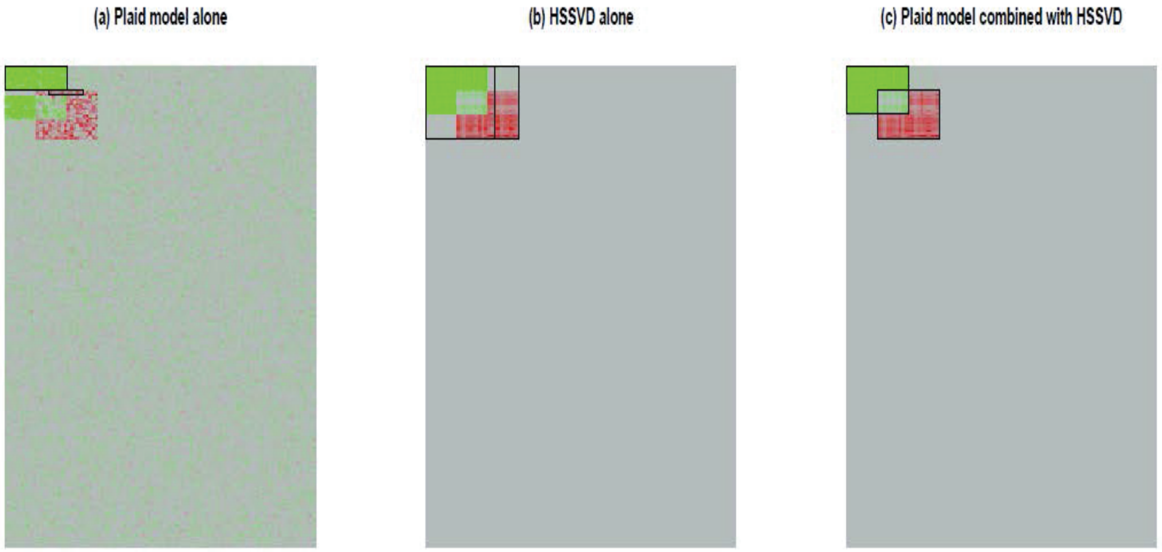

Figure 2.5: Overlapped bicluster detection by the plaid model and HSSVD. In panel (a), we draw the original data and the plaid model detection result is highlighted with a black frame. In panel (b), the HSSVD detection result is highlighted with a black frame. In panel (c), we obtain the mean approximation by HSSVD first and then apply the plaid model detection result onto the mean approximation data.

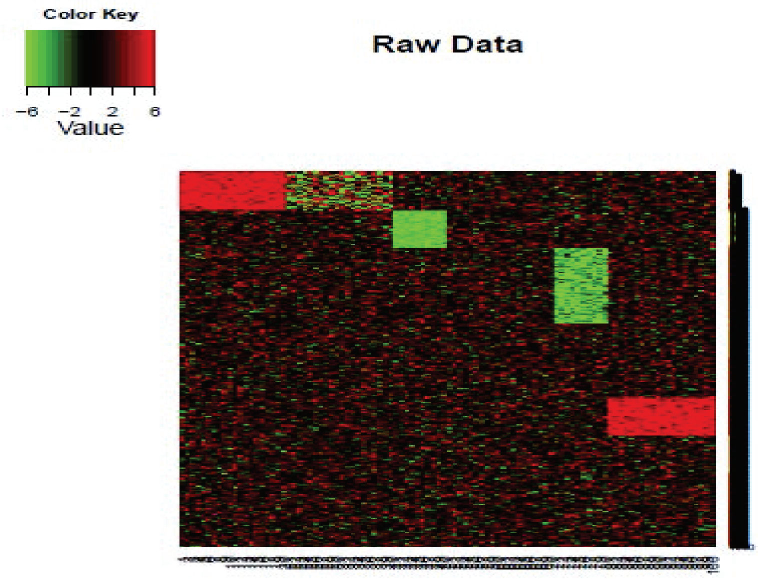

Figure 2.6: The image of raw data for Example 1. There are five biclusters. Red is for positive values and green is for negative values.

with input rank equal to 6 versus HSSVD with an estimated rank (in Figure 2.9). We can see that there are four mean biclusters from Figure 2.6. As the input rank is greater than the mean bicluster number, there will be several pseudo-mean biclusters (layer 5 and layer 6) in the FIT-SSVD result. For HSSVD, there will be both mean biclusters and variance biclusters. In Figure 2.7, it appears that the pseudo mean biclusters can detect part of the variance biclusters, however, this is because we know the "correct" order to display the graph. However, in practice, we would not know this order. Moreover, we can see that the pseudo-mean structure can be confounded with a type of true bicluster.

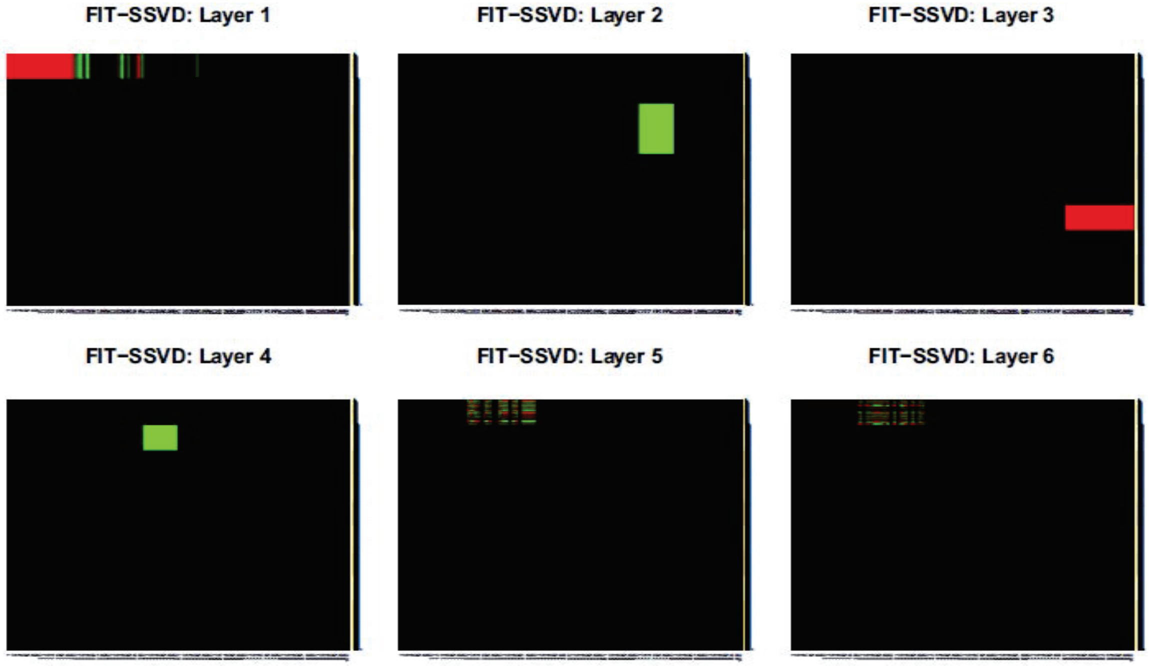

Figure 2.7: FIT-SSVD for Example 1. Each layer represents one bicluster. Layer 5 and 6 are pseudo mean biclusters.

For example, let $\mathbf{X}_0 = \mathbf{u}\mathbf{v}^T + \mathbf{\Phi}$,$\mathbf{u} = (\text{rep}(1,50), \text{rep}(1,50), \text{rep}(0,900))$, $\mathbf{v} = (\text{rep}(1,5), \text{rep}(-1,5), \text{rep}(0,90))$, where $\text{rep}(1,5) = (1,1,1,1,1)$ and $\mathbf{\Phi}$ is a 1000 matrix with entries follow i.i.d N(0,1). There is only a mean bicluster for X_0, and we can apply FIT-SSVD with input rank = 1. When we compare the mean bicluster result for X_0 and the pseudo-mean biclusters, layer 5 or layer 6 for Example 1, graphically, the resulting heatmaps, given in Figure 2.8 (rows and columns are reordered by hierarchical clustering), are very similar. This indicates that pseudo-mean biclusters can be confounded with certain true mean biclusters. This issue is probably even more complicated for real data settings.

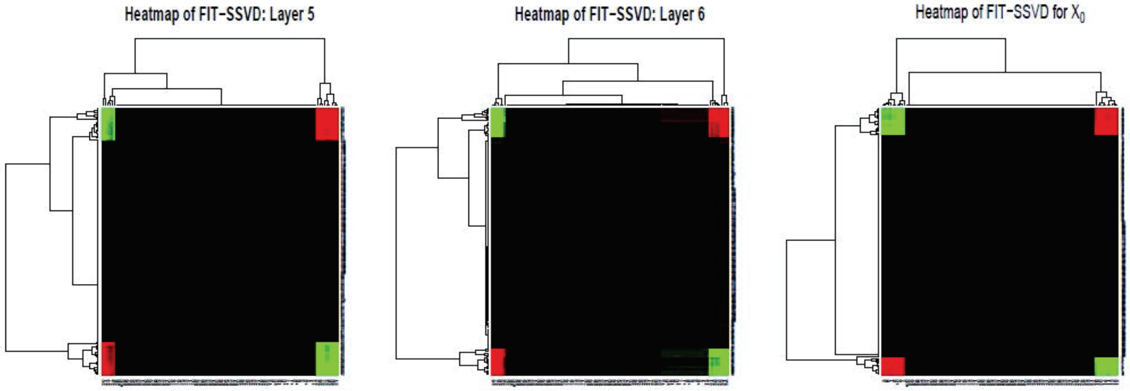

Figure 2.8: Heatmaps of two pseudo mean biclusters and a true mean bicluster. The rows and columns are reorders by hieratical clustering. Only the first 200 rows (original order) are shown for better display (the remaining rows are all 0).

In contrast, HSSVD can provide more accurate large variance bicluster detection (layer 1 for variance) and small variance bicluster detection (layer 2 for variance), as shown in Figure 2.9. Lastly, we want to emphasize that the BCV method (Owen and Perry 2009) can be quite helpful for preventing the type of pseudo-mean detection which can weaken variance detection in the latter steps.

2.7 Conclusion and Discussion

In this chapter, we introduced HSSVD, a statistical framework and its implementation to detect biclusters with potentially heterogeneous variances. Compared to existing methods, HSSVD is both scale invariant and rotation invariant (as the quantity for scaling is the same for all matrix entries and does not vary by row or column). HSSVD also

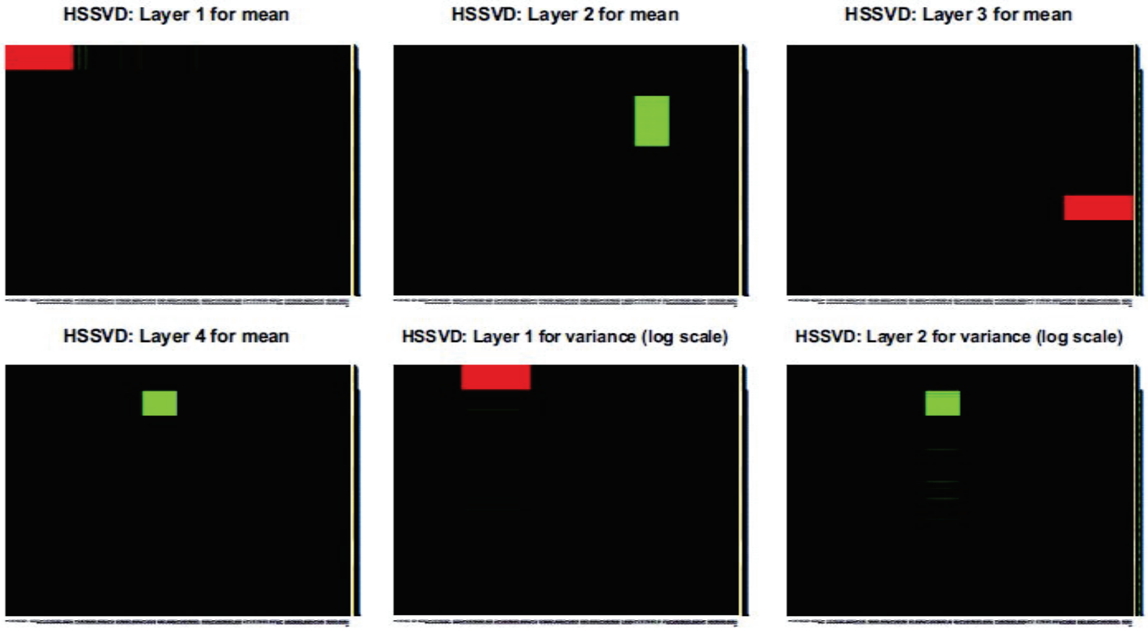

Figure 2.9: HSSVD results for Example 1. Each layer represents one bicluster. There are four mean biclusters and two variance biclusters.

has the advantage of working on the log scale in estimating the variance components: the log scale makes detection of low variance (less than 1) biclusters possible, and any traditional sparse singular value decomposition method can be naturally utilized in our variance detection steps. The new method confirms the existence of methylation hypervariability in the methylation data example, something which cannot be done with other existing biclustering methods. Although we use the FIT-SSVD method in our implementation, other low rank matrix approximation methods are applicable. Moreover, the software implementing our proposed approach was computationally comparable to the other approaches we evaluated.

A potential shortcoming of SVD based methods is their inability to detect overlapping biclusters. We show that our method can serve as a denoising process for overlapping bicluster detection. In particular, we can first apply the HSSVD method on the raw data to obtain the mean approximation. Then we can apply a suitable approach, such as the widely used plaid model (Lazzeroni and Owen 2002, Turner et al. 2005), on the mean approximation to detect overlapping biclusters. This combined procedure improves on the performance of the plaid model when the overlapping biclusters have heterogeneous variance. Hence our method remains useful in the present of overlapping biclusters.

Another potential issue for HSSVD is the question of whether a low rank mean approximation plus a low rank variance approximation could be alternatively represented by a higher rank mean approximation. In another words, is it possible to detect variance biclusters through mean biclusters only, even though the mean clusters which form the variance clusters would be pseudo-mean clusters. Our conclusion is that the variance detection step in HSSVD is necessary for the following two reasons: First, pseudo-mean biclusters are completely unable to capture small variance biclusters. Second, although pseudo-mean biclusters are able to capture some structure from large variance biclusters, such structure is much less accurate than that provided by HSSVD, and can be confounded with one or more true mean biclusters.

Although HSSVD works well in practice, there are a number of open questions that are important to address in future studies. For example, it would be worthwhile to modify the method to allow non-negative matrix approximations in order to better handle count data such as next-generation sequencing data (RNA-seq). Additionally, the ability to incorporate data from multiple "omic" platforms is becoming increasingly important in current biomedical research, and it would be useful to extend this work to simultaneous analysis of methylation, gene expression, and miRNA data.

CHAPTER3: COMPOSITE LARGE MARGIN CLASSIFIERS WITH LATENT SUBCLASSES

High dimensional classification problems are prevalent in a wide range of modern scientific applications. Despite a large number of candidate classification techniques available to use, practitioners often face a dilemma of the choice between linear and general nonlinear classifiers. Specifically, simple linear classifiers have good interpretability, but may have limitations in handling data with complex structures. In contrast, general nonlinear kernel classifiers are more flexible but may lose interpretability and have higher tendency for overfitting. In this chapter, we consider data with potential latent subgroups in the classes of interest. We propose a new group of methods, namely the Composite Large Margin Classifier (CLM) to address the issue of classification with latent subclasses. The CLM aims to find three linear functions simultaneously: one linear function to split the data into two parts, with each part being classified by a different linear classifier. Our method has comparable prediction accuracy to a general nonlinear kernel classifier without overfitting the training data, at the same time maintaining the interpretability of traditional linear classifiers. We demonstrate the competitive performance of the CLM through comparisons with several existing linear and nonlinear classifiers and through the analysis of Monte Carlo experiments. Finally, applications to Alzheimer's disease classification and cancer subtype prediction further demonstrate the usefulness of our proposed CLM.

3.1 Introduction

In biomedical research, it can be useful to discriminate the patients with high-risk of disease from the patients with low-risk of disease using biomarkers. Obtaining accurate prediction is very important, as the follow-up treatment plan can largely depend on such diagnosis. For example, Alzheimer's disease (AD) is one of the most common mental diseases which causes memory, thinking and behavior problems. Between normal aging and Alzheimer's disease, there exists a transitional stage, amnestic mid cognitive impairment (MCI). Although AD cannot be cured currently, proper early therapy can slow down the progress and alleviate symptoms. Thus, it is vitally important to accurately diagnose AD, especially for MCI. The classification task can be achieved by building classifiers and the biomarkers can come from various resources such as microarray or imaging data (fMRI). There are a large number of classifiers available in the literature, for example linear discriminative analysis (LDA) (Fisher 1936), Support Vector Machines (SVM) (Vapnik 1995), Distance Weighted Discrimination (DWD) (Marron et al. 2007), Random Forests (RF) (Breiman 2001), and the Large Margin Unified Machines (LUM) (Liu et al. 2011). Hastie et al. (2009) provide a comprehensive review of many machine learning techniques mentioned above. Among various classification tools, linear classifiers are popular especially for high dimensional problems, due to their simplicity and good interpretability. While widely used, linear classifiers can be improved upon for an important collection of problems (Huang et al. 2012). The problem we are interested in is classification in the presence of latent subgroups. In the AD example, patients with MCI at the first clinical visit potentially contain latent subgroups with different rates of disease progression. Another important biological application area is in cancer classification. It is known that cancer can be very heterogeneous and many types of cancer have distinct subtypes (Soslow 2008, Verhaak et al. 2010). Accurate classification of cancer subtypes can be very important since different cancer subtypes can have heterogenous response to treatment (Liu et al. 2010, Lehmann et al. 2011). In general, complex examples with data heterogeneity such as AD and cancer subtype classification can pose challenges for linear classifiers because of their insufficient flexibility for capturing potential data heterogeneity.

To further illustrate the problem of interest, we show a simple two dimensional toy example in Figure 3.1. This is a binary classification problem with X_1 and X_2 as predictors, and two classes are labeled in grey plus and black cross signs. As we can see from the plots, each class has two latent subclasses. A linear SVM model is fitted to the data and its decision boundary is shown in the solid line in Figure 3.1(a). Note that although linear methods for classification are not able to effectively capture the difference between classes in this example, the classification task becomes much easier if we divide the data by the line $X_2 = 0$. In practice, when in-class heterogeneity is ignored, traditional procedure may have poor classification performance. This motivates us to introduce the idea of a splitting function to divide the data into two parts so that we can handle the classification task with two separate linear classifiers.

As briefly mentioned earlier, despite good properties such as simplicity and interpretability of linear classifiers, they can be insufficient for problems with nonlinear decision boundaries. Using the kernel trick with a nonlinear kernel function (see Hastie et al. (2009) for details), one can extend a linear large margin classifier to a non-linear classifier to gain more flexible classification boundaries. However, the corresponding functional space can be much larger and consequently control of overfitting becomes more challenging. Although regularization is commonly used to control overfitting, finding the optimal tuning parameters can be difficult for nonlinear kernel classifiers, especially for high dimensional data. Furthermore, compared to simple linear classifiers, results from non-linear classifiers are more difficult to interpret in general. As shown in Figure 3.1, two non-linear classifiers, quadratic (panel b) and Gaussian (panel c) kernel

SVMs work reasonably well. However, their decision boundaries are quite complicated. Moreover, we will show in Section 3.4 that the performance of kernel SVMs can deteriorate rapidly when the dimension of covariates increases.

To solve the classification problem for complex structures, we propose a new group of methods, namely the Composite Large Margin Classifier (CLM). The CLM aims to find three linear functions simultaneously: one linear function to split the data into two parts, with each part being classified by a different linear classifier. We denote these three linear functions as $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$, respectively. Because of the split function, the CLM method provides a natural solution to the classification problem with latent subgroups. In Figure 3.1(d), we plot the decision boundary of the CLM with $\hat{f}_1(\mathbf{x}) = 0$ using a solid line, and both $\hat{f}_2(\mathbf{x}) = 0$ and $\hat{f}_3(\mathbf{x}) = 0$ using dashed lines. The function $\hat{f}_1(\mathbf{x})$ helps capture the hidden structure and divide the data into two parts, one part with $\hat{f}_1(\mathbf{x}) > 0$ and the other part with $\hat{f}_1(\mathbf{x}) < 0$. With this division, we can use two separate linear classifiers on each part. In particular, we predict the label $\hat{y} = \text{sign}(\hat{f}_2(\mathbf{x}))$ for the part on the top, and $\hat{y} = \text{sign}(\hat{f}_3(\mathbf{x}))$ for the part on the bottom. Thus, our CLM method makes use of three linear functions simultaneously to classify the data and capture the latent subclasses.

The CLM method has some advantages over both traditional linear and nonlinear methods. Compared to linear methods, the CLM is more flexible for classifying data with complex structure. On the other hand, unlike general nonlinear methods, the CLM only depends on the linear combination of the features and thus retains most of the simplicity and interpretability of linear methods. Furthermore, the functional space of interest for the CLM can be much smaller than is the case for general kernel methods. As a consequence of its relative simplicity, the CLM can perform better than the kernel based method in high dimensions, as we will show below in Section 3.4. In addition, the splitting function for the CLM has a natural latent variable interpretation, and it can

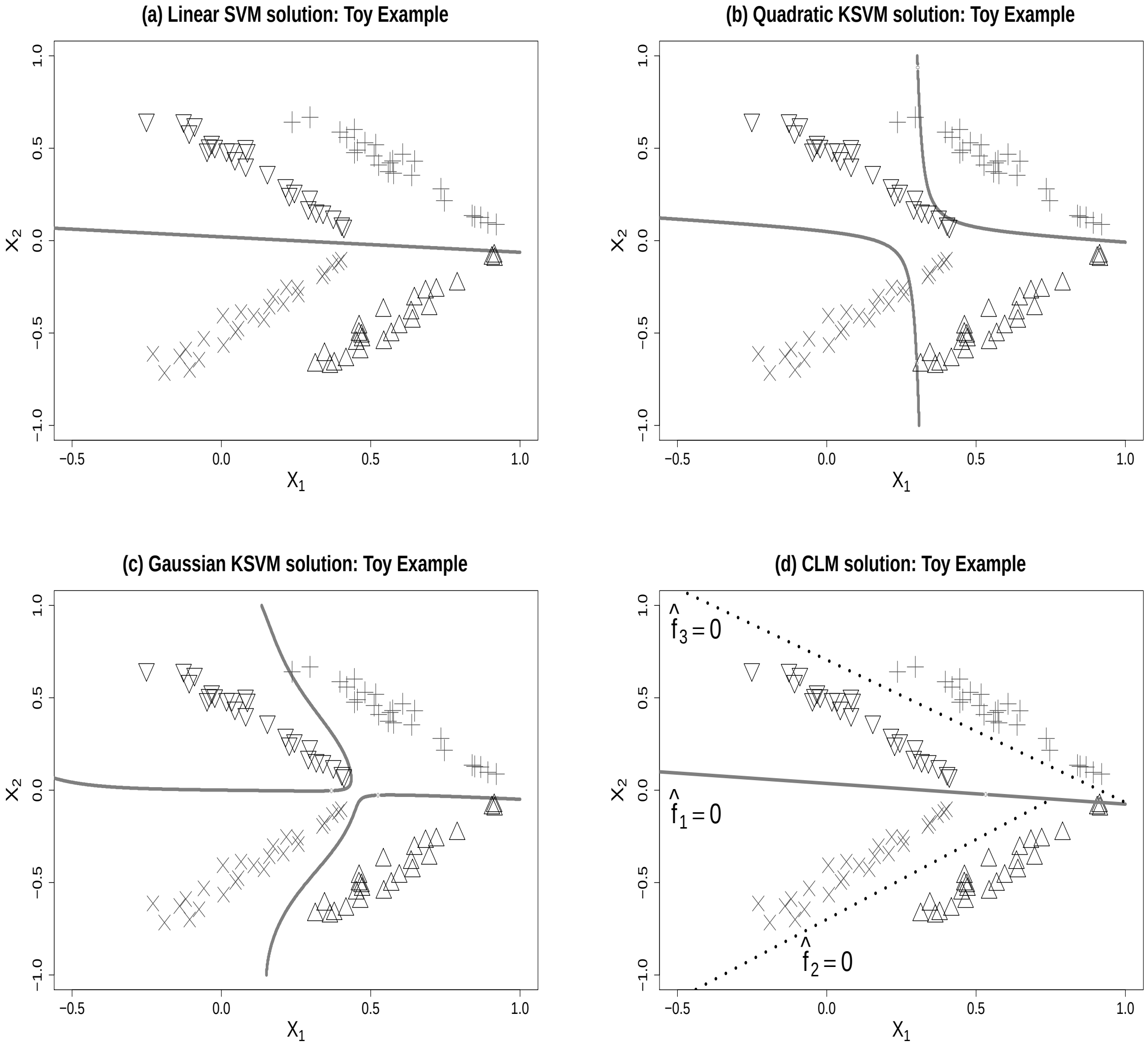

Figure 3.1: Illustration of a two dimensional toy example. Grey (+ and ×) represents the positive class and black (∇ and $\triangle$) represents the negative class. In panels (a), (b) and (c), the decision boundaries are drawn with wide grey lines. In panel (d) for the CLM method, the wide grey line splits the data into two parts and in each part the dashed line is the separating hyperplane for the corresponding classifier.

be viewed as a change-plane problem – an extension of the well studied change-point problem (Carlstein et al. 1994) and the recent change-line problem (Kang 2011). The latent variable identified by our methods in the AD and cancer data examples turn out to be scientifically meaningful and can be used for disease prognosis predication and treatment selection.

Although our CLM method is motivated by large margin classifiers, the fundamental concept is more general and can be applied to many other linear classifiers as well. Furthermore, besides classification, we can also generalize the CLM method to regression. In this article, we only focus on the implementation of the LUM loss and the logistic loss for classification and use them as examples to illustrate how the CLM method works.

The rest of chapter is organized as follows. In Section 3.2.1, we briefly review binary classification methods. The CLM framework is introduced in Section 3.2.2 and the properties of the CLM and its connection to existing methods are discussed in Section 3.2.3. In Section 3.3, we present a principal component analysis (PCA) based computational strategy for non-sparse solutions and a refitting procedure for sparse solutions. We demonstrate the effectiveness of our method with simulated data and the apply the method to the analysis of Alzheimer's disease and cancer data in Sections 3.4 and 3.6, respectively. Concluding comments are given in Section 3.7.

3.2 Methodology

We first review binary classification and large margin classifiers in Section 3.2.1. Due to the limitation of existing large margin classifiers for classification with latent subclasses, we propose the Composite Large Margin (CLM) classifier in Section 3.2.2. Furthermore, we discuss the properties of the CLM with two particular loss functions, the LUM and logistic losses, in Section 3.2.3.

3.2.1 Review of Binary Classification

Suppose we have a training data set $\{(\mathbf{x}_i, y_i); i = 1, 2, \dots n\}$ available. The class label $y \in \{\pm 1\}$, and the predictor $\mathbf{x}$ is a p-dimensional vector. Our goal is to build a classifier based on the training data for prediction of data points with $\mathbf{x}$ only. For a given binary classification problem, there are many techniques available in the literature and our focus in this chapter is on large margin classifiers (Hastie et al. 2009). Given the training data set, a large margin classifier is trained to obtain $f(\mathbf{x}) : \Re^k \to \Re$, such that the predicted class label is assigned using the sign of $f(\mathbf{x})$. Note that we correctly predict the class label of $\mathbf{x}$ when $yf(\mathbf{x})$ is positive. The term $yf(\mathbf{x})$ is known as the functional margin. In general, the objective function of a large margin classifier can be written in the regularization framework of a loss plus a penalty. The loss is a measure of the goodness of fit between the model and data, and the penalty controls the complexity of the model to avoid overfitting. Specifically, the optimization problem of a large margin classifier can be expressed as follows:

$$\min_{f \in \mathcal{F}} J(f) + \lambda \sum_{i=1}^{n} L(y_i f(\mathbf{x}_i)),$$

where $\mathcal{F}$ is the function class that all candidate solution functions belong to, $J(f)$ is a regularization term penalizing the complexity of f, $L(\cdot)$ is the loss function, and λ is a tuning parameter balancing the two terms. When the function $f(\mathbf{x})$ is linear with the form $w^T\mathbf{x}$, a common choice of $J(f)$ is the ℓ_2 penalty, i.e. $\|w\|_2^2$. A natural loss function is the so called $0-1$ loss with value 1 if $yf(\mathbf{x}) \leq 0$, and 0 otherwise, i.e. $L_{0-1}(yf(\mathbf{x})) = I\{yf(\mathbf{x}) \leq 0\}$. However, the $0-1$ loss is difficult for optimization due to its nonconvexity. Consequently, various convex surrogate loss functions have been proposed in the literature to alleviate the computational problem (Zhang 2004). For example, SVM uses hinge loss, penalized logistic regression uses logistic loss, and AdaBoost uses exponential loss (Friedman et al. 2000).

Recently, Liu et al. (2011) proposed a unified large margin machine (LUM) with a family of convex loss functions which contains DWD and SVM as special cases. The LUM loss function is differentiable everywhere, hence it has some computational advantage. As an important component of our proposed method, we will describe the

LUM loss function in detail. The LUM loss is indexed by two parameters a and c with the following explicit form:

$$V(u) = \begin{cases} 1-u & \text{if } u \leq \frac{c}{1+c}, \\ \frac{1}{1+c}(\frac{a}{(1+c)u-c+a})^a & \text{if } u > \frac{c}{1+c}. \end{cases} \tag{3.1}$$

The left piece of $V(u)$ with $u \leq \frac{c}{1+c}$ is the same as the hinge loss used in the SVM. The right piece is a convex curve whose shape is controlled by c with rate of decay controlled by a. With $a > 0$ and $c \to \infty$, LUM is equivalent to standard SVM. With $a \to \infty$ and fixed c, LUM loss is a hybrid of SVM and AdaBoost.

The techniques discussed above work well in many traditional classification problems. For large margin classifiers, it is common to use linear learning. One advantage of linear learning is its simple interpretation. Once the function $f(\mathbf{x}) = \mathbf{x}^T \mathbf{w}$ is obtained, one can examine the importance of each dimension in $\mathbf{x}$ through its corresponding coefficients $\mathbf{w}$. When a linear function is insufficient, one can map the original linear space to a higher dimensional nonlinear space using kernel methods (Hastie et al. 2009). Despite its flexibility, it is typically more difficult to interpret. Our goal is to propose a class of classifiers which maintain sufficient flexibility to incorporate latent subclasses without losing the interpretability of linear classifiers.

3.2.2 The Composite Large Margin (CLM) framework

In this section, we describe the CLM framework for binary classification with latent subclasses in detail. We assume there does not exist a global single linear classifier that can well separate the positive and the negative classes due to the existence of heterogenous subclasses. However, the data can be divided by a simple function (e.g. a linear function) into two parts, and each part can be classified relatively easily.

Next, we describe our proposed CLM method. To that end, we first define the generalized $0-1$ latent classification loss as $W_{0-1}(y,\mathbf{x}) = I(f_1(\mathbf{x}) \leq 0)I(yf_2(\mathbf{x}) \leq 0) + I(f_1(\mathbf{x}) > 0)I(yf_3(\mathbf{x}) \leq 0)$, where $f_1(\mathbf{x})$ is the splitting function, $f_2(\mathbf{x})$ is the classifier for data points with $f_1(\mathbf{x}) \leq 0$, and $f_3(\mathbf{x})$ is the classifier for data points with $f_1(\mathbf{x}) > 0$.

The generalized $0-1$ latent classification loss is the composition of two $0-1$ standard binary classification loss functions with weights $I(f_1(\mathbf{x}) \leq 0)$ and $I(f_1(\mathbf{x}) > 0)$. Similar to the standard $0-1$ loss, it is hard to optimize the generalized $0-1$ loss due to its discontinuity. In practice, a surrogate loss function is often used instead. For illustration, we use logistic and LUM losses as surrogate loss functions for the indicators $I(yf_2(\mathbf{x}) \leq 0)$ and $I(yf_3(\mathbf{x}) \leq 0)$. For weight functions $I(f_1(\mathbf{x}) \leq 0)$ and $I(f_1(\mathbf{x}) > 0)$, we use $G(-f_1(\mathbf{x}))$ and $G(f_1(\mathbf{x}))$ as their corresponding smooth approximations, where $G(u)$ is defined as:

$$G(u) = \begin{cases} 1 & \text{if } u \geq \epsilon, \\ 1 - 0.5(1-u/\epsilon)^2 & 0 \leq u \leq \epsilon \\ 0.5(1+u/\epsilon)^2 & -\epsilon \leq u \leq 0 \\ 0 & u \leq -\epsilon. \end{cases} \tag{3.2}$$

ϵ is tuning parameter, but we can set it to be small. Note that $G(u) + G(-u) = 1$, also as $\epsilon \to 0$, $G(u)$ converges to $I(u > 0)$ pointwisely. Note that this choice is not unique, and there are many other possible approximations such as sigmoid functions.

The loss functions W_{log} and W_{lum} for the latent classification problem are defined as follows:

$$W_{lum}(y_i,\mathbf{x}_i) = \alpha_i V\big(y_i f_2(\mathbf{x}_i)\big) + (1-\alpha_i) V\big(y_i f_3(\mathbf{x}_i)\big), \tag{3.3}$$

$$W_{log}(y_i,\mathbf{x}_i) = \alpha_i \log(1+e^{-y_i f_2(\mathbf{x}_i)}) + (1-\alpha_i)\log(1+e^{-y_i f_3(\mathbf{x}_i)}), \tag{3.4}$$

where $\alpha_i = G(-f_1(\mathbf{x}_i))$. Note that the W_{log} and W_{lum} losses are the compositions of two logistic and LUM loss functions, respectively. The α_i and $1 - \alpha_i$ are the weights. Furthermore, We assume that f_1, f_2, f_3 are all linear, i.e. $f_j(\mathbf{x}) = \mathbf{x}\,\mathbf{w}_j^T + b_j, j = 1, 2, 3$, to maintain the interpretability of linear classifiers.

With the loss function $L(y, \mathbf{x})$ defined, we can express the optimization problem for CLM as $\min_{\mathbf{w},\mathbf{b}} Q(\mathbf{w}, \mathbf{b}|\mathbf{Y}, \mathbf{X}) = \frac{1}{2}\sum_{j=1}^{3} ||\,\mathbf{w}_j\,||_2^2 + \lambda \sum_{i=1}^{n} L(y_i, \mathbf{x}_i)$, where λ is the tuning parameter, and $(\mathbf{w}, \mathbf{b}) = (\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, b_1, b_2, b_3)$. We will discuss the algorithm for obtaining the CLM solution in Section 3.3. Next, we briefly describe the properties of the CLM method.

3.2.3 Connections with existing literature

To handle the data heterogeneity and identify the potential subtypes, there are two main categories of methods in the literature. One is tree-based methods (Breiman et al. 1984), and the other is likelihood based mixture models (Fraley and Raftery 2002).

For tree-based methods, several techniques (Kim and Loh 2001, Loh 2010) were proposed to overcome the potential problems of splitting variables with only local effects. For example, GUIDE allows searching for linear splits using two variables at one time when the marginal effects of both covariates are weak, while the pairwise interaction effect is strong (Loh 2010). Both GUIDE and CLM can work well for certain problems as illustrated in Section 3.4, while traditional tree-based methods cannot. Unlike the tree-based GUIDE, CLM is a composite large margin classifier motivated by latent variables. Furthermore, with the use of three linear functions, the interpretation of CLM is relatively simple. Lastly, our numerical examples indicate that CLM is more competitive for high dimensional data.

The likelihood based mixture models assume that the data are coming from several mixture components (with the number of components known) and the model usually has a hierarchical structure. The hierarchical mixture of experts (HME) introduced by Jordan and Jacobs (1994) is one such example described as follows. Assume that there are two layers and four components. Then, the parametric likelihood of Y given X can be written as:

$$P(Y|X,\theta) = \sum_{i=1}^{2} g(i) \sum_{j=1}^{2} g(j|i) \mu_{ij}^{I(y=1)} (1-\mu_{ij})^{I(y=-1)},$$

where $g(i) = \exp(q_i(x,\theta))/(\exp(q_1(x,\theta)) + \exp(q_2(x,\theta)))$,

$$g(j|i) = \exp(q_{j|i}(x))/(\exp(q_{1|i}(x,\theta)) + \exp(q_{2|i}(x,\theta))), \mu_{ij} = E(I(Y=1)|i,j,X,\theta).$$

The $g(i)$ and $g(j|i)$ are the proportions for the four components, and μ_{ij} is the model for a given component. The task is to calculate the MLE for θ, and techniques such as the EM algorithm can be employed. If we consider a specific form of CLM without the penalty term, i.e. we use logistic loss function for $f_2(x)$ and $f_3(x)$, and approximate $I(f_1(x) > 0)$ with $\exp(f_1(x))/(\exp(f_1(x)) + \exp(f_1(-x)))$, then the one layer HME model has the same objective function as that of CLM. Despite the interesting connection, the motivations of CLM and HME are different. In particular, the CLM method is motivated from the perspective of latent subclasses and is a generalization of change-point models to the change-plane, while the HME is a likelihood based mixture model. When making a decision, CLM can be viewed as a “hard” classifier in the sense that it targets directly on estimating the decision boundary of the latent classification problem represented by $I(f_1(x) > 0)$. In contrast, HME is similar to a “soft” classifier which first estimates the conditional class probability and then converts the probability into the decision. More details about “hard” and “soft” classifiers can be found in Wahba (2002). In addition, the CLM is broader than likelihood-based methods and allows for more general loss functions. In particular, a general loss function for CLM may provide better classification performance for complex problems as shown in Section 3.4. We show that CLM with LUM delivers smaller classification errors than CLM with logistic loss in both of our application settings studied in Section 5.

3.3 Computational Algorithms for CLM

In this section, we discuss implementation of CLM. In particular, we describe a gradient based algorithm in Section 3.3.1. To tackle the difficulty of high dimensional problems, a PCA based algorithm is given in Section 3.3.2. Based on this algorithm, we further describe a refitting procedure to achieve variable selection in Section 3.3.3.

3.3.1 Gradient based algorithm for the CLM

To describe the algorithm, we use CLM with logistic loss as an example. The corresponding objective function can be written as:

$$Q_1^\lambda(\mathbf{w}, \mathbf{b}) = \frac{1}{2}\sum_{j=1}^{3} \parallel \mathbf{w}_j \parallel_2^2 + \lambda \sum_{i=1}^{n} [\alpha_i \log(1+e^{-y_i f_2(\mathbf{x}_i)}) + (1-\alpha_i)\log(1+e^{-y_i f_3(\mathbf{x}_i)})], \quad (3.5)$$

where λ, α_i and $f_j(\mathbf{x})$ are as defined in (3.4). Since the objective function is continuously differentiable, many general optimization algorithms such as the conjugate gradient method or the quasi-Newton method (Nocedal and Wright 1999) are applicable.

To apply these algorithms, we first need to derive the corresponding gradient functions. Once the gradient is given, we can iteratively update the solution. For example, for the gradient descent algorithm, the $(t+1)$-th step solution $(\mathbf{w}^{t+1}, \mathbf{b}^{t+1})$ based on the t-th step solution $(\mathbf{w}^{t+1}, \mathbf{b}^{t+1})$ is given by $\mathbf{w}_i^{t+1} = \mathbf{w}_i^t - \gamma \frac{\partial Q_1^\lambda(\mathbf{w}^t, \mathbf{b}^t)}{\partial \mathbf{w}_i}$, $b_i^{t+1} = b_i^t - \gamma \frac{\partial Q_1^\lambda(\mathbf{w}^t, \mathbf{b}^t)}{\partial b_i}$, $i = 1 \ldots 3$, where γ is a small positive number known as the learning rate. Similar calculations can be done for other loss functions such as the W_{lum} loss in (3.3).

3.3.2 PCA algorithm for the CLM

The direct optimization strategy in Section 3.3.1 works well for low or moderate-size dimensional problems. However, direct optimization encounters significant challenges for high dimensional data since the computational burden of most general optimization methods increases dramatically with dimension. To alleviate the severity of this problem, we incorporate the principal component idea, i.e., we predict the class label using the CLM method by a reduced rank design matrix instead of the original design matrix. The reduced rank matrix comprises the first k principal component scores of the original design matrix with $k < d$. The steps of this PCA-based algorithm for CLM are given in Algorithm 1 below.

We now describe the algorithm in detail. Denote the design matrix as $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T = [\mathbf{X}_1, \ldots, \mathbf{X}_d]$, which is an $n \times d$ matrix. The eigen-decomposition of $\mathbf{X}$ can be written as $\mathbf{X}^T\mathbf{X} = \mathbf{P}\,\Lambda\,\mathbf{P}^T$, where the $\mathbf{P}$ is a matrix with orthonormal columns ($[\mathbf{P}_1, ..., \mathbf{P}_d]$) and Λ is the diagonal matrix with the eigenvalues as the diagonal elements. Furthermore, we define $\mathbf{P}^k = [\mathbf{P}_1, ..., \mathbf{P}_k]$, and $\mathbf{X}^k = \mathbf{X}\,\mathbf{P}^k = [\mathbf{x}_1^k, \ldots, \mathbf{x}_n^k]^T$. Our idea is to work with the k-dimensional space spanned by the first k principal component dimensions. In particular, instead of working with d-dimensional $\mathbf{x}$, we work with k-dimensional $\mathbf{x}^k$. If we replace the corresponding elements in Q_1^λ in (3.5) with the new linear functions $\tilde{f}_j(\mathbf{x}_i) = \mathbf{x}_i^k\,\tilde{\mathbf{w}}_j^T + \tilde{b}_j$ $(j = 1, 2, 3)$, then we can obtain a new objective function $\tilde{Q}_1^{k,\lambda}$:

$$\tilde{Q}_1^{k,\lambda}(\tilde{\mathbf{w}}, \tilde{\mathbf{b}}) \;=\; \frac{1}{2}\sum_{j=1}^{3} \parallel \tilde{w}_j \parallel_2^2 + \lambda \sum_{i=1}^{n} W_{log}(y_i, \mathbf{x}_i^k). \tag{3.6}$$

We minimize $\tilde{Q}_1^{k,\lambda}$ instead of Q_1^λ and get the minimizer $(\tilde{\mathbf{w}}^*, \tilde{\mathbf{b}}^*)$. Consequently, we can calculate the solution for the original problem ($\mathbf{w}^*$,$\mathbf{b}^*$) by setting $\mathbf{w}^* = \tilde{\mathbf{w}}^*[\mathbf{P}^k]^T$, $\mathbf{b}^* = \tilde{\mathbf{b}}^*$. This strategy reduces the dimension of the problem from $3d$ to $3k$. If $k \ll d$, then we can handle high dimensional data relatively efficiently.

We would like to point out that although reducing the dimension greatly helps the computational efficiency, we may lose important classification information. Thus the choice of k is very important. We assume that the important classification information is mostly contained in the space spanned by the first k PC dimensions. Under this assumption, finding the right k helps to eliminate noise dimensions and improve computational efficiency as well as accuracy of the resulting classifier. Therefore, we need to measure the information of Y contained in $\mathbf{X}^k$ for various k. The traditional Pearson correlation is not appropriate for this purpose, since it restricts the two random vectors to be one dimensional and only measures the linear dependence. To address this problem, we make use of a recently proposed "distance correlation" (dcor) (Székely et al. 2008) for choosing the number of leading principal components k. The dcor measures arbitrary types of dependence between two random vectors. In particular, the distance covariance (*dcov*) between two random vectors $\mathbf{u}$ and $\mathbf{v}$ with finite first moments is written as $dcov(\mathbf{u},\mathbf{v}) = \sqrt{\int_{R^{d_u+d_v}} \|\phi_{\mathbf{u},\mathbf{v}}(\mathrm{t},\mathrm{s}) - \phi_{\mathbf{u}}(\mathrm{t})\phi_{\mathbf{v}}(\mathrm{s})\|^2 \, w(\mathrm{t},\mathrm{s}) \, dt \, d\mathrm{s}}$, where $\phi(.)$ represents the characteristic function, d_u and d_v are the dimensions of $\mathbf{u}$ and $\mathbf{v}$, and $w(\mathrm{t},\mathrm{s})$ is a properly defined weight function. The distance correlation between $\mathbf{u}$ and $\mathbf{v}$ is defined as $dcor(\mathbf{u},\mathbf{v}) = dcov(\mathbf{u},\mathbf{v})/\sqrt{dcov(\mathbf{u},\mathbf{u})dcov(\mathbf{v},\mathbf{v})}$. Unlike Pearson correlation, dcor is 0 if and only if the two random vectors are independent and it does not restrict the dimensions of the two random vectors (Székely et al. 2008). From its properties, the dcor is feasible and robust for screening information for classification problems (Li et al. 2012). We measure the information of Y contained in the leading k-PCs of $\mathbf{X}$ as $dcor(Y, \mathbf{X}^k)$. Consequently, the optimal $k_{opt} = \underset{k=1,2,\ldots,d}{\arg\max} dcor(Y, \mathbf{X}^k)$. In practice, we find that using $\mathbf{X}^{k_{opt}}$ as the predictor may not always yield the lowest classification error, so in the implementation, we treat k as a tuning parameter from the set $\{k_{opt}, k_{opt}+1, k_{opt}+2\}$. This strategy appears to work well in practice.

PCA algorithm for the CLM method

(0) Initialization: Denote the training data set as $\{\mathbf{Y}_1, \mathbf{X}\}$ and the tuning data set $\{\mathbf{Y}_2, \mathbf{U}\}$. Let $k_{opt} = \arg\max_{k=1,2,\ldots,d} dcor(\mathbf{Y}_1, \mathbf{X}^k)$.

(1) Model Training: For fixed k and λ, solve $(\tilde{\mathbf{w}}^{k,\lambda}, \tilde{\mathbf{b}}^{k,\lambda}) = \arg\min_{\tilde{\mathbf{w}},\tilde{\mathbf{b}}} Q_1^{k,\lambda}(\tilde{\mathbf{w}}, \tilde{\mathbf{b}} |\, \mathbf{Y}_1, \mathbf{X}^k)$.

(2) Solution Reconstruction: Calculate $\mathbf{w}^{k,\lambda} = \tilde{\mathbf{w}}^{k,\lambda}[\mathbf{P}^k]^T, \mathbf{b}^{k,\lambda} = \tilde{\mathbf{b}}^{k,\lambda}$.

(3) Prediction: The tuning error $\xi(k, \lambda)$ is the classification error on predicting $\mathbf{Y}_2$ using $\mathbf{U}$ by the model from Step 2.

(4) Iteration: Repeat Steps $1-3$ for all $k \in \{k_{opt}, k_{opt}+1, k_{opt}+2\}$ and $\lambda \in \{\lambda_1, \ldots, \lambda_n\}$.

(5) Output: The final solution $(\mathbf{w}^*, \mathbf{b}^*)$ is $(\mathbf{w}^{k,\lambda}, \mathbf{b}^{k,\lambda})$ from the model with the lowest $\xi(k, \lambda)$. For the choice of λ, we use a warm-start strategy such that for the sequence of model fittings from λ_1 to λ_n $(\lambda_1 > \lambda_2 > \ldots > \lambda_n)$, the solution based on λ_i is provided as the starting point for λ_{i+1}. This approach helps to improve the convergence of the algorithm.

3.3.3 Refitting algorithm for sparse CLM

Variable selection can be important for the analysis of high dimensional data. The PCA algorithm we proposed above does not have variable selection capability due to the choice of the ℓ_2 penalty and the solution reconstruction step. As $\mathbf{P}^k$ is full rank, $\mathbf{w} = \tilde{\mathbf{w}}[\mathbf{P}^k]^T$ is not sparse in general, even if $\tilde{\mathbf{w}}$ is. Consequently, we can not achieve sparsity on $\mathbf{w}$ by simply replacing the ℓ_2 penalty with the lasso penalty in $\tilde{Q}_1^{k,\lambda}$ in (3.6).

To achieve variable selection, we propose a refitting procedure. We first identify the informative variables from the output of the PCA algorithm with all variables as predictors. Then, we refit the PCA algorithm using the informative variables only. The key step is to identify the informative variables. For this purpose, we fit three penalized logistic regression (PLR) models with the elastic net penalty (Zou and Hastie 2005, Park and Hastie 2007, Friedman et al. 2010) separately for f_1, f_2 and f_3. The details are given in Algorithm 2:

Refitting algorithm for the CLM method

(0) Initialization: Denote the training data set as $\{\mathbf{Y}, \mathbf{X}\}$ and the solution from Algorithm 1 as $\hat{f}_1$,$\hat{f}_2$, $\hat{f}_3$. Let PLR$(\mathbf{Y}, \mathbf{X})$ be the solution of the PLR fitting, and let the initial value of the active variable index set be $A = \{1, 2, \ldots, p\}$.

(1) Approximate f_1: Let $Z_i = \text{sign}(\hat{f}_1(\mathbf{x}_i^A))$. Then obtain $(\mathbf{w}_1^s, b_1^s) = \text{PLR}(\mathbf{Z}, \mathbf{X}^A)$, where "s" represents the sparse solution.

(2) Approximate f_2 and f_3: The samples in $\{\mathbf{Y}, \mathbf{X}^A\}$ are divided into Set 1 and Set 2 by the sign of $f_1^s(\mathbf{X})$. Using Set 1 data, we have $(\mathbf{w}_2^s, b_2^s) = \text{PLR}(\mathbf{Y}_{\text{Set1}}, \mathbf{X}_{\text{Set1}}^A)$, and similarly $(\mathbf{w}_3^s, b_3^s) = \text{PLR}(\mathbf{Y}_{\text{Set2}}, \mathbf{X}_{\text{Set2}}^A)$.

(3) Select active variables: Let $A = I_1 \cup I_2 \cup I_3$, where $I_1 = \{j : w_{1j}^s \neq 0\}$, $I_2 = \{j : w_{2j}^s \neq 0\}$, $I_3 = \{j : w_{3j}^s \neq 0\}$.

(4) Refit: Refit Algorithm 1 using X^A as predictors.

(5) Iteration: Repeat Steps $1 - 4$ until the active variable index set A stabilizes.

(6) Output: The final solution is obtained from the most recent Step 4. Based on our experience, the algorithm converges within several steps of refitting. In our simulation studies and application settings, we observe that the refitting procedure performs well overall. Depending on the computational budget, one may also couple our algorithm with the stability selection procedure proposed by Meinshausen and Bühlmann (2010) to select the active variables.

3.4 Simulation Studies

We now investigate the performance of the proposed methods on two synthetic examples. We simulate both low- and high-dimensional situations. The training and testing data are generated from the same distributions with sample sizes 200 and 20000, respectively. For each example, there are several scenarios with different Bayes errors (the error rate of the optimal Bayes rule) and different numbers of noise variables. In both examples, the data contain four clusters of equal sizes. To reflect the latent subclass structure, two clusters are "+" class and the others are "−" class. We compare our CLM methods with linear SVM, quadratic and Gaussian kernel SVM, Random Forest, HME and ℓ_1-penalized logistic regression. In addition, we also include an enhanced tree-based method GUIDE (Loh 2010) for comparison. We select tuning parameters for all methods via five-fold cross validation.

Example 1 (Twisted Case): The four clusters are sampled from four bi-variate normal distributions with corresponding means (μ, μ) and $(-\mu, -\mu)$ for the "+" class, $(\mu, -\mu)$ and $(-\mu, \mu)$ for the "−" class, and the identity covariance matrix. In addition to the two informative variables, we generate random noise variables from $N(0, 0.5^2)$. We present the scenarios with $\mu = 2.24$ or 1.2, where larger μ makes the classification task easier with smaller Bayes error. We also compare the performance of different methods on the scenarios with the same μ but different numbers of noise variables. Note that although the Bayes error only depends on μ in our example, the classification problem becomes more challenging when more noise variables are added. The scatter plot for the no noise variables scenario ($\mu = 2.24$) with the CLM solution boundary is shown in Figure 3.2(a).

Example 2 (Parallel Case): The four clusters are sampled from four bi-variate normal distributions whose means are $(\mu, 0)$, $(-\mu, 0)$ for the "+" class, and $(0, 0)$, $(2\mu, 0)$ for the "−" class. The covariance matrix is $\mathbf{\Sigma} = 0.6\mathbf{I}_{2\times2} + 0.4\mathbf{J}_{2\times2}$, where $\mathbf{I}$ is the identity matrix, $\mathbf{J}$ is a matrix with all elements equal to 1. We set $\mu = 3.90$ or 2 and the additional random noise variables follow i.i.d $N(0, 0.5^2)$. As in Example 1, a similar scatter plot ($\mu = 3.90$, no noise variables) is shown in Figure 3.2(b), where the four clusters are parallel to each other. The testing errors of the CLM and other methods in both examples are reported in Table 3.1.

Table 3.1: Average testing errors in the simulation data with standard deviations in parentheses

Twisted Case						
Bayes Error (ε)	$\mu = 2.24$, $\varepsilon = 0.025$			$\mu = 1.2$, $\varepsilon = 0.204$		
Dimension	$p = 2$	$p = 100$	$p = 1000$	$p = 2$	$p = 100$	$p = 1000$
LSVM	.363 (.073)	.499 (.007)	.499 (.006)	.444 (.034)	.499 (.007)	.499 (.007)
KSVM(quadratic)	.035 (.006)	.040 (.008)	.072 (.020)	.221 (.010)	.270 (.012)	.390 (.024)
KSVM(Gaussian)	.032 (.004)	.120 (.011)	.454 (.004)	.232 (.010)	.409 (.012)	.496 (.002)
Random Forest	.033 (.004)	.491 (.004)	.500 (.001)	.231 (.012)	.495 (.003)	.500 (.001)
PLR	.500 (.001)	.500 (.001)	.500 (.002)	.500 (.002)	.500 (.001)	.500 (.002)
GUIDE	.031 (.003)	.039 (.020)	.496 (.004)	.220 (.008)	.308 (.021)	.498 (.002)
HME	.034 (.004)	.042 (.018)	.091 (.024)	.220 (.019)	.254 (.024)	.332 (.036)
CLM_{log}	.028 (.002)	.028 (.003)	.033 (.003)	.218 (.009)	.219 (.008)	.251 (.011)
CLM_{lum}	.028 (.002)	.029 (.003)	.033 (.003)	.215 (.006)	.220 (.008)	.249 (.009)
CLM_{log} Sparse	.028 (.002)	.029 (.003)	.030 (.004)	.218 (.009)	.220 (.010)	.221 (.011)
CLM_{lum} Sparse	.028 (.002)	.028 (.003)	.028 (.003)	.215 (.006)	.219 (.009)	.220 (.011)

Parallel Case						
Bayes Error(ε)	$\mu = 3.9$, $\varepsilon = 0.024$			$\mu = 2$, $\varepsilon = 0.206$		
Dimension	$p = 2$	$p = 100$	$p = 1000$	$p = 2$	$p = 100$	$p = 1000$
LSVM	.487 (.006)	.428 (.013)	.382 (.004)	.423 (.008)	.437 (.013)	.397 (.007)
KSVM(quadratic)	.377 (.048)	.435 (.013)	.342 (.016)	.378 (.018)	.421 (.008)	.395 (.013)
KSVM(Gaussian)	.035 (.004)	.258 (.010)	.385 (.003)	.286 (.029)	.401 (.008)	.380 (.003)
Random Forest	.040 (.006)	.289 (.065)	.333 (.082)	.259 (.041)	.372 (.034)	.415 (.051)
PLR	.332 (.012)	.352 (.025)	.405 (.040)	.378 (.027)	.401 (.032)	.405 (.032)
GUIDE	.042 (.003)	.047 (.006)	.052 (.006)	.238 (.009)	.281 (.027)	.304 (.030)
HME	.055 (.004)	.063 (.007)	.072 (.012)	.236 (.012)	.284 (.032)	.310 (.033)
CLM_{log}	.032 (.003)	.032 (.003)	.041 (.004)	.233 (.025)	.234 (.014)	.260 (.013)
CLM_{lum}	.031 (.003)	.032 (.004)	.041 (.004)	.225 (.014)	.227 (.011)	.257 (.010)
CLM_{log} Sparse	.032 (.003)	.034 (.005)	.037 (.008)	.233 (.025)	.241 (.019)	.260 (.024)
CLM_{lum} Sparse	.031 (.003)	.034 (.003)	.039 (.003)	.225 (.014)	.242 (.009)	.263 (.012)

The results of Examples 1 and 2 show that our methods outperform the competitors in most scenarios. Both examples have latent subclasses and our method is well suited for these problems. The results show that our method is very effective in detecting the true latent structure.

In the twisted case, linear SVM and PLR methods fail to detect the pattern. Although the quadratic kernel method works, it still has larger testing errors than the CLM methods. In Example 1, we can divide the samples into four clusters by lines $X_1 = 0$ and $X_2 = 0$. Since these lines can be approximated by a quadratic function, the quadratic kernel can work well in the twisted case. For the parallel case, we need three parallel lines to separate the four clusters, so methods such as quadratic kernel SVM, linear SVM, and PLR do not perform well. For both examples, the Gaussian kernel SVM and Random Forest work well under the low dimensional setting. When the dimension is high, Random Forest fails to choose the right covariate to split among all covariates. In addition, the Random Forest method will only split variables with strong marginal effect, hence the performance of Random Forest is better in the parallel case than in the twisted case. The model structure of Gaussian kernel SVM is flexible enough to separate the two classes for low dimensions. However, its performance decays rapidly in the high dimensional setting, possibly due to overfitting. In contrast, the results of the CLM method under the high dimensional setting are still comparable to those under the low dimensional setting. The CLM method uses linear classifiers, and hence it may alleviate the overfitting problem compared to kernel based methods.

When there are no latent subclasses or only one class has subclasses, we show our method is still comparable to the competitors (see Section 3.5 for details). Note that for the parallel example, X_2 carries most information about Y. As a result, GUIDE performs well for the parallel case in general due to its ability to generate unbiased splits regardless of whether it searches for pairwise interactions or not. In the twisted example, the marginal effect of X_1 and X_2 is weak, while the interaction effect is strong. However, when the dimension is high, due to the high computational demand, the implementation of GUIDE does not perform interaction tests for determining which covariates to split. Hence, GUIDE does not perform very well in the twisted case with

$p = 1000$. In both example, HME performs worse than CLM when the dimension is high or the classification is challenging, as the CLM can better identify the latent variable (see Section 3.5). Overall, the performance of CLM is the best among all methods.

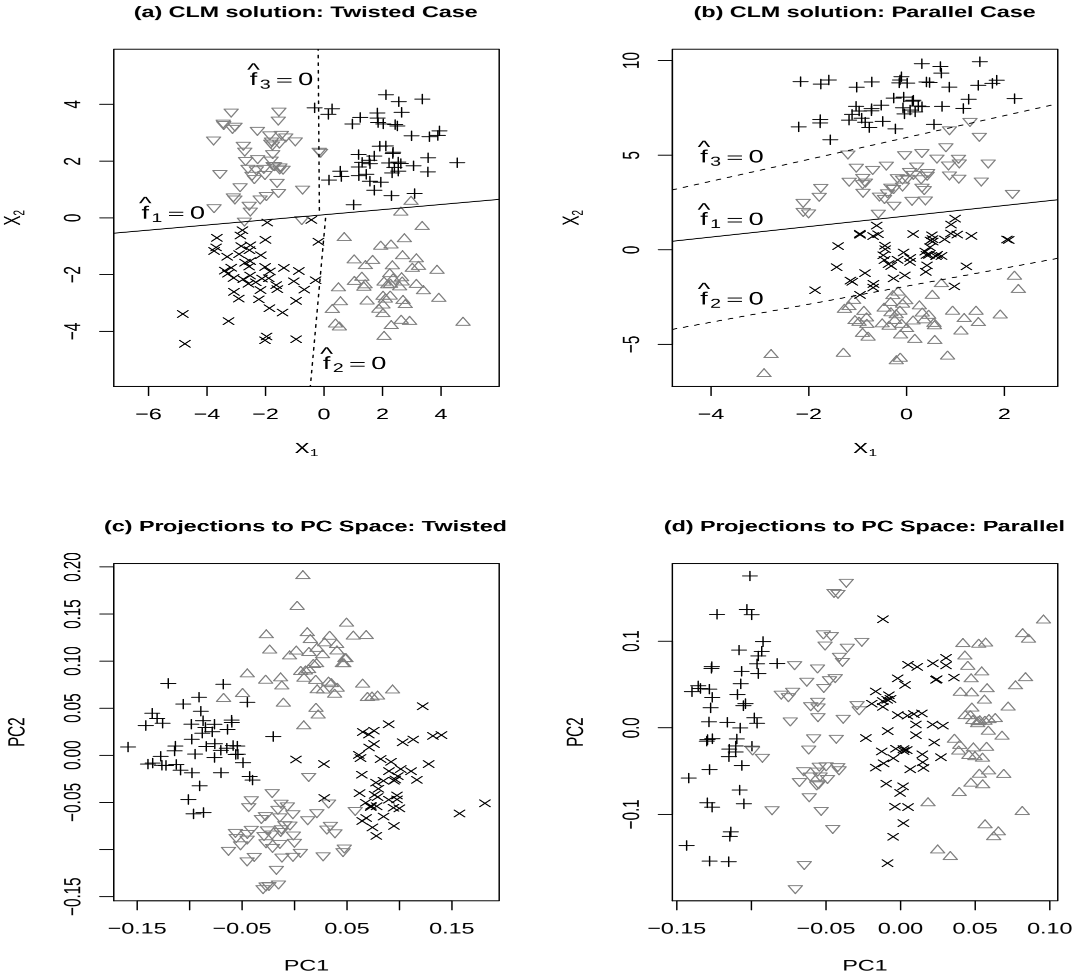

Figure 3.2: Plots for CLM methods in twisted and parallel cases. Black color (+ and ×) represents the positive class and grey color (∇ and $\triangle$ represents the negative class. Different symbols in the same class indicates the latent subclasses. In both panels (a) and (b), the boundary of $\hat{f}_1$ is shown as a solid line and the boundaries of $\hat{f}_2$ and $\hat{f}_3$ are given as dashed lines. In panels (c) and (d), we show projections onto the space spanned by the first 2 principle components for the twisted and parallel cases. We apply PCA on the data which contains the informative variables X_1 and X_2 as well as an additional 998 noise variables. The four-clusters structure in the original space (as observed in panels (a) and (b)) is preserved in the PC space.

Note that we do not lose much information by applying the PCA strategy. This can be seen by comparing the data projections in the space spanned by X_1 and X_2 in Figures 3.2(a) and 3.2(b) with those in the space spanned by the first two principal components in Figures 3.2(c) and 3.2(d). The structure of the four clusters is preserved after applying the PCA strategy.

3.5 Additional Examples

Example 3: The two equal size clusters are sampled from two bi-variate normal distributions with corresponding means $(1, 1)$ for the "+" class, $(-1, 1)$ for the "−" class, and the identity covariance matrix. In addition to the two informative variables, we generate random noise variables from $N(0, 0.5^2)$. We report the results with $p = 1000$. The training and testing sample sizes are 200 and 20000 respectively. In principle, one linear function is sufficient for classifying the data. In this case, there does not exist latent subclasses, and we show that the CLM methods still gives competitive performance in Table 3.2.

Example 4: Three equal sized clusters are sampled from three bivariate normal distributions with means $(1, 1)$ and $(-1, -1)$ for the "+" class, and $(1, -1)$ for the "−" class. The training and testing sample sizes are 150 and 20000 respectively. Other settings for this example are the same as that of Example 1. The numerical results of different methods are presented in the right side of Table 3.2. In this example, only the "+" class has latent subclasses. Similar to the results in Section 4, we can see that sparse CLM with LUM loss and logistic loss deliver the lowest classification errors.

In addition, we compare the performance of CLM and HME in terms of identifying the latent structure. As both methods identify a linear function of X as surrogate for

Table 3.2: Average testing error rates in additional examples with standard errors in parentheses

Methods	Example 3: $p = 1000$		Example 4: $p = 1000$	
LSVM	.245	(.011)	.302	(.011)
KSVM(quadratic)	.341	(.008)	.333	(.002)
KSVM(Gaussian)	.220	(.008)	.293	(.009)
Random Forest	.299	(.114)	.333	(.001)
PLR	.161	(.003)	.230	(.008)
GUIDE	.166	(.005)	.284	(.005)
HME	.197	(.008)	.264	(.025)
W_{log}	.191	(.007)	.250	(.018)
W_{lum}	.191	(.008)	.246	(.017)
W_{log} Sparse	.166	(.008)	.206	(.015)
W_{lum} Sparse	.166	(.006)	.207	(.015)

the latent class variable that can be used to separate the data into two parts, we can calculate the angle between the oracle linear function and the estimated function by CLM and HME. Consider the twisted case with $p = 1000$ and $\mu = 1.2$ as an example (results given in Table 3.3). In this setting, the oracle linear function is either $X_1 = 0$ or $X_2 = 0$. In each simulation, if the angle between $X_1 = 0$ and the estimated line is small, then we use $X_1 = 0$ as the truth, otherwise we use $X_2 = 0$. We also calculate the classification error for the latent class variable. We can see that CLM better identifies the latent structure.

Table 3.3: Average testing errors in angle and classification with standard deviations in parentheses

Method	sin(angle)	Error
HME	.201 (.08)	.197 (.05)
CLM$_{log}$	.134 (.04)	.121 (.02)
CLM$_{lum}$	.122 (.04)	.112 (.02)
CLM$_{log}$ Sparse	.098 (.03)	.093 (.01)
CLM$_{lum}$ Sparse	.093 (.03)	.082 (.01)

3.6 Applications

In this section, we apply our methods to the analysis of Alzheimer's disease and ovarian carcinoma data. The first data set involves imaging data while the second involves gene expression data. We are interesting in the ability of our method to detect meaningful latent heterogenous groups while simultaneously delivering competitive classification accuracy. The performance of different methods is evaluated by cross validation (CV) errors of 100 random divisions of the data. We use 75% of data for training and 25% for testing. We also keep the original class proportions within both the training and testing sets. In the training steps, tuning parameters are selected by 5-fold CV. We compare the same methods evaluated in Section 3.4 and report the average testing errors with the corresponding standard deviations in Table 3.4.

3.6.1 Alzheimer's disease

Our first application is to a longitudinal study designed for Alzheimer's disease detection and tracking, the Alzheimer's Disease Neuroimaging Initiative (ADNI) (Mueller et al. 2005). See www.adni-info.org for additional details. There are 226 normal controls and 393 MCI patients in total. The primary goal of our analysis is to discriminate between MCI samples and normal control samples using imaging data collected at their first visit. As mentioned previously, such a discrimination tool could help physicians know when to begin intervention. Each sample is characterized by features extracted from structural MR imaging (MRI) which measure brain atrophy (a known AD related factor). The image pre-processing and feature extraction follow the procedure described in Wang et al. (2011) and Zhang et al. (2011). Basically, each processed image is divided into 93 regions-of-interest (ROI) and then the volume of grey matter tissue in each ROI region is computed as a feature (Zhang et al. 2011). The results of classifying MCI versus normal controls using the 93 MRI features are shown in the left panel of Table 3.4. We can see that sparse CLM with the LUM loss provides the smallest average testing error among all methods.

Table 3.4: Testing errors on classifying Alzheimer's disease and ovarian cancer data with standard deviations in parentheses

	Alzheimer Disease		Ovarian Cancer	
LSVM	.322	(.028)	.075	(.018)
KSVM (quadratic)	.384	(.032)	.148	(.021)
KSVM (Gaussian)	.333	(.033)	.065	(.017)
Random Forest	.323	(.027)	.080	(.022)
PLR	.323	(.030)	.078	(.019)
GUIDE	.323	(.029)	.083	(.021)
HME	.331	(.031)	.066	(.022)
CLM$_{log}$	.330	(.030)	.054	(.020)
CLM$_{lum}$	.314	(.030)	.042	(.014)
CLM$_{log}$ Sparse	.315	(.025)	.071	(.021)
CLM$_{lum}$ Sparse	.297	(.024)	.058	(.017)

Beside accurate diagnosis of AD/MCI, another important quest in AD research is predicting whether MCI patients will convert to AD. Clinically, AD and MCI are defined according to certain severity measures of symptoms of dementia at the baseline based on the Clinical Dementia Rating (CDR) scale (Misra et al. 2009). In particular, CDR $= 0$ is considered normal, CDR $= 0.5$ is considered MCI, and CDR $= 1, 2, 3$ is considered AD with different levels of severity. If an MCI patient's condition is getting worse during the follow-up (i.e., they receive a CDR higher than 1), then he/she is considered a converter (to AD), otherwise they are considered nonconverters. For the data set we are analyzing, there are 167 converters and 226 nonconverters. We found that if we predict MCI patients with $\hat{f}_1(\mathbf{x}) < 0$ as converters, and the other MCI patients as nonconverters, then the average prediction accuracy among 100 replications is 0.799 (sd=0.14). This prediction result is greater than the accuracy of directly predicting MCI subclasses using the same baseline MRI data. In particular, the linear classifier constructed from l_1-penalized logistic regression has a ten fold cross validation error value of 0.69. This result indicates that heterogeneity among MCI patients can be characterized by MRI data, and this result provides new insights for prognosis of MCI. Therefore, besides successfully classifying MCI versus normal control patients, the proposed CLM also provides a good prognostic tool for MCI patients in terms of separating converters from nonconverters. Our method discovers meaningful latent subclasses of MCI without using the follow-up clinical information.

3.6.2 Ovarian Carcinoma

The second application is to an ovarian carcinoma data set containing $11,864$ genes from three different platforms (TCGA, 2011). There are four cancer subtypes: immunoreactive (sample size is 107), differentiated (135), proliferative (138) and mesenchymal (109). We focus on classifying the proliferative samples ("+" class) versus non-proliferative samples ("$-$" class). We select 3520 genes with the largest median absolute deviance (MAD) value for further classification analysis. From the testing errors shown in the right part of Table 3.4, we can see that CLM with LUM loss performs the best, followed by CLM with logistic loss. Both methods perform better than the other competitors.

To better understand the results, we can visualize the results of the CLM method by projecting the samples onto the space spanned by $\hat{f}_1, \hat{f}_2$ if $\hat{f}_1(\mathbf{x}) < 0$, and onto the space spanned by $\hat{f}_1, \hat{f}_3$ otherwise. Figure 3.3 suggests that there may exist subgroups within the proliferative samples (Pro(A) and Pro(B)). Additionally, the "$-$" class samples are grouped into two parts: one consists of some immunoreactive samples (Imm(B)) and all differentiated samples, the other consists of the remaining immunoreactive samples (Imm(A)) and almost all mesenchymal samples.

To confirm statistical stability of the subgroups found in proliferative and immunoreactive samples, the Sigclust method proposed by Liu et al. (2008) is applied for testing

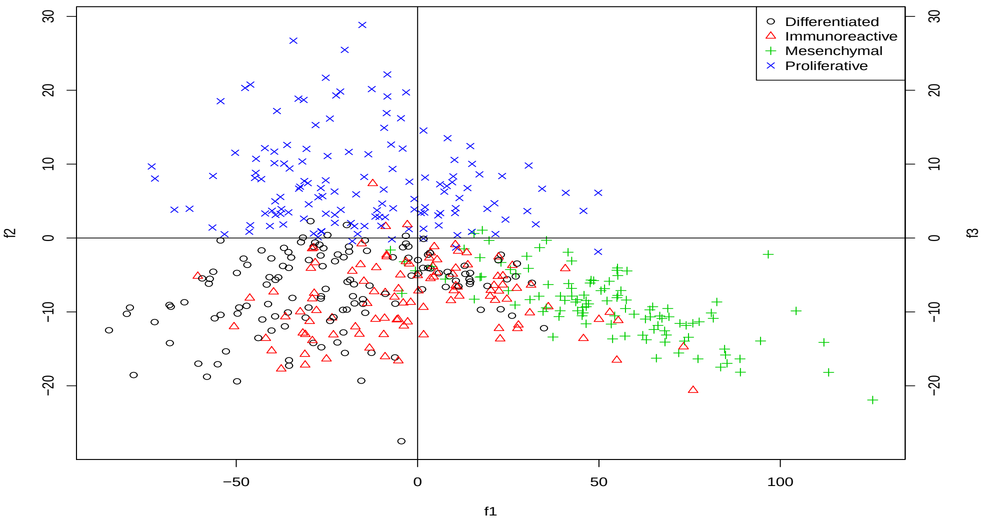

Figure 3.3: Visualization of latent subclasses in the ovarian cancer dataset. The x-axis is the $\hat{f}_1$ value, the y-axis displays the $\hat{f}_2$ value of the points for which $\hat{f}_1$ is less than 0, otherwise it displays the $\hat{f}_3$ value. The plot indicates that there exist subclasses within both the proliferative and immunoreactive types of ovarian cancer.

whether the difference between the two subgroups is significant. The subgroups are determined by the average of the sign of $\hat{f}_1$ in the 100 simulations given by the CLM method with the LUM loss. The p-value for the subclasses within the immunoreactive subtype is very significant, i.e., < 0.001, while the p-value for the subclasses within the proliferative samples is not.

The detected subgroups can also be visualized from the heatmap of gene expression of a subset of genes (size $= 153$) in Figure 3.4. These genes were selected more than 15 times as “active variables” by sparse CLM with LUM loss among the 100 random splits. The genes are displayed in rows, and samples are shown in columns where the red line separates samples by the sign of $\hat{f}_1$ (positive on the left). The plot shows that there exists a clear distinction between Imm(A) and Imm(B), as well as between Pro(A) and Pro(B) in the gene expression level, which suggests our latent subclass findings are not random. Additionally, the plot indicates that the sign of $\hat{f}_1$ is driven by 20 genes.

We apply a gene functional enrichment analysis using DAVID (Huang et al. 2009) on these 20 genes, and we find that most of them (17 with adjusted p-value $\leq 3 \times 10^{-4}$) are related to glycoprotein and secreted protein. These findings suggest that further biological investigation may be worth pursuing. Note that the selected active genes only consist of 5 percent of the genes in the training data set, so our methods not only decrease classification error and detect new subclass structure, but they also facilitate variable selection.

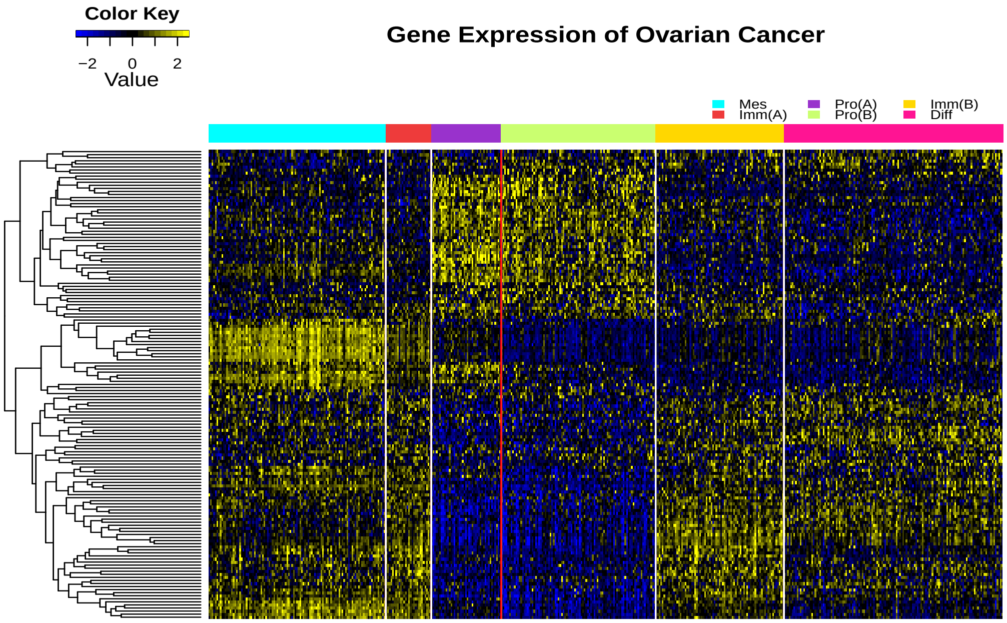

Figure 3.4: Heatmap of ovarian cancer data using 153 active genes selected by sparse CLM with LUM loss. Samples are displayed in columns by subtypes. Genes are ordered by hierarchical clustering. Nearly all samples on the left of the red line have average f_1 greater than 0, and the remaining samples have average f_1 less than 0. We can see a clear distinction between Imm(A) and Imm(B), and a mild difference between Pro(A) and Pro(B), which suggests that subclasses exist in the immunoreactive and differentiated subtypes.

3.7 Discussion

In this article, we propose Composite Large Margin classifiers to address the classification problem with latent subclasses by splitting the data and classifying the subsets using separate linear classifiers. Our approach inherits the nice interpretability of the standard linear approach while maintaining flexibility to handle complex data structures. At the same time, our classifier is simpler than more complex methods such as kernel techniques and tree based methods. Consequently, it may have less tendency for overfitting. In addition, the CLM method not only detects latent subclasses but also enables visualization of high dimensional data using low dimensional plots.

To achieve feature selection, we also propose a refitting algorithm for CLM. One future direction is to explore variable selection consistency. Another direction is to make use of other penalties such as the group-lasso penalty (Yuan and Lin 2006) in CLM besides the ℓ_2 penalty for selecting groups of variables.

Although our focus is on the CLM with the LUM and logistic losses, the basic idea can be implemented with other linear classifiers as well. Currently, our method is designed for binary classification with up to two latent subclasses in each class, so only one splitting function is needed. We can also extend the CLM method to permit multiple cuts to handle data with potentially multiple latent subclasses.

CHAPTER4: PERSONALIZED DOSE FINDING USING OUTCOME WEIGHTED LEARNING

In dose-finding clinical trials, there is a growing recognition of the importance to consider individual level heterogeneity when searching for optimal treatment doses. Such an optimal individualized treatment rule (ITR) for dosing should maximize the expected clinical benefit. In this chapter, we consider a randomized trial design where the candidate dose levels are continuous. To find the optimal ITR under such a design, we propose an outcome weighted learning method which directly maximizes the expected clinical beneficial outcome. This method converts the individualized dose selection problem into a penalized weighted regression with a truncated ℓ_1 loss. A difference of convex functions (DC) algorithm is adopted to efficiently solve the associated non-convex optimization problem. The consistency and convergence rate for the estimated ITR are derived and small-sample performance is evaluated via simulation studies. We demonstrate that the proposed method outperforms competing approaches. We illustrate the method using data from a clinical trial for Warfarin (an anti-thrombotic drug) dosing.

4.1 Introduction

Dose finding plays an important role in modern clinical trials aiming to assess the toxicity of drugs, to identify the maximum tolerated dose for safety usage, and to determine the efficiency of the drug. For example, a double-blinded Phase II trial for dose-finding is usually conducted to identify the no-effect, the mean effective and the maximal effective dosages (Chevret 2006). Optimal treatment dose identification is essential for the drug use for patients, since an over-dose can increase the occurrence of side effects, while an under-dose can weaken the therapeutic effect of the drug.

In a traditional randomized dose trial, patients are randomized to several fixed safe dose levels, and the single optimal dose level is typically determined by comparing the average outcomes across each dose level. However, such one-size-fit-all therapy can be inefficient when the responses of a medicine are heterogenous among the patients, i.e. what works for some patients may not work for others. Hence, personalized medicine, which tailors the treatment according to individual health condition and disease prognosis, is necessary. Taking the breast cancer treatment as an example, research suggests that individualized treatment can be more beneficial for young women with early-stage (lymph-nodenegative) breast cancer, such that surgery and localized radiation treatment can lead to a lower rate of recurrence than the traditional adjuvant chemotherapy (Van't Veer and Bernards 2008). On the other hand, continuous individualized dose levels are needed for practical dosing problem. For example, it is known that various dose levels for warfarin (a commonly used medicine for preventing thrombosis and thromboembolism) are needed for treating individuals with different clinical and genetical conditions (Klein et al. 2009) and the optimal dose can vary from 10mg to 80mg per day.

To achieve personalized interventions, many methods have been developed to group patients into subgroups according to their risk levels as estimated by some parametric or semiparametric models (Cai et al. 2010). However, these methods require prior knowledge on the optimal treatment, which is usually not available in typical randomized clinical trials. For estimating an optimal individualize treatment rule (ITR) with binary candidate treatment (denoted as A) of a single decision problem, recently, two main methods have been proposed: namely indirect methods and direct methods.

The indirect methods first model the reward (denoted as R) as a parametric or semiparametric function of the treatment and predictive covariates (denoted as X), and then the treatment is determined by the sign of the estimated treatment difference for $E(R|X, A = 1) - E(R|X, A = -1)$ (Robins 2004, Moodie et al. 2009, Qian and Murphy 2011). In particular, Qian and Murphy (2011) used ℓ_1 penalized least square, and Robins (2004) employed g-estimation. These indirect methods have drawbacks due to the different goals between the reward modeling and optimal treatment rule finding. For example, if the model of the reward is misspecified or overfitted, the two-step procedure may yield a suboptimal treatment decision. Instead, Zhao et al. (2012) introduced the framework of outcome weighted learning (O-Learning) for finding the optimal binary treatments directly. Specifically, Zhao et al. (2012) formulated the problem as a weighted binary classification with the rewards as weights. The authors demonstrated the superior performance of O-Learning over the indirect methods especially when the sample size is small, which is not uncommon in clinical trials. In another work, Zhang, Tsiatis, Davidian, Zhang and Laber (2012), Zhang, Tsiatis, Laber and Davidian (2012) proposed a robust semiparametric regression functions for maximization to infer the optimal rule; however, the learning approach in Zhao et al. (2012) is nonparametric and robust and can handle high-dimensional X commonly seen in practice. For estimating ITR with continuous dose, Rich et al. (2013) proposed a structured nested mean model for multiple decision problem. The method is still indirect method. Furthermore, Zhao et al. (2012) cannot directly be applied to the dose finding problem we are interested in. This is because the treatment option A is continuous so $P(A = a) = 0$.

In this chapter, we consider a single-stage trial where treatment is provided once. Our goal is to propose a trial for personalized dose finding and provide a robust analysis method based on the outcome weighted learning method. We assume the data is from randomized clinical trial with continuous dose level, and our method estimate a treatment policy for dosage based on patient's information such as age, gender, and genetic information. Our proposed method is a non-trivial extension of O-Learning for binary candidate treatments proposed by Zhao et al. (2012). The background information for the individual treatment rule and O-Learning are introduced in Section 4.2.1. Under the O-Learning framework, we demonstrate that the dose finding problem is a weighted regression with random treatment as the outcome, prognosis factors as predictors and the individual responses as weights in Section 4.2.2. We propose a nonconvex loss function for the weighted regression model, and provide a difference convex (DC) algorithm to solve the corresponding optimization problem in Section 4.3. The theoretical property of our method is provided in Section 4.4. In particular, we show that our loss function can lead to consistent estimation of the optimal dose. In addition, we derive the convergence rates for the estimated treatment rule. We demonstrate that our proposed method can identify individual treatment rule with better predicted rewards than the indirect methods through simulation studies in Section 4.5. A real data application is given in Section 4.6. Lastly, some future work including dose finding for a multi-stage treatment setting is discussed.

4.2 Methodology

4.2.1 Individualized Treatment Rule

We assume that the data are collected from a randomized trial with treatment assignment A. The traditional dose finding trial is usually designed to identify the best dose level among some small number of fixed dose levels. In contrast, we allow A to be continuous and within a bounded interval (the safe dose range). For simplicity we assume that $\mathcal{A} = [0, 1]$ and are independent of any patient's prognostic variables which is a d-dimensional vector $X = (X_1, X_2, ..., X_d)^T \in \mathcal{X}$. We define the observed clinical beneficial outcome as R, which is also known as "reward". We assume that reward is bounded and a large reward value is desired. Hence, an individualized treatment rule (ITR) is a map $f : \mathcal{X} \to \mathcal{A}$, and the optimal ITR is a rule that maximizes the expected reward if implemented.

Following the similar notation in Qian and Murphy (2011) and also Zhao et al. (2012), we denote the distribution of (X, A, R) as P and the expectation with respect to P as E. We denote the expected reward under the ITR f as $\mathcal{V}(f)$, which is also called value function. Furthermore, we define potential outcomes $R^*(a)$ representing the outcome that would be observed if treatment a is given to a subject. Under the Stable Unite Treatment Value Assumption (SUTVA) as in Rubin (1978): $R = \int I(A = a)R^*(a)p(a|x)da$, so that the observed outcome is the potential outcome that would be seen under the treatment actually received and noting that $R^*(a), a \in \mathcal{A}$ is independent of A given X due to the randomization, for any rule $f(X)$, we obtain

$$\begin{aligned}\mathcal{V}(f) &= E(R^*(f)) \\ &= E_X[E\{R^*(f)|X\}] = E_X[E\{R^*(f)|A = f(X), X\}] \\ &= E_X[E\{R|A = f(X), X\}], \qquad (4.1)\end{aligned}$$

where the first equation follows from the randomization and (4.1) follows from the SUTVA assumption. As a result, the optimal rule $f_{opt} = \arg\max_f \mathcal{V}(f) = \arg\max_f E(R|A = f(X))$. Note that f_{opt} does not change if R is placed by $R + c$ for any constant c, also the f_{opt} is invariant of the scale of R. Thus, without loss of generality, we assume that R is positive by subtracting R from its lower bound.

4.2.2 Outcome Weighted Learning

Since A is continuous, the chance of observing $A = f(x)$ is zero so $\mathcal{V}(f)$ cannot be obtained directly form data. This is the key challenge for the dose finding as compared to the situation when the treatment takes discrete values (c.f. Zhao et al. (2012)). To handle this, we propose the following approximation to estimate $\mathcal{V}(f)$. First, we note

$$\begin{aligned}
\mathcal{V}(f) &= E(R|A = f(X)) \\
&= \lim_{\phi \to 0^+} E_X \left(\frac{\int_{a \in (f(X)-\phi, f(X)+\phi)} E(R|a, x) p(a, X) da}{\int_{a \in (f(X)-\phi, f(X)+\phi)} p(a, X) da} \right) \\
&= \lim_{\phi \to 0^+} E_X \left(\frac{\int E[RI(a \in (f(X) - \phi_n, f(X) + \phi_n))|a, X] p(a|X) da}{\int_{a \in (f(X)-\phi, f(X)+\phi)} p(a|X) da} \right) \\
&= \lim_{\phi \to 0^+} E \left(\frac{RI[A \in (f(X) - \phi, f(X) + \phi)]}{2\phi p(A|X)} \right),
\end{aligned}$$

where $p(a|x)$ is the density of $A = a$ given $X = x$ which is known in the randomized design and assumed to be bounded by a positive constant from below.

Thus, if we let $\widetilde{\mathcal{V}}(f) = E\left(\frac{RI[A \in (f(X)-\phi, f(X)+\phi)]}{2\phi p(A|X)}\right)$, we may consider to maximize $\widetilde{\mathcal{V}}(f)$ to obtain an approximate optimal ITR. Equivalently, we aim to minimize $E[RI(|A - f(X)| > \phi)/(2\phi p(A|X))]$. However, the $0 - 1$ loss $I(|A - f(X)| > \phi)$ is difficult to optimize due to its discontinuity in f (Zhang 2004). In contrast, in machine learning context, one usually chooses a continuous surrogate loss for optimization. For our purpose, we choose the surrogate loss to be

$$\ell_\phi(A - f(X)) = \min(|A - f(X)|/\phi, 1).$$

Its corresponding objective function for minimization is then $\mathcal{R}_\phi(f) = E\left(\frac{R\ell_\phi(A - f(X))}{\phi_n p(A|X)}\right)$ and the counterpart to $\widetilde{\mathcal{V}}(f)$ is $1 - \mathcal{R}_\phi(f)$, which is denoted as

$$V_\phi(f) = E\left[R \max(1 - |A - f(X)|/\phi, 0)/(\phi p(A|X))\right]$$

. The graphical presentation of ℓ_ϕ can be found in the right panel of Figure 1, which can also be written as the difference of the two convex functions shown in the left panel. When $\phi = 1$, then the corresponding ℓ_ϕ loss is also called truncated L_1 penalty, which Shen et al. (2012) used as a surrogate for l_0-penalty in a variable selection problem. On the other hand, it can also be treated as a kernel estimator of $0 - 1$ loss using the triangular kernel. The theoretical justification for using such a loss function will be further given in Section 4.4.

Using n observed data $(A_i, X_i, R_i), i = 1, ..., n$, given a fixed $\phi = \phi_n$ which may vary with n, we attempt to minimize

$$\sum_{i=1}^{n} \frac{R_i \ell_\phi [A_i - f(X_i)]}{\phi_n p(A_i|X_i)}.$$

To prevent overfitting, we penalize the complexity of $f(X)$. Hence, our O-learning framework have a loss plus penalty form as follows:

$$\min_f \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{R_i \ell_\phi (A_i - f(X_i))}{2\phi_n p(A_i|X_i)} + \lambda_n \|f\|^2 \right\}. \tag{4.2}$$

where $\|f\|$ is some norm for f, and λ_n controls the severity of the penalty on f. For example, if we assume a linear decision rule: $f(X) = X^T \mathbf{w} + b$, then $\|f\|$ is the Euclidean norm of $\mathbf{w}$.

4.3 Computation Algorithm

The objective function (4.2) is nonconvex, and the nonconvex optimization is known to be difficult. In the following section, we adopt the DC algorithm (An and Tao 1997, Wu and Liu 2007) to tackle this nonconvex optimization. We first discuss the algorithm for the linear learning rule, where $f(x)$ is a linear function of x, and then extend it to the nonlinear learning where $f(x)$ is chosen from a reproducing kernel Hilbert space.

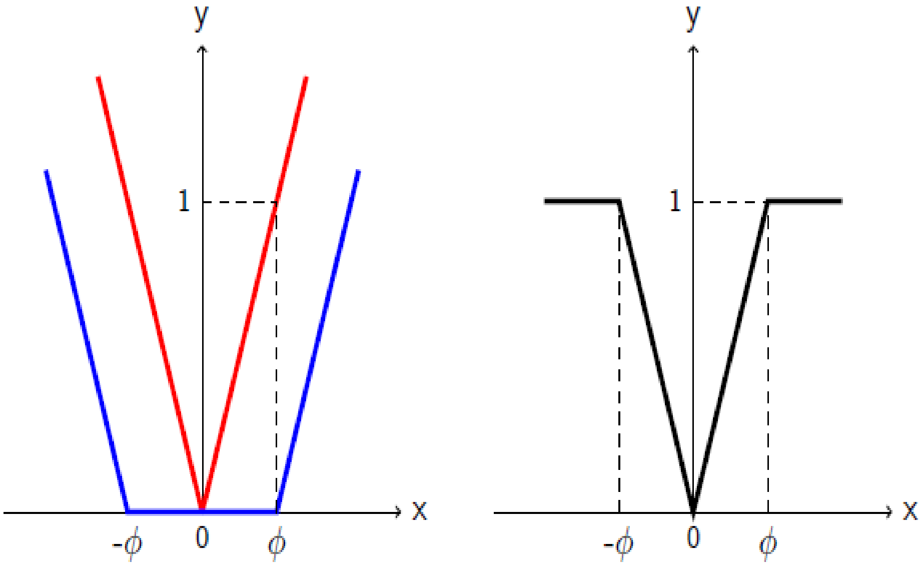

Figure 4.1: Loss function of ℓ_ϕ for dose finding in ITR

4.3.1 Linear Learning

Considering $f(x) = x^T \mathbf{w} + b$. We can formulate the objective function as follows:

$$S = \frac{\lambda_n}{2} || \mathbf{w} ||_2^2 + \frac{1}{\phi_n} \sum_{i=1}^{n} R_i \min(\frac{| A_i - \mathcal{D}(X_i) |}{\phi_n}, 1).$$

where λ_n is the tuning parameter that balances the bias and variance. Denote that $\Theta = (\mathbf{w}, b)$. The objective function S can be expressed as the difference of two convex functions: $S(\Theta) = S_1(\Theta) - S_2(\Theta) = \big(\frac{\lambda_n}{2} || \mathbf{w} ||_2^2 + \frac{1}{\phi_n} \sum_{i=1}^{n} R_i \frac{|A_i - \mathcal{D}(X_i)|}{\phi_n}\big) - \frac{1}{\phi_n} \sum_{i=1}^{n} R_i \big(\frac{|A_i - \mathcal{D}(X_i)|}{\phi_n} - 1\big)_+$. Then the DC algorithm minimizes a sequence of convex subproblems to solve the original nonconvex minimization problem: initialize Θ^0 then repeat updating Θ by $\Theta^{t+1} = \text{argmin}_\Theta(S_1(\Theta) - \langle \nabla S_2(\Theta^t), \Theta - \Theta^t \rangle)$ until the convergence of Θ. For the initial value of Θ, we use observations with large reward assuming that the observations with large reward are more likely to receive treatment doses close to the optimal ones. Specifically, we use least square estimator to predict A

with X as predictors if R is large, e.g. observations with R in the upper 50 percentile of the training data.

Define $Q_i^{(t)} = I(|a_i - \mathbf{x}_i^T \mathbf{w}^t - b^t| \leq \phi_n)$, where $\mathbf{x}_i^T \mathbf{w}^t + b^t$ is the temporary predicted optimal dose with $\mathbf{w}^t$ and b^t as the solution from the t-th iteration. After some algebra manipulations, the objective function of $t+1$-th iteration $S^{(t+1)}(\Theta)$ equals to $\frac{\lambda_n}{2} || \mathbf{w} ||_2^2 + \frac{1}{\phi_n^2} \sum_{i=1}^n R_i Q_i^{(t)} |a_i - \mathbf{x}_i^T \mathbf{w} - b|$. Consequently, the convex subproblem is a weighted penalized median regression problem. Note that the t-th iterations' result only impacts the S^{t+1} through Q_i^t. Thus, if the observation receive a dose that is close to the surrogate optimal dose ($Q_i^{(t)} = 1$), then the observation will contribute to the objection function of the $t+1$ step subproblem otherwise it will not contribute. Let $\mathcal{T} = \{i : Q_i^{(t)} = 1\}$. Divide the objective function by λ_n and plug slack variables into $S^{(t+1)}(\Theta)$ to get rid of the absolute function, then the primary optimization problem of $t+1$-th iteration is

$$\min_{\mathbf{w},b,\xi,\tilde{\xi}} \frac{1}{2} || \mathbf{w} ||_2^2 + \frac{1}{\lambda_n \phi_n^2} \sum_{i \in \mathcal{T}} (\xi_i + \tilde{\xi}_i) R_i, \tag{4.3}$$

subject to $\xi_i, \tilde{\xi}_i \geq 0$, $a_i - \mathbf{x}_i^T \mathbf{w} - b \leq \xi_i$, $-(a_i - \mathbf{x}_i^T \mathbf{w} - b) \leq \tilde{\xi}_i$, $\forall i \in \mathcal{T}$. After using the Lagrangian multiplies and some algebra, we have

$$\min_{\mathbf{w},b,\xi,\tilde{\xi}} L = \frac{1}{2} || \mathbf{w} ||_2^2 + \frac{1}{\lambda_n \phi_n^2} \sum_{i \in \mathcal{T}} (\xi_i + \tilde{\xi}_i) R_i - \sum_{i \in \mathcal{T}} \alpha_i (\xi_i - a_i + \mathbf{x}_i^T \mathbf{w} + b)$$

$$- \sum_{i \in \mathcal{T}} \tilde{\alpha}_i (\tilde{\xi}_i + a_i - \mathbf{x}_i^T \mathbf{w} - b) - \sum_{i \in \mathcal{T}} u_i \xi_i - \sum_{i \in \mathcal{T}} \tilde{u}_i \tilde{\xi}_i$$

subject to

$$\frac{\partial L}{\partial \mathbf{w}} = w - \sum_{i \in \mathcal{T}} \alpha_i \mathbf{x}_i + \sum_{i \in \mathcal{T}} \tilde{\alpha}_i \mathbf{x}_i = 0; \; \frac{\partial L}{\partial b} = - \sum_{i \in \mathcal{T}} \alpha_i + \sum_{i \in \mathcal{T}} \tilde{\alpha}_i = 0;$$

$$\frac{\partial L}{\partial \xi_i} = \frac{R_i}{\lambda_n \phi_n^2} - \alpha_i - u_i = 0; \; \frac{\partial L}{\partial \xi_i} = \frac{R_i}{\lambda_n \phi_n^2} - \tilde{\alpha}_i - \tilde{u}_i = 0.$$

By plugging the equations got from the above constraint, we obtain a convex dual problem as follows:

$$\min_{\alpha,\tilde{\alpha}} \frac{1}{2} \sum_{i\in\mathcal{T}} \sum_{j\in\mathcal{T}} (\alpha_i - \tilde{\alpha}_i) < x_i, x_j > (\alpha_j - \tilde{\alpha}_j) - \sum_{i\in\mathcal{T}} (\alpha_i - \tilde{\alpha}_i) a_i$$

subject to

$$\sum_{i\in\mathcal{T}} (\alpha_i - \tilde{\alpha}_i) = 0;\ 0 \le \alpha_i \le \frac{R_i}{\lambda_n \phi_n^2},\ 0 \le \tilde{\alpha}_i \le \frac{R_i}{\lambda_n \phi_n^2},\ \forall i \in \mathcal{T}$$

The $< . >$ denotes the inner product. This dual problem is a quadratic programming (QP) problem and can be solved by many standard optimization packages. Once its solution is obtained, the coefficients $\mathbf{w}$ can be recovered by the relation that $\mathbf{w} = \sum_{i\in\mathcal{T}} (\alpha_i - \tilde{\alpha}_i) \mathbf{x}_i$. After the solution of $\mathbf{w}$ is derived, b can be obtained by solving either a sequence of Karush-Kuhn-Tucker conditions or a linear programming (LP) problem (Boyd and Vandenberghe 2004), i.e.

$$\min_{b,\eta} \sum_{i\in\mathcal{T}} \eta_i R_i,$$

subject to $\eta_i > 0$, $a_i - \mathbf{x}_i^T \mathbf{w} - b \le \eta_i$, $a_i - \mathbf{x}_i^T \mathbf{w} - b \ge -\eta_i$, $\forall i \in \mathcal{T}$.

The algorithm would stop when $|| \mathbf{w}^{t+1} - \mathbf{w}^t ||$ is smaller than a pre-specified small constant (10^{-8} in our simulations). Note that as the convex function S_2 is replaced by its affine majorization, the DC algorithm can be also regarded as a special case of the minorize-maximize or majorize-minimize (MM) algorithm (Hunter and Lange 2004). As the objective function S is bounded below by 0 and $S^{(t)}$ is descending after each iteration, it guaranteed that the DC algorithm converges to an local minimizer in finite steps (An and Tao 1997, Wu and Liu 2007).

4.3.2 Nonlinear Learning

For nonlinear learning, the decision function $f(X) = \mathbf{w}^T \Phi(X)$, where $\Phi(.)$ is an unknown transformation on X. We further define the kernel $K(\cdot, \cdot)$ as a positive definite function mapping from $\mathbb{R}^d \times \mathbb{R}^d$ to $\mathbb{R}$ such that $< \Phi(x_i), \Phi(x_j) >= K(x_i, x_j)$. At each iteration in the previous section, since the QP problem depends on the inner products between x_i, similar derivation as in the linear learning leads to the following dual problem for nonlinear learning

$$\min_{\alpha,\tilde{\alpha}} \frac{1}{2} \sum_{i\in\mathcal{T}} \sum_{j\in\mathcal{T}} (\alpha_i - \tilde{\alpha}_i) K(x_i, x_j)(\alpha_j - \tilde{\alpha}_j) - \sum_{i\in\mathcal{T}} (\alpha_i - \tilde{\alpha}_i) a_i$$

subject to

$$\sum_{i\in\mathcal{T}} (\alpha_i - \tilde{\alpha}_i) = 0;\ 0 \le \alpha_i \le \frac{R_i}{\lambda_n \phi_n^2},\ 0 \le \tilde{\alpha}_i \le \frac{R_i}{\lambda_n \phi_n^2},\ \forall i \in \mathcal{T}.$$

After solving the above QP problem, we can recover the coefficients $\mathbf{w}$,

$$\mathbf{w} = \sum_{i=1}^{n} I(i \in \mathcal{T})(\alpha_i - \tilde{\alpha}_i).$$

The intercepts b can be solved using LP as in the linear learning. Due to the representation theorem of Kimeldorf and Wahba (1971) and the derivation from above, we obtain $f(X) = \sum_{i=1}^{n} I(i \in \mathcal{T})(\alpha_i - \tilde{\alpha}_i) K(X, x_i) + b$. In this chapter, we focus on the implementation of using the Gaussian kernel, i.e. $K(x_i, x_j) = \exp(-\gamma^{-2} ||x_i - x_j||_2^2)$, where $\gamma > 0$ and is known as bandwidth. We denote the RKHS induced by Gaussian kernel as H_γ. In our subsequent implementation, the bandwidth parameter γ for the Gaussian kernel is tuned between the first quartile, the median, and the third quartile of all pairwise Euclidean distances of training inputs. Note that the resulting function may not always generate a dose within the target range. If out of range, we can either do a truncation afterwards or do logarithm transformation of dose level before model fitting.

4.3.3 Tuning Parameter

For the tuning parameter, we choose them by cross validation. Specifically, we utilize the loss function $L_{f(X),\phi_n}$ to help tune the parameter, the strategy are as follows:
(a) Divide the data into training, tuning sets. (b) Denote candidate values of ϕ_n as $\Phi = \{0.4, 0.35, 0.3, 0.25, 0.2, 0.15, 0.1, 0.05\}$. (c) ϕ_n, find optimal rule $\widehat{f}_\phi(X)$ by cross validation with a pair of (λ_n, γ) such that the criterion $L_{f,\phi_n}(R_{train}, A_{train})$ is minimized using the training data. (d) $\phi_{opt} = \arg\min_{\phi_n \in \Phi} L_{\widehat{f}_\phi(X),0.01}(R_{tune}, A_{tune})$. (e) Given ϕ_{opt}, identify the best (λ_n, γ) by cross validation.

4.4 Theoretical Results

In this section, we study the asymptotic behavior of the minimizer $(\hat{f}_n)$ of the optimization problem proposed in Section 4.3. In particular, we will show that $\mathcal{V}(\hat{f}_n)$ converge to $\mathcal{V}(f_{opt})$ with certain rate. Our first result shows the approximation of the value function is valid.

Theorem 4.4.1. *For any measurable function $f : \mathcal{X} \to \mathbb{R}$ and any probability distribution P, if $\sup_{(a,x)\in\mathcal{A}\times\mathcal{X}} |\frac{\partial}{\partial a}E(R|A = a, X = x)| \leq C$ for a bounded constant C, then $|\mathcal{V}_\phi(f) - \mathcal{V}(f)| \leq (C + 1)\phi_n$.*

Proof: Clearly,

$$
\begin{aligned}
\mathcal{V}_{\phi_n}(f) &= E\left(R\max(1 - \frac{|A - f(x)|}{\phi_n}, 0)/\phi_n p(A|X)\right) \\
&= \int \left\{\frac{1}{\phi_n^2}\int_{|f(x)-a|\leq\phi_n} [\phi_n - |a - f(x)|]E(R|a, x)da\right\} p(x)dx \\
&= \frac{1}{\phi_n^2}\int_{|z|\leq\phi_n} [\phi_n - |z|]E(R|z + f(x), x)dz.
\end{aligned}
$$

By Taylor expansion, we obtain

$$|\mathcal{V}_{\phi_n}(f) - \mathcal{V}(f)| \le \frac{1}{\phi_n^2}\int_0^{\phi_n} 2z dz \le C\phi_n.$$

Our main result concerns about the convergence rate of $\mathcal{V}(\widehat{f}_n) - \mathcal{V}(f_{opt})$.

Theorem 4.4.2. *Assume that the Bayes decision rule $f_{opt} \in B^{\alpha}_{1,\infty}(\mathbb{R}^d)$, a Besov space. Then, for any $\epsilon > 0$, $d/(d+\tau) < p < 1$, the tail component $\tau > 0$, and the bandwidth γ_n for the Gaussian kernel, the following inequality holds:*

$$\mathcal{V}(f_{opt}) - \mathcal{V}(\widehat{f}_n) \le c_1 \left[\frac{1}{\gamma_n^{(1-p)(1+\epsilon)d}\lambda_n^p n}\right]^{\frac{1}{2-p}} + c_2(\tau/n)^{\frac{1}{2}} + c_3(\tau/n) + c_4\lambda_n\gamma_n^{-d} + c_5\frac{\gamma_n^{\alpha}}{\phi_n} + c_6\phi_n$$

with probability P_n not less than $1 - 3e^{-\tau}$.

With properly chosen parameters γ_n, λ_n, ϕ_n, the right hand side of the inequality will go to 0 as n goes to infinity. The sixth term is due to the usage of $\mathcal{V}_\phi$ to approximate $\mathcal{V}$. Others terms are from approximation error due to using H_k and stochastic error due to finite sample size. Choose p close to 0, and $(1-p)(1+\epsilon) = 1$, and we can further choose λ_n, γ_n, and ϕ_n to balance the approximation accuracy and the stochastic variability as follows:

$$\lambda_n = \left(\frac{1}{n}\right)^{\frac{d+0.5\alpha}{d+\alpha}}, \gamma_n = \left(\frac{1}{n}\right)^{\frac{1}{d+\alpha}}, \phi_n = \left(\frac{1}{n}\right)^{\frac{\alpha}{2(d+\alpha)}}$$

Then, the optimal rate for the value function is

$$\mathcal{V}(f_{opt}) - \mathcal{V}(\widehat{f}_n) = O_p\left(\left(\frac{1}{n}\right)^{\frac{\alpha}{2(d+\alpha)}}\right)$$

Theorem 2 implies that the value of estimated rule $\widehat{f}_n$ from O-learning converges to the optimal value function. Clearly, the convergence rate decreases as the dimension of prognostic variables increases. Furthermore, if α goes to infinity and d stays constant, the optimal convergence rate of O-learning with Gaussian kernel is close to the parametric rate $n^{-1/2}$. Note that the rate can not be close to n^{-1} as the rate proved by Zhao et al. (2012) for binary treatment option problem. The reason is due to the continuous nature of the dose, i.e. the data are not complete separated.

4.5 Simulation Study

We have conducted extensive simulations to access the performance of the proposed method with various training sample sizes. In these simulations, we generate 30−dimensional vectors of prognostic variables, $X_1, \ldots, X_{30}$, consisting of independent $U[-1, 1]$ covariates, where U represent the uniform distribution. The treatment A is generated from $U[0, 2]$ independently of X. The response R is normally distributed in $N(Q_0(\mathbf{X}, A), 1)$ with mean $Q_0(\mathbf{X}, A)$ and standard deviation 1, where $Q_0(X, A)$ reflects the interaction between treatment and prognostic variables and is chosen to vary according to the following scenarios:

$$\begin{aligned}
&(1)Q_0(\mathbf{X}, A) = 8 + 4X_1 - 2X_2 - 2X_3 - 25 \times (f_{opt}(X) - A)^2,\\
&\quad f_{opt}(X) = 1 + 0.5X_1 + 0.5X_2.\\
&(2)Q_0(\mathbf{X}, A) = 8 + 4X_2 - 2X_4 - 8X_5^3 - 15 \times |f_{opt}(X) - A|,\\
&\quad f_{opt}(X) = \frac{1}{3}|(X_1^2 - X_7^3 + \exp(X_3^3 - X_6 - 2 + \sin(X_2))|.
\end{aligned}$$

In Example 1, the optimal dose is a linear function of X, and in Example 2, the optimal dose is a non-linear function of X.

We evaluate the performance of the proposed method by comparing the expected value function. The empirical expected optimal value function is calculated from the average value functions estimation of 200 simulations, where each estimation is based on the mean estimated reward of testing sets with 5000 samples. For Each sample, the estimated reward is calculated by plug in the estimated optimal dose and the observed diagnosis value into the true underline value function. The tuning parameter is selection by five fold cross validation.

Two competitive methods are considered in this chapter. One is a modified LASSO based method proposed by Qian and Murphy (2011). It is a two-stage procedure that it first build a model for $E(R|A, X)$, the conditional expected reward given treatment and other covariates. Given the model, for each individual, the estimated optimal dose is determined individually such that the dose maximizes the estimated reward. The design matrix used in the LASSO model for both examples is (X, A, X^2, XA, A, A^2). In other words, we assume that given the diagnosis variables the treatment dose and the reward have a quadratic relationship. This assumption is different from of the original two-stage method by Qian and Murphy (2011) for binary treatment problem which assumed that $E(R|A, X)$ is linear in A and X. The reason is that if we also assume the linear relation for dose finding problem then the predicted optimal dose will only be the two extreme value of the feasible treatment dose. Hence, to prevent such trivial result, we need to assume a higher order relation between dose and reward. Note that maximizing the predicted reward is also more complicated than that for binary treatment. We suggest a grid search procedure to find the optimal treatment dose for the two-stage procedure. In particular, if the feasible dose level is $(0, 2)$, we can choose 400 equally spaced grids within the interval $(0, 2)$ as candidate doses. Given patient diagnosis variables, we can comparing the predicted rewards by plugging the candidate doses into the model. As a result, the grid that maximizes the predicted reward is recorded as the optimal dose for that patient. In the same spirit of the two stage methods, we can model $E(R|A, X)$ by nonparametric function instead of parametric function. For example, we can use the supporting vector regression (SVR) with the gaussian kernel to reflect the potential non-linear relation between R and A as well as the interaction between A and X. A similar grid search for optimal dose can be done for the two-stage method based on SVR as well. For both LASSO and SVR, the tuning parameters are selected by five fold cross validation.

Table 4.1: Average $\widehat{\mathcal{V}}(f)$ across 200 simulations under different treatment rules

	n	K-O-Learning	L-O-Learning	SVR	LASSO
Example 1	50	3.58 (0.80)	3.47 (0.82)	-6.01 (2.39)	-5.87 (2.41)
$\mathcal{V}(f_{opt}) = 8$	100	3.98 (0.62)	3.99 (0.67)	-0.24 (1.41)	-0.12 (1.45)
$\mathcal{V}(A = 1) = 3.8$	200	4.80 (0.47)	4.69 (0.45)	3.05 (0.91)	3.87 (0.71)
	400	5.58 (0.35)	5.61 (0.33)	3.64 (0.71)	4.91 (0.48)
	800	7.03 (0.27)	7.10 (0.23)	5.84 (0.49)	6.65 (0.40)
Example 2	50	3.04 (0.90)	3.02 (0.95)	-4.02 (1.48)	-5.67 (1.41)
$\mathcal{V}(f_{opt}) = 8$	100	3.69 (0.61)	3.48 (0.65)	0.28 (0.89)	-1.12 (0.92)
$\mathcal{V}(A = 0.8) = 2.6$	200	4.50 (0.45)	4.09 (0.51)	3.05 (0.81)	2.67 (0.64)
	400	5.38 (0.30)	4.76 (0.35)	4.14 (0.73)	3.71 (0.48)
	800	6.69 (0.21)	5.85 (0.26)	5.32 (0.62)	4.40 (0.23)

We can see that in both examples, our method works well especially in the low dimensional case. In example 1, L-O-Learning and K-O-Learning perform equally well, and in example 2, K-O-Learning perform better especially when sample size is large. In both cases, our methods outperform the two stage methods, such different is more significant for small sample size situations. In practice, the sample size for clinical trial is usually small, hence our method can be more useful than the two-stage method. Note that although LASSO model in example 1 correctly specify the quadratic relationship between the reward and the treatment, but it still suffers from overfitting the data when the sample size is not large enough. As a result, it performs worse than our method when the sample size is small and comparable to our method when the sample size is large. The optimal value functions for both examples are 8. As the sample size increases, the value functions of estimated rules from all methods increase. Moreover, for example 1 and example 2, the respective optimal fixed dose is 1 and 0.8 with the value function equal to 3.8 and 2.6 (the value is independent of the training sample size). When sample size is greater than 100, our methods yield a larger value function than the optimal value of fixed dose. The result indicates that our proposed randomized trial design with our proposed O-learning method can yield a better rule than the fixed dose rule identified by the traditional dose finding trial. For scenarios similar to the simulations, the personalized treatment may be beneficial and worth consideration.

4.6 Warfarin Dosage

Warfarin is commonly used medicine for preventing thrombosis and thromboembolism. Proper dosage of warfarin is vitally important, as overdose predisposes to a high risk of bleeding, while underdose is insufficient to protect against thrombosis. International normalized ratio (INR) is a measure of how fast the blood can clot, and is used to monitor whether the dose of warfarin is safe. For normal people, the INR is referred as 1, and for patient taking warfarin, the targeted INR range is typically between 2 to 3 (Klein et al. 2009). Predicting the optimal dose for warfarin is still an open problem in the medical community and a lot of methods are have been proposed to refine the optimal dose rule (Klein et al. 2009, Hu et al. 2012). Klein et al. (2009) compared three methods for predicting warfarin dose: models by clinical data, models by pharmacogenetic data or a single fixed dose rule. The paper concluded that pharmacogenetic data had better performance in predicting the optimal dose. In the Klein et al. (2009), samples are used for training the prediction model only if their INR are stable and between 2 to 3 after receiving the Warfarin treatment. The authors fitted a linear model with the received dose as response and pharmacogenetic data as predictor. Such steps are valid when the doses received by the patients in the training data are optimal (optimal dose assumption). Later studies found that the pharmacogenetic model for optimal Warfarin dose identification by Klein et al. (2009) can be suboptimal for elder patients. Hence, it is reasonable to assume that optimal dose assumption may be violated in practice.

To estimate the optimal dose, we utilize both the pharmacogenetic and clinical variables including age, height, weight, race, CYP2C9 genotype, and VKORC1 genotype, and the usage of two types of medicines: Cytochrome P450 enzyme (including phenytoin, carbamazepine, and rifampin) and Amiodarone (Cordarone)). After removing observations with missing data in these covariates, there are total 1744 patients with 400 patients with INR not in range 2 to 3. Instead of using patients with INR between 2 to 3 after treatment and making the optimal dose assumption, we include all these 1744 patients for analysis. To convert the INR to direct measure of reward, we code $R_i = -|INR_i - 2.5|$ for i-th individual, as INR = 2.5 is ideal. Note that the study is a observational rather than randomized trial, so we need to address the potential sampling bias of the dose assignment. We estimate the propensity score and use the estimated propensity score to scale the reward for each patient. We first fit a linear model with the assumption that given the covariates the assigned dose follows lognormal distribution. Then the propensity score can be calculated by plugging the fitted value into the normal density formula (Imai and Van Dyk 2004). We randomly split the data into training and testing sets 100 times independently. We consider the scenario that the training set contains 800 samples and the testing set contains the rest. The performances of different methods are evaluated by the value functions estimated from the testing set averaged across these 100 splits.

For the real data, the true relationship between dose and reward is unknown, hence we need to estimate the value function for a given treatment rule. A potential criterion is using $\mathcal{V}_\phi(.)$ as criterion, however as our method directly maximize the approximated value function, it is unfair to compare different methods under such criterion. For binary treatment problems, the value function is unbiasedly estimated by $\mathbb{P}_n^*[I(A = f(X))R/P(A)]/\mathbb{P}_n^*[I(A = f(X))/P(A)]$ (Murphy 2003), where P_n^* denotes the empirical average using the validation data and $P(A)$ is the probability of being assigned treatment A. We dichotomize the observed and predicted treatment into $sign(A - c)$ and $sign(f(X) - c)$, then the method for value function estimation with binary treatment can be applied. We choose different cutoff point c, and if one method have larger values than the other methods under most of c value, then such method is better than the others. In the Warfarin example, the c is chosen to be 20,40 and 60. The Table 4.2 reflects the performance of difference methods. From the Table 4.2, we can see that our method is competitive to the two stage methods with the different choice of c.

Table 4.2: Numerical comparison of $\widehat{\mathcal{V}}(\mathbf{D}(X))$ average across 100 random splits

c	K-O-Learning	L-O-Learning	SVR	LASSO
20	-0.34 (0.06)	-0.35 (0.07)	-0.39 (0.09)	-0.40 (0.10)
40	-0.23 (0.05)	-0.25 (0.06)	-0.30 (0.08)	-0.33 (0.08)
60	-0.31 (0.06)	-0.30 (0.06)	-0.38 (0.09)	-0.38 (0.09)

4.7 Discussion

The proposed O-Learning method appears to be more effective in both simulations and real data, especially when the training sample size is small, which is not uncommon in clinical trial. Our method has advantages over two stage methods through directly estimating the optimal dose. As a result, our method is more robust to the model specification of the reward. Unlike O-Learning, to successfully identify the optimal dose for two-stage methods, one needs to correctly specify the model between reward (outcome) and treatment together with diagnosis variables (covariates). In practice, it is possible that variables that can affect the outcome are not observed. When these variables have no interaction with the treatment, O-Learning can be unaffected, while the two stage method can perform badly. Moreover, it is difficult to convert the reward-treatment model for the two stage method to find the optimal treatment for the dose finding problem. Under the binary treatment setting, one can compare the contrast between the reward given treatment A versus treatment B to find the better treatment. For dose finding, one can only enumerate the possible reward by plugging into a fine grid dose levels into the reward-treatment model, which can be computational intensive. In contrast, our method can find the optimal rule directly. Note that the loss function proposed in the paper can be further generalized to $|A - f(X)|_{\phi_1} - |A - f(X)|_{\phi_2}$ with $0 \leq \phi_1 < \phi_2$. The current loss function is a special case of the generalized one with $\phi_1 = 0$. In practice, such a generalization can provide further robustness to our methods with the expense of adding another tuning parameters.

In practice, the reward can be censored as happens, for example, with the survival time of cancer patients. Techniques such as inverse probability of censoring weighting can be estimated to weight the observations, however such a procedure can yield a less efficient rule. Similar problems can occur when the training data comes from an observational study. For both situations, we need to develop doubly robust estimators for ITR. In addition, the toxicity of the drug may need to be considered when identifying the optimal treatment dose (Thall and Russell 1998, Laber et al. 2014).

Another future extension of our method is to have variable selection for the implementation of our method especially for linear learning. For some complex diseases, sequential treatments are needed, hence dynamic treatment regimes instead of optimal single stage treatment is more useful. For warfarin dosing, patients need to take medicine on a regular basis, and the optimal doses may vary over time. Recently, Rich et al. (2013) proposed an adaptive strategy under the framework of a structured nested mean model. On the other hand, other methods for estimating dynamic treatment regimes under a reinforcement learning framework (Sutton and Barto 1998) have been proposed in several papers including Murphy (2003), Zhao et al. (2009) to solve multiple stage therapy problems. Extensions of our proposed method for dynamic treatment regimes could be of great interest.

CHAPTER5: CONCLUSION AND FUTURE RESEARCH PLAN

In the dissertation, I proposed three statistical learning methods: HSSVD for biclustering with heterogeneity of variance, Composite Large Margin Classifier for latent subclasses and O-learning for personalized dose finding. All three methods deal with various forms of data heterogeneity and are proven to be useful either empirically or theoretically. In the immediate future, I would like to explore the following additional topics:

Personalized Medicine and Dynamic Treatment Regimens: Treatments tailored for individuals have great potential to improve patient outcome and can bring new insight to drug evaluation. My future research on personalized medicine involves both designing ethical and efficient clinical trials to identify optimal personalized treatment, as well as developing estimation and inference methods using data from traditional and new trial data for personalized medicine. As many diseases require time-varying treatment, it will be of great interest to construct customized sequential decision rules. While I have focused on estimating optimal single-stage decisions, the O-learning framework can be generalized to the multi-stage or even continuous-stage decision settings setup with customized adaptive treatments according to the prognosis of the disease. Estimation and inference for dynamic treatment regimes is challenging as it is a non-regular estimation problem. I plan to combine empirical process methods, semiparametric inference tools and resampling methods to solve the problem. There are also other interesting subproblems for dynamic treatment regimes, e.g. dimension reduction and variable selection.

Supervised learning for data with complex structure: Due to the complex nature of big data, the outcome we are interested in can be multivariate or functional data like imaging data. In addition, the predictors may be no longer vectors but matrices or curves. It would be of great interest to develop supervised learning techniques to analyze these more complex data structures. This requires more comprehensive statistical modeling, while maintaining ease of interpretation. One potential solution is extending the latent supervised learning method to data with complex structure. Currently, the latent supervised learning method assumes there exists only one latent variable, hence one splitting function is enough. We need to allow multiple splitting functions for more complex data. Moreover, after dividing the data by splitting functions, we rely on existing statistical models, e.g., the CLM can use logistic regression with vectors as predictors. For problems with matrixes as predictors like imaging data, we need to develop new matrix regression method or incorporate existing matrix regression methods (Zhou et al. 2013) into the latent supervised learning framework.

Network data: Studying network data can help us understand some fundamental problems in biological and clinical science, e.g. how the gene is regulated. In particular, I plan to work on two subareas. One is constructing directed graphs from intervention data for casual inference. Directed acyclic graphs (DAGs) under the Gaussian assumption have been intensively studied in the literature, however, estimating DAGs under non-Gaussian assumptions is still open question. Beyond the estimation, I am also interested in testing and inference for DAGs. The other is clustering network data, also known as community/module detection. One interesting question to ask is that if we start with gene expression data, what is the best strategy to detect the gene communities? I plan to systemically study the combination of network structure extraction methods and module detection methods empirically, and provide some theoretical insights using random matrix theory.

APPENDIX : Asymptotic Results

A.1 Proofs of Theorem 4.4.2

Let f^*_ϕ is the minimizer of $\mathcal{R}_\phi(f)$ and by definition $\mathcal{R}_\phi(f) = E(R/\phi_n) - \mathcal{V}_\phi(f)$.

$$
\begin{aligned}
&\mathcal{V}(f_{opt}) - \mathcal{V}(\widehat{f}_n) = E(R|A = f_{opt}) - E(R|A = \widehat{f}_n) \\
&\leq \mathcal{V}_\phi(f_{opt}) - \mathcal{V}_\phi(\widehat{f}_n) + 2C\phi_n \leq \mathcal{R}_\phi(\widehat{f}_n) - \mathcal{R}_\phi(f_{opt}) + 2C\phi_n \\
&\leq \mathcal{R}_\phi(\widehat{f}_n) - \mathcal{R}_\phi(f^*_\phi) + c_6\phi_n \qquad (5.1)
\end{aligned}
$$

The first inequality is due to Theorem 4.4.1, and the second follows by the definition of $\mathcal{R}_\phi$. C is the same constant as in Theorem 4.4.1 and $c_6 = 2C$. Denote that $\mathcal{R}^*_\phi = \mathcal{R}_\phi(f^*_\phi)$, then we can see that: $\mathcal{R}_\phi(\hat{f}_n) - \mathcal{R}^*_\phi \leq \lambda_n ||\hat{f}_n||^2_k + \mathcal{R}_\phi(\hat{f}_n) - \mathcal{R}^*_\phi = LHS$.

In the following context, let $\widehat{f}_n = f_{D,\lambda_n}$, and $\mathcal{R}_{L,P}$ corresponds to $\mathcal{R}_\phi$ in our problem. To bound the LHS, we rely on the theorem proved by Steinwart and Christmann (2008), which is displayed as follows:

Theorem 7.23 (Oracle inequality for SVMs, Steinwart and Christmann 2008

Let $L : X \times Y \times \mathbb{R} \to [0, \infty)$ be a loss function. Also let H be a separable RKHS of a measurable kernel over X and P be a distribution on $X \times Y$. If the following conditions are satisfied: (A1) L is a locally Lipschitz continuous loss that can be clipped at $M > 0$. (A2) L satisfies the superemum bound $L(x, y, t) \leq B$ for a $B > 0$. (A3) The variance bound $\mathbb{E}_P(L\circ\tilde{f} - L\circ f^*_{L,P})^2 \leq V \cdot (\mathbb{E}_P(L\circ\tilde{f} - L\circ f^*_{L,P}))^\upsilon$ is satisfied for constants $\upsilon \in [0, 1]$, $V \geq B^{2-\upsilon}$, and all $f \in H$. (A4) For fixed $n \geq 1$, there exist constants $p \in (0, 1)$ and $a \geq B$ such that the entropy number $\mathbb{E}_{D_X \sim P^n_X} \mathfrak{e}_i(id : H \to L_2(D_X)) \leq a i^{-\frac{1}{2p}}$, $i \geq 1$. (A5) Fix an $f_0 \in H$ and a constant $B_0 \geq B$ such that $Lf_0 \leq B_0$.

Then, for all fixed $\tau > 0$ and $\lambda_n > 0$, the SVM using H and L satisfies:

$$\lambda_n||f_{D,\lambda_n}||_H^2 + \mathcal{R}_{L,P}(\tilde{f}_{D,\lambda_n}) - \mathcal{R}^*_{L,P} \le 9\big(\lambda_n||f_0||_H^2 + \mathcal{R}_{L,P}(f_0) - \mathcal{R}^*_{L,P}\big)$$
$$+ K_0\big(\frac{a^{2p}}{\lambda_n^p n}\big)^{\frac{1}{2-p-\upsilon+\upsilon p}} + 3\big(\frac{72V\tau}{n}\big)^{\frac{1}{2-\upsilon}} + \frac{15B_0\tau}{n}$$

with probability P^n not less than $1 - 3e^{-\tau}$, where $K_0 \ge 1$ is a constant only depending on p, M, B, υ, and V.

We will verify the conditions (A1) - (A5) as follows:

Note that $\tilde{f}$ is a clipped version of f (Winsorization) for some value M, such that $\tilde{t} = I(|t| \le M)t + I(|t| > M)\text{sign}(t)M$. It is easy to check that risks of $L_\phi(.)$ loss satisfy that $\mathcal{R}(\tilde{f}) \le \mathcal{R}(f)$, if we set M to be some large value, i.e. larger than the range of the dose. Hence, $L_\phi(.)$ can be clipped. That implies we can investigate the clipped version of loss instead of the origin loss function without loss of generality (Steinwart and Christmann 2008). The loss function we use is Lipschitz continuous with Lipschitz constant equal to $1/\phi_n$ hence it is also locally Lipschitz continuous. As a result, condition (A1) is satisfied.

For our problem, it is reasonable to assume the rewards are bounded such that $R \in [0, B]$, where B is some constant. Hence, we have $L(x, y, t) \le B$ for (A2). Furthermore, (A3) is true since $\mathbb{E}_P(L \circ \tilde{f} - L \circ f^*_{L,P})^2 \le 2\mathbb{E}_P[(L \circ \tilde{f})^2 + (L \circ f^*_{L,P})^2] \le 4B^2$. The benign kernel we implement in the algorithm is the Gaussian kernel. By theorem 7.34 of Steinwart and Christmann (2008), we have for the constant a is equal to $c_{\epsilon,p}\gamma_n^{-\frac{(1-p)(1+\epsilon)d}{2p}}$ for (A4). By far, all conditions needed for Theorem 7.23 are satisfied, hence plug in $a = c_{\epsilon,p}\gamma_n^{-\frac{(1-p)(1+\epsilon)d}{2p}}$, $\upsilon = 0$, $V = 4B^2$, and $B_0 = B$, we have

$$LHS \le 9A(\lambda_n) + K[\frac{1}{\gamma_n^{(1-p)(1+\epsilon)d}\lambda_n^p n}]^{\frac{1}{2-p}} + 36\sqrt{2}B(\tau/n)^{\frac{1}{2}} + 15B(\tau/n), \qquad (5.2)$$

where $A(\lambda_n) = \lambda_n||f_0||^2_{H_{\gamma_n}} + \mathcal{R}_{L,P}(f_0) - \mathcal{R}^*_{L,P}$.

The next step is to bound the approximation error $A(\lambda_n)$. Since any $f_0 \in H_\gamma$ is valid, we discuss a specific choice of f_0 and the corresponding bound for $A(\lambda_n)$. To construct this f_0, we define that for $r \in \mathbb{N}$ and $\gamma > 0$, the function $K : \mathbb{R}^d \to \mathbb{R}$ Let us assume that there exists a function $f^*_{L,P}$ is a Bayes decision function, i.e. and such that $f^*_{L,P} \in L_2(\mathbb{R}^d) \bigcap L_\infty(\mathbb{R}^d)$. Then we define f_0 by convolving K with this Bayes decision function, that is

$$f_0 := K * f^*_{L,P} = \int_{\mathbb{R}^d} K(X-t) f^*_{L,P}(t)dt, \ x \in \mathbb{R}^d. \tag{5.3}$$

With the help of two theorems in Eberts and Steinwart (2013), we can show that f_0 is contained in H_γ, and it is a suitable function to bound $A(\lambda_n)$.

By the construction of f_0, the approximation error for our problem can be written as:

$$A(\lambda_n) = \lambda_n ||f_0||^2_{H_{\gamma_n}} + \mathcal{R}_{L,P}(f_0) - \mathcal{R}^*_{L,P} = \lambda_n ||K * f^*_{L,P}||^2_{H_{\gamma_n}} + \mathcal{R}_{L,P}(K * f^*_{L,P}) - \mathcal{R}^*_{L,P}$$

By theorem 2.3 of Eberts and Steinwart (2013), we have:

$$A(\lambda_n) \leq \lambda_n (\gamma_n \sqrt{\pi})^{-d} (2^r - 1)^2 ||f||^2_{L_2(\mathbb{R}^d)} + \mathcal{R}_{L,P}(K * f^*_{L,P}) - \mathcal{R}^*_{L,P}$$

By the Lipschitz continuity property of the loss function L_ϕ:

$$A(\lambda_n) \leq \lambda_n (\gamma_n \sqrt{\pi})^{-d} (2^r - 1)^2 ||f||^2_{L_2(\mathbb{R}^d)} + \frac{B}{\phi_n} |K * f^*_{L,P} - f^*_{L,P}|_{L_1(P_X)}$$

By theorem 2.2 of Eberts and Steinwart (2013) with $q = 1$:

$$A(\lambda_n) \leq \lambda_n (\gamma_n \sqrt{\pi})^{-d} (2^r - 1)^2 ||f||^2_{L_2(\mathbb{R}^d)} + \frac{B}{\phi_n} C_{r,1} ||g|| L_p(P_X) \omega_{r, L_1(\mathbb{R}^d)}(f^*_{L,P}, \gamma_n/2) \tag{5.4}$$

If we further assume $f^*_{L,P} \in B^\alpha_{1,\infty}(\mathbb{R}^d)$, a Besov space, i.e. $B^\alpha_{1,\infty}(\mathbb{R}^d) = \{f \in L_\infty((\mathbb{R}^d)) : sup_{t>0}(t^{-\alpha}\omega_{r,L_1((\mathbb{R}^d))}(f,t)) < \infty\}$. Then $\omega_{r,L_1(\mathbb{R}^d)}(f^*_{L,P}, \gamma_n/2) < c_0\gamma_n^\alpha$, c_0 is a constant. Plug it into the inequality 5.4 and merge the constant, we have:

$$A(\lambda_n) \leq c_1\lambda_n\gamma_n^{-d} + c_2\gamma_n^\alpha\phi_n^{-1} \tag{5.5}$$

Combine Equation (5.1), Equation (5.2) and Equation (5.5), we get the theorem proved.

A.2 Theoretical Results with other losses in Chapter 4

The following lemma proves that the theoretical minimizer for the problem with absolute deviation loss (a special case for ϵ-insensitive loss) is not consistent, i.e. not the same function that maximizes $\mathcal{V}(f)$.

Lemma 1

Let $f_{abs}(X) = \arg\min_f E(\frac{R|A-f(X)|}{p(A|X)})$*, then* $f_{abs}(X) \neq f_{opt}(X)$*.*

Proof: By definition: $E(\frac{R|A-f(X)|}{p(A|X)}) = \int E(R|a,x)|a - f(x)|p(x)dadx$. Let $\tilde{p}(a|x) \propto E(R|a,x)p(x)$, then for any given x, $f_{abs}(x)$ is the median of a with respect to density $\tilde{p}(a|x)$. On the other side, we have proved that $V(f) = E(R|A = f) = E_X[E(R|A = f(x), x)] = \int E(R|A = f(x), x)p(x)dx$. That implies, for any given x, $f_{opt}(x)$ is the mode of a with respect to density $\tilde{p}(a|x)$. If the $\tilde{p}(a|x)$ is not symmetric, then $f_{abs}(x) \neq f_{opt}(x)$. As a result, $f_{abs}(X) \neq f_{opt}(X)$ almost surely. Hence, it is not proper to use the absolute deviance loss in our O-learning for dose finding. In addition, it demonstrates that the trivial extension of Zhao et al. (2012) (weighted supporting vector regression) does not work for the dose finding. By the similar argument, we can show that the quadratic loss function also does not work. This follows by the argument that $f_{abs}(X) = \arg\min_f E(\frac{R(A-f(X))^2}{p(A|X)})$, and for given x, $f_{abs}(x)$ is the mean of a with respect to density

$\tilde{p}(a|x)$.

A.3 Other theorems for Proving Theorem 4.4.2

For completeness, we includes the theoretical results used for the proof of Theorem 4.4.2 as follows:

Theorem 7.34 (Steinwart and Christmann 2008)

Let μ be a distribution on $\mathbb{R}^d$ having tail exponent $\tau \in (0, \infty]$. Then, for all $\epsilon > 0$ and $d/(d + \tau) < p < 1$, there exists a constant $c_{\epsilon,p} \geq 1$ such that

$$\mathfrak{e}_i(id : H_\gamma(\mathbb{R}^d) \to L_2(\mu)) \leq c_{\epsilon,p}\gamma^{-\frac{(1-p)(1+\epsilon)d}{2p}} i^{-\frac{1}{2p}}.$$

for all $i \geq 1$ and $\gamma \in (0, 1]$.

Theorem 2.2 (Eberts and Steinwart 2013)

Let us fix some $q \in [1, \infty)$. Furthermore, assume that P_X is a distribution on $\mathbb{R}^d$ that has a Lebesgue density $g \in L_p(\mathbb{R}^d)$ for some $p \in [1, \infty)$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be such that $f \in L_p(\mathbb{R}^d) \bigcap L_\infty(\mathbb{R}^d)$. Then, for $r \in \mathbb{N}$, $\gamma > 0$, and $s \geq 1$ with $1 = 1/s + 1/p$, we have

$$||K * f - f||^q_{L_q(P_X)} \leq C_{r,q}||g||L_p(P_X)\omega^q_{r,L_{q^s}(\mathbb{R}^d)}(f, \gamma/2),$$

where $C_{r,q}$ is a constant only depending on r and q and $\omega^q_{r,L_{q^s}(\mathbb{R}^d)}(f, \gamma/2)$ is the r-th modulus of smoothness of f (see Definition 2.1 in Eberts and Steinwart (2013) for detailed definition).

Theorem 2.3 (Eberts and Steinwart 2013)

Let $f \in L_2(\mathbb{R}^d)$, H_γ be the RKHS of the Gaussian RBF kernel k_γ over $X \subset \mathbb{R}^d$ with $\gamma > 0$ and $K :$ be the same as in Equation 5.3 for a fixed $r \in \mathbb{N}$. Then we have $K * f \in H_\gamma$ with

$$||K * f(x)||_{H_\gamma} \leq (\gamma\sqrt{\pi})^{-\frac{d}{2}}(2^r - 1)||f||_{L_2(\mathbb{R}^d)}$$

Moreover, if $f \in L_\infty(\mathbb{R}^d)$, we have

$$|K * f(x)| \leq (2^r - 1)||f||_{L_\infty(\mathbb{R}^d)}$$

REFERENCE

Allen, G. A., Grosenick, L., and J., T. (2014), "A Generalized Least-Square Matrix Decomposition," *Journal of the American Statistical Association*, 109, 145–159.

Allen, G. I., and Maletic-Savatic, M. (2011), "Sparse non-negative generalized PCA with applications to metabolomics," *Bioinformatics*, 27, 3029–3035.

An, T. H., and Tao, P. D. (1997), "Solving a Class of Linearly Constrained Indefinite Quadratic Problems by D . C . Algorithms," *Journal of Global Optimization*, 11, 253–285.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J., and Meyerson, M. (2001), "Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses.," *Proceedings of the National Academy of Sciences of the United States of America*, 98, 13790–13795.

Boyd, S., and Vandenberghe, L. (2004), *Convex Optimization*, Vol. 25 Cambridge University Press.

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

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984), *Classification and Regression Trees*, Vol. 19 CRC Press.

Busygin, S., Jacobsen, G., and Kramer, E. (2002), "Double Conjugated Clustering Applied to Leukemia Microarray Data," *Proc 2nd SIAM ICDM Workshop on clustering high dimensional data*, .

Cai, T., Tian, L., Uno, H., Solomon, S. D., and Wei, L. J. (2010), "Calibrating parametric subject-specific risk estimation.," *Biometrika*, 97(2), 389–404.

Carlstein, E., Müller, H., and Siegmund, D. (1994), *Change-Point Problems*, number v. 23 Institute of Mathematical Statistics.

Chen, G., Sullivan, P. F., and Kosorok, M. R. (2013), "Biclustering with heterogeneous variance," *Proc Natl Acad Sci U S A*, 110, 12253–12258.

Cheng, Y., and Church, G. M. (2000), "Biclustering of expression data.," *Proceedings of International Conference on Intelligent Systems for Molecular Biology*, 8, 93–103.

Chevret, S. (2006), *Statistical Methods for Dose Finding Experiments.* John Wiley-Sons, New York.

Eberts, M., and Steinwart, I. (2013), “Optimal regression rates for SVMs using Gaussian kernels,” *Electronic Journal of Statistics*, 7, 1–42.

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.

Fisher, R. (1936), “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, 7(2), 179–188.

Fraley, C., and Raftery, A. (2002), “Model-Based Clustering, Discriminant Analysis, and Density Estimation,”.

Friedman, J., Hastie, T., and Tibshirani, R. (2000), “Additive logistic regression: a statistical view of boosting,” *Annals of Statistics*, 28(2), 337–407.

Friedman, J., Hastie, T., and Tibshirani, R. (2010), “Regularization Paths for Generalized Linear Models via Coordinate Descent.,” *Journal Of Statistical Software*, 33(1), 1–22.

Hansen, K. D., Timp, W., Bravo, H. C., Sabunciyan, S., Langmead, B., McDonald, O. G., Wen, B., Wu, H., Liu, Y., Diep, D., Briem, E., Zhang, K., Irizarry, R. A., and Feinberg, A. P. (2011), “Increased methylation variation in epigenetic domains across cancer types.,” *Nature genetics*, 43, 768–775.

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning*, Vol. 1 Springer New York.

Hoff, P. (2010), “Model Averaging and Dimension Selection for the Singular Value Decomposition,” *Journal of the American Statistical Association*, 102, 674–685.

Hu, Y.-H., Wu, F., Lo, C.-L., and Tai, C.-T. (2012), “Predicting warfarin dosage from clinical data: a supervised learning approach.,” *Artificial intelligence in medicine*, 56(1), 27–34.

Huang, D., Sherman, B., and Lempicki, R. (2009), “Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources.,” *Nature protocols*, 4(1), 44–57.

Huang, H., Liu, Y., and Marron, J. (2012), “Bidirectional Discrimination with Application to Data Visualization,” *Biometrika*, 99(4), 851–864.

Hunter, D., and Lange, K. (2004), “A Tutorial on MM Algorithms,” *American Statistician*, 58(1), 30–37.

Imai, K., and Van Dyk, D. A. (2004), “Causal Inference with General Treatment Treatment Regimes: Generalizing the Propensity Score,” *Journal of the American Statistical Association*, 99(467), 854–866.

Irizarry, R. A., Ladd-Acosta, C., Wen, B., Wu, Z., Montano, C., Onyango, P., Cui, H., Gabo, K., Rongione, M., Webster, M., Ji, H., Potash, J. B., Sabunciyan, S., and Feinberg, A. P. (2009), "The human colon cancer methylome shows similar hypo- and hypermethylation at conserved tissue-specific CpG island shores.," *Nature genetics*, 41, 178–186.

Johnstone, I. M., and Lu, A. Y. (2009), "On Consistency and Sparsity for Principal Components Analysis in High Dimensions," *Journal of the American Statistical Association*, 104, 682–693.

Jordan, M., and Jacobs, R. (1994), "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation*, 6(2), 181–214.

Kang, C. (2011), "New Statistical Learning Methods for Chemical Toxicity Data Analysis.," *PhD thesis, Univerty of North Carolina at Chapel Hill*, .

Kim, H., and Loh, W. (2001), "Classification Trees With Unbiased Multiway Splits," *Journal of the American Statistical Association*, 96(454), 589–604.

Kimeldorf, G., and Wahba, G. (1971), "Some results on Tchebycheffian spline function-s," *Journal of Mathematical Analysis and Applications*, 33(1), 82–95.

Klein, T. E., Altman, R. B., Eriksson, N., Gage, B. F., Kimmel, S. E., Lee, M.-T. M., Limdi, N. A., Page, D., Roden, D. M., Wagner, M. J., Caldwell, M. D., and Johnson, J. A. (2009), "Estimation of the warfarin dose with clinical and pharmacogenetic data.," *The New England Journal of Medicine*, 360(8), 753–764.

Kriegel, H.-P., Kröger, P., and Zimek, A. (2009), "Clustering high-dimensional data," *ACM Transactions on Knowledge Discovery from Data*, 3(1), 1–58.

Laber, E. B., Lizotte, D. J., and Ferguson, B. (2014), " Set-valued dynamic treatment regimes for competing outcomes.," *Biometrics*, 70(1), 53–61.

Lazzeroni, L., and Owen, A. (2002), "Plaid models for gene expression data," *Statistica Sinica*, 12(1), 61–86.

Lee, M., Shen, H., Huang, J. Z., and Marron, J. S. (2010), "Biclustering via Sparse Singular Value Decomposition," *Biometrics*, 66, 1087–1095.

Lehmann, B., Bauer, J., Chen, X., Sanders, M., Chakravarthy, A., Shyr, Y., and Pietenpol, J. (2011), "Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies," *J Clin Invest*, 121(7), 2750–2767.

Li, R., Zhong, W., and Zhu, L. (2012), "Feature Screening via Distance Correlation Learning," *Journal of the American Statistical Association*, 107(499), 1129–1139.

Liu, C., Prior, J., Piwnica-Worms, D., and Bu, G. (2010), "LRP6 overexpression defines a class of breast cancer subtype and is a target for therapy.," *Proceedings of the National Academy of Sciences of the United States of America*, 107(11), 5136–5141.

Liu, Y., Hayes, D., Nobel, A., and Marron, J. (2008), "Statistical Significance of Clustering for High-Dimension, Low-Sample Size Data," *Journal of the American Statistical Association*, 103(483), 1281–1293.

Liu, Y., Zhang, H., and Wu, Y. (2011), "Hard or Soft Classification? Large-Margin Unified Machines," *Journal of the American Statistical Association*, 106(493), 166–177.

Loh, W. (2010), "Improving the precision of classification trees," *The Annals of Applied Statistics*, 3(4), 1710–1737.

Ma, Z. (2013), "Sparse principal component analysis and iterative thresholding," *The Annals of Statistics*, 41, 772–801.

Marron, J., Todd, M., and Ahn, J. (2007), "Distance Weighted Discrimination," *Journal of the American Statistical Association*, 102(480), 1267–1271.

Meinshausen, N., and Bühlmann, P. (2010), "Stability selection," *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 72(4), 417–473.

Misra, C., Fan, Y., and Davatzikos, C. (2009), "Baseline and longitudinal patterns of brain atrophy in MCI patients, and their use in prediction of short-term conversion to AD: results from ADNI.," *NeuroImage*, 44(4), 1415–1422.

Moodie, E. M., Platt, R. W., and Kramer, M. S. (2009), "Estimating Response-Maximized Decision Rules With Applications to Breastfeeding," *Journal of the American Statistical Association*, 104(485), 155–165.

Mueller, S., Weiner, M., Thal, L., Petersen, R., Jack, C., Jagust, W., Trojanowski, J., Toga, A., and Beckett, L. (2005), "Ways toward an early diagnosis in Alzheimer's disease: the Alzheimer's Disease Neuroimaging Initiative (ADNI).," *Alzheimer's & dementia*, 1(1), 55–66.

Murphy, S. A. (2003), "Optimal dynamic treatment regimes," *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 65(2), 331–355.

Nocedal, J., and Wright, S. (1999), *Numerical Optimization*, Vol. 43 Springer, New York, NY.

Owen, A. B., and Perry, P. O. (2009), "Bi-cross-validation of the SVD and the nonnegative matrix factorization.," *The Annals of Applied Statistics*, 3, 564–594.

Park, M., and Hastie, T. (2007), "L1-regularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 69(4), 659–677.

Qian, M., and Murphy, S. A. (2011), "Performance guarantees for individualized treatment rules," *Annals of Statistics*, 39(2), 1180–1210.

Rich, B., Moodie, E. E., and Stephens, D. (2013), "Adaptive individualized dosing in pharmacological studies: Generating candidate dynamic dosing strategies for warfarin treatment.," *Technical report*, .

Robins, J. M. (2004), "Optimal structural nested models for optimal sequential decisions," *Proceedings of the Second Seattle Symposium in Biostatistics Analysis of Correlated Data*, .

Rubin, D. B. (1978), "Bayesian Inference for Causal Effects: The Role of Randomization," *The Annals of Statistics*, 6(1), 34–58.

Shabalin, A. A., Weigman, V. J., Perou, C. M., and Nobel, A. B. (2009), "Finding large average submatrices in high dimensional data.," *The Annals of Applied Statistics*, 3, 985–1012.

Shen, H., and Huang, J. Z. (2008), "Sparse principal component analysis via regularized low rank matrix approximation," *Journal of Multivariate Analysis*, 99, 1015–1034.

Shen, X., Pan, W., and Zhu, Y. (2012), "Likelihood-based selection and sharp parameter estimation.," *Journal of the American Statistical Association*, 107(497), 223–232.

Soslow, R. (2008), "Histologic subtypes of ovarian carcinoma: an overview.," *International Journal of Gynecological Pathology*, 27(2), 161–174.

Steinwart, I., and Christmann, A. (2008), *Support vector machines*, Information science and statistics Springer-Verlag New York.

Sutton, R. S., and Barto, A. G. (1998), "Reinforcement Learning: An Introduction," *IEEE Transactions on Neural Networks*, 9(5), 1054–1054.

Székely, G., Rizzo, M., and Bakirov, N. (2008), "Measuring and testing dependence by correlation of distances," *Annals of Statistics*, 35(6), 2769–2794.

Thall, P. F., and Russell, K. E. (1998), "A strategy for dose-finding and safety monitoring based on efficacy and adverse outcomes in phase I/II clinical trials.," *Biometrics*, 54(1), 251–264.

Tibshirani, R. (1994), "Regression selection and shrinkage via the lasso," *Journal of the Royal Statistical Society B*, 58, 267–288.

Turner, H., Bailey, T., and Krzanowski, W. (2005), “Improved biclustering of microarray data demonstrated through systematic performance tests,” *Computational Statistics and Data Analysis*, 48, 235–254.

Van’t Veer, L., and Bernards, R. (2008), “Enabling personalized cancer medicine through analysis of gene-expression patterns.,” *Nature*, 452(7187), 564–570.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Vol. 8 Springer, New York, NY.

Verhaak, R., Hoadley, K., Purdom, E., Wang, V., Qi, Y., Wilkerson, M., Miller, C., Ding, L., Golub, T., Mesirov, J. P., Alexe, G., Lawrence, M., O&apos;Kelly, M., Tamayo, P., Weir, B. A., Gabriel, S., Winckler, W., Gupta, S., Jakkula, L., Feiler, H. S., Hodgson, J. G., James, C. D., Sarkaria, J. N., Brennan, C., Kahn, A., Spellman, P. T., Wilson, R. K., Speed, T. P., Gray, J. W., Meyerson, M., Getz, G., Perou, C. M., and Hayes, D. N. (2010), “Integrated Genomic Analysis Identifies Clinically Relevant Subtypes of Glioblastoma Characterized by Abnormalities in PDGFRA, IDH1, EGFR, and NF1,” *Cancer Cell*, 17(1), 98–110.

Wahba, G. (2002), “Soft and hard classification by reproducing kernel Hilbert space methods,” *Proceedings of the National Academy of Sciences of the United States of America*, 99(26), 16524–16530.

Wang, Y., Nie, J., Yap, P., Shi, F., Guo, L., and Shen, D. (2011), *Robust deformable-surface-based skull-stripping for large-scale studies*, Vol. 6893 Springer Berlin / Heidelberg, Toronto, Canada.

Witten, D. M., and Tibshirani, R. (2010), “A Framework for Feature Selection in Clustering,” *Journal of the American Statistical Association*, 105, 713–726.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009), “A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis,” *Biostatistics*, 10, 515–534.

Wu, Y., and Liu, Y. (2007), “Robust Truncated Hinge Loss Support Vector Machines,” *Journal of the American Statistical Association*, 102(479), 974–983.

Yang, D., Ma, Z., and Buja, A. (2014), “A Sparse SVD Method for High-dimensional Data,” *Journal of Computational and Graphical Statistics: DOI:10.1080/10618600.2013.858632*, .

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.

Zhang, B., Tsiatis, A. A., Davidian, M., Zhang, M., and Laber, E. (2012), “Stat Estimating optimal treatment regimes from a classification perspective Stat,” *Stat*, 1(1), 103–114.

Zhang, B., Tsiatis, A. A., Laber, E. B., and Davidian, M. (2012), "A Robust Method for Estimating Optimal Treatment Regimes.," *Biometrics*, 68(4), 1010–1018.

Zhang, D., Wang, Y., Zhou, L., Yuan, H., and Shen, D. (2011), "Multimodal classification of Alzheimer's disease and mild cognitive impairment.," *NeuroImage*, 55(3), 856–67.

Zhang, T. (2004), "Statistical behavior and consistency of classification methods based on convex risk minimization," *Annals of Statistics*, 32(1), 56–85.

Zhao, Y., Kosorok, M. R., and Zeng, D. (2009), "Reinforcement learning design for cancer clinical trials.," *Statistics in Medicine*, 28(26), 3294–3315.

Zhao, Y., Zeng, D., Rush, J., and Kosorok, M. R. (2012), "Estimating Individualized Treatment Rules Using Outcome Weighted Learning.," *Journal of the American Statistical Association*, 107(449), 1106–1118.

Zhou, H., Li, L., and Zhu, H. (2013), "Tensor Regression with Applications in Neuroimaging Data Analysis," , 108, 540–552.

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.

Zou, H., Hastie, T., and Tibshirani, R. (2006), "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, 15, 265–286.