MACHINE LEARNING TECHNIQUES FOR HETEROGENEOUS DATA SETS

Jingxiang Chen

A dissertation submitted to the faculty at 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
2017

Approved by:

Yufeng Liu

Michael R. Kosorok

Stephen R. Cole

Eric Laber

Donglin Zeng

ABSTRACT

Jingxiang Chen: Machine Learning Techniques for Heterogeneous
Data Sets
(Under the direction of Yufeng Liu and Michael R. Kosorok)

Over the past few decades, machine learning tools are under rapid development in various application fields to support statistical decision making. In this dissertation, we aim at investigating new supervised machine learning techniques which can contribute to analysis of complex datasets.

First, we discuss a new learning method under Reproducing Kernel Hilbert Spaces (RKHS) to achieve variable selection and data extraction simultaneously. In particular, we propose a unified RKHS learning method, namely, DOuble Sparsity Kernel (DOSK) learning, to overcome this challenge. We prove that under certain conditions, our new method can asymptotically achieve variable selection consistency. Numerical study results demonstrate that DOSK is highly competitive among existing approaches for RKHS learning.

Second, we study on how machine learning can be applied to heterogeneous data analysis by detecting an optimal individual treatment rule for the ordinal treatment case. One of the primary goals in precision medicine is to obtain an optimal individual treatment rule (ITR). Recently, outcome weighted learning (OWL) has been proposed to estimate such an optimal ITR in a binary treatment setting by maximizing the expected clinical outcome. However, for the ordinal treatment settings such as dose level finding, it is unclear how to use OWL. We propose a new technique for estimating ITR with ordinal treatments. Simulated examples and an application to a type-2 diabetes study demonstrate the highly competitive performance of the proposed method.

Third, we also focus on analyzing the heterogeneous data but in a different point of view.

In particular, we develop a new exploratory machine learning tool to identify the heterogeneous subpopulations without much prior knowledge. To achieve this goal, we formulate a regression problem with subject specific regression coefficients and use adaptive fusion to cluster the coefficients into subpopulations. This method has two main advantages. First, it relies on little prior knowledge on the underlying subpopulation structure. Second, it makes use of the outcome-predictor relationship and hence can have competitive estimation and prediction accuracy. To estimate the parameters, we design a highly efficient accelerated proximal gradient algorithm. Numerical studies show that the proposed method has competitive estimation and prediction accuracy.

To my parents.

ACKNOWLEDGEMENTS

I would like to express my deepest and sincerest gratitude to my advisors Dr. Yufeng Liu and Dr. Michael R. Kosorok, for guiding, training, and helping me, not only in research but also in many aspects of my life. Their supervisions and encouragements motivated me to explore many problems that I never thought I could handle. Moreover, they demonstrated the most charming side of data science in front me, and that made me determine firmly on what to pursue as my career goal.

I am very grateful to my committee members: Dr. Donglin Zeng, Dr. Eric Laber, and Dr. Stephen R. Cole. Dr. Zeng guided me not only in my thesis project but also in other collaborative works. I benefited a lot from his wise instructions, insightful comments, and broaden knowledge. Dr. Laber impacted and motivated me by the recent development of statistical learning. Dr. Cole offered me numerous detailed suggestions on my thesis which improved the presentation significantly.

I would like to thank my collaborators: Dr. Haoda Fu, Dr. Xuanyao He, Dr. Naim Rashid, Dr. Quoc Tran-Dinh, and Dr. Chong Zhang. In particular, Dr. Zhang helped me overcome some of the most challenge technical problems in my first topic. Dr. Fu and Dr. He shared their data with me and provided many useful analysis ideas. Dr. Tran-Dinh encouraged me explore many computational strategies and techniques that significantly improved the efficiency of the algorithm. I also learned a lot from Dr. Rashid in a very interesting clinical research project.

My gratitude also goes to my supervisors during my graduate research assistantship (GRA) and summer internship: Dr. Bob Rodriguez, Dr. Jianwen Cai, Dr. Neil Hayes, Dr. Dacheng Liu, Dr. Ceib Phillips, Dr. John Preisser, and Dr. Haibo Zhou. My special thank goes to Dr. Bob Rodriguez for offering me great GRA experience in the SAS institute.

I am also deeply grateful for the invaluable friendship in both the Department of Biostatistics and the Department of Statistics and Operation Research. Most importantly, I would like to thank all the members in both Dr. Kosorok and Dr. Liu's research groups, for their valuable suggestions and inspiring discussions. My gratitude also goes to all the friends that I made in UNC-Chapel Hill. In addition, I would like to thank SC Boy for every moment in the past years when they helped me relax from pressure.

Last but not the least, I would like to present special appreciations to my family for their selfless love and support. I would like to thank Shuang Lu, who has stayed by my side and kept encouraging me along my entire journey of the thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Over the past few decades, machine learning tools are under rapid development in various application fields to support statistical decision making. Covering a board range of methods, machine learning contains but is not limited to supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. In particular, when each observation in the data set has at least one response variable, This type of problems is often referred to as supervised learning. Typical examples of supervised learning include regression and classification. On the other hand, when there are no response variables, the goal is often to study the intrinsic pattern of predictors. This group of learning techniques are often called unsupervised learning, which include principal component analysis and clustering, among others, as special cases. When some observations have responses and some do not, it is often named semi-supervised learning. In dynamic systems, sometimes the goal is to train a machine that can determine the ideal behavior within a specific context to maximize its performance according to the feedback from the environment. This group of learning methods is called reinforcement learning. See Hastie et al. (2011), Chapelle et al. (2006), and Sutton and Barto (1998) for a comprehensive review. However, there can still be some cases that remain unclear on how to be categorized. For example, Wei and Kosorok (2013) introduced a new category named latent supervised learning, which aims to identifying the subgroup structure at the cases when the underlying group labels remain unmeasured and can only be induced from a surrogate outcome and predictors.

Overall speaking, many of the machine learning methods aim at solving certain statistical problems by formulating them into machines, each of whose cores includes one or a series of optimization problems. Training such a machine is actually the process of implementing certain numerical algorithm to solve the corresponding optimization problem that depends on certain training datasets. In many cases, such optimization problems consist of two components: the loss term and the penalty term. The loss term is usually used to achieve the goal of fitting, such as the quadratic loss under least squares estimate, the check loss under quintile regression, and the hinge loss under support vector machines. The penalty term is used to pursue alternative learning goals such as the L_1 penalty for variable selection, L_2 penalty or group lasso penalty for grouping effect, and fusion penalty for sparsity of coefficient differences. In addition, many penalty terms in nonparametric models, such as kernel learning and smoothing splines, can be used to prevent overfitting by control the model complexity. In this thesis, the primary focus is to investigate on how machine learning techniques can contribute to analysis of complex datasets, with special interest in heterogeneous datasets.

1.1 Double Sparsity Kernel Learning

Recent advances in technology have enabled scientists to collect massive datasets with high dimensions. For example, in online movie evaluation systems, the data sets can contain rating information from millions of users on thousands of movies. Extracting knowledge from such large data sets poses unprecedented challenges to existing learning techniques. To overcome new difficulties in mining big data sets, in the last few decades, many methodologies have been proposed in the machine learning literature. In the first part, we focus on supervised learning with one response variable. In particular, the learning goal is often to train a function using a training data set, such that for new observations, one can use this function to predict the unobserved responses. See Hastie et al. (2011) for a comprehensive review of supervised learning techniques.

For many applications in supervised learning, appropriate variable selection is very important to the prediction performance of the estimated function. In particular, for real data sets, many predictors do not contain useful information with respect to the response. Hence, these redundant predictors should be excluded when we make further prediction. For instance, in classification problems, Fan and Lv (2008) showed that prediction using all variables may behave similarly to random guessing, due to the noise accumulation. How to perform variable selection has drawn much attention in the literature. Traditional methods for variable selection include forward and backward selections, among others. Recently, model fitting using sparse regularization has become very popular in the learning framework. The corresponding optimization problems of these techniques are equivalent to minimizing objective functions in the *loss* + *penalty* form. The loss term measures the goodness of fit of the estimated function, and the penalty term aims to select important variables in the learning problem, which further controls the complexity of the function space to prevent overfitting.

For different learning tasks, one uses different loss functions. For example, in least squares regression, one uses the squared error loss, and in standard Support Vector Machines (SVM, Boser et al., 1992), we use the hinge loss. For the penalty term, the choice depends on the corresponding functional space. In particular, if the response depends on the predictors linearly, linear learning should be used. Otherwise, one can employ various nonlinear learning methods such as splines (De Boor, 2001) in regression. In this paper, we focus on learning in Reproducing Kernel Hilbert Spaces (RKHS, Aronszajn, 1950; Kimeldorf and Wahba, 1971). This is a very general setting, and many nonlinear learning techniques can be regarded as special cases of RKHS learning. For example, it covers penalized linear regression, additive spline models with or without interactions, and the entire family of smoothing splines. RKHS learning has been extensively used in the literature, and has achieved great successes. See, for example, Schölkopf and Smola (2002), Shawe-Taylor and Cristianini (2004a), and Hastie et al. (2011).

For linear learning, variable selection with sparse regularization has been extensively studied. See, for example, Tibshirani (1996), Fan and Li (2001), Zou and Hastie (2005), Wu et al. (2009), Zhang (2010), Fan and Lv (2010), and the references therein. For RKHS learning, however, the problem of variable selection has received much less attention. In the literature, Guyon et al. (2002) suggested an extension of variable selection from linear learning to kernel learning using the Recursive Feature Elimination (RFE) approach. Lin and Zhang

(2006) developed the Component Selection and Smoothing (COSSO), and proposed to use the sum of component norms as the sparse penalty, instead of the squared norm penalty in standard RKHS learning. Zhang et al. (2011) proposed a structure selection method that can automatically determine whether the signal for one predictor is linear or nonlinear. Recently, Allen (2012) developed an interesting framework of variable selection in RKHS learning. In particular, Allen (2012) imposed a weight on each predictor, and proposed to train the model with a sparse penalty on the weight vector. When a fitted weight is zero, the corresponding predictor is regarded as unimportant in the learning problem, and is removed from further analysis. Allen (2012) provided the Kernel Iterative Feature Extraction (KNIFE) algorithm to solve the corresponding optimization.

Despite the current progress in variable selection for RKHS learning, many challenges remain. First, theoretical properties of sparse penalties in linear learning have been well studied in the literature. For example, Fan and Li (2001) and Zou (2006) proved the oracle property of their proposed methods, and Zhao and Yu (2006) showed selection consistency for LASSO problems. In contrast, theoretical properties of existing variable selection approaches for RKHS learning are much less developed. In particular, it is desirable to explore conditions under which one can have consistency for kernel variable selection. Moreover, Allen (2012) proposed to use the standard squared norm penalty on the learning function to avoid overfitting, besides the sparse penalty on the variable weight vector. However, as Zhang et al. (2015) pointed out, this approach uses all observations to represent the fitted function. This can lead to suboptimal prediction performance as the underlying function can be well approximated by a data sparse representation in the dual space (see Zhang et al., 2015, and Section 2.2.2 for more details). Therefore, it can be beneficial to have a regularization method that can automatically select data points for RKHS learning. To circumvent this difficulty, Zhang et al. (2015) proposed a data sparsity constraint for data extraction. However, Zhang et al. (2015) did not consider the problem of kernel variable selection, and the data sparsity method can have suboptimal performance when there are noisy covariates. Therefore, it is desirable to design a new method that can perform variable selection and data extraction simultaneously. Motivated by the two dimensions of redundancy, Chapter 2 of this thesis aims to develop a machine learning technique, named double sparsity kernel learning, that can perform both variable selection and data extraction simultaneously.

1.2 Generalized Outcome Weighted Learning

In clinical research, precision medicine is a medical paradigm that promotes personalized health care to individual patients. Its recent development originates from the fact that treatment effects can vary widely from subject to subject due to individual level heterogeneity. For example, Ellsworth et al. (2010) found that women whose CYP2D6 gene has a certain mutation state are not able to metabolize Tamoxifen efficiently, and this makes them an improper target group for this therapy. In this way, one of the primary goals for precision medicine is to establish rules so that patients level characteristics can be used directly to find optimal treatments (Mancinelli et al., 2000). Recent literature indicates that statistical machine learning tools can be useful in building such rules. However, the primary focus has been on the binary treatment case, and the ordinal setting has not been fully explored. Ordinal treatments are commonly seen in practice. For example, some drugs for the same disease can be ranked by their medicinal strengths and multiple doses of the same treatment can be ranked by the dose level. However, the dose-response relationship is usually discussed from a population perspective in practice (Robins et al., 2008). In precision medicine, it is desirable to pursue the dose level that is best suited for each individual patient. In this paper, we develop a statistical learning model which can properly handle optimal treatment detection for both binary and ordinal treatment scenarios.

Various quantitative methods have been proposed in the statistical learning literature to estimate ITRs. For example, one group of methods aims to construct interpretable results by using tree-based methods to explore heterogeneous treatment effects (Su et al., 2009; Laber and Zhao, 2015). Another group of methods focuses on establishing a scoring system to evaluate patients' benefits from certain treatments (Zhao et al., 2013). However, these two groups of methods do not propose any optimization function from which the optimal treatment solution can be found. As an alternative, Qian and Murphy (2011) proposed a value function of the average reward that patients receive from their assigned treatments so that the rule discovery process is transformed into an optimization problem. Zhang et al. (2012) developed inverse probability of treatment weights to robustly estimate such value functions, and Zhao et al. (2012) proposed outcome weighted learning (OWL) to transform the rule detection problem into a weighted classification problem. In particular, the OWL approach uses a hinge loss function to replace the original 0-1 loss function in Qian and Murphy (2011), and thus the corresponding computation becomes feasible. Recently, Chen et al. (2016) adjusted OWL to continuous dose cases to find the best dose.

Although Zhao et al. (2012) proposed a smart idea on how the ITR can be estimated, there are still some challenges in practice. The first challenge is that OWL's ITR estimate might be suboptimal when some patient rewards are less than zero. In this setting, a global minimization of the loss function cannot be guaranteed since the objective function is no longer convex. If one chooses to manually shift all of the rewards to be positive, the estimated ITR tends to retain what is actually assigned (Zhou et al. (2017)). This phenomenon can become more severe when the sample size is small and the covariate dimension is large. To alleviate this problem, Zhou et al. (2017) recently proposed residual weighted learning. However, their resulting object function is non-convex, and consequently, global minimization is still not guaranteed.

When we have multiple ordinal treatments, it would be useful to extend the objective function of OWL to solve the ITR estimation problem. Under this case, direct extensions of binary OWL may not work well because it ignores how different the actual assigned treatment is from the optimal treatment. This can lead to information loss. An ordinal treatment, a categorical treatment with a defined order to its categories, can be different from nominal treatment and continuous dose in precision medicine. On one hand, an ordinal treatment can give more restrictions on treatment effect estimate when compared with nominal treatments; on the other hand, it is not appropriate to simply consider an ordinal treatment as a continuous variable because the labels do not contain information about difference scales between each two treatment levels. In that case, the discussion remains valuable that how to extend the objective function of OWL to solve the ITR estimation problem for ordinal treatments. Such an extension is non trivial in practice. This is because the objective function of standard OWL maximizes the average reward by adjusting only the observations where the optimal treatment is identical to the actually assigned treatment. In other words, it ignores how different the actual assigned treatment is from the optimal treatment, which leads to information loss. Several methods have been proposed to consider such differences among treatments. In the literature of standard ordinal classification, one idea in statistical learning is the data duplication strategy introduced by Cardoso and Pinto da Costa (2007). This strategy borrows the idea from proportional odds cumulative logistic regression, which restricts the estimated boundaries not to cross with each other. Furthermore, the ordinal response is relabeled as a binary variable and duplicated in the covariate data to generate a higher dimensional sample space. Then, an all-at-once model is fitted in the transformed sample space to produce a corresponding ranking rule for the original response. Although such data duplication methods are shown to be effective in solving complex ordinal classification problems, it remains unclear how this idea can be utilized in OWL to help find the optimal ITR among multiple ordinal treatments. In the second part of the thesis, we propose a new method called generalized outcome weighted learning (GOWL), which aims to fill the gap of how to use OWL for ordinal treatments.

1.3 Latent Supervised Clustering

Except for estimating the ITR, another goal of precision medicine is to identify the heterogeneity in the population and achieve subpopulation detection. Recently, various machine learning methods have been introduced and applied to investigate on this problem. In supervised learning field, linear regressions with two-way interactions between predictors are widely used but are restricted to certain parametric assumptions, such as that the underlying heterogeneity is determined by those interactions (Greenland (2009)). Besides, some nonparametric methods such as random forest are also popular in literature while the results remain less interpretable in practice (Wager and Athey (2015)). In addition, there are also numerous studies on unsupervised learning field in which clustering analysis can be a good representative. Clustering analysis is commonly used to detect the similarity of features that can lead to underlying subpopulation structures such as producing the heatmaps for gene expression results (Perou et al. (2000)). Some traditional clustering methods, such as hierarchical clustering, enjoy the benefit of weak parametric assumptions on the features and also does not require the number of clusters to be specified ahead of time. Recently, Guo et al. (2010) , Hocking et al. (2011) and Chi and Lange (2015) suggested a new clustering method named convex clustering to formulate clustering as a convex optimization problem via pairwise fusion penalty. The algorithm they proposed tremendously boosted the efficiency of the clustering process especially for large data sets. However, such unsupervised machine learning tools can produce meaningful and desirable results only when the subpopulations are determined by the features alone. In practice, the subpopulation identification can also heavily depend on the outcomes or even the relationship between outcomes and predictors, as many examples show in precision medicine that aim to detect the targeted subpopulations of certain drugs.

Other than supervised learning and unsupervised learning, Wei and Kosorok (2013) recently introduced a new category of machine learning methods named latent supervised learning to relax the parametric assumptions while keep interpretability in the supervised learning tools as mentioned previously. This category of methods assumes that each observation corresponds to an unobserved index label, i.e. the latent outcome, which identifies the subpopulation that it belongs to and also determines the underlying distribution of the observed outcome when adjusting for predictors. In particular, Wei and Kosorok (2013) further assumes that the distribution of the observed outcome follows a mixture Gaussian distribution with two latent components. Furthermore, these latent groups are determined by a linear combination of observed features, such as gene expression information. In this way, studying such a linear combination can help divide the entire population into subpopulations that correspond to different treatment effects. There are also some extensions on the original latent supervised learning idea. For example, Altstein and Li (2013) applied this idea to the time-to-event response, and Shen and He (2015) suggested a logistic-normal mixture model rather than the Gaussian model for a better performance. These methods all showed competitive performance to detect the underlying subpopulation boundaries.

However, some drawbacks still exist for latent supervised learning. First, most of such methods still require certain parametric assumptions for the underlying subpopulation boundaries and can only deal with the cases when the number of latent subpopulations is known. In exploratory studies, it can be very common that such latent subpopulation information is hard to induce from the observed data directly. Second, all the latent supervised learning methods so far rely on certain distribution assumptions on the outcomes as well. Such assumptions can be too strong in practice especially at the time of complex heterogeneous structures. In the third part of the thesis, we focus on similar datasets with unobserved subpopulation label as latent supervised learning but plan to address these two drawbacks simultaneously. In particular, we would like to propose a novel exploratory tool, named latent supervised clustering, to estimate the heterogeneous effects at the same time of clustering the samples into subpopulations without much prior knowledge on the underlying boundaries.

1.4 Outline of Thesis

The outline of the thesis is as follows. In Chapter 2, we discuss a new learning method to achieve variable selection and data extraction simultaneously. In Chapter 3, we study on how machine learning can be applied to heterogeneous data analysis in the first direction discussed. In particular, we are interested in detecting an optimal individual treatment rule for the ordinal treatment case. In Chapter 4, we follow the second direction and propose a new machine learning method, named latent supervised clustering, whose goal is to identify the heterogeneous effect of the predictors on outcomes by clustering the subject-specific regression coefficients. In Section 5, we discuss some future work that can be further explored. Technique lemmas and proofs are provided in the appendices.

CHAPTER 2: DOUBLE SPARSITY KERNEL LEARNING

2.1 Introduction

In this chapter, we propose a new DOuble Sparsity Kernel (DOSK) learning method to fill this gap. We provide an efficient algorithm to solve the corresponding optimization problem. Through numerical examples, we show that our DOSK method can often select useful predictors accurately, and the sparsely represented functions can have very good prediction performance. Moreover, under some conditions, we prove that our DOSK method can enjoy many desirable statistical properties, including variable selection consistency.

The rest of the chapter is organized as follows. In Section 2.2, we briefly introduce standard kernel learning methods, and discuss variable selection and data extraction for learning in a RKHS. Then, we propose our DOSK method, and develop our algorithm for the corresponding optimization problem. We establish some theoretical properties of DOSK, such as selection consistency, in Section 2.3. Simulated and real data examples are used to demonstrate the effectiveness of our new method in Section 2.4. We provide some discussions in Section 2.5. All technical proofs are collected in the appendix.

2.2 Methodology

We first give a brief review of standard kernel learning in Section 2.2.1. Then we propose our DOSK method in Section 2.2.2. We discuss how to solve the corresponding optimization problem in Section 2.2.3.

2.2.1 Standard Learning in RKHS

Suppose each observation in the training data set $(\boldsymbol{x}_i, y_i);\ i = 1, \dots, n$ is obtained from a fixed but unknown distribution $P(\boldsymbol{X}, Y)$, where $\boldsymbol{X} \in \mathbb{R}^p$ is a vector of predictors, and Y is the response. The learning goal is to find $f(\cdot)$ based on the training data set, so that for a new observation with only $\boldsymbol{x}$ available, the prediction of Y based on $f(\boldsymbol{x})$ can be accurate.

For example, in regression, one often uses $f(\boldsymbol{x})$ to estimate the response Y, and in binary margin-based classification where $Y \in \{+1, -1\}$, one can let $\text{sign}\{f(\boldsymbol{x})\}$ be the predicted label for $\boldsymbol{x}$. For many learning problems, the goodness of fit of f can be measured by a loss function $L\{Y, f(\boldsymbol{X})\}$. For different learning tasks, one uses different loss functions. For instance, in standard regression problems where the goal is to estimate the conditional mean of Y with given $\boldsymbol{x}$, it is common to use the squared error loss $L\{Y, f(\boldsymbol{X})\} = \{Y - f(\boldsymbol{X})\}^2$. In classification problems, one can use the hinge loss $L\{Y, f(\boldsymbol{X})\} = \{1 - Yf(\boldsymbol{X})\}_+$ for support vector machines (SVM, Boser et al., 1992), and the deviance loss $L\{Y, f(\boldsymbol{X})\} = \log[1 + \exp\{-Yf(\boldsymbol{X})\}]$ for logistic regression (Lin et al., 2000).

The optimization problem of a learning technique typically involves minimizing an objective function in the form of *loss* + *penalty.* In particular, the objective function can be written as

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L\{y_i, f(\boldsymbol{x}_i)\} + \lambda J(f), \tag{2.1}$$

where $\mathcal{F}$ is the function space for learning. Here the penalty term $J(f)$ regularizes $f(\cdot)$ in order to prevent overfitting, and the tuning parameter λ balances $L(\cdot, \cdot)$ and $J(f)$ with the aim to achieve a good prediction performance. The choice of the penalty term varies based on $\mathcal{F}$. For example, in standard linear regression, one often assumes that the conditional mean of Y is a linear function of $\boldsymbol{x}$, and it is common to use $\mathcal{F} = \{f : \ f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\beta} + \beta_0; \ \boldsymbol{\beta} \in \mathbb{R}^p, \beta_0 \in \mathbb{R}\}$. There are many popular choices for $J(f)$ in the linear learning literature. See, for example, Tibshirani (1996), Fan and Li (2001), Zou and Hastie (2005), Zhang (2010), among others. If a linear function cannot estimate the response well, one often considers a nonlinear function space $\mathcal{F}$. In this chapter, we focus on learning in RKHS. For more details about RKHS, we refer the readers to Wahba (1990), Shawe-Taylor and Cristianini (2004a), and the references therein.

For learning in a RKHS $\mathcal{H}$, it is common to use the squared norm penalty $J(f) = \|f\|_{\mathcal{H}}^2$, where $\|f\|_{\mathcal{H}}$ is the norm of f in $\mathcal{H}$. In other words, (2.1) can be written as

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} L\{y_i, f(\boldsymbol{x}_i)\} + \lambda \|f\|_{\mathcal{H}}^2. \tag{2.2}$$

Kimeldorf and Wahba (1971) showed that under mild conditions on L, the estimated function $\hat{f}$ from (2.2) has the form $\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \hat{\alpha}_i K(\boldsymbol{x}_i, \boldsymbol{x})$, where $K(\cdot, \cdot)$ is the kernel function associated with $\mathcal{H}$, $\boldsymbol{x}_i$'s are the observed predictor vectors in the training data set, and α_i's are the parameters to estimate. Moreover, define $\boldsymbol{K}$ to be the gram matrix with the (i, j)th element $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$; $i, j = 1, \ldots, n$, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T$. One can verify that the penalty $\|f\|_{\mathcal{H}}$ in (2.2) can be written as $\hat{\boldsymbol{\alpha}}^T \boldsymbol{K} \hat{\boldsymbol{\alpha}}$. Consequently, (2.2) is equivalent to the following problem,

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} L\{y_i, f(\boldsymbol{x}_i)\} + \lambda \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}.$$

In practice, however, many commonly used kernel spaces, for example the well known Gaussian RKHS, do not include offsets or intercepts (Minh, 2010). This can lead to suboptimal results for some learning problems. For instance, in quantile regression, if one is interested in estimating the $100\tau\%$ quantile of the response with τ close to 0 or 1, a regression function without an intercept can have inferior performance. Therefore, in this chapter, we consider learning in RKHS with intercepts. In particular, in (2.1), we assume that $f = \tilde{f} + b \in \mathcal{H} \oplus \mathbb{R}$, and let $J(f)$ be the squared norm of $\tilde{f}$, where $\tilde{f}$ is the projection of f onto $\mathcal{H}$. The Representer's Theorem (Kimeldorf and Wahba, 1971) shows that under mild conditions, $\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \hat{\alpha}_i K(\boldsymbol{x}_i, \boldsymbol{x}) + \hat{b}$, where b is the intercept term, and $J(\hat{f}) = \hat{\boldsymbol{\alpha}}^T \boldsymbol{K} \hat{\boldsymbol{\alpha}}$. Hence, for standard RKHS learning, the optimization problem (2.2) with an intercept in f can be written as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} L\{y_i, \sum_{j=1}^{n} \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b\} + \lambda \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}. \tag{2.3}$$

2.2.2 Double Sparsity Kernel Learning

Despite the success of standard kernel learning methods, many challenges remain. First, the standard squared norm penalty cannot perform automatic variable selection. When the underlying signal depends only on a small fraction of the predictors (note that the corresponding relationship can be nonlinear), learning with all predictors can lead to overfitting, and consequently unsatisfactory results. In the literature, Zhang et al. (2011) and Allen (2012), among others, proposed different methods for variable section in RKHS learning. In particular, to perform variable selection in kernel learning, Allen (2012) proposed the idea of variable weighted kernel learning as follows. For a weight vector $\mathbf{w} \in \mathbb{R}^p$ and any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^p$, we define the variable weighted kernel function $K_{\mathbf{w}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = K(\mathbf{w} \odot \boldsymbol{x}_1, \mathbf{w} \odot \boldsymbol{x}_2)$, where $\mathbf{w} \odot \boldsymbol{x}$ denotes the element-wise product of vectors. In other words, the jth element of $\mathbf{w}$, w_j, represents the weight of the jth predictor of $\boldsymbol{X}$ in the kernel function. For any positive definite kernel function K, one can verify by Mercer's Theorem that the newly defined variable weighted kernel $K_{\mathbf{w}}(\cdot, \cdot)$ naturally introduces a RKHS over the domain of $\boldsymbol{X}$. For identifiability, we impose the constraint that $w_j \in [0, 1]$ for all j. In the variable weighted kernel function, if $w_j = 0$, then the jth predictor of $\boldsymbol{X}$ has no impact on f or the prediction. Therefore, one can impose an L_1 type penalty on the vector $\mathbf{w}$ to achieve variable selection in RKHS learning. In particular, Allen (2012) proposed KNIFE for learning in a RKHS with variable selection, with the following optimization

$$\min_{\alpha,b,w} \left[\frac{1}{n} \sum_{i=1}^{n} L\Big\{ y_i, \sum_{j=1}^{n} K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_j + b \Big\} + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \boldsymbol{\alpha}^T K_{\mathbf{w}} \boldsymbol{\alpha} \right], \tag{2.4}$$

where λ_1 and λ_2 are tuning parameters, and $\mathbf{w} \in [0, 1]^p$.

To better illustrate the variable weighted kernel function, we consider several commonly used RKHSs as examples. Define x_{ik} to be the kth element of $\boldsymbol{x}_i$. The linear variable weighted kernel is $K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{p} w_k^2 x_{ik} x_{jk}$, the polynomial variable weighted kernel is $K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \{c + \sum_{k=1}^{p} w_k^2 (x_{ik} x_{jk})\}^d$ with $c \in \mathbb{R}$ and $d \in \mathbb{N}$, the Gaussian variable weighted kernel is $K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\{-\gamma \sum_{k=1}^{p}(w_k x_{ik} - w_k x_{jk})^2\}$ with $\gamma \in \mathbb{R}^+$, and the Laplacian variable weighted kernel is $K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \sum_{k=1}^{p} |w_k x_{ik} - w_k x_{jk}|)$ with $\gamma \in \mathbb{R}^+$.

Recently, Zhang et al. (2015) showed that in some cases, using the squared norm penalty $\| \cdot \|_{\mathcal{H}}^2$ for learning in RKHS can lead to suboptimal results. In particular, in a given learning problem, let $f^*(\boldsymbol{x})$ be the minimizer of the conditional expected loss. In other words, $f^*(\boldsymbol{x}) = E[L\{Y, f(\boldsymbol{X})\} \mid \boldsymbol{X} = \boldsymbol{x}]$ for any $\boldsymbol{x}$ (e.g., $f^*(\boldsymbol{x})$ is the conditional mean of $Y(\boldsymbol{x})$ in standard regression). Zhang et al. (2015) observed that if $f^*(\boldsymbol{x})$ can be well approximated by a function with a sparse representation in the RKHS (in other words, $f^*(\cdot)$ can be well approximated by $\sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \cdot) + b$ for only some nonzero α_i), learning with the squared norm penalty can have the potential danger of overfitting. To overcome this difficulty, one can apply an L_1 penalty on the vector $\boldsymbol{\alpha}$ for data selection of the estimated function. For RKHS learning problems, Zhang et al. (2015) proposed the data sparsity constraint with the following optimization

$$\min_{\alpha, b} \left[\frac{1}{n} \sum_{i=1}^{n} L\Big\{ y_i, \sum_{j=1}^{n} K(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_j + b \Big\} + \lambda \|\boldsymbol{\alpha}\|_1 \right], \tag{2.5}$$

where $K(\cdot, \cdot)$ is the standard kernel function and $\|\boldsymbol{\alpha}\|_1 = \sum_{i=1}^{n} |\alpha_i|$. Using the quantile regression as an example, Zhang et al. (2015) showed that, in certain cases, learning with the data sparsity constraint in (2.5) can improve the prediction performance.

Although data extraction was used in Zhang et al. (2015), their method does not consider variable selection. Hence, when there are noise predictors in $\boldsymbol{x}$, the proposed approach can be suboptimal. To our knowledge, not much work has been done on simultaneous data extraction and variable selection in the literature. To fill this gap, we propose our DOuble Sparsity Kernel learning (DOSK) method as follows

$$\min_{\boldsymbol{\alpha}, b, w} \left[\frac{1}{n} \sum_{i=1}^{n} L\Big\{ y_i, \sum_{j=1}^{n} K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_j + b \Big\} + \lambda_1 \|\boldsymbol{\alpha}\|_1 + \lambda_2 \|\mathbf{w}\|_1 + \lambda_3 \boldsymbol{\alpha}^T K_{\mathbf{w}} \boldsymbol{\alpha} \right], \tag{2.6}$$

with $\lambda_i \geq 0;\ i = 1, 2, 3,\ K_{\mathbf{w}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = K(\mathbf{w} \odot \boldsymbol{x}_1, \mathbf{w} \odot \boldsymbol{x}_2)$ as defined earlier with $\mathbf{w} \in [0, 1]^p$.

The framework of our DOSK (2.6) is very general, in the sense that it includes many existing approaches as special cases. In particular, when $\lambda_1 = \lambda_2 = 0$, (2.6) reduces to the standard squared norm penalized kernel learning (2.3). When $\lambda_1 = 0$, (2.6) reduces to the KNIFE approach (2.4) proposed by Allen (2012). If $\lambda_2 = \lambda_3 = 0$, (2.6) becomes the data sparsity learning (2.5) in Zhang et al. (2015). Because DOSK is a general framework of RKHS learning, one can use various approaches to solve the optimization problem (2.6), based on the choice of the loss function $L(\cdot, \cdot)$, $\mathbf{w}$ and λ_l; $l = 1, 2, 3$. For example, in linear kernel learning with $\lambda_2 \neq 0$, one can verify that (2.6) is a biconvex problem with respect to $(\boldsymbol{\alpha}^T, b)^T$ and $\mathbf{w}$, and can be solved by the alternate convex search algorithm (Gorski et al., 2007). For more general DOSK problems, we propose a unified algorithm to solve (2.6) in the Section 2.3.

Note that although we impose multiple penalties in (2.6), our DOSK method can circumvent the difficulty of over-penalization by choosing $(\lambda_1, \lambda_2, \lambda_3)$ carefully. In particular, in Section 2.3, we show that if the tuning parameters are chosen appropriately, our DOSK method can enjoy many desirable theoretical properties.

2.2.3 Computational Algorithm for DOSK

The major difficulty of solving the optimization (2.6) is that even L is convex, the composite loss function $L\Big\{y, \sum_{j=1}^n K_{\mathbf{w}}(\boldsymbol{x}, \boldsymbol{x}_j)\alpha_j + b\Big\}$ may not be convex with respect to $(\mathbf{w}^T, \boldsymbol{\alpha}^T, b)^T$. Consequently, many existing algorithms for convex optimizations (Boyd and Vandenberghe, 2004) cannot be used directly. On the other hand, one can verify that if the loss function L is convex, the optimization (2.6) is convex respect to $(\boldsymbol{\alpha}^T, b)^T$ for a fixed $\mathbf{w}$. Hence, a natural way to circumvent the difficulty of non-convex optimization is to update $\mathbf{w}$ and $(\boldsymbol{\alpha}^T, b)^T$ recursively. This, however, cannot be done directly, as for a general kernel function $K(\cdot, \cdot)$, $L\Big\{y, \sum_{j=1}^n K_{\mathbf{w}}(\boldsymbol{x}, \boldsymbol{x}_j)\alpha_j + b\Big\}$ is not biconvex with respect to $\mathbf{w}$ and $(\boldsymbol{\alpha}^T, b)^T$. One way to tackle this problem is that for fixed $(\boldsymbol{\alpha}^T, b)^T$, we can find a linear approximation of the variable weighted kernel function $K_{\mathbf{w}}$ in a small neighbourhood of $(\mathbf{w}^T, \boldsymbol{\alpha}^T, b)^T$ (Allen, 2012). Thus, to update $\mathbf{w}$, one can employ the linear approximation of $K_{\mathbf{w}}$ to make the corresponding objective function convex. Note that in the literature, the idea of local linear approximation has been widely used to solve optimizations for many learning problems. See, for example, An and Tao (1997), Zou and Li (2008), Lee et al. (2012), among others.

To introduce our algorithm for DOSK, we need some further notation. Let the objective function in (2.6) be $\phi(\boldsymbol{\alpha}, b, \mathbf{w})$. Define an $n \times p$ matrix $A(\mathbf{w})$, whose ith row is $\sum_{j=1}^{n} \alpha_j \nabla_{\mathbf{w}} K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)^T$, and an $n \times n$ matrix $B(\mathbf{w})$ with the (i,j)th element $B(i,j) = K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \nabla K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)^T \mathbf{w}$. Here $\nabla_{\mathbf{w}} K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the gradient vector of $K_{\mathbf{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with respect to $\mathbf{w}$. By Taylor's expansion, one can verify that for $\mathbf{w}_1$ and $\mathbf{w}_2$, we have

$$K_{\mathbf{w}_1} \boldsymbol{\alpha} = A(\mathbf{w}_2)\mathbf{w}_1 + B(\mathbf{w}_2)\boldsymbol{\alpha} + o(\|\mathbf{w}_1 - \mathbf{w}_2\|_2). \tag{2.7}$$

Define $\boldsymbol{c}_{\mathbf{w}_2}(\mathbf{w}_1) = A(\mathbf{w}_2)\mathbf{w}_1 + B(\mathbf{w}_2)\boldsymbol{\alpha}$, which is a linear function of $\mathbf{w}_1$. When $\mathbf{w}_1$ and $\mathbf{w}_2$ are close, we can use $\boldsymbol{c}$ as the local linear approximation of $K_{\mathbf{w}}\boldsymbol{\alpha}$ in our DOSK optimization algorithm. In particular, we outline the general algorithm to solve (2.6) in the algorithm of DOSK below.

Algorithm of DOSK:

```
1. Initialize $\mathbf{w}^{(0)}$, $\boldsymbol{\alpha}^{(0)}$ and $b^{(0)}$ with $w_j \in [0,1]$ for $1 \le j \le p$.
2. The $\boldsymbol{\alpha}$ step: fix $\mathbf{w}^{(t-1)}$ and $b^{(t-1)}$, and find $\boldsymbol{\alpha}^{(t)} = \mathrm{argmin}_{\boldsymbol{\alpha}} \phi(\boldsymbol{\alpha}, b^{(t-1)}, \mathbf{w}^{(t-1)})$.
   The optimization problem is convex, and independent of the $\lambda_2\|\mathbf{w}\|_1$ term in (2.6).
3. The $b$ step: fix $\mathbf{w}^{(t-1)}$ and $\boldsymbol{\alpha}^{(t)}$, and find
   $b^{(t)} = \mathrm{argmin}_b \frac{1}{n}\sum_{i=1}^{n} L\Big\{y_i, \sum_{j=1}^{n} K_{\mathbf{w}^{(t-1)}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_j^{(t)} + b\Big\}$. This is a convex
   optimization with one parameter, and can be solved by standard methods.
4. The $\mathbf{w}$ step: fix $b^{(t)}$ and $\boldsymbol{\alpha}^{(t)}$, and define $\boldsymbol{c}_{\mathbf{w}^{(t-1)}}(\mathbf{w}) = A(\mathbf{w}^{(t-1)})\mathbf{w} + B(\mathbf{w}^{(t-1)})\boldsymbol{\alpha}^{(t)}$.
   Let $\{\boldsymbol{c}_{\mathbf{w}^{(t-1)}}(\mathbf{w})\}_i$ be the $i$th element of $\boldsymbol{c}_{\mathbf{w}^{(t-1)}}(\mathbf{w})$. Under the constraint $\mathbf{w}^{(t)} \in [0,1]^p$,
   find
   $\mathbf{w}^{(t)} = \mathrm{argmin}_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} L[y_i, \{\boldsymbol{c}_{\mathbf{w}^{(t-1)}}(\mathbf{w})\}_i + b^{(t)}] + \lambda_2\|\mathbf{w}\|_1 + \lambda_3 \mathbf{w}^T A(\mathbf{w}^{(t-1)})\boldsymbol{\alpha}^{(t)}$.
   This is a standard quadratic programming problem.
5. Repeat steps 2-4 until convergence.
```

In the $\boldsymbol{\alpha}$ and b steps in the algorithm above, the corresponding objective functions are convex, therefore after updating the parameters, the value of ϕ decreases. On the other hand, in the $\mathbf{w}$ step, we replace the original objective function ϕ by its local linear approximation, and solve a quadratic programming problem. Denote the solution to this quadratic programming problem by $\mathbf{w}^{(QP)}$. In the algorithm of DOSK, the updated $\mathbf{w}^{(t)} = \mathbf{w}^{(QP)}$ can have some distance from $\mathbf{w}^{(t-1)}$, hence the original ϕ function is not guaranteed to decrease. One possible way to overcome this difficulty is that in the $\mathbf{w}$ step, instead of having $\mathbf{w}^{(t)} = \mathbf{w}^{(QP)}$, we can treat $\mathbf{w}^{(QP)} - \mathbf{w}^{(t-1)}$ as a direction in which ϕ tends to decrease, and determine the appropriate step size by conducting a line search. In particular, we present the revised algorithm as below.

Revised Algorithm of DOSK:

```
1. Initialize $\mathbf{w}^{(0)}$, $\boldsymbol{\alpha}^{(0)}$ and $b^{(0)}$ with $w_j \in [0,1]$ for $1 \leq j \leq p$.
2. The $\boldsymbol{\alpha}$ step: fix $\mathbf{w}^{(t-1)}$ and $b^{(t-1)}$, and find $\boldsymbol{\alpha}^{(t)} = \text{argmin}_{\boldsymbol{\alpha}}\, \phi(\boldsymbol{\alpha}, b^{(t-1)}, \mathbf{w}^{(t-1)})$.
   The optimization problem is convex, and independent of the $\lambda_2\|\mathbf{w}\|_1$ term in (2.6).
3. The $b$ step: fix $\mathbf{w}^{(t-1)}$ and $\boldsymbol{\alpha}^{(t)}$, and find
   $b^{(t)} = \text{argmin}_b \sum_{i=1}^n L\Big\{y_i, \sum_{j=1}^n K_{\mathbf{w}^{(t-1)}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_j^{(t)} + b\Big\}$. This is a convex
   optimization with one parameter, and can be solved by standard methods.
4. The $\mathbf{w}$ step: fix $b^{(t)}$ and $\boldsymbol{\alpha}^{(t)}$, and define $\mathbf{w}^{(\text{temp})} = \mathbf{w}^{(t-1)}$.
   (a) Define $\boldsymbol{c}_{\mathbf{w}^{(\text{temp})}}(\mathbf{w}) = A(\mathbf{w}^{(\text{temp})})\mathbf{w} + B(\mathbf{w}^{(\text{temp})})\boldsymbol{\alpha}^{(t)}$. Let $\{\boldsymbol{c}_{\mathbf{w}^{(\text{temp})}}(\mathbf{w})\}_i$ be the $i$th
       element of $\boldsymbol{c}_{\mathbf{w}^{(\text{temp})}}(\mathbf{w})$. Under the constraint $\mathbf{w} \in [0,1]^p$, find
       $\mathbf{w}^{(QP)} = \text{argmin}_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^n L[y_i, \{\boldsymbol{c}_{\mathbf{w}^{(\text{temp})}}(\mathbf{w})\}_i + b^{(t)}] + \lambda_2\|\mathbf{w}\|_1 +$
       $\lambda_3 \mathbf{w}^T A(\mathbf{w}^{(\text{temp})})\boldsymbol{\alpha}^{(t)}$.
   (b) Define $\Delta\mathbf{w} = \mathbf{w}^{(QP)} - \mathbf{w}^{(\text{temp})}$. Find the best step size $s$ by
       $s = \text{argmin}_{u \geq 0}\, \phi(\boldsymbol{\alpha}^{(t)}, b^{(t)}, \mathbf{w}^{(\text{temp})} + u\Delta\mathbf{w})$.
   (c) Set $\mathbf{w}^{(\text{temp})} = \mathbf{w}^{(\text{temp})} + s\Delta\mathbf{w}$.
   (d) Repeat steps (a)-(c) until convergence, and set $\mathbf{w}^{(t)} = \mathbf{w}^{(\text{temp})}$.
5. Repeat steps 2-4 until convergence.
```

In the revised algorithm of DOSK, one can verify that after updating the parameters, the ϕ function value would not increase. This helps to guarantee that we can obtain a stationary point of the objective function using the revised algorithm. In particular, we have the following theorem.

Theorem 2.2.1. *Suppose that the loss function L in (2.6) is a convex and continuously differentiable function, and the variable weighted kernel $K_{\mathbf{w}}$ is a convex or concave and continuously differentiable function of $\mathbf{w}$. Then the solution from the revised algorithm is a stationary point of the objective function.*

Remark 1: Theorem 2.2.1 is valid for many loss functions, e.g., the squared error loss in standard regression, and the deviance loss in logistic regression. For many other loss functions that are not differentiable, such as the hinge loss in SVM, or the check loss function in quantile regression, one can consider an alternative continuous approximation to the loss function. For example, Wang et al. (2007) proposed the hybrid huberized hinge loss for SVM. One can verify that the hybrid huberized loss meets the condition in Theorem 2.2.1, and the corresponding solution is a stationary point. Moreover, for many commonly used kernel functions, the assumptions on $K_{\mathbf{w}}$ in Theorem 2.2.1 are satisfied. For example, one can verify that the variable weighted kernel introduced by the Laplacian RKHS, or by the linear kernel when all elements in $\boldsymbol{x}$ are non-negative, is convex with respect to $\mathbf{w}$.

Remark 2: The revised algorithm of DOSK replaces the quadratic programming step in the first algorithm of DOSK by the descent direction and line search method. This approach is guaranteed to decrease the objective function value at each iteration step, at the cost of a more complex computation. On the other hand, our numerical experience shows that the first algorithm almost always decreases the objective for commonly used kernels and loss functions. Therefore, we use the first algorithm of DOSK in the numerical examples, whereas in each step we check if the objective function decreases. If not, we then employ the line search approach as in the revised algorithm instead.

Remark 3: Since the objective function can be non-convex, it is possible that the numerical solution is just a stationary point, not the global minimum. To increase the chance of finding the optimal solution, we suggest to use multiple different starting points, compare the corresponding results, and choose the fitted model with the smallest objective function value.

2.3 Statistical Learning Theory

In this section, we explore the theoretical properties of the proposed DOSK method. In particular, we first study the convergence rate of the excess risk for various learning problems under certain conditions, and then show that DOSK can enjoy selection consistency for high dimensional learning problems. Moreover, we show that the expected loss using the estimated function $\hat{f}$, $E[L\{y, \hat{f}(\boldsymbol{X})\}]$, can be well approximated by the empirical loss on the training data, in the sense that the corresponding difference converges to zero with a fast convergence rate.

To state our theory, we first introduce some technical assumptions, and provide detailed discussions on why these conditions are needed. We also discuss some cases where these conditions are met. We would like to point out that most of the assumptions in this chapter are mild and reasonable, which can be satisfied or checked for various real applications.

To begin with, we need to present some further notation. Let $\mathbf{w}^* = (\mathbf{w}_{(1)}^T, \mathbf{w}_{(0)}^T)^T$ be the underlying variable weight vector, where elements in $\mathbf{w}_{(1)}$ are non-zero, and elements in $\mathbf{w}_{(0)}$ are zero. In other words, the predictors in $\boldsymbol{x}$ that correspond to $\mathbf{w}_{(0)}$ are noise covariates. Accordingly, one can define $\boldsymbol{x} = (\boldsymbol{x}_{(1)}^T, \boldsymbol{x}_{(0)}^T)^T$, such that predictors in $\boldsymbol{x}_{(1)}$ contain useful information for the learning problem. In this chapter, we focus on the case that the number of useful predictors is finite (i.e., $|\mathbf{w}_{(1)}| < \infty$). Furthermore, with a little abuse of notation, we let $\|f\|_{\mathcal{H}} = \|\tilde{f}\|_{\mathcal{H}}$, where $\tilde{f}$ is the projection of f onto $\mathcal{H}$.

We impose our first assumption on the distribution of $\boldsymbol{X}$ and $\boldsymbol{X}_{(1)}$, where $\boldsymbol{X}$ and $\boldsymbol{X}_{(1)}$ correspond to the p dimension random vector and the vector containing important variables.

Assumption 1: Every element in $\boldsymbol{X}$ ranges in $[0, 1]$. Furthermore, the distribution of $\boldsymbol{X}_{(1)}$ is absolutely continuous with respect to the Lebesgue measure, where the corresponding Radon-Nikodym derivative is bounded away from 0.

In Assumption 1, we restrict our consideration on $\boldsymbol{X} \in [0, 1]^p$. One can verify that our theory can be naturally generalized to the case where the elements in $\boldsymbol{X}$ are uniformly bounded. We defer the discussion on the second part of Assumption 1 until after Assumption 4.

In the next assumption, we impose some constraints on the kernel function $K(\cdot, \cdot)$.

Assumption 2: The kernel function $K(\cdot, \cdot)$ is separable and $\sup K(\cdot, \cdot) < \infty$. Furthermore, the kernel function $K_{\mathbf{w}^*}(\boldsymbol{x}, \cdot)$ is Lipshcitz with respect to $\boldsymbol{x}_{(1)}$, i.e. the useful variables vector, in terms of the L_2 norm.

The first part of Assumption 2 is very mild, and has been frequently used in the literature. See, for example, Steinwart and Scovel (2007a), Blanchard et al. (2008a), Zhang et al. (2015), among others. It suggests that the corresponding RKHS $\mathcal{H}$ is not too complex, in the sense that its diameter would not be infinity. The second part is used to ensure that the best learning function using n observations can converge to the underlying function in a fast rate. See the proof of Lemma A.0.2 for more details. This assumption is valid for many commonly used kernel functions such as the Gaussian kernel and the polynomial kernel.

In Assumption 3, we assume that L can be treated as a univariate function. This is a very mild condition, and is valid for many learning problems. For example, in standard least squares regression, we have $L(u) = u^2$ where $u = (f - y)$, and in logistic regression, $L(u) = \log\{1 + \exp(-u)\}$ where $u = yf$ and $y \in \{+1, -1\}$.

Assumption 3: The loss function $L(u)$ has a second order derivative with $0 < L''(u) < \infty$ for every u.

Assumption 3 is needed to ensure that the expected loss function is strictly convex around the underlying optimal solution. Moreover, the second order differentiability helps to control the convergence rate of the estimated function $\hat{f}$ to the best function. See the discussion of Assumption 5 for more details.

Next, we consider assumptions on the function $f(\boldsymbol{x})$. Recall that the learning goal is to obtain $\hat{f}(\boldsymbol{x})$ from the training data set for good prediction performance. Therefore, we consider the “best" function f_0, in the sense that its corresponding expected loss $E[L\{Y, f_0(\boldsymbol{X})\}]$ is the minimum among all possible $E[L\{Y, f(\boldsymbol{X})\}]$. Consequently, f_0 can have the best prediction performance under mild conditions. For instance, in classification, f_0 can achieve the minimal classification error rate, given that the loss function L is Fisher consistent (Liu, 2007). We will prove that under certain conditions on f_0, the estimated function $\hat{f}$ would converge to f_0 with a desirable convergence rate.

Assumption 4: The underlying function f_0 has a sparse representation in the RKHS. In particular, there exist $\gamma_1, \ldots, \gamma_m$, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$, and b_0 such that $f_0(\boldsymbol{x}) = \sum_{j=1}^{m} \gamma_j K_{\mathbf{w}^*}(\boldsymbol{z}_j, \boldsymbol{x}) + b_0$. Here m is a fixed integer, $\gamma_j \neq 0$, and $\boldsymbol{z}_j \in [0,1]^p$ for $j = 1, \ldots, m$.

As a remark, we note that some RKHSs are very rich, in the sense that many functions can be well approximated by $f \in \mathcal{H}$. For example, Steinwart and Scovel (2007a) proved that all step functions can be approximated by f in the Gaussian RKHS arbitrarily well under mild conditions, and this result can be generalized to the case of continuous functions. However, if f_0 does not have a sparse representation in the RKHS, the function in $\mathcal{H}$ that approximates f_0 well may have an infinite norm. When $\hat{f}$ approaches f_0 as $n \to \infty$, $\|\hat{f}\|_{\mathcal{H}}$ would be unbounded. Consequently, the variation of $\hat{f}$ due to the randomness of the sample can be very large. In the literature, Bartlett et al. (2005), among others, pointed out that large variation of $\hat{f}$ can lead to suboptimal prediction performance. Assumption 4 ensures that the underlying function f_0 has a finite norm in the RKHS. In the proof of Theorem 2.3.1, we show that with an appropriate λ_1, the data selection can provide a sparsely represented function $\hat{f}$ whose norm can be bounded away from infinity. This is crucial to prove the convergence of $\hat{f}$ to f_0, which further leads to the selection consistency of our DOSK method.

The next assumption ensures that in the updating scheme, $\hat{f}$ would converge to the global solution, once we are at a point that is close enough. To state this assumption, we first introduce some further notation. Define $\|\cdot\|_{*,2}$ to be the restricted L_2 norm with respect to the partition of $\mathbf{w}$. In particular, $\|\boldsymbol{x} - \boldsymbol{z}\|_{*,2} = \|\boldsymbol{x}_{(1)} - \boldsymbol{z}_{(1)}\|_2$. For any $n \gg m$, we define $(\boldsymbol{\alpha}_n^*, b_n^*)$ as follows. Notice that the empirical loss function value does not change if we switch the order of the pairs $(\boldsymbol{x}_i, y_i)$ and $(\boldsymbol{x}_j, y_j)$ for $i \neq j$. Hence, without loss of generality, we can assume that $\boldsymbol{x}_j$ is the observation that is closest to $\boldsymbol{z}_j$ in terms of the $\|\cdot\|_{*,2}$ norm among the training data set $\{(\boldsymbol{x}_i, y_i);\ i = 1, \ldots, n\}$, for $j = 1, \ldots, m$. When $n \gg m$, we can assume that each $\boldsymbol{x}_j$ is distinct (in other words, $\boldsymbol{x}_j$ would not be closest to $\boldsymbol{z}_u$ and $\boldsymbol{z}_v$ simultaneously, compared to other observations). Next, define $(\boldsymbol{\alpha}_n^*, b_n^*)$ such that $\boldsymbol{\alpha}_n^* = (\gamma_1, \ldots, \gamma_m, 0, \ldots, 0)^T$ with length n, $b_n^* = b_0$, and let $f_{\boldsymbol{\alpha}_n^*, b_n^*}(\boldsymbol{x}) = \sum_{i=1}^n \alpha_j^* K_{\mathbf{w}^*}(\boldsymbol{x}_i, \boldsymbol{x}) + b_n^*$. The definition of $(\boldsymbol{\alpha}_n^*, b_n^*)$ helps to show that the approximation error of the DOSK method under Assumption 4 converges to 0 very quickly. See the proof of Lemma A.0.4 in the appendix for more discussions.

Before stating Assumption 5, we would like to discuss the second part of Assumption 1, which ensures that with large enough n, the underlying function can be well approximated by the sparsely represented function $f_{\boldsymbol{\alpha}_n^*, b_n^*}(\boldsymbol{x})$ from our training data. In particular, Assumption 1 guarantees that as $n \to \infty$, $f_{\boldsymbol{\alpha}_n^*, b_n^*}(\boldsymbol{x})$ can approach $f_0(\boldsymbol{x})$ with a rate very close to $O_P(n^{-1})$ in terms of the $\|\cdot\|_2$ norm. See Lemma A.0.2 and the corresponding proof for more discussions.

Assumption 5: For any p and $n \gg m$, there exists a neighborhood $\mathcal{N}$ of $\left((\mathbf{w}^*)^T, (\boldsymbol{\alpha}_n^*)^T, b_n^*\right)^T$, such that in $\mathcal{N}$, the expected loss function $E\left[\sum_{i=1}^n L\{Y_i, f(\boldsymbol{X}_i)\}\right]$ is strictly convex with respect to $(\mathbf{w}^T, \boldsymbol{\alpha}^T, b)^T$.

Assumption 5 is necessary for our theory, because if the loss function is not strictly convex, a small perturbation in the training data set can lead to a significant change of $\hat{f}$. See, for example, the discussion on a similar issue for quantile regression using the check loss function in Li and Zhu (2008). Consequently, the convergence rate of $\hat{f}$ to f_0 can be difficult to obtain. To our knowledge, there has been no theoretical result on selection consistency that does not rely on the assumption or fact of local convexity. Notice that Assumption 3 is important to the validity of Assumption 5, because if L is not strictly convex, it is likely that the expected loss function is not convex even if the kernel function is locally convex. For instance, if we use the hinge loss $L(u) = [1-u]_+$ which is piecewise linear, Assumption 5 cannot be satisfied.

Next, we impose constraints on the signal strength in the learning problem. For variables weighted learning, the jth predictor provides useful information if and only if the weight w_j is positive. Variable selection consistency means that $\text{sign}(\hat{w}_j) = \text{sign}(w_j)$ for all j with a high probability, where $\text{sign}(0) = 0$. The next assumption is an important part of sufficient conditions for variable selection consistency.

Assumption 6: For any w_j in $\mathbf{w}_{(1)}$, $\frac{\partial E[L\{Y, f_0(\boldsymbol{X})\}]}{\partial w_j} |_{w_j=0,\ w_i=w_i^*,\ i\neq j} < 0$, and for any w_j in $\mathbf{w}_{(0)}$, $\frac{\partial E[L\{Y, f_0(\boldsymbol{X})\}]}{\partial w_j} |_{w_j=0,\ w_i=w_i^*,\ i\neq j} \geq 0$. Here w_i^* is the ith element of $\mathbf{w}^*$.

In Assumption 6, we measure the signal strength of w_j by its partial derivative with respect to the expected loss function evaluated at $\mathbf{w}^*$ (except the jth weight is at zero). In the literature, there are many existing assumptions on the signal strength that are (essentially) similar to Assumption 6. For example, one can verify that for regular linear regression with the squared error loss, Assumption 6 reduces to that the non-zero coefficients are bounded away from zero. This is analogous to the assumptions considered in Fan and Peng (2004) and Fan and Lv (2010), among others. Furthermore, we require the partial derivative with respect to the noise covariates are non-negative.

In the last assumption, we focus on regression problems, where $Y = f_0(\boldsymbol{X}) + \epsilon(\boldsymbol{X})$ with $\epsilon(\boldsymbol{X})$ being the random error term. Notice that we include both the homoscedastic and the heteroscedastic cases here, as ϵ can have different distributions for different $\boldsymbol{X}$. If the distribution of ϵ has a very heavy tail, there is a large probability that we observe a y_i that is very far away from $f_0(\boldsymbol{x}_i)$. This outlier can lead to a severely biased estimation $\hat{f}$. Assumption 7 aims to control the probability of an extreme y_i, which can help to bound the magnitude of the estimated $\hat{b}$. Recall that if a random variable U is sub-Gaussian with parameter s, then $\mathrm{pr}(|U| > u) \leq 2\exp(-u^2/s)$ for large enough u.

Assumption 7: In a regression problem, the error term $\epsilon(\boldsymbol{X})$ follows a sub-Gaussian distribution with a universal parameter $s < \infty$ for any $\boldsymbol{X}$.

Assumption 7 is very general, as many distributions are sub-Gaussian. For example, in linear regression, we often assume that $\epsilon \sim N(0, \sigma^2)$ with finite σ. This is a homoscedastic case of Assumption 7, and normal random variables are known to be sub-Gaussian. Furthermore, all random variables with bounded ranges are sub-Gaussian, and distributions with small kurtosis are sub-Gaussian.

We are ready to present our main theorems. The first theorem studies the convergence rate of $\hat{f}$ to f_0. Recall that $a \vee b = \max(a, b)$ for $a, b \in \mathbb{R}$.

Theorem 2.3.1. *Suppose Assumptions 1-7 hold, and* $\log(p)/\sqrt{n} \to 0$ *as* $n \to \infty$. *If we choose* $\lambda_1 = O\{\log(n)^{-1}\}$, $\lambda_2 = O[\{\log(p) \vee \log(n)\}/\sqrt{n}]$, *and* $\lambda_3 = o(\lambda_1)$ *in (2.6), we have that the corresponding global solution* $(\hat{\mathbf{w}}^T, \hat{\boldsymbol{\alpha}}^T, \hat{b})^T$ *to (2.6) satisfies that* $\|\hat{f} - f_0\|_2 = O_P\{\log(n)/\sqrt{n}\}$, *where* $\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{n} \hat{\alpha}_j K_{\hat{\mathbf{w}}}(\boldsymbol{x}, \boldsymbol{x}_j) + \hat{b}$.

Theorem 2.3.1 suggests that $\hat{f}$ converges to f_0 at a rate very close to the “parametric rate" $O_P(n^{-1/2})$. Comparing Theorem 2.3.1 with the theoretical results in Zhang et al. (2015), one can see that the multiple penalties in (2.6) do not affect the performance of $\hat{f}$, as long as the corresponding λ's are appropriately selected. This helps to justify that our DOSK method can avoid the issue of over-penalization by carefully choosing the tuning parameters.

Next, we study the selection consistency of our DOSK method. Our results suggest that we can have selection consistency if p is of a polynomial order of n.

Theorem 2.3.2. *Suppose Assumptions 1-7 hold. Furthermore, assume that* $\log(p)/\sqrt{n} \to 0$ *as* $n \to \infty$. *If we choose* $\lambda_1 = O\{\log(n)^{-1}\}$, $\lambda_2 = O[\{\log(p) \vee \log(n)\}/\sqrt{n}]$, *and* $\lambda_3 = o(\lambda_1)$ *in (2.6), we have that the corresponding global solution* $(\hat{\mathbf{w}}^T, \hat{\boldsymbol{\alpha}}^T, \hat{b})^T$ *to (2.6) satisfies that, with probability tending to 1 as* $n \to \infty$, *sign*$(\hat{w}_j) =$ *sign*(w_j^*) *for* $j = 1, \ldots, p$, *where* w_j^* *is the* j*th element of* $\mathbf{w}^*$.

Theorem 2.3.2 shows that our DOSK method can enjoy the desirable asymptotic selection consistency at the global solution. In other words, if the sample size is large, one can often correctly identify the important and unimportant variables in the learning problem. This can help researchers to obtain a better understanding of the relationship between predictors and the response, and provide a more interpretable model for future prediction.

The next theorem studies the prediction performance of the obtained $\hat{f}$. In particular, since one uses the loss function L to measure the goodness of fit of $\hat{f}$, it is desirable to obtain a bound for the expected loss $E[L\{Y, \hat{f}(\boldsymbol{X})\}]$. For example, in regression problems, $E[L\{Y, \hat{f}(\boldsymbol{X})\}]$ indicates the average prediction error using $\hat{f}$. In margin-based classification where the loss function L dominates the $0-1$ loss function (which is further equivalent to the prediction error rate), $E[L\{Y, \hat{f}(\boldsymbol{X})\}]$ can be regarded as an upper bound of the future misclassification rate. In the next theorem, we show that under the assumptions specified above, the empirical measurement $n^{-1}\sum_{i=1}^{n}[L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]$ converges to its expectation $E[L\{Y, \hat{f}(\boldsymbol{X})\}]$ at the rate $O_P[\{\log(p) \vee \log(n)\}/\sqrt{n}]$.

Theorem 2.3.3. *Suppose Assumptions 1-7 hold. Furthermore, assume that* $\log(p)/\sqrt{n} \to 0$ *as* $n \to \infty$. *If we choose* $\lambda_1 = O\{\log(n)^{-1}\}$, $\lambda_2 = O[\{\log(p) \vee \log(n)\}/\sqrt{n}]$, *and* $\lambda_3 = o(\lambda_1)$ *in (2.6), we have that the corresponding global solution* $(\hat{\mathbf{w}}^T, \hat{\boldsymbol{\alpha}}^T, \hat{b})^T$ *to (2.6) satisfies that,* $|E[L\{Y, \hat{f}(\boldsymbol{X})\}] - n^{-1}\sum_{i=1}^{n}[L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]| = O_P[\{\log(p) \vee \log(n)\}/\sqrt{n}]$, *where* $\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{n}\hat{\alpha}_j K_{\hat{\mathbf{w}}}(\boldsymbol{x}, \boldsymbol{x}_j) + \hat{b}$.

Theorem 2.3.3 shows that the empirical average loss $n^{-1}\sum_{i=1}^{n}[L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]$ from the training data set, can be a good estimate of the expected loss $E[L\{Y, \hat{f}(\boldsymbol{X})\}]$. As discussed above, this empirical loss can provide valuable information on the prediction performance of $\hat{f}$.

As a remark, we would like to point out that our theorems can be generalized to the case of local solutions, provided that similar conditions as in Assumptions 4-6 are met. For example, the convexity of local solutions can be stated in an analogous manner as in Assumption 5, and the corresponding signal strength can be measured by the partial derivatives as in Assumption 6.

2.4 Numerical Analysis

In this section, we use regression and classification as examples of learning techniques, and explore the numerical performance of our proposed DOSK method using simulated and real data sets. In Section 2.4.1, we study the empirical prediction behavior of DOSK using synthetic data sets, and in Section 2.4.2, we examine the performance of DOSK in real data applications. We compare our method with some existing approaches in the literature. In particular, for regression problems, we compare our DOSK method with the standard linear ridge regression, LASSO, standard L_2 kernel learning as in (2.3), COSSO and KNIFE.

Moreover, we implement the Sure Independence Screening (SIS) and Recursive Feature Elimination (RFE) methods with L_2 kernel learning. Notice here the generalization of SIS from linear learning to kernel learning is analogous to the approach discussed in Guyon et al. (2002). We employ the squared error loss function for all regression techniques. For classification methods, we use the SVM hinge loss for DOSK, and compare with the standard kernel SVM, kernel SIS SVM, kernel RFE SVM and KNIFE SVM.

In all numerical examples, we select the tuning parameters as follows. For our DOSK method, because there are three tuning parameters λ_1-λ_3 and potential kernel parameters (such as the γ parameter in the Gaussian kernel), we fix $\lambda_3 = 0.5$, and let other parameters be selected from a set of candidates. In particular, we let λ_1 vary in $\{0, 0.25, 0.5\}$, and let λ_2 vary in $\{2^i;\ i = -3, -2, \ldots, 2, 3\}$. As we will show in Section 4.1 that the selection of λ_3, the tuning parameter for the quadratic kernel regularization term, does not appear to play an essential role in maximizing the prediction accuracy of DOSK as long as its value is taken within a certain range. For the kernel parameters, because we use the Gaussian and Laplacian kernels (whose kernel functions are discussed in Section 2.2.2) in our analysis, we let the parameter γ vary in $\{0.1, 0.2, \ldots, 0.9, 1\}$, a candidate set whose range always covers $1/2\hat{\sigma}^2$ where $\hat{\sigma}$ is the median of the Euclidean distances between each pair of the observations. In our experience, this tuning procedure works reasonably well for the numerical examples in this chapter. For real applications, one can perform finer tuning procedures using a larger candidate set of tuning parameters. For other existing approaches except SIS and RFE, the tuning parameters are chosen in an analogous manner. The best set of tuning parameters that minimizes the prediction error in five fold cross validations on the training data set is then selected, and we report the corresponding prediction errors on a separate testing data set. Here the prediction error for regression examples is measured by the Mean Prediction Error (MPE, Hastie et al., 2011), $\frac{1}{n}\sum_{i=1}^{n}\{\hat{f}(\boldsymbol{x}_i) - y_i\}^2$. The error measure for classification problems is the misclassification rate (MCR), $\frac{1}{n}\sum_{i=1}^{n} I[y_i \neq \text{sign}\{\hat{f}(\boldsymbol{x}_i)\}]$, where $I(\cdot)$ is the indicator function.

2.4.1 Simulated Examples

In this section, we conduct four simulated examples to demonstrate the performance of our DOSK method. The first two examples are regression problems, and the last two are classification problems. In each example, we let the responses depend only on several predictors, and we add noise covariates in the date sets. We denote by p_0 the number of noise predictors. To assess various methods, we repeat each example 50 times and report the average prediction errors on the training and testing data sets. Furthermore, for all the methods that have variable selection, we report the True Positive (TP) rates and False Negative (FN) rates of predictors to compare the corresponding performance on variable selection.

Regression Example 1: For this example, the response depends only on one predictor. In particular, we have $y_i = 10\sin(x_{i1})I(0 < x_{i1} < 2\pi) + \epsilon_i$ where x_{i1} is the first predictor of the ith observation. Here x_{ij} follows a uniform distribution within $[-2\pi, 4\pi]$ for $j = 1, \cdots, 1 + p_0$, and the error term ϵ is generated from the standard normal distribution. In this example, we let $p_0 = 2$ and $p_0 = 8$, and choose the size of the training data set to be 50 and 100. The size of the testing set is 10 times larger than that of the training set. We use the Laplacian kernel in this example.

The numerical results for Regression Example 1 are reported in Table 2.1. One can see that the ridge regression and LASSO perform poorly using linear learning, as the underlying function f_0 is highly nonlinear. Note that the standard kernel learning method with the L_2 penalty has very small prediction error rate on the training data set. This shows that the corresponding models can fit the training observations very well. However, the errors on the testing data set are very large. This suggests that without appropriate variable selection, the performance of standard kernel learning can be greatly undermined by overfitting. Moreover, the SIS and RFE approaches can also have overfitting issues, which are partly due to their large FN rates. Compared to these methods, KNIFE and our DOSK work competitively. Note that the prediction error of COSSO is also good with a large sample size ($n = 100$). However, the corresponding variation is significantly larger than that of KNIFE or DOSK. This suggests that decomposing the nonlinear function into a sum of orthogonal components can be instable for some kernels. Furthermore, as the underlying function can be well approximated by functions that have sparse presentations, our DOSK method works better than KNIFE. This is similar to the findings in Zhang et al. (2015). To demonstrate the effect of data selection, in Figure 2.1, we plot the fitted regression function $\hat{f}$ from our DOSK method in a typical replicate, and the underlying function f_0 as a comparison. Moreover, we plot all the training observations, and highlight the selected ones, whose corresponding $\hat{\alpha}_j$'s are non-zero. One can see that because we are using the Laplacian kernel which has a singularity at 0 and smooth elsewhere, the data sparsity penalty tends to choose the observations that are closer to the "sharp turns" of f_0 for representation. This helps to build a model that is smooth when the curvature of f_0 is small, thus prevents overfitting from using all observations in the kernel function representation. In addition, according to Figure 2.1, some points located in the fluctuating region of the curve were selected. This might be due to the existence of other eight noisy variables. This further shows that variable sparsity can be important besides data sparsity.

Regression Example 2: In this example, the response Y depends on 4 predictors. In particular,

$$y_i = 10 \sum_{j=1}^{4} \exp(-x_{ij}^2) + \epsilon_i,$$

where the error term follows standard normal distribution, and x_{ij} follows a uniform distribution in $[-6, 6]$ for $j = 1, \ldots, 4$. The number of noise covariates and sizes of the training and testing data sets are the same as in Regression Example 1. We use the Gaussian kernel in this example. The prediction performance and variable selection results for Regression Example 2 are reported in Table 2.2, and one can draw similar conclusions as in Regression

p_0	Method	$n = 50$				$n = 100$			
		Train MPE	Test MPE	TP	FN	Train MPE	Test MPE	TP	FN
2	Linear Ridge	15.89 (4.46)	17.96 (1.33)	-	-	16.29 (3.46)	17.82 (1.16)	-	-
	LASSO	15.89 (4.47)	17.96 (1.32)	1	0.49	16.29 (3.46)	17.82 (1.17)	**1**	0.5
	L_2 Kernel	2.06 (0.45)	11.17 (2.00)	-	-	2.09 (0.38)	7.36 (1.55)	-	-
	SIS	8.22 (5.50)	12.20 (7.13)	0.42	0.29	5.39 (5.85)	7.54 (7.51)	0.68	0.16
	RFE	4.77 (3.91)	10.57 (6.05)	0.44	0.30	3.10 (3.51)	5.44 (5.02)	0.7	0.16
	COSSO	7.05 (6.56)	11.99 (10.32)	0.56	0.39	0.96 (1.29)	1.99 (2.58)	0.98	0.53
	KNIFE	3.66 (0.48)	6.14 (2.00)	**1**	0.14	2.35 (0.19)	3.03 (0.57)	**1**	**0**
	DOSK	1.42 (0.21)	**3.40 (2.92)**	**1**	**0.04**	0.92 (0.13)	**1.42 (0.19)**	**1**	**0**
8	Linear Ridge	13.77 (2.89)	18.09 (1.55)	-	-	16.11 (2.78)	17.68 (1.03)	-	-
	LASSO	13.77 (2.89)	18.12 (2.15)	**1**	0.87	16.13 (2.77)	17.61 (1.02)	**1**	0.88
	L_2 Kernel	0.05 (0.01)	17.26 (1.52)	-	-	0.05 (0.01)	15.76 (1.05)	-	-
	SIS	3.94 (2.04)	16.18 (4.44)	0.46	0.31	3.07 (1.90)	9.01 (3.95)	0.86	0.26
	RFE	9.83 (4.97)	16.18 (12.30)	0.54	**0.24**	6.44 (5.73)	10.29 (6.03)	0.86	0.25
	COSSO	12.27 (40.97)	19.93 (12.30)	0.54	**0.24**	6.44 (5.73)	10.29 (8.66)	0.76	0.25
	KNIFE	2.40 (0.53)	13.89 (3.64)	**1**	0.42	1.58 (0.18)	2.69 (1.99)	**1**	0.22
	DOSK	2.70 (0.59)	**10.80 (5.59)**	0.95	0.29	1.12 (0.20)	**2.15 (2.81)**	**1**	**0.20**

Table 2.1: Results of Regression Example 1. The numbers in parentheses show the corresponding standard deviations. MPE stands for mean prediction error, TP and FN represent true positive rates and false negative rates, respectively.

Example 1.

Classification Example 1: In this example, we consider a binary classification problem, where the prior probabilities $\mathrm{pr}(Y = +1) = \mathrm{pr}(Y = -1) = 1/2$. The posterior probabilities $\mathrm{pr}(Y = +1 \mid \boldsymbol{X} = \boldsymbol{x})$ depend on two predictors. In particular, the distribution of $x_{\cdot 1}$ and $x_{\cdot 2}$ for the first class is $N\{(0,0)^T, I_2\}$, where $x_{\cdot j}$ represents the jth predictor, and I_2 is the 2×2 identity matrix. For the second class, the distribution of $x_{\cdot 1}$ and $x_{\cdot 2}$ is proportional to the restricted joint normal distribution $N\{(0,0)^T, I_2\} \mid 9 < (x_{\cdot 1}^2 + x_{\cdot 2}^2) < 16$. To illustrate the marginal distribution of $x_{\cdot 1}$ and $x_{\cdot 2}$, we plot the first two covariates for a typical sample in Figure 2.2. In this example, we let $p_0 = 0, 4, 8$, and add independent noise variables following $N\,(0, 0.1)$ in the data set. The number of observations in the training data set is 200, and in the testing 2000. Note that a similar example was previously used in Hastie et al. (2011). The Gaussian kernel is used.

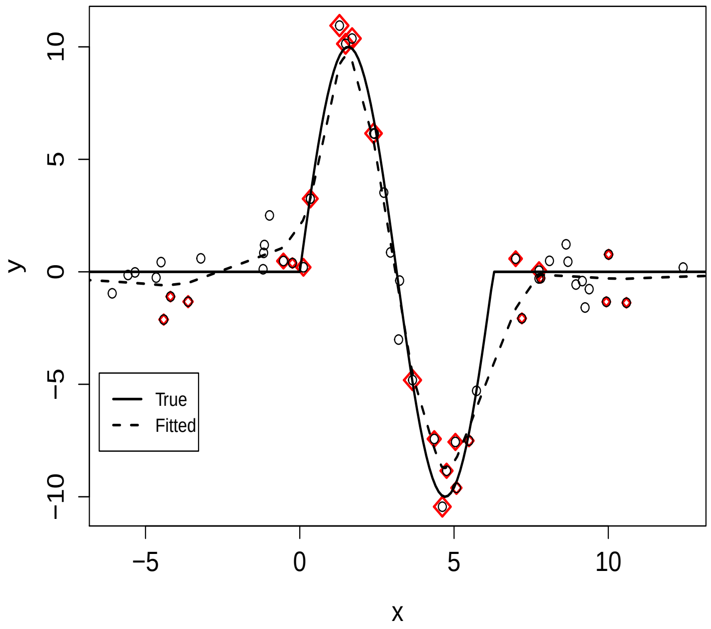

Figure 2.1: Plot of the underlying f_0 (solid) and fitted $\hat{f}$ by DOSK (dashed) when $n = 100$ and $p_0 = 2$. Observations with non-zero $\hat{\alpha}_j$'s are highlighted in red. One can see that the data sparsity penalty tends to choose observations that are closer to 0, $\pi/2$, $3\pi/2$ and 2π for the function representation.

The simulation results are reported in Table 2.3. One can see that when there are no noise predictors, all the methods can provide similar classification performance, with our DOSK method being slightly better. When the number of noise covariates increases, the prediction performance of L_2 kernel SVM, SIS and RFE deteriorates. On the other hand, the KNIFE method and our DOSK work competitively. Moreover, in this example, the classification boundary ($x_{\cdot 1}^2 + x_{\cdot 2}^2 = 9$) is relatively simple (see Figure 2.2 for an illustration). Hence, functions with sparse representations in the dual space can separate the two classes well. Consequently, our DOSK method works better than the KNIFE approach. In terms of variable selection, KNIFE and DOSK both perform very well, and are significantly better than the other methods.

Classification Example 2: We consider a similar example as in Classification Example 1.

p_0	Method	$n = 50$				$n = 100$			
		Train MPE	Test MPE	TP	FN	Training MPE	Test MPE	TP	FN
2	Linear Ridge	30.20 (7.34)	35.01 (2.54)	-	-	32.11 (5.91)	33.97 (2.09)	-	-
	LASSO	30.19 (7.34)	35.00 (2.57)	0.99	0.50	32.10 (5.91)	33.97 (2.09)	1	0.50
	L_2 Kernel	0.05 (0.02)	28.01 (2.57)	-	-	0.04 (0.01)	23.94 (2.09)	-	-
	SIS	1.07 (2.09)	30.92 (3.53)	0.34	0.31	1.92 (2.70)	29.61 (3.62)	0.29	0.41
	RFE	8.75 (8.22)	32.15 (4.05)	0.34	0.32	14.32 (9.20)	30.34 (3.53)	0.30	0.27
	COSSO	14.56 (4.60)	31.45 (11.10)	0.49	**0.17**	16.33 (8.93)	21.09 (9.62)	0.48	**0.11**
	KNIFE	6.56 (1.33)	21.26 (3.12)	**1**	0.49	5.99 (0.54)	12.99 (1.29)	**1**	0.18
	DOSK	2.14 (0.61)	**18.25 (3.70)**	**1**	0.54	2.60 (0.31)	**9.86 (1.44)**	**1**	0.12
8	Linear Ridge	26.28 (7.09)	33.95 (3.05)	-	-	30.06 (5.60)	34.21 (1.73)	-	-
	LASSO	26.26 (7.07)	33.94 (3.04)	1	0.88	29.06 (5.41)	33.17 (1.69)	1	0.88
	L_2 Kernel	0.05 (0.02)	33.97 (3.05)	-	-	0.04 (0.01)	26.23 (1.73)	-	-
	SIS	0.05 (0.03)	33.63 (2.94)	0.32	0.33	0.04 (0.01)	33.71 (1.84)	0.31	0.35
	RFE	10.54 (7.79)	32.90 (3.50)	0.33	**0.18**	13.92 (10.32)	32.25 (3.30)	0.32	0.19
	COSSO	18.36 (7.82)	35.54 (6.68)	0.31	0.25	16.41 (7.13)	27.14 (7.13)	0.51	0.18
	KNIFE	5.47 (0.78)	25.53 (4.03)	**0.99**	0.46	5.53 (0.50)	14.52 (2.41)	**1**	0.17
	DOSK	1.54 (0.33)	**23.97 (6.10)**	**0.99**	0.36	2.37 (0.28)	**10.70 (3.20)**	**1**	**0.15**

Table 2.2: Results of Regression Example 2. The numbers in parentheses show the corresponding standard deviations. MPE stands for mean prediction error, TP and FN represent true positive rates and false negative rates, respectively.

In particular, we let the classification signal depend on 4 predictors. For the first class, the distribution of $x_{\cdot 1}$ to $x_{\cdot 4}$ is $N\{(0,0,0,0)^T, I_4\}$. The corresponding distribution of the second class is proportional to $N\{(0,0,0,0)^T, I_4\} \mid 9 < \sum_{j=1}^{4} x_{\cdot j}^2 < 16$. We let $p_0 = 0, 4, 8$ in this example. The classification results are reported in Table 2.4, and one can draw a similar conclusion as that of Classification Example 1.

Next, we would like to use simulated examples to discuss the computational complexity and the compare the runtime of DOSK with other methods. According to the algorithm of DOSK, the linear approximation in the $\mathbf{w}$ step simplifies the original non-convex optimization problem into a quadratic programming program with linear constraints. Similar to KNIFE, the order of the computational cost per iteration of DOSK should be equivalent to that of the kernel regression using the quadratic loss. Similarly, the computational cost of DOSK would perform the same as the standard SVM using the hinge loss. In practice, the actual runtime of DOSK can depend on the number of iterations used before convergence. Therefore, a

p_0	Method	Train MCR	Test MCR	TP	FN
0	L_2 Kernel	2.94 (0.93)	2.92 (0.50)	-	-
	SIS	2.94 (0.93)	2.92 (0.50)	**1**	**0**
	RFE	2.94 (0.93)	2.92 (0.50)	**1**	**0**
	KNIFE	4.00 (2.92)	4.32 (3.94)	0.98	**0**
	DOSK	1.63 (0.73)	**1.72 (0.34)**	**1**	**0**
4	L_2 Kernel	1.63 (0.89)	6.68 (0.75)	-	-
	SIS	2.31 (1.22)	5.23 (1.50)	**1**	0.69
	RFE	9.48 (12.84)	12.02 (12.40)	0.8	0.36
	KNIFE	3.33 (1.30)	3.31 (0.50)	**1**	**0**
	DOSK	2.07 (0.12)	**2.02 (0.56)**	**1**	**0**
8	L_2 Kernel	0.08 (0.21)	15.07 (1.89)	-	-
	SIS	0.96 (1.00)	9.53 (4.45)	**1**	0.66
	RFE	5.42 (8.97)	12.18 (9.16)	0.86	0.46
	KNIFE	3.48 (1.87)	3.89 (2.97)	0.99	**0**
	DOSK	1.58 (1.63)	**1.79 (0.34)**	**1**	**0**

Table 2.3: Results of Classification Example 1. The numbers in parentheses show the corresponding standard deviations. MSC stands for Mis-Classification Rate, TP and FN represent true positive rates and false negative rates, respectively.

proper starting point $\mathbf{w}^{(0)}$ can save the computational time significantly.

In order to assess the actual runtime performance of DOSK, we use the same four simulated examples above and fix the noise dimension as $p_0 = 8$. We also include two real data applications: the CPUs and Ecoli datasets. To have a general idea of the runtime in finding the best tuning parameters, we record the average time (in seconds) that each method takes for each tuning parameter value combination. For regression examples, the linear ridge and LASSO are implemented by the R package glmnet. The L_2 Kernel method is also implemented by glmnet but includes some extra kernel matrix calculation. SIS, RFE and COSSO are implemented by the corresponding R packages SIS, caret, and COSSO respectively. KNIFE and DOSK are implemented using R entirely. For classification examples, L_2 Kernel, SIS and RFE are all primarily fitted by the R package e1071 with some extra matrix calculation. KNIFE and DOSK are implemented by a R wrapper of the Matlab package CVX to conduct the two quadratic programmings in each iteration. As to the stopping criterion, we always use the default settings when there is a corresponding R package. For KNIFE and

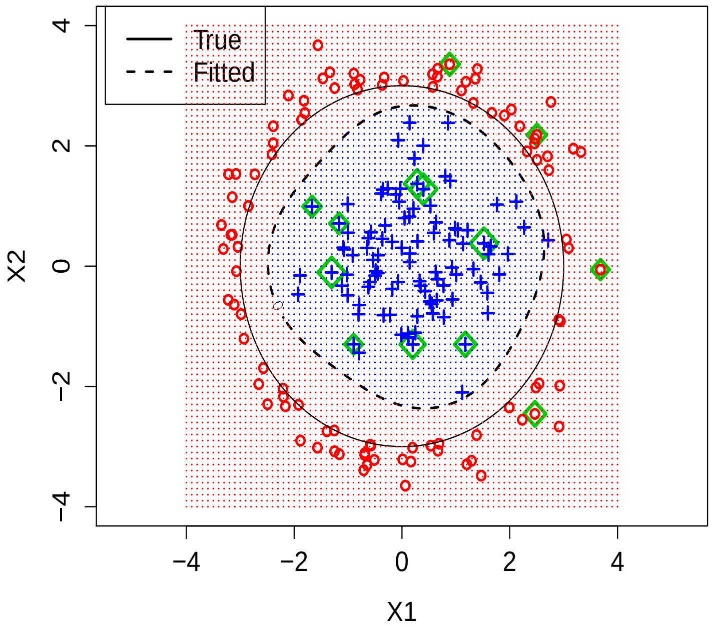

Figure 2.2: Plot of the underlying classification boundary (solid circle) and estimated boundary by DOSK (dashed circle) when $n = 200$ and $p_0 = 8$. Observations with non-zero $\hat{\alpha}_j$'s are highlighted in green.

DOSK, we set the maximum iteration number to be 300 and the stopping rule as when the L_2-norm of the objective function change is less than 0.001. The average runtime of all the methods for each tuning parameter set is listed in Table 2.5.

Based on the results in Table 2.5, it is not surprising to see that the linear ridge and LASSO take much less time than all the other methods since the core of the package glmnet contains a set of Fortran subroutines, which is much faster than the corresponding R code. The L_2 kernel method, SIS, and RFE are slower not only because they have more complexity but also due to the extra matrix calculation in R. Similar arguments can also be made for these methods in classification, which are implemented by the libsvm C++ code. The results of COSSO heavily depend on the selection of the knots number. As to KNIFE and DOSK, they perform almost equivalently in terms of computational time under both the regression and classification examples. This comparison result is consistent to our previous discussion on

p_0	Method	Train MCR	Test MCR	TP	FN
0	L_2 Kernel	6.34 (0.15)	8.08 (0.80)	-	-
	SIS	6.34 (0.15)	8.08 (0.80)	1	0
	RFE	6.34 (0.15)	8.08 (0.80)	1	0
	KNIFE	7.30 (1.70)	8.85 (0.87)	1	0
	DOSK	4.37 (1.74)	**5.81 (0.73)**	**1**	**0**
4	L_2 Kernel	1.58 (1.08)	14.56 (1.23)	-	-
	SIS	2.59 (1.02)	13.49 (1.87)	1.00	0.84
	RFE	10.82 (3.96)	19.96 (6.87)	0.76	0.52
	KNIFE	7.73 (1.88)	9.41 (1.66)	1	0
	DOSK	4.94 (1.68)	**6.00 (0.84)**	**1**	**0**
8	L_2 Kernel	0.02 (0.01)	22.28 (1.65)	-	-
	SIS	2.02 (5.64)	19.60 (3.72)	0.96	0.72
	RFE	8.12 (2.10)	22.93 (6.21)	0.76	0.50
	KNIFE	7.21 (1.72)	9.03 (1.20)	1	0
	DOSK	5.04 (1.75)	**5.93 (0.64)**	**1**	**0**

Table 2.4: Results of Classification Example 2. The numbers in parentheses show the corresponding standard deviations. MSC stands for Mis-Classification Rate, TP and FN represent true positive rates and false negative rates, respectively.

the comparable computational complexity. Note that KNIFE and DOSK have long runtime under classification examples because there is some additional communication cost needed for calling the Matlab package CVX from R.

As to the tuning parameter selection, we fix $\lambda_3 = 0.5$ to save the computational time. Note that there are three tuning parameters $\lambda_1, \lambda_2, \lambda_3$ in (2.6) for the proposed DOSK. Based on our numerical experiment, the performance of DOSK is not sensitive to the choice of λ_3, the tuning parameter for the quadratic penalty term. For illustration, we draw four contour plots of the mean prediction errors for Regression Example 2 when $p_0 = 8$ in Figure 2.3. In particular, we set λ_3 as $\{0, 0.25, 0.5, 1\}$ respectively for each plot and calculate the optimal prediction error among all combinations of λ_1 and λ_2 with τ being $1/2\hat{\sigma}^2$, where $\hat{\sigma}$ is the median of the pairwise Euclidean distances for the simulated samples. From the result, one can observe that the best $(\lambda_1,\, \lambda_2)$ combination is almost always near the coordinate $(0.5, 0.5)$ for all these λ_3 values. Because we fix λ_3 to be 0.5 in DOSK, KNIFE and DOSK have the identical number of parameters to be tuned in practice. This choice appears to work well in all the experiments we tried. As a consequence, these two methods need approximately the same time in finding the best λ's.

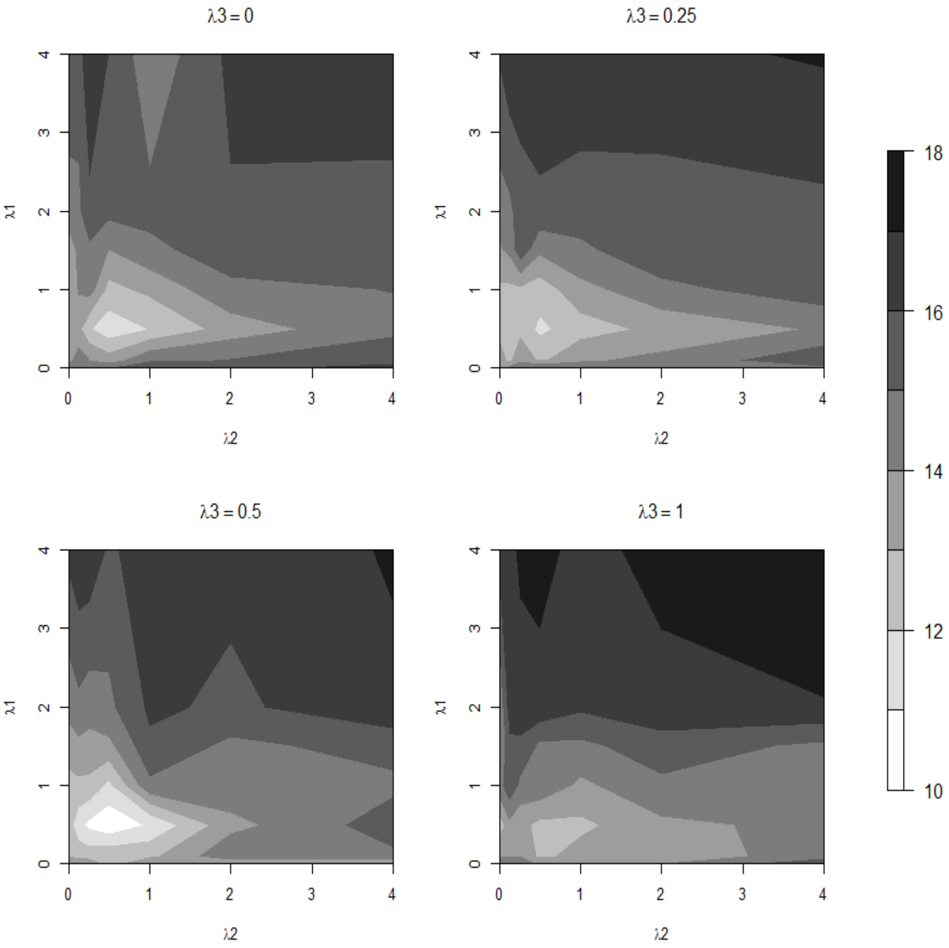

Figure 2.3: Contour plots of the mean prediction errors of DOSK for Regression Example 2 where $p_0 = 8$. Here λ_3 is set as $\{0, 0.25, 0.5, 1\}$ for the four panels and the kernel bandwidth $\tau = 1/2\hat{\sigma}^2$, where $\hat{\sigma}$ is the median of the pairwise Euclidean distances of the simulated samples.

2.4.2 Real Data Applications

In this section, we apply our DOSK method to four real data sets and explore the corresponding prediction performance. In particular, the first two real data sets are about regression problems, and the last two are for classification applications.

Regression Examples: Ozone Data and CPUs Data

We consider the ozone pollution data in Los Angels (Breiman and Friedman, 1985), and the Central Processing Units (CPUs) performance prediction data (Ein-Dor and Feldmesser, 1987) as our regression applications. The ozone data set includes 330 observations, and each observation contains the daily measurement of ozone reading (the response) in 1976. Furthermore, 8 predictors that have potential impact on the ozone readings are also available, such as temperature, inversion base height, etc. The CPUs performance data set can be found in the UCI machine learning Repository (Bache and Lichman, 2015). The corresponding response variable contains 209 different CPUs' published relative performance on a benchmark mix. The data set also includes 7 predictors, such as the cache size, minimum main memory, and cycle time, among others, which may be useful in predicting a computer's performance.

Before the analysis, we standardize the data sets, such that the range of each predictor is in $[0, 1]$. Because we do not have separate training and testing data sets, for each replicate we randomly split the data into two equal parts, and use one for training and the other for testing. We choose the best tuning parameters in a similar way as in the simulated examples, by 5-fold cross validations on the training sets. The Laplacian kernel is used for both examples. We compare our DOSK method with LASSO, standard L_2 kernel learning, SIS regression with L_2 kernel learning, RFE with L_2 kernel learning, COSSO and KNIFE.

The average prediction errors in 50 replicates are summarized in Table 2.6. For the ozone data, the DOSK method performs better than the existing approaches in terms of the average prediction error. For the CPUs data, one can see that the standard L_2 kernel learning may have a potential overfitting issue, which is similar to the simulation results. In terms of variable selection, we report the predictors that are selected more than 45 times out of the 50 replicates. In the CPUs data set, each method selects a small subset of the predictors in the models. In particular, SIS tends to fit a model with minimum main memory and maximum main memory. The RFE and LASSO approaches select maximum main memory, cache size, and maximum number of channels as the important variables. For COSSO, KNIFE and our DOSK methods, the maximum main memory and cache size are the selected variables. This is

Examples	Reg-1	Reg-2	CPUs		Class-1	Class-2	Ecoli
Methods	Time	Time	Time	Methods	Time	Time	Time
Linear Ridge	0.26	0.36	0.22				
LASSO	1.12	0.87	0.57				
L_2 Kernel	13.65	13.09	11.94	L_2 Kernel	4.39	4.41	2.18
SIS	11.18	13.31	13.50	SIS	17.13	17.91	13.73
RFE	41.25	69.27	57.71	RFE	28.42	39.87	16.72
COSSO	34.23	39.37	42.84				
KNIFE	82.2	83.88	82.16	KNIFE	145.68	162.41	86.10
DOSK	98.46	97.36	81.25	DOSK	153.94	156.16	91.45

Table 2.5: Average runtime (in second) of each method per tuning parameter combination in the selected numerical studies. Here $n = 100$ and $p_0 = 8$ for all simulated examples.

consistent with the insights given in Ein-Dor and Feldmesser (1987). In other words, to specify the performance of a computer, only a few components are necessary. Interestingly, LASSO works slightly better than SIS, RFE, or the COSSO methods in prediction. One possible explanation is that the response is not highly nonlinear in this example, and kernel learning methods without stable variable selection can lead to suboptimal results. In contrast, KNIFE performs competitively, while our DOSK enjoys the best accuracy. This suggests that variable weighted kernel learning can provide stable selection performance for real applications.

	Ozone		CPUs	
Methods	Train MPE	Test MPE	Train MPE	Test MPE
L_2 Kernel	12.51 (1.27)	17.37 (1.68)	0.01 (0.002)	0.40 (0.24)
LASSO	19.34 (1.36)	20.80 (1.69)	0.11 (0.04)	0.21 (0.09)
SIS	18.72 (1.61)	21.47 (1.78)	0.11 (0.03)	0.33 (0.21)
RFE	13.89 (1.44)	18.37 (1.73)	0.02 (0.01)	0.35 (0.20)
COSSO	17.56 (2.14)	20.45 (1.96)	0.12 (0.07)	0.28 (0.12)
KNIFE	11.03 (1.09)	17.08 (1.90)	0.10 (0.01)	0.17 (0.08)
DOSK	11.21 (1.41)	**16.92 (1.65)**	0.09 (0.02)	**0.16 (0.10)**

Table 2.6: The mean prediction error (MPE) for the ozone and CPUs data sets.

Classification Examples: Breast Cancer Wisconsin Data and Ecoli Data

For classification applications, we use the diagnostic Wisconsin breast cancer data set (Street et al., 1993) and the Ecoli data set (Nakai and Kanehisa, 1991) for illustration. These

	Breast Cancer		Ecoli	
Methods	Train MCR	Test MCR	Train MCR	Test MCR
L_2 Kernel	0.39 (0.24)	7.78 (1.42)	0.22 (0.33)	13.24 (4.42)
SIS	1.27 (0.73)	4.20 (1.09)	0.95 (0.68)	2.13 (1.21)
RFE	1.33 (0.56)	4.26 (1.00)	0.95 (0.68)	2.13 (1.25)
KNIFE	1.77 (0.54)	4.04 (0.78)	1.69 (0.81)	2.26 (1.27)
DOSK	2.40 (0.60)	**3.97 (1.11)**	1.52 (1.02)	**1.95 (1.02)**

Table 2.7: The Mis-Classification Rate (MCR, in percentages) for the breast cancer and Ecoli data sets.

two data sets can also be found in the UCI machine learning Repository. The breast cancer data set has diagnosis results (malignant or benign) for 569 patients. The data also contain 30 predictors computed from a digitized image of a fine needle aspirate of a breast bass, such as mean distances from center to points on the perimeter, standard deviation of gray-scale values, etc. The Ecoli data set has 8 categories of proteins, and we use two categories, namely, cytoplasmic proteins and inner membrane proteins without signal sequence, for demonstration in our analysis. The total number of samples of these two classes is 220, and the data set includes 7 predictors, such as different measures of signal protein sequence recognition, consensus sequence score, amino acid content in certain outer proteins, among others.

We use DOSK with the SVM hinge loss, and compare our method with standard L_2 kernel SVM, SIS, RFE and KNIFE. Similar to the regression examples, we standardize all the predictors before our analysis. Furthermore, we randomly split the data sets into two equal parts, and use one for training (5 fold cross validations to select the best tuning parameters) and the other for testing. We report the average prediction error rates for various methods in Table 2.7, and one can see that the standard kernel SVM with the L_2 norm penalty can have a potential overfitting issue on these two data sets, which is consistent with the simulation results. Compared with other methods, our DOSK performs competitively.

2.5 Discussion

In this chapter, we propose a new DOSK method in kernel learning that can perform variable selection and data extraction simultaneously. We show that under certain conditions, the new DOSK method can achieve selection consistency, and the estimated function can converge to the underlying function with a fast rate. We also develop an efficient algorithm to solve the corresponding optimization, which is guaranteed to converge to a local optimum. Numerical results show that our DOSK method is highly competitive among existing approaches.

As a remark, our DOSK method can be generalized to alleviate the computational burden for applications with massive data sets. Without loss of generality, take regression as an example. Suppose one needs to estimate a nonlinear underlying function, and the data set contains many observations and predictors. To perform kernel regression with such big data can be computationally inefficient. One way to circumvent this difficulty is to split the predictors into several parts or dividing the observations into several subsets, learn on each part individually, and then combine the results. In particular, each time one can perform our DOSK method on one piece of the data set. Because our DOSK method can have double sparsity in predictors and dual variables, for each sub-regression, it is possible to find a sparsely represented function that only involves a subset of observations and predictors. Then we can combine the selected observations and predictors to train for a global estimator. One can see that this approach can greatly reduce the computational time for problems with massive data sets. Further research can be pursued in this direction.

CHAPTER 3: ESTIMATING INDIVIDUALIZED TREATMENT RULES FOR ORDINAL TREATMENTS

3.1 Introduction

Heterogeneous data analysis becomes popular recently due to the development of precision medicine. Precision medicine is a medical paradigm which suggests personalized health care to different patients. Its recent development originates from the fact that the treatment effect can vary widely from subject to subject due to individual level heterogeneity. For example, Ellsworth et al. (2010) found that women whose CYP2D6 gene has a certain mutation are not able to digest Tamoxifen efficiently and this makes them an improper target group for this therapy. In this way, one of the primary goals for precision medicine is to establish rules such that patient personal characteristics can be used directly to find optimal treatments (Mancinelli et al. (2000); Simoncelli (2014). From statistics perspective, there are at least two directions that one can follow to achieve this goal. The first direction is to estimate the optimal individual treatment rule (ITR) directly regardless of how different the treatment effect is between the observations that correspond to different optimal treatments. Some representative literature contains Qian and Murphy (2011); Zhao et al. (2012); Zhang et al. (2012). In contrast, the other direction focuses on detecting the optimal treatments by estimating the heterogeneous outcome-predictor relationship and then derive the optimal treatment rule, such as (Su et al. (2009, 2011); Lipkovich and Dmitrienko (2014); Zhao et al. (2013); Shen and He (2015)). These methods usually ask for stronger assumptions on the datasets than those in the first direction.

In this chapter, we focus on the first direction as mentioned above. In particular, we propose a new method called generalized outcome weighted learning (GOWL). Specifically, our first contribution is to create a new objective function for ITR estimation based on the value function definition in Qian and Murphy (2011) through making use of the data duplication idea. We then formulate the optimal ordinal treatment rule detection problem into an aggregation of several optimal binary treatment rule detection subproblems. Furthermore, considering that each subproblem corresponds to a level of the ordinal treatment, we prevent estimated decision boundaries from the subproblems intersecting with each other to circumvent contradictory results. The second contribution of the chapter is to modify the loss function in Zhao et al. (2012) to maintain convexity regardless of whether the value of the reward is positive or negative. This loss function enables GOWL to penalize the treatments corresponding to negative reward values properly to avoid the rewards shift problem previously described.

To estimate the optimal individual treatment rule in the new optimization problem, we provide an efficient algorithm using the primal-dual formulation. Moreover, we show that our method achieves Fisher consistency under mild conditions, which means that the true optimal treatment can be reached if the entire population is used. In addition, we prove that the estimated intercepts of the decision functions are monotonic along the treatment level, which will make the decision boundaries interpretable in practice. We also show that the proposed method with the Gaussian kernel has the asymptotic convergence rate of $n^{-1/2}$ for a well-separated data set under the geometric noise condition (Steinwart and Scovel (2007b)).

The remainder of the chapter is organized as follows. In Section 2, we review the OWL method and then explain how the modified loss function for GOWL works under the binary treatment setting. In Section 3, we illustrate how GOWL works for the ITR estimate in the ordinal treatment setting based on the necessary background information for the data duplication method. In Section 4, we establish the statistical learning properties of GOWL. Simulated data examples are used in Section 5, and two applications to an irritable bowel syndrome problem and a type 2 diabetes mellitus observational study are provided in Section 6. We then provide some discussions and conclusions in Section 7. A separate supplemental material includes the computational algorithm, additional numerical results and proofs of the theorems.

3.2 Generalized Outcome Weighted Learning for Binary Treatments

In this section, we give a brief review of OWL and its corresponding optimization problem. Motivated by the limitations of OWL, we propose a generalized version of OWL for the binary treatment case using a modified loss function.

3.2.1 Outcome Weighted Learning

Suppose that we collect the data from a two-arm clinical study where the binary treatment is denoted by $A \in \mathcal{A} = \{-1, 1\}$. We assume that the patients' covariates are represented by an n by p matrix $X \in \mathcal{X}$, where $\mathcal{X}$ denotes the covariate space, n is the number of patients enrolled, and p corresponds to the number of covariates. We also use a bounded random variable R to represent the clinical outcome reward and assume a larger R value is more desirable. Note that R can depend on both X and A. Under this framework, the ITR is a mapping from $\mathcal{X}$ to $\mathcal{A}$. According to Qian and Murphy (2011), the goal of an optimal ITR is to find the mapping $\mathcal{D} = \mathcal{D}^*$ such that

$$\mathcal{D}^*(X) = \arg\min_{\mathcal{D}} \left\{ E\left(\frac{R \cdot I(A \neq \mathcal{D}(X))}{P(A|X)} |X, \mathcal{D} \right) \right\}, \tag{3.1}$$

where $P(A|X)$ is the prior probability of treatment A for X. Note that $P(A|X) = P(A)$ under the independence assumption between A and X. Furthermore, the expectation operation in (3.1) is conditional on X and $\mathcal{D}$. From now on, we will omit the conditional part of the expectation to simplify the expressions. To estimate the optimal treatment rule $\mathcal{D}^*$, one needs to obtain a classifier function $f(x)$ such that $\mathcal{D}(x) = \text{sign}(f(x))$. Thus, we have that $I(A \neq \mathcal{D}(X)) = I(A \cdot f(X) \leq 0)$. To alleviate the non-deterministic polynomial-time (NP) computational intensity in (3.1), Zhao et al. (2012) proposed OWL by replacing the 0-1 loss above with the hinge loss used in the Support Vector Machine (SVM, Cortes and Vapnik (1995)) together with a regularization term to control model complexity. As a consequence, the regularized optimization problem becomes a search for the decision rule f which minimizes the objective function

$$\frac{1}{n}\sum_{i=1}^{n}\frac{r_i}{P(a_i|x_i)}\left[1-a_i f(x_i)\right]_{+}+\lambda||f||^2, \tag{3.2}$$

where $(x_i, a_i, r_i); i = 1, \cdots, n$, is a realization of (X, A, R) with $a_i \in \{-1, 1\}$, the function $[u]_+ = \max(u, 0)$ denotes the positive part of u, $||f||^2$ is the squared L_2 norm of f and λ is the tuning parameter used to control the model complexity and avoid overfitting. Notice that to maintain the convexity of the objective function, OWL requires all rewards to be non-negative.

In practice, when there are negative rewards, one can shift them by a constant to ensure positiveness. Zhou et al. (2017) noted that such a constant shift process for the rewards may lead to suboptimal estimates. They noted that the optimal treatment estimates tend to be the same as the random treatments that are originally assigned. This situation can be further illustrated by a toy example as follows. Suppose we have two intervention groups (treatment and placebo) and two patients both being assigned to the treatment group and receiving rewards of -10 and 10, respectively. Such results imply that the first patient may not benefit from the treatment due to the corresponding negative feedback. If we follow the reward shift idea as mentioned above and add 15 to both rewards, then the model will probably draw an incorrect conclusion that both patients benefit from the treatment since both shifted rewards are positive. Another drawback of this rewards-shift strategy comes from the fact that there are an infinite number of constants one can choose for the shift. Different shift constants can lead to different coefficient estimates when the decision rule f has a certain parametric or nonparametric form in problem (3.2). To solve this problem, we propose a generalized OWL in Section 2.2 which does not require rewards to be positive.

3.2.2 Generalized Outcome Weighted Learning

For problem (3.2), note that the OWL objective function is convex only when all of the rewards are non-negative and such a restriction could make OWL suboptimal when there are negative rewards, as discussed earlier. To remove such a restriction, we first consider reformulating the minimization problem (3.1) into two pieces as

$$\arg\min_{\mathcal{D}} E\left\{\frac{|R|}{P(A|X)}\left[I(R\geq 0)I\left(A\neq\mathcal{D}(X)\right)+I(R<0)I\left(A=\mathcal{D}(X)\right)\right]\right\}. \tag{3.3}$$

Note that (3.3) is equivalent to (3.1) because the term $\frac{R\cdot I(R<0)}{P(A|X)}$ is free of $\mathcal{D}(X)$. Similar to the discussion in Section 2.1, we can rewrite the optimization problem in (3.3) as follows, with $\mathcal{D}(X)=\text{sign}(f(X))$:

$$\arg\min_{\mathcal{D}} E\left\{\frac{|R|}{P(A|X)}\left[I(R\geq 0)I\left(A\cdot f(X)\leq 0\right)+I(R<0)I\left(A\cdot f(X)>0\right)\right]\right\}. \tag{3.4}$$

Furthermore, to alleviate the computational intensity of solving (3.4), we use a modified loss function to be minimized with the population form expressed as

$$E\left\{\frac{|R|}{P(A|X)}\left[I(R\geq 0)\left[1-Af(X)\right]_{+}+I(R<0)\left[1+Af(X)\right]_{+}\right]\right\}. \tag{3.5}$$

Here the ITR $\mathcal{D}$ in (3.3) is the sign function of the decision rule f in (3.5) by definition. Therefore, the corresponding empirical sum on the training data becomes

$$\sum_{i=1}^{n}\left\{\frac{|r_i|}{P(a_i|x_i)}\left[I(r_i\geq 0)\left[1-a_if(x_i)\right]_{+}+I(r_i<0)\left[1+a_if(x_i)\right]_{+}\right]\right\}. \tag{3.6}$$

Note that the loss in (3.6) has two parts according to the sign of r_i. For observations with positive rewards, we use r_i as their weights for the corresponding loss function and penalize the mis-classification by the standard hinge loss function $l_1(u)=[1-u]_+$ (see the left panel in Figure 3.1 for how the hinge loss approaches the 1-0 loss). This part is identical to the hinge loss in OWL. However, for observations with negative rewards, we use $-r_i$ as their weights instead and employ a modified hinge loss $l_2(u)=[1+u]_+$ (see the right plot in Figure 3.1 for how the modified hinge loss approaches the 0-1 loss) which assigns a larger loss to the observations whose estimated treatment $f(x_i)$ matches the observed treatment a_i. As

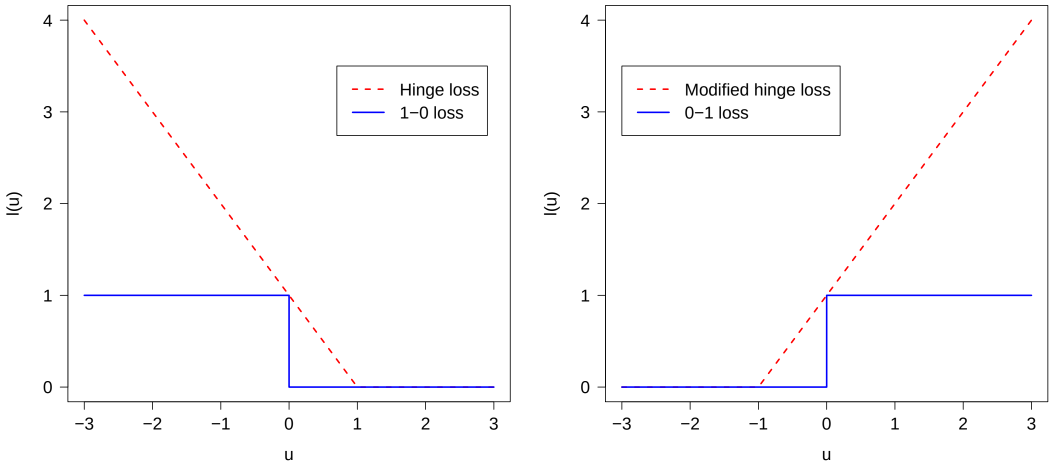

Figure 3.1: Standard hinge loss $l_1(u) = [1 - u]_+$ versus 1-0 loss (left) and modified hinge loss $l_2(u) = [1 + u]_+$ versus 0-1 loss (right). The modified hinge assigns large loss values to those observations whose estimated treatment rule matches the actual treatment assigned.

a consequence, the modified loss function in (3.6) is piecewise convex in terms of $a_i f(x_i)$. Therefore, a global optimization of the objective function could be guaranteed when standard convex optimization algorithms are applied. One advantage of using the modified hinge loss is that the observed rewards are no longer required to be positive so that the problem caused by the non-unique reward shift can be circumvented. In addition, one can see that the loss function reduces to the standard hinge loss when all $r_i > 0$. As a remark, we note that Laber and Murphy (2011) previously used a similar surrogate loss for construction of the adaptive confidence intervals for the test error in classification.

3.3 Generalized Outcome Weighted Learning for Ordinal Treatments

In this section, we discuss how to extend GOWL from binary treatments to ordinal treatments. For problems with multiple ordinal treatments, it is important to utilize the ordinal information. To this end, we borrow the idea of data duplication in standard ordinal classification and develop our new procedure for GOWL with ordinal treatments.

3.3.1 Classification on Ordinal Response with Data Duplication

For an ordinal response problem, suppose each observation vector is $\left(x_i^T, y_i\right)$ where $i = 1, \cdots, n$, the predictor x_i contains p covariates, and the response $y_i \in \{1, \cdots, K\}$.

Cardoso and Pinto da Costa (2007) proposed a data duplication technique to address this problem. To apply this idea, one first needs to generate a new data set written as $(x_i^{(k)^T}, y_i^{(k)})$, where $x_i^{(k)} = (x_i^T, e_k^T)^T$, $y_i^{(k)} = \text{sign}\,(y_i - k)$, e_k^T is a $K-1$ dimensional row vector whose kth element is 1 while others are zeros, and $k = 1, \cdots, K-1$. Thus, $y_i^{(k)}$ defines a new binary response indicating $1, \cdots, k$ versus $k+1, \cdots, K$. Here the sign(x) function is defined to be 1 when $x > 0$ and -1 otherwise. Then, the goal of the classification method is to find a surrogate binary classifier $f(x^{(k)})$ to minimize $\sum_{i=1}^{n} \sum_{k=1}^{K-1} l(y_i^{(k)}, f(x^{(k)})) + J(f)$, where $l(\cdot)$ is the pre-defined loss and $J(f)$ is a penalty term. Once these $f(x_i^{(k)})$ are obtained for $k = 1, \cdots, K-1$, then the predicted rule $\hat{\mathcal{D}}(x_i)$ for the original ordinal outcome y_i can be calculated by $\hat{\mathcal{D}}(x_i) = \sum_{k=1}^{K-1} I(f(x_i^{(k)}) > 0) + 1$, where $I(\cdot)$ is the indicator function.

3.3.2 Generalized Outcome Weighted Learning

Now consider an extended version of clinical data (X, A, R) in Section 2 with X and R the same as before but with A being an ordinal treatment with $A \in \mathcal{A} = \{1, \cdots, K\}$. In contrast to standard multicategory treatment scenarios, the K categories of treatments are ordered in a way that 1 and K are most different, For example, these treatments may represent different discrete dose levels with $A = 1$ being the lowest dose and $A = K$ being the highest dose. Similar to Section 3.1, we define the duplicated random set $\left(X^{(k)}, A^{(k)}, R^{(k)}\right)$ with its ith realization defined as $x_i^{(k)} = (x_i^T, e_k^T)^T$, $a_i^{(k)} = \text{sign}\,(a_i - k)$, and $r_i^{(k)} = r_i$ for $k = 1, \cdots, K-1$. According to the value function definition from Qian and Murphy (2011), we let $P^{\mathcal{D}_k}$ denote the conditional distribution of (X, A, R) on $A^{(k)} = \mathcal{D}(X^{(k)})$. Then, with the duplicated data set and a map $\mathcal{D}$ from each $X^{(k)}$ to $\{-1, 1\}$ for $k = 1, \cdots, K-1$, we propose a new conditional expected reward to be maximized as follows:

$$\begin{aligned} \sum_{k=1}^{K-1} E\left(R|A^{(k)} = \mathcal{D}(X^{(k)}), X\right) &= \sum_{k=1}^{K-1} \int R \frac{dP^{\mathcal{D}_k}}{dP} dP \\ &= \sum_{k=1}^{K-1} \int R \frac{I(A^{(k)} = \mathcal{D}(X^{(k)}))}{P(A|X)} dP \\ &= \sum_{k=1}^{K-1} E\left(\frac{R \cdot I(A^{(k)} = \mathcal{D}(X^{(k)}))}{P(A|X)}\right). \end{aligned} \tag{3.7}$$

Similar to Qian and Murphy (2011) and Zhao et al. (2012), we refer to (3.7) as the value function of $\mathcal{D}$ and denote it by $\mathcal{V}(\mathcal{D})$. In this way, the optimal map $\mathcal{D}^*$ is defined as

$$\mathcal{D}^* = \arg\min_{\mathcal{D}} \sum_{k=1}^{K-1} E\left(\frac{R \cdot I(A^{(k)} \neq \mathcal{D}(X^{(k)}))}{P(A|X)}\right). \tag{3.8}$$

Once the map $\mathcal{D}$ is estimated, one can obtain the corresponding ITR estimate of X by using $\hat{\mathcal{D}}(X) = \sum_{k=1}^{K-1} I(f(X^{(k)}) > 0) + 1$.

Notice that optimal treatment estimation through (3.8) can be effective when the treatment is ordinal due to the way it utilizes the ordinality information. In particular, the new minimization problem considers the distance between the estimated optimal treatment and the actually assigned treatment by counting the number of mismatches between each $\mathcal{D}(X^{(k)})$ and each $A^{(k)}$ for $k = 1, \cdots, K-1$. In the extreme case when a certain subject has an extremely large positive reward value, the estimated $\mathcal{D}(X^{(k)})$ would be likely to match $A^{(k)}$ for all $k = 1, \cdots, K-1$, which results in $\hat{\mathcal{D}}(X) = A$. In contrast, it may imply that the actually assigned treatment is suboptimal when the reward outcome takes a small value. Some of the estimated $\mathcal{D}(X^{(k)})$ will not match the observed $A^{(k)}$ as the estimated rule approximates the global minimizer of (3.8).

To alleviate the computational intensity of the minimization problem in (3.8), we replace the 0-1 loss with the modified loss in (3.6) proposed in Section 2.2 and add the model complexity penalty term to avoid overfitting. Thus, the new objective function on $(x_i^{(k)}, a_i^{(k)}, r_i^{(k)})$ becomes

$$\sum_{i=1}^{n} \sum_{k=1}^{K-1} \frac{|r_i|}{P(a_i|x_i)} \left[I(r_i \geq 0)\left[1 - a_i^{(k)} f(x_i^{(k)})\right]_+ + I(r_i < 0)\left[1 + a_i^{(k)} f(x_i^{(k)})\right]_+\right] + \lambda||f||^2, \tag{3.9}$$

where $x_i^{(k)}$ is the kth duplication of the ith original subject and $f(x_i^{(k)})$ is the corresponding binary classifier. Similarly, the predicted optimal ITR of the ith subject x_i can be obtained by $\hat{\mathcal{D}}(x_i) = \sum_{k=1}^{K-1} I(f(x_i^{(k)}) > 0) + 1$. In Section 4, we show that our method is Fisher consistent in the sense that the estimate matches $\arg\max_{\mathcal{D}} E(R|X, \mathcal{D})$ asymptotically under certain mild conditions.

To solve the optimization problem in (B.1), we develop an algorithm based on the primal-dual formula for the SVM (Vazirani, 2013). In particular, due to the convexity of the objective function, we reformulate (B.1) into a minimization problem with linear constraints, and then derive the corresponding Lagrange function for the primal and dual problems. To implement the quadratic programming in the dual problems above, we use the open source package CVXOPT based on the Python programming in practice. We discuss situations for both linear and nonlinear decision functions. For the nonlinear case, we apply the kernel learning approach in Reproducing Kernel Hilbert Spaces (RKHS, Kimeldorf and Wahba (1970)). Due to the space limitation, we leave all the details of the algorithm into the supplemental material.

3.4 Statistical Learning Theory

In this section, we show Fisher consistency of the estimated ITR, the monotonic property of the intercepts, consistency and convergence rate of the risk bound for the estimated ITR using GOWL. We define some essential notation before getting into the details. First, we define the risk associated with 0-1 loss in (3.3) as $\mathcal{R}(f) = \sum_{k=1}^{K-1} \mathcal{R}^{(k)}(f) = E\{\sum_{k=1}^{K-1} \frac{R}{P(A|X)} I(A^{(k)} \neq \text{sign}(f(X^{(k)})))\}$, where $\mathcal{R}^{(k)}(f) = E[\frac{R}{P(A|X)} I(A^{(k)} \neq \text{sign}(f(X^{(k)})))]$ for $k = 1, \cdots, K-1$ and $f(X^{(k)})$ is an ITR associated decision function. According to the 0-1 risk above, we define its Bayes risk as $\mathcal{R}(f^*) = \inf_f \{\mathcal{R}(f) | f : \mathcal{X} \to \mathbb{R}\}$ and the corresponding optimal ITR as $\mathcal{D}^*(X) = \sum_{k=1}^{K-1} I(f^*(X^{(k)}) > 0) + 1$. Correspondingly, we define the $\phi-$risk associated with the surrogate loss in (3.5) as $\mathcal{R}_\phi(f) = \sum_{k=1}^{K-1} \mathcal{R}_\phi^{(k)}(f) = E\{\sum_{k=1}^{K-1} \frac{|R|}{P(A|X)} [\phi(A^{(k)} f(X^{(k)}), R)]\}$ where $\mathcal{R}_\phi^{(k)}(f) = E[\frac{|R|}{P(A|X)} \phi(A^{(k)} f(X^{(k)}), R)]$ and $\phi(u, r) = I(r \geq 0)[1-u]_+ + I(r < 0)[1+u]_+$. We also define the minimal $\phi-$risk as $\mathcal{R}_\phi(f_\phi^*) = \inf_f \{\mathcal{R}_\phi(f) | f : \mathcal{X} \to \mathbb{R}\}$ and the corresponding surrogate optimal ITR as $\mathcal{D}_\phi^*(X) = \sum_{k=1}^{K-1} I(f_\phi^*(X^{(k)}) > 0) + 1$. Furthermore, we assume that the number of treatment levels K is finite in the following discussions. All the details of theorem proofs are included in the supplemental material.

3.4.1 Fisher Consistency

Recall that the optimal ITR always corresponds to the treatment that can produce the best expected clinical reward, i.e. $\mathcal{D}^*(x) = \arg\max_{k\in\mathcal{A}} [E(R|X=x, A=k)]$. To derive Fisher consistency, we need to show that by using the suggested loss ϕ to replace the 0-1 loss, the surrogate optimal ITR $\mathcal{D}^*_\phi(x)$ matches $\mathcal{D}^*(x)$. We divide the process into two steps: first, we show in Lemma 3.4.1 that $\mathcal{D}^*_\phi(x) = \mathcal{D}^*(x)$ for binary treatments. Second, the result can be generalized into ordinal treatments with an additional assumption in Theorem 3.4.1.

Lemma 3.4.1. *When $A \in \{1, 2\}$, for any measurable function f, we have $\mathcal{D}^*_\phi(x) = I(f^*_\phi(X^{(1)}) > 0) + 1 = \mathcal{D}^*(X)$, where f^*_ϕ is the minimizer of $\mathcal{R}_\phi(f)$ in $\phi-$risk with $K = 2$.*

To prove Lemma 3.4.1, one can show that the minimizer f^*_ϕ should be within the range of $[-1, 1]$ and then we can show $\text{sign}(f^*_\phi) = \text{sign}(E[R|A=2] - E[R|A=1])$.

Theorem 3.4.1. *When $A \in \{1, \cdots, K\}$ and K is an integer greater than 2, we have $\mathcal{D}^*_\phi(x) = \sum_{k=1}^{K-1} I(f^*_\phi(X^{(k)}_i) > 0) + 1 = \mathcal{D}^*(X)$ under the assumption that $E(R|X, A > k) > E(R|X, A \le k)$ if and only if $\mathcal{D}^*(X) \ge k$ for $k = 1, \cdots, K-1$, where f^*_ϕ is a measurable function that minimizes $\mathcal{R}_\phi(f)$ in $\phi-$risk.*

To show Theorem 3.4.1, we start from the conclusion in Lemma 3.4.1 and obtain $\mathcal{D}^*(X)$ by summing all binary decision functions across $k = 1, \cdots, K-1$. The assumption on $E(R|X)$ in Theorem 3.4.1 is necessary when one needs to accumulate all $f^*_\phi(X^{(k)})$ correctly to reach $\mathcal{D}^*(X)$. Essentially, this assumption requires the reward curve decreases at a similar rate when the treatment is away from the optimal one at both sides of its peak (see the R1 curve in Figure 3.2). According to this assumption, each binary surrogate classifier $I(f^*_\phi(X^{(k)}) > 0)$ matches the corresponding optimal binary classifier $I(f^*(X^{(k)}) > 0)$ in each binary subproblem. We would like to point out that even when the assumption fails in real applications, Fisher consistency could still be guaranteed by modifying the data duplication strategy into $r^{(k)}_i = r_i$ only if $a_i \in \{k, k+1\}$. The modified strategy uses partial data in each binary treatment subproblem so that we only need the reward curve to be monotonically

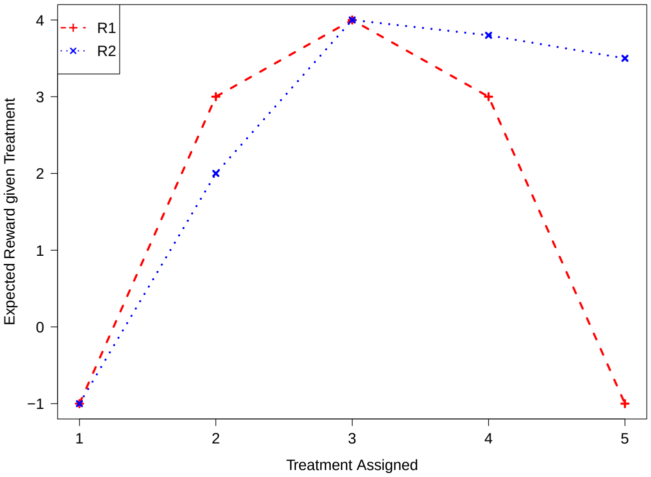

Figure 3.2: Examples when the assumption holds and fails for Theorem 3.4.1. In this case, $\mathcal{D}^*(X) = 3$ and the assumption in Theorem 3.4.1 holds for curve R1 but fails for curve R2.

decreasing when the assigned treatment moves away from the true optimal treatment $\mathcal{D}^*(X)$. Note that the modified duplication strategy uses subsets of data and may work well for large sample problems. In particular, it is well suited for the cases where there is a sufficient sample size within each treatment group.

3.4.2 Monotonic Boundary

In Section 3.3, we discussed that the decision function $f(X^{(k)})$ can be expressed as $g(X)+b_k$ for both linear and nonlinear cases. The following theorem shows that the intercepts b_k for $k = 1, \cdots, K-1$ can have the monotonic property under certain assumptions so that the resulting rule has no contradiction. Note that it is only meaningful to consider the monotonic property of the intercepts when $K \geq 3$.

Theorem 3.4.2. *If we write the decision function as $f(X^{(k)}) = g(X) + b_k; k = 1, \cdots, K-1$, and assume that the signs of $E[R|A=k]$ are the same for $k = 1, \cdots, K$, then the optimal solution (g, b) for minimizing the ϕ-risk $\mathcal{R}_\phi(f)$ has monotonic b values. In particular, we have $b_k > b_{k+1}$ ($b_k < b_{k+1}$) for $k = 1, \cdots, K-2$ when $E[R|A=k] > 0$ (< 0) for $k = 1, \cdots, K$.*

To understand the condition in Theorem 3.4.2, note that the value of $E[R|A=k]$ is the average benefit patients receive from taking the treatment k. Violating the conditions in

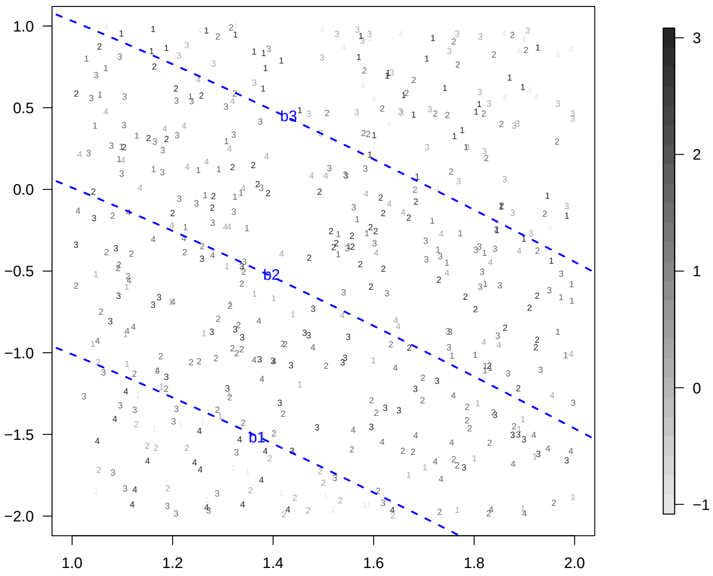

Figure 3.3: A simulation example explaining how the monotonic property works. In this case, there are two covariates and four treatment levels where the numbers represent the actually assigned treatments. The gray-scale of the numbers indicates the clinical outcome value and a darker color means a larger reward (see the gray-scale strip). The dashed lines indicate how the optimal ITR boundaries split the input space. When $E\left[R|A=2\right]$ reduces to a certain negative value that has a large magnitude, the margin between the estimated b_1 and b_2 boundaries would decrease to zero and then the monotonic property no longer holds.

Theorem 3.4.2 could destroy the monotonic order of b. For example, when $E\left[R|A=m\right]$ for certain $m \in \{2, \cdots, K-1\}$ is observed to be negative while all the other $E[R|A=k]$ are positive, no patient will be assigned with the treatment m as the optimal treatment and the corresponding b would not be monotonic. The covariate information has been integrated out in the condition because a monotonic boundary can guarantee a minimal risk which only depends on the decision function f but not the X by definition.

To further illustrate the condition in Theorem 3.4.2, Figure 3.3 demonstrates a simulated example with two covariates and four treatment levels where the numbers represent the actually assigned treatments. The gray-scale of the numbers indicates the clinical outcome value and a darker color means a larger reward (see the gray-scale strip). The dashed lines indicate how the optimal ITR boundaries split the input space into four regions where the optimal treatment rule changes from $\mathcal{D}^*(x) = 1$ in the top right area to $\mathcal{D}^*(x) = 4$ in the bottom left. Starting with all positive $E[R|A=k]$, if we decrease $E[R|A=2]$ while keeping the other $E[R|A=k]$ values constant, the margin between b_1 and b_2 will be narrower. Such a change indicates that a smaller proportion of the population will be assigned $A=2$ as the optimal treatment. In the extreme case where $E[R|A=2]$ is negative and small enough compared with the other two treatments, the boundaries of b_1 and b_2 will overlap, violating the monotonic property. Under this circumstance, the rewards can contradict the ordinality of the treatments.

Finally, we would like to emphasize that Theorem 3.4.2 only presents a sufficient condition for the monotonicity of the intercepts. In particular, the signs of $E[R|A=1]$ and $E[R|A=K]$ do not impact the monotonicity of the intercepts. For example, if $E[R|A=1]$ is the only non-positive one among all $E[R|A=k], k=1,\cdots,K$, the first boundary b_1 would need to be at an extreme large value to prevent any subject from choosing $A=1$ as the optimal treatment. Such a scenario is not interesting in practice despite the fact that the monotonicity still holds with $b_k > b_{k+1}$ for $k=1,\cdots,K-2$.

3.4.3 Excess 0-1 Risk and Excess $\phi-$Risk

The following theorem shows that for any decision function f, the excess risk of f under the 0-1 loss, $\mathcal{R}(f)-\mathcal{R}(f^*)$, can be bounded by the excess risk of f under the surrogate loss, $\mathcal{R}_\phi(f)-\mathcal{R}_\phi(f^*_\phi)$.

Theorem 3.4.3. *For any measurable function $f:\mathcal{X}\to\mathbb{R}$ and any probability distribution of (X,A,R), we have $\mathcal{R}_\phi(f)-\mathcal{R}_\phi(f^*_\phi)\geq\mathcal{R}(f)-\mathcal{R}(f^*)\geq 0$.*

Because some of our theoretic discussions are based on the $\phi-$risk, it is necessary to first show how the 0-1 loss risk $\mathcal{R}(f)$ could be controlled accordingly. The proof of Theorem 3.4.3 uses the idea of partition and integration by dividing $\mathcal{R}_\phi(f)$ into $K-1$ parts with $\mathcal{R}_\phi(f)=\sum_{k=1}^{K-1}\mathcal{R}_\phi^{(k)}(f)$. For each part $\mathcal{R}_\phi^{(k)}(f)$, we generalize the idea of Zhao et al. (2012) and make use of the risk bound theories in Bartlett et al. (2006) to derive the relationship between the two excess risks.

3.4.4 Consistency and Convergence Rate

Denote $\hat{f}_n$ as the sample solution for our proposed GOWL as a minimizer of (B.1) with $f \in \mathcal{H}$. We next discuss the consistency of $\phi-$risk with $\hat{f}_n$ in the following Theorem 3.4.4.

Theorem 3.4.4. *Assume the tuning parameter λ_n is selected such that $\lambda_n \to 0$ and $n\lambda_n \to \infty$. Then for any distribution of (X, A, R), we have that $\mathcal{R}_\phi(\hat{f}_n) \to \inf\limits_{f \in \bar{\mathcal{H}}} \mathcal{R}_\phi(f)$ in probability as $n \to \infty$, where $\hat{f}_n$ is the empirical minimizer of (B.1) and $\bar{\mathcal{H}}$ denotes the closure of a selected space $\mathcal{H}$.*

By theorem 3.4.4, minimization of the $\phi-$risk depends on the selection of $\mathcal{H}$. Additionally, if f^*_ϕ, the global minimizer of $\phi-$risk, belongs to the closure of $\limsup\limits_{n\to\infty}\mathcal{H}$, where $\mathcal{H}$ can depend on n, then we have $\inf\limits_{f \in \bar{\mathcal{H}}} \mathcal{R}_\phi(f) = \mathcal{R}_\phi(f^*_\phi)$ and thus $\liminf\limits_{n\to\infty}\mathcal{R}_\phi(\hat{f}_n) = \mathcal{R}_\phi(f^*_\phi)$ in probability. This result will lead to $\liminf\limits_{n\to\infty}\mathcal{R}(\hat{f}_n) = \mathcal{R}(f^*)$ in probability by Theorem 3.4.3. In particular, the above conditions are met when $\mathcal{H}$ is an RKHS with the Gaussian kernel of which the bandwidth decreases to zero as $n \to \infty$ (see Zhao et al. (2012) for a related discussion).

In the next theorem, we discuss the convergence rate of the excess 0-1 risk $\mathcal{R}(\hat{f}_n) - \mathcal{R}(f^*)$ based on the geometric noise assumption for each measure $P^{(k)}$ introduced in Steinwart and Scovel (2007b). For our problem, we define the decision boundary for the optimal ITR as $\{2\eta(x^{(k)})-1 = 0\}$ in each classification subproblem between $\{1, \cdots, k\}$ and $\{k+1, \cdots, K\}$ for $k = 1\cdots, K-1$, where $\eta(x^{(k)}) = \frac{E[R|X^{(k)}=x^{(k)},A^{(k)}=1]-E[R|X^{(k)}=x^{(k)},A^{(k)}=-1]}{E[R|X^{(k)}=x^{(k)},A^{(k)}=1]+E[R|X^{(k)}=x^{(k)},A^{(k)}=-1]} + \frac{1}{2}$. Furthermore, we define the purity of the corresponding data set as $\Delta(x^{(k)}) = |2\eta(x^{(k)}) - 1|$. Note that $\Delta(x^{(k)})$ can be viewed as a measure of closeness of $x^{(k)}$ to the corresponding kth decision boundary. Using these notations, we state the geometric noise assumption in our problem for each duplicate k for $k = 1, \cdots, K-1$ as follows: Let $X^{(k)} \in \mathbb{R}^p$ be compact, we define that the measure $\mathcal{P}^k$ has geometric noise exponent $q_k > 0$ if there exists a constant $C_k > 0$ such that $E[|2\eta(X^{(k)}) - 1|\exp(-\frac{\Delta(X^{(k)})^2}{t})] \leq C_k t^{q_k p/2}$, for $t > 0$. According to Steinwart and Scovel (2007b), the geometric noise exponent describes the concentration and the noise level of the data generating distribution near the decision boundary. As we will discuss further, the geometric noise exponent q_k of the distribution of $(X^{(k)}, A^{(k)}, R^{(k)})$ depends on how the density of the data set decreases when the point gets close to the boundary. In the extreme case, q_k can be arbitrarily large when $\eta(x^{(k)})$ is continuous and $\Delta(x^{(k)}) > \delta > 0$ for some constant $\delta > 0$ (i.e., the distinctly separable case). In addition to the geometric noise condition, we also consider the RKHS associated with the Gaussian kernel as in Steinwart and Scovel (2007b) in Theorem 3.4.5. We use σ_n to denote the kernel bandwidth for the Gaussian kernel.

Theorem 3.4.5. *Suppose that the distribution of $(X^{(k)}, A^{(k)}, R^{(k)})$ satisfies the geometric noise assumption with exponent $q_k \in (0, \infty)$ for $k = 1, \cdots, K-1$. Then for any $\delta > 0$ and $\nu \in (0, 2)$, there exists a C, which depends on ν, δ, the dimension p, and $P(A|X)$, such that for $\forall \tau \geq 1$ and $\sigma_n = \lambda_n^{-\frac{1}{(q+1)p}}$ for the Gaussian kernel, we have $Pr^*(\mathcal{R}(\hat{f}_n) \leq \mathcal{R}(f^*) + \epsilon) \geq 1 - e^{-\tau}$, where $q = \arg\max_{q_k} \lambda_n^{q_k/(q_k+1)}$, Pr^* denotes outer probability and $\epsilon = C(\lambda_n^{-\frac{2}{2+\nu}+\frac{(2-\nu)(1+\delta)}{(2-\nu)(1+q)}} n^{-\frac{2}{2+\nu}} + \frac{\tau}{n\lambda_n} + \lambda_n^{\frac{q}{q+1}})$.*

Taking a closer look at the ϵ expression in Theorem 3.4.5, we can find that the first two terms can be treated as the bound for the stochastic error, whereas the last term is an error bound for the noise associated with the corresponding RKHS. There is a trade off between the two components. For example, the noise bound term will decrease and the stochastic error will inflate if the RKHS is selected to be more complex. Based on the ϵ expression, one can tell that an optimal choice of λ_n is $n^{-\frac{2(1+q)}{(4+\nu)q+2+(2-\nu)(1+\delta)}}$ and the corresponding rate of the excess risk can be expressed as $\mathcal{R}(\hat{f}_n) - \mathcal{R}(f^*) \leq O_p(n^{-\frac{2q}{(4+\nu)q+2+(2-\nu)(1+\delta)}})$. By the geometric noise exponent property, such q can be sufficiently large when different optimal treatment groups are separated well enough just as in the distinctly separable case we discuss previously. In this way, the rate of convergence can be almost $O_p\left(n^{-1/2}\right)$ when we let δ and ν be small.

3.5 Simulation Study

In this section, we conduct simulation studies with both linear and nonlinear ITR boundaries to assess the finite sample performance of the proposed GOWL. In both cases, we first generate a training set with the covariates $X_1, \cdots, X_p$ from a uniform distribution $U(-1, 1)$ and the treatment A from a discrete uniform distribution ranging from 1 to K, where $K = 2, 3, 5$ and 7 respectively. In each example, X and A are independent. For each K, we choose two training sample sizes to represent the small and large sample scenarios. The reward R follows $N(Q(X, A), 1)$ with $Q(X, A) = \mu(X) + t(X, A)$, where $\mu(X)$ is the overall effect of X and $t(X, A)$ is the interaction that determines the true optimal treatment. We maintain approximately 70% of the generated rewards as positive. For simplicity in simulation studies, except for the training set, we also generate an independent equal-size tuning set and a much larger testing set (10 times as large as the training set) with the same variables in each scenario. The tuning set is used to select the optimal tuning parameter λ and the Gaussian kernel bandwidth σ_n. In particular, we choose λ from $\{\frac{i}{n}; i = 0.1, 1, 10, 100, 500\}$ and σ_n from $\{0.1, 1, 10\}$, where n is the tuning size. The testing set is used to check the prediction performance of the models. For real data application, cross-validations are used for tuning parameter selection.

For comparisons, we manually modify some existing methods so that they can be used to detect the ITR for ordinal treatments. Specifically, we pick OWL and l_1 penalized least squares including one way covariate-treatment interaction terms (PLS-l_1, Qian and Murphy (2011)) to conduct a series of pairwise comparisons between $\{1, \cdots, k\}$ and $\{k+1, \cdots, K\}$ for $k = 1, \cdots, K-1$. The final estimated optimal treatment is obtained by summing through all pairwise prediction results. For OWL, the original reward outcome is shifted to be all positive. For both OWL and GOWL, both the linear kernel (OWL-Linear and GOWL-Linear) and the Gaussian kernel (OWL-Gaussian and GOWL-Gaussian) are used for estimating the classifier. We select two criteria to evaluate the model performance: the misclassification rate (MISC), and the MSE of the value function (Value), i.e., the mean of squares of the difference between the Values under the estimated ITR versus under the optimal ITR for all replicates. Smaller values are preferred for both criteria by definition. In particular, the first criterion measures the proportion of correct treatment assignments. The second criterion is a more comprehensive measure on how close the estimated ITR is to the true optimal ITR. The value function estimate is defined as

$\mathbb{P}_n^* \left[\sum_{k=1}^{K-1} I\left(A^{(k)} = \mathcal{D}(X^{(k)})\right) R/P(A)\right] / \mathbb{P}_n^* \left[\sum_{k=1}^{K-1} I\left(A^{(k)} = \mathcal{D}(X^{(k)})\right) / P(A)\right]$, where $\mathbb{P}_n^*$ denotes the empirical average of the testing data set.

3.5.1 Linear Boundary Examples

We consider the following four scenarios with $\mu(X)$ and $t(X, A)$ defined as,

1. $K = 2$: $\mu(X) = 1 + X_1 + X_2 + 2X_3 + 0.5X_4$ and $t(X, A) = 1.8\,(0.3 - X_1 - X_2)\,(2A - 3)\,;$

2. $K = 3$: $\mu(X) = 2 + 2X_1 + X_2 + 0.5X_3$, $t(X, A) = 4\sum_{i=1}^{3} I(g(X) \in (b_{i-1}, b_i])(2 - |A - i|)$, $g(X) = -X_1 + 2X_2 + X_3 + 0.6X_4 - 1.5(X_5 + X_6)$, $b_0 = -\infty$, $b_1 = -0.5$, $b_2 = 1$, $b_3 = \infty$;

3. $K = 5$: $\mu(X) = 2 + 2X_1 + X_2 + 0.5X_3$ and $t(X, A) = 4\sum_{i=1}^{5} I\left(g(X) \in (b_{i-1}, b_i]\right)(2 - |A - i|)$, where $g(X) = -X_1 + 2X_2 + X_3 + 0.6X_4 - 1.5(X_5 + X_6)$, $b_0 = -\infty$, $b_1 = -1.9$, $b_2 = -0.5$, $b_3 = 0.5$, $b_4 = 1.7$ and $b_5 = \infty$;

4. $K = 7$: $\mu(X) = 2 + 2X_1 + X_2 + 0.5X_3$ and $t(X, A) = 4\sum_{i=1}^{7} I\left(g(X) \in (b_{i-1}, b_i]\right)(2 - |A - i|)$, where $g(X) = -X_1 + 2X_2 + X_3 + 0.6X_4 - 1.5(X_5 + X_6)$, $b_0 = -\infty$, $b_1 = -2.1$, $b_2 = -1.2$, $b_3 = -0.4$, $b_4 = 0.4$, $b_5 = 1$, $b_6 = 2.1$ and $b_7 = \infty$.

The simulated data sets satisfy that the true boundaries are parallel to each other. The cut-off values b are set to encourage an evenly distributed true optimal treatment from 1 to K in samples. Furthermore, $t(X, A)$ are set to ensure that the reward outcome decreases symmetrically when the assigned treatment moves away from the optimal treatment towards high or low levels. The training sample sizes are listed in Table 3.1, which range from 30 to 500. We repeat the simulation 50 times and present the testing prediction results in Table 3.1.

Methods		PLS-l_1		OWL-Linear		OWL-Gaussian		GOWL-Linear		GOWL-Gaussian	
K	n	MISC	Value	MISC	Value	MISC	Value	MISC	Value	MISC	Value
2	30	**0.117**	**0.128**	0.198	0.464	0.196	0.454	0.155	0.166	0.122	0.138
		(0.107)	(0.111)	(0.168)	(0.327)	(0.148)	(0.290)	(0.121)	(0.133)	(0.087)	(0.145)
	300	0.130	0.018	0.055	0.081	0.105	0.084	0.077	0.014	**0.032**	**0.012**
		(0.045)	(0.005)	(0.024)	(0.054)	(0.073)	(0.036)	(0.034)	(0.009)	(0.011)	(0.006)
3	30	0.269	0.450	0.425	0.620	0.422	0.633	**0.220**	**0.270**	0.235	0.273
		(0.152)	(0.288)	(0.349)	(0.413)	(0.350)	(0.315)	(0.150)	(0.198)	(0.157)	(0.118)
	300	0.285	0.044	0.261	0.398	0.243	0.468	**0.032**	**0.028**	0.055	0.029
		(0.071)	(0.019)	(0.165)	(0.271)	(0.176)	(0.364)	(0.021)	(0.012)	(0.043)	(0.013)
5	50	0.608	0.616	0.589	0.878	0.355	0.758	0.351	0.290	**0.337**	**0.267**
		(0.241)	(0.432)	(0.330)	(0.320)	(0.329)	(0.345)	(0.256)	(0.175)	(0.229)	(0.145)
	500	0.436	0.272	0.303	0.305	0.344	0.295	0.163	0.042	**0.118**	**0.030**
		(0.122)	(0.129)	(0.263)	(0.319)	(0.184)	(0.283)	(0.095)	(0.033)	(0.095)	(0.018)
7	50	0.672	1.609	0.707	0.910	0.721	1.625	**0.414**	0.404	0.420	**0.375**
		(0.327)	(0.855)	(0.317)	(0.480)	(0.303)	(0.575)	(0.282)	(0.244)	(0.290)	(0.308)
	500	0.587	0.371	0.491	0.364	0.522	0.365	**0.210**	**0.098**	0.227	0.103
		(0.247)	(0.280)	(0.247)	(0.282)	(0.179)	(0.219)	(0.161)	(0.072)	(0.145)	(0.040)

Table 3.1: Results of linear boundary examples: K represents the number of treatment levels; n represents the training set size; the MISC column gives the mean and standard deviation of the misclassification rate; and the Value column gives the mean and standard deviation of the value function MSE. PLS$-l_1$ represents penalized least squares including covariate-treatment interactions with l_1 penalty (Qian and Murphy, 2011); OWL represents the outcome weighted learning and GOWL represents the proposed generalized outcome weighted learning. In each scenario, the model producing the best criterion is in bold.

As shown in Table 3.1, the proposed GOWL reveals competitive accuracy rate in predicting

ITR for testing data sets in most of the cases. In general, when both the sample size n and number of treatment classes K are small, the PLS-l_1 can be competitive because the true decision boundary is linear. However, when K increases to 5 or 7, GOWL outperforms all the other methods, especially in terms of the value function of the estimated ITR. Moreover, for the binary treatment with small n, GOWL performs comparable to PLS-l_1 whereas OWL shows relatively worse results with a larger MSE for the corresponding value function. When the number of treatment category K increases, the advantage of GOWL becomes more significant in terms of both the misclassification and value function comparisons. For example, GOWL can maintain an average misclassification rate as 21% even when K increases to 7. One reason can be that the parallel decision boundary assumption of GOWL matches the underlying truth and this can lead to robust estimate even when K is large. Furthermore, under the true linear boundary cases, the performance of GOWL with the Gaussian kernel can be comparable to the case with the linear kernel when a proper tuning parameter is used. Thus a flexible nonparametric estimation procedure can be considered in practice when there is no prior knowledge about the shape of the underlying ITR boundaries.

3.5.2 Nonlinear Boundary Examples

We also assess the performance of GOWL using nonlinear boundary examples and compare it with other methods used previously. The results are provided in Table 3.2, where we find that none of the method performs well when the sample size is very small because the true boundaries have complex structures. When n becomes large, GOWL with the Gaussian kernel outperforms PLS-l_1 in all cases due to PLS-l_1's wrong model specification. GOWL-Gaussian shows better performance than OWL-Gaussian in terms of both classification accuracy and value functions. More details are provided in the supplemental material.

Methods		PLS-l_1		OWL-Linear		OWL-Gaussian		GOWL-Linear		GOWL-Gaussian	
K	n	MISC	Value	MISC	Value	MISC	Value	MISC	Value	MISC	Value
2	30	0.496	2.107	0.412	1.791	**0.353**	**1.301**	0.438	1.846	0.423	1.580
		(0.130)	(0.366)	(0.086)	(0.574)	(0.091)	(0.580)	(0.074)	(0.300)	(0.069)	(0.548)
	300	0.396	1.983	0.374	1.815	0.184	0.110	0.339	1.510	**0.089**	**0.015**
		(0.08)	(0.134)	(0.076)	(0.357)	(0.06)	(0.096)	(0.045)	(0.438)	(0.024)	(0.005)
3	30	0.461	1.191	0.470	2.640	0.468	1.574	0.403	1.214	**0.370**	**0.909**
		(0.225)	(0.347)	(0.107)	(0.538)	(0.106)	(0.608)	(0.094)	(0.445)	(0.066)	(0.218)
	300	0.345	0.645	0.361	1.495	0.362	0.861	0.224	0.403	**0.146**	**0.048**
		(0.18)	(0.239)	(0.092)	(0.527)	(0.089)	(0.448)	(0.08)	(0.136)	(0.04)	(0.018)
5	50	0.578	0.690	0.642	1.586	0.624	2.020	**0.521**	1.059	0.525	**0.950**
		(0.226)	(0.519)	(0.483)	(0.713)	(0.179)	(1.073)	(0.124)	(0.588)	(0.109)	(0.326)
	500	0.548	0.316	0.468	1.812	0.396	1.348	0.412	0.358	**0.246**	**0.185**
		(0.28)	(0.028)	(0.149)	(1.035)	(0.133)	(0.384)	(0.193)	(0.078)	(0.119)	(0.136)
7	50	0.727	3.489	0.707	4.172	0.716	2.412	0.590	0.695	**0.563**	**0.503**
		(0.319)	(0.989)	(0.578)	(0.923)	(0.266)	(0.685)	(0.178)	(0.561)	(0.163)	(0.388)
	500	0.665	2.754	0.722	1.757	0.541	1.414	0.610	1.378	**0.445**	**0.795**
		(0.287)	(0.798)	(0.238)	(0.424)	(0.21)	(0.253)	(0.244)	(0.146)	(0.168)	(0.17)

Table 3.2: Results of nonlinear boundary examples: K represents the number of treatment levels, n represents the training set size, MISC column gives the mean and standard deviation of the misclassification rate and Value column gives the mean and standard deviation of the value function MSE

So far, our focus has been on examples with parallel boundaries. We would like to point out that the proposed GOWL could also work well when the parallel assumption of the true boundaries does not hold. Under these circumstances, one can consider using nonlinear learning so that the estimated boundaries would be flexible enough to approach the underlying true boundaries. To illustrate the idea, we use a case with $n = 300$, $p = 2$ and $K = 3$ and show that the estimated boundaries produced by GOWL can capture the underlying pattern of the optimal ITR well. More details are given in the supplemental material.

3.6 Dataset Applications

We apply GOWL to an irritable bowel syndrome clinical data set and a type 2 diabetes mellitus clinical observational study to assess its performance in real studies.

3.6.1 Irritable Bowel Syndrome Dataset

This dataset consists of a dose ranging trial that aims to develop a treatment for irritable bowel syndrome (IBS) (see Biesheuvel and Hothorn (2002) for more details). The clinical study enrolled four active treatment arms, corresponding to doses 1, 2, 3, 4 and placebo. The primary endpoint is a baseline adjusted abdominal pain score with larger values corresponding to a better treatment effect. There are 369 patients completing the study, with an almost balanced allocation across the groups of different doses. The final data set only contains three variables: patients' gender, treatment, and the adjusted abdominal pain score. Approximately 72% of the observed pain scores are greater than 0.

Given the small covariate dimension, we merge doses 1 and 2 together as the low dose group and merge doses 3 and 4 together as the high dose group. The average adjusted abdominal pain scores of the total data set is 0.475, with standard deviation equal to 0.769. To estimate the optimal ITR, we apply methods including PLS-l_1, OWL-Gaussian, and GOWL-Gaussian, and modify the first two methods in the same way as in the simulation study. As to the evaluation criterion, we calculate the empirical value functions with the following cross-validation strategy. In particular, we randomly partition the dataset into 5 equal-sized parts, train the model based on every 4 of them, and predict the value function using the remaining part. We repeat the partition 50 times and the corresponding means and standard deviations of the predicted value functions are 0.491(0.029), 0.503(0.004), and 0.537(0.011) for PLS-l_1, OWL-Gaussian, and GOWL-Gaussian.

The result shows that GOWL returns the highest predicted value function with a moderately low standard deviation. By reassigning the treatment, GOWL could improve the predicted value function by approximately 13%. Furthermore, as to the estimated optimal treatment assignment, PLS-l_1 suggests the optimal treatment to be either placebo or low dose. OWL assigns almost all the patients to the low dose group whereas GOWL suggests about 60% patients in high dose and 40% in low dose. In particular, around 70% patients are female for those recommended to be in high dose group. This conclusion appears consistent to what Biesheuvel and Hothorn (2002) reported.

3.6.2 Type 2 Diabetes Mellitus Clinical Observational Study

In this section, we apply the proposed method to a type 2 diabetes mellitus (T2DM) observational study to assess its performance in real life data application. This study includes people with T2DM during 2012-2013, from clinical practice research datalink (CPRD) (Herrett et al., 2015). Three anti-diabetic therapies have been considered in this study: glucagon-like peptide-1 (GLP-1) receptor agonist, long-acting insulin only, and a regime including short-acting insulin. The primary target variable is the change of HbA1c before and after the treatment, and seven clinical factors are used including age, gender, ethnicity, body mass index, high-density lipoprotein cholesterol (HDL), low-density lipoprotein cholesterol (LDL) and smoking status. In total, 634 patients satisfying aforementioned requirements are while around 5% have complete observations

To handle the missing data issue before analysis, we first remove all the covariates with missing proportions greater than 70%. Then, we conduct a t test for each covariate to detect whether its missing pattern impacts the mean of outcome significantly. According to the Bonferroni multiple-testing adjusted p value, we remove all of the covariates with insignificant test results. For the continuous variables with significant test p values, we remove all of their incomplete observations. For categorical covariates having significant test results, we relabel the missing value as a new class when encoding the covariate. After the data preprocessing, there are 10 covariates with 142 observations in total.

Similar to the previous analysis, we apply PLS-l_1, OWL-Linear, OWL-Gaussian, GOWL-

Linear, and GOWL-Gaussian to estimate the ITR with the first three methods modified in the same way. We use the inverse value of the HbA1c change as the reward in estimating the ITR since a smaller HbA1c is desired. In order to obtain the propensity score $P(A|X)$ before using OWL and GOWL, we fit an ordinal logistic model with the cleaned data set using the treatment as the response and all 10 covariates as predictors. As to the criterion, we calculate the predicted value function using the same formula as in the simulation study over 50 replications of 5-fold cross-validation. Table 3.3 summarizes the means and standard deviations of the empirical value functions from the training and validation sets.

Model	Training	Testing
PLS-l_1	2.257 (0.001)	2.206 (0.059)
OWL-Linear	2.335 (0.017)	2.305 (0.072)
OWL-Gaussian	2.456 (0.011)	2.285 (0.049)
GOWL-Linear	2.378 (0.047)	2.332 (0.095)
GOWL-Gaussian	**2.486 (0.025)**	**2.383 (0.060)**

Table 3.3: Analysis Results for the T2DM Dataset. Empirical Value Function Results using 5-fold Cross-Validation with 50 Replications are reported. For comparison, the original assigned treatment strategy has the value function 2.205 and the randomly assigned treatment method has average value function 2.104 in testing sets with standard deviation 0.131.

To further demonstrate how much improvement the proposed method obtain, we also calculate the value function with the original treatments and the average value function with treatment being randomly assigned 50 times. The empirical means of the value functions are 2.205 and 2.104 with the standard deviation for the random assignment to be 0.131.

According to Table 3.3, GOWL achieves both the highest mean and the lowest standard deviation of the empirical value function in the prediction results. In addition, the three linear models are outperformed by the nonlinear models, possibly due to their suboptimal model specification for this application. As to the distribution of estimated optimal treatment assignments, the PLS-l_1 only includes long-acting insulin as the optimal treatment. OWL-Gaussian chooses approximately 83% of the patients to be in either the GLP-1 group or short-acting insulin group. GOWL-Gaussian assigns approximately 50% patients into the short-acting insulin group while assigning the rest into one of the other two groups in a more even way. This conclusion is consistent to some literature on short-acting insulins, which shows the benefit of reducing HbA1c (Holman et al., 2007). Moreover, it is worth noting that prandial insulins also have elevated risk of hypo and weight gain, which are crucial safety and efficacy measurements for diabetes patients. Our study only considers HbA1c change as the outcome. One can consider more composite metrics, including HbA1c change, hypo events, and weight gain, to find the corresponding optimal treatment rules.

3.7 Conclusion

In this chapter, we use a modified loss function to improve the performance of OWL and then generalize OWL to solve the ordinal treatment problems. In particular, the proposed GOWL converts the optimal ordinal treatment finding problem into multiple optimal binary treatment finding subproblems under certain restrictions. The estimating process produces a group of estimated optimal treatment boundaries with monotonic intercepts that never cross. Such boundaries can make the ITR estimates more stable and interpretable in practice.

There are various possible extensions for GOWL that could be considered. For example, one can incorporate a variable selection component into the objective function. In the literature, Xu et al. (2015) proposed variable selection in the linear case and Zhou et al. (2017) extended the idea for kernel learning. According to their ideas, one nature extension for GOWL is to include an l_1 penalty of the parameters into its optimization problem. In this way, variable sparsity could be achieved simultaneously when detecting the optimal ITR. The second possible extension that might improve the performance of GOWL is to modify the outcome in its optimization problem which is originally the reward R. Specifically, according to Fu et al. (2016), one can consider fitting a model with R versus X and then put the residuals as the outcome in the optimization problem of GOWL instead. Such an adjustment is likely to further improve the ITR estimation results for some finite sample scenarios.

CHAPTER 4: IDENTIFYING HETEROGENEOUS EFFECT THAT USES LATENT SUPERVISED CLUSTERING WITH ADAPTIVE FUSION

4.1 Introduction

In clinical research, precision medicine aims to develop the optimal treatment for each individual according to subject's personal characteristics. The motivation originates from the findings that different groups of patients can respond dramatically different to the same health care intervention due to specific body mechanism. Take a drug market product for example, Tamoxifen is developed to prevent and treat breast cancer in women. However, it is proved to be entirely useless for the subpopulation who has the gene CYP2D6 mutated because they are not able to digest the effective ingredients (Ellsworth et al. (2010)). From the example, one can be aware that failure to find the proper subpopulation that the intervention targets at can lead to overkill of effective drugs due to the false negative results obtained by testing efficiency on the whole population data instead. In real life, achieving precision medicine goal is an arduous work because it can take tremendous efforts to detect the targeted subpopulation for certain health care interventions (Brookes et al. (2004); Lagakos (2006)). One important reason attributed to such difficulty is the reality that the primary features distinguishing targeted subpopulation from others are usually either hidden among numerous other collected features or even remain unmeasured. Therefore, it always remains desirable and interesting to develop methods that can automatically detect such subpopulations.

In this chapter, we focus on similar datasets with unobserved subpopulation label as latent supervised learning but plan to address these two drawbacks simultaneously. In particular, we would like to propose a novel exploratory tool, named latent supervised clustering, to estimate the heterogeneous effects at the same time of clustering the samples into subpopulations without much prior knowledge on the underlying boundaries. To achieve these two learning goals, we formulate the regression problem that latent supervised learning discusses as a model that has subject-specific coefficients, which can be treated as the subject-specific relationships between the outcomes and predictors from the observed data. Then we cluster such relationships from the perspective of clustering analysis. This method inherits the advantages of both the latent supervised learning and traditional clustering analysis. On one hand, it does not require much prior knowledge on the latent subpopulation structures and instead let the data orient the learning process. On the other hand, it utilize the information of both predictors and outcomes so that it can be used to fit the predictor effects as well as produce competitive prediction results.

Note that clustering such outcome-predictor relationships can be very challenge because they are not observed directly but can only be derived from the observed data. In this way, one of the most important questions is how to define a distance properly so that the clustering pattern will be encouraged accordingly. At this point, we would like to adapt the idea of convex clustering. Convex clustering formulate the clustering process as a minimization problem with a loss+penalty form with a tuning parameter λ_n balancing the two terms. The loss term is the Euclidean distance between the covariates of each observation $\boldsymbol{x}_i$ and its corresponding subject-specific centroid $\boldsymbol{\mu}_i$. The penalty term includes sum of the fusion penalty between each pair of $(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$ to encourage sparsity of differences. In latent supervised clustering, we formulate the optimization using a similar form of loss+penalty but with different definitions on the loss and penalty. For the loss term, we instead define the distance using the loss function value of the outcome y_i and its fitted value by a certain model $f(\boldsymbol{x}_i|\boldsymbol{\beta}_i)$ with subject-specific parameter $\boldsymbol{\beta}_i$. In this way, a smaller distance of the ith observation indicates better goodness of fit. The model $f(\boldsymbol{x}_i|\boldsymbol{\beta}_i)$ can be either parametric or non-parametric such as smoothing spline while we only discuss linear functions in this chapter, i.e. $f(\boldsymbol{x}_i|\boldsymbol{\beta}_i) = \boldsymbol{x}_i^T\boldsymbol{\beta}_i$. For the subject-specific coefficient $\boldsymbol{\beta}_i$, we assume that observations coming from the same subpopulation share the identical value of $\boldsymbol{\beta}_i$. To encourage such estimation pattern, we impose an adaptive pairwise fusion penalty on $\boldsymbol{\beta}_i$ in the penalty term.

The weights for the adaptive terms are determined by the estimators of $\boldsymbol{\beta}_i$ and needs to be updated every time the value of $\boldsymbol{\beta}_i$ is renewed. In summary, such an optimization formula can lead to maximize the overall goodness of fit at the same time of minimize the heterogeneity within each cluster.

The main contributions of this chapter are as follows. First, we propose a novel machine learning method that aims to identify the heterogeneity by clustering the defined outcome-predictor relationship. We borrow the convex clustering idea but with different loss and penalty terms to encourage statistical consistency properties as well as computational efficiency. Second, we design a novel algorithm to solve the optimization problem. This algorithm has theoretical guarantee showing that the obtain approximate solution converges to the true solution of the underlying optimization model at a competitive rate. We would like to point out that the proposed latent supervised clustering covers the problem that Ma and Huang (2016) discussed as a special case. They focused on the case when the subpopulations can be determined by a variant intercept term of a linear model.

The remainder of the chapter is organized as follows. In Section 4.2, we illustrate formulation of the latent supervised clustering and discuss the corresponding statistical consistency properties of the estimators. In Section 3, we present the proposed accelerated proximal gradient algorithm to solve the optimization problem and show its convergence rate properties. Moreover, we briefly discuss how the starting points of the algorithm are calculated in practice and how the estimated results can be used for predictions. Simulated examples and data applications are presented in Section 4 to demonstrate the performance of the proposed method under finite samples. Some discussions are made in the last section as a conclusion and more technical details on the proofs are left in the appendix.

4.2 Methodology

This section illustrates the idea of latent supervised clustering and formulates the optimization problem. The statistical consistency properties are also presented with the technical proofs in the appendix.

4.2.1 Latent Supervised Clustering

We use the notation $(\boldsymbol{x_i}, y_i)$, $i = 1 \cdots, n$, to represent the ith observation of the collected data set $(X, \boldsymbol{y})$. In particular, $\boldsymbol{x_i}$ is a p dimensional vector that indicates the covariates with heterogeneous effect on the response y_i. Now we consider the model,

$$y_i = f(\boldsymbol{x}_i; \boldsymbol{\beta}_i) + \varepsilon_i, \tag{4.1}$$

where the subject index $i = 1, \cdots, n$, $\boldsymbol{x_i}$ always contains a subject-specific intercepts as the first column, $\boldsymbol{\beta}_i$ is the coefficient vector of $\boldsymbol{x}$ for the ith observation, and ε_i is the noise term that has zero expectation and bounded variance. We further assume that $\boldsymbol{x_i}$, and ϵ_i are independent of each other. To describe the heterogeneity, we let the predictors $\boldsymbol{x_i}$ have subject-specific effect on the response, which means $\boldsymbol{\beta}_i$ can take different values for different indices i. Our goal is, as mentioned previously, to estimate and cluster the n coefficient vectors $\boldsymbol{\beta}_i$ simultaneously, and let the clustering results guide on subpopulation identification. For model (4.1), the important assumption on the subgroup structure is that the value of $\boldsymbol{\beta}_i$ for $i = 1, \cdots, n$ only depends on the underlying subpopulation that the corresponding ith subject belongs to. In other words, if we denote a partition of $\{1, \cdots, n\}$ with $\mathcal{S} = (\mathcal{S}_1, \cdots, \mathcal{S}_K)$ where K represents the number of subpopulations, then the coefficient vector $\boldsymbol{\beta}_i$ of all the subjects from the same latent subgroup, i.e. $\forall i \in \mathcal{S}_m$ for some $m \in \{1, \cdots, K\}$, are supposed to be identical to each other. Note that the true value of K is usually unknown in practice, and this could bring difficulty in estimation as previously discussed. In this chapter, we only consider the class of linear functions, i.e. $f(\boldsymbol{x}_i; \boldsymbol{\beta}_i) = \boldsymbol{x}_i^T \boldsymbol{\beta}_i$. Note that when the linear assumption is too strong, one can extend the function to nonlinear such as using the smoothing spline that $f(\boldsymbol{x}_i; \boldsymbol{\beta}_i) = \sum_{j=1}^m \boldsymbol{\beta}_{ij} g_j(\boldsymbol{x}_i)$, where $g_1, \cdots, g_m$ are basis functions.

To achieve the learning goal of estimation and clustering for the coefficients $\boldsymbol{\beta}_i$, we would like to borrow the idea of the convex clustering (Chi and Lange (2015)) and consider optimization problem as below:

$$\min_{\boldsymbol{\beta}_i} \left\{ Q_n(\boldsymbol{\beta}_i; \lambda_n) \triangleq \sum_{i=1}^{n} \left[\ell(y_i, \boldsymbol{x}_i^T \boldsymbol{\beta}_i) + \lambda_n \sum_{i<j} w_{ij} ||\boldsymbol{\beta}_i - \boldsymbol{\beta}_j||_1 \right] \right\}, \tag{4.2}$$

where ℓ represents a preselected loss function to measure the goodness of fit and the penalty term is the pairwise fusion penalty adapted by certain nonnegative weights w_{ij} to guarantee consistency as well as achieve satisfactory estimation and computational results. For the loss term, we compare two popular options: the check loss (Koenker (2005)) that is used in quantile regression and the quadratic loss that is used in the standard regression. For the penalty term, we use an adaptive pairwise fusion penalty to adjust for the biasness made by the L_1 penalty. In particular, we suggest $w_{ij} = \min\{B_w, \frac{\iota^m_{\{i,j\}}}{||\tilde{\boldsymbol{\beta}}_i - \tilde{\boldsymbol{\beta}}_j||_1}\}$, where $\iota^m_{\{i,j\}}$ is a indicator that the observation j is among i's m-nearest neighbors by Euclidean distance, and $\tilde{\boldsymbol{\beta}}$ is an current estimate of $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1^T, \cdots, \boldsymbol{\beta}_n^T\}^T$ which can start as the local regression coefficients. The upper bound B_w is added in case some pairs of $(\tilde{\boldsymbol{\beta}}_i, \tilde{\boldsymbol{\beta}}_j)$ have values that are too close to each other. Due to the numerous terms of the fusion penalty, this m-nearest neighbors idea can save tremendous computational time when solving (4.2) while maintaining competitive performance. We also find that it can perform better if w_{ij} is updated in each iteration by using the latest estimated $\boldsymbol{\beta}_i$ instead of sticking to the initial values.

In practice, one may find some prior knowledge available to show that certain components in $\boldsymbol{x}_i$ can have homogeneous effect on the outcome y_i, i.e. the coefficients of some predictors remain constant across all the subpopulations. In this case, we recommend conducting latent supervised clustering only for the variables with heterogeneous effect while adjusted for such homogeneous effect. To illustrate this idea with the similar notations, we suppose the collected data is $(\boldsymbol{x}_i, \boldsymbol{z}_i, y_i)$ where $\boldsymbol{z}_i$ is a q dimensional predictor vector known to have homogeneous effect. Then, we can rewrite the model (4.1) into $y_i = f(\boldsymbol{x}_i; \boldsymbol{\beta}_i) + h(\boldsymbol{z}_i; \boldsymbol{\gamma}) + \varepsilon_i$ where $\boldsymbol{\gamma}$ is the same coefficient vector for all the observations and h is a measurable function that can lead to both parametric and nonparametric models. Similar to f, we restrict our coverage to a simple but common case when $h(\boldsymbol{z}_i;\boldsymbol{\gamma}) = \boldsymbol{z}_i^T\boldsymbol{\gamma}$. In this scenario, the optimization problem in (4.2) can be expressed as,

$$\min_{\boldsymbol{\beta}_i}\left\{Q_n(\boldsymbol{\beta}_i;\lambda_n) \overset{\triangle}{=} \sum_{i=1}^{n}\left[\ell(y_i,\boldsymbol{x}_i^T\boldsymbol{\beta}_i + \boldsymbol{z}_i^T\boldsymbol{\gamma}) + \lambda_n\sum_{i<j} w_{ij}||\boldsymbol{\beta}_i - \boldsymbol{\beta}_j||_1\right]\right\}. \tag{4.3}$$

Moving some redundant components from $\boldsymbol{x}_i$ to $\boldsymbol{z}_i$ can be crucial in achieving a good prediction results at the same time of speeding up the computation when the dimension of the predictors is very large. Even through no prior knowledge can be obtained in distinguish z_i from x_i, we find an idea still helpful that is similar to the forward variable screening. In particular, we can start with a parsimonious x_i model and then put one variable from $\boldsymbol{z}_i$ to $\boldsymbol{x}_i$ that boosts the prediction the best. We repeat the process until the model performance is not longer improved significantly by adding more variables into x_i. We leave details of illustrating this idea in the data application section.

As a special case of latent supervised clustering, Ma and Huang (2016) focused on the problem when the subpopulation can be determined a subject-specific intercept. They suggested using a concave fusion penalty in the objective function and applied the alternating direction method of multipliers (ADMM) algorithm to solve the optimization problem. Our method enjoys several advantages even if one extends their method into model (4.1). First, our method has significant computational benefits because (4.3) is still convex and our proposed algorithm does not need to generate the $p \cdot n^2$ additional intermediate parameters that ADMM does. Second, the quadratic loss suggested by Ma and Huang (2016) can be a suboptimal choice due to its sensitivity to the outliers. For example, if a subject actually coming from the first subpopulation is wrongly assigned into the second subpopulation, it can highly impact the coefficient estimates of the second subpopulation when quadratic loss is applied. This can be partly attributed to the fact that the least square estimators have breakdown point to be zero (Huber (2004)). We compare the results of the qudratic loss and check loss and find the latter one can significantly improve the model performance. Third, it is not desirable to penalize all the pairs of $(\boldsymbol{\beta}_i, \boldsymbol{\beta}_j)$ equally. In the ideal case, we hope that large weights are assigned to the pairs coming from the same subpopulation while zero weights are used for those coming from different subpopulations. That is one of the motivations to suggest the adaptive fusion penalty instead and such a proposal could maintain the desirable consistency properties of the estimators at the same time of bringing in a convex objective function so that the global minimization can be guaranteed.

4.2.2 Making Predictions on New Observations

The proposed method can also be used to decide the subpopulations and predict the response of a new observation. One can treat the latent supervised clustering labels in the training datasets as the estimated underlying outcome, and fit a standard classification model using the observed predictors. Some popular choices include discriminant analysis (McLachlan (2004)), k-nearest-neighbor and random forest. Then, one can used the fitted model to make predictions on which subpopulation the new observed data should be assigned to. To predict the response y_i, one can plug in the corresponding estimated $\boldsymbol{\alpha}_i$ and $\boldsymbol{\gamma}$.

4.3 Computational optimization algorithms

In Section 4.3, we present our proposed algorithm to solve the optimization problem. Primarily, we design an accelerated proximal gradient variant of Fast Iterative Shrinkage-Thresholding Algorithm (FISTA, Beck and Teboulle (2009)) which is shortened as APG. In addition, a restart step is integrated in to further speed up the convergence rate in practice. For the check loss scenario, in which the objective function is nonsmooth, we proposed a surrogate loss for approximation and show that it can also achieve the same estimation results with certain adjustments on APG. The corresponding convergence rate properties are also discussed below.

4.3.1 Properties of the new model

For our notational simplicity, we rewrite the model in a compact form. Let us concatenate the variables $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ into one vector $\boldsymbol{\zeta} \triangleq (\boldsymbol{\beta}^T, \boldsymbol{\gamma})^T$. Now, we define

$$f_n(\boldsymbol{\zeta}) \triangleq \sum_{i=1}^{n} \ell(y_i, \boldsymbol{x_i^T}\boldsymbol{\beta_i} + \boldsymbol{z_i^T}\boldsymbol{\gamma}) \quad \text{and} \quad J_n(\boldsymbol{\zeta}) \triangleq \lambda_n \|D_w \boldsymbol{\beta}\|_1 \equiv \lambda_n \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} \sum_{k=0}^{p} |\beta_{ik} - \beta_{jk}|. \tag{4.4}$$

Then, we can rewrite the optimization problem into the following compact form

$$\min_{\boldsymbol{\zeta} \in \mathbb{R}^{np+q}} \left\{ F_n(\boldsymbol{\zeta}) \triangleq f_n(\boldsymbol{\zeta}) + J_n(\boldsymbol{\zeta}) \right\}. \tag{4.5}$$

Under the choice of our loss function ℓ, we make the following observation.

- Problem (4.5) is convex, i.e., both f_n and J_n are convex (f_n is convex if $f_n((1-\alpha)\boldsymbol{\zeta} + \alpha\hat{\boldsymbol{\zeta}}) \leq (1-\alpha) f_n(\boldsymbol{\zeta}) + \alpha f_n(\hat{\boldsymbol{\zeta}})$ for all $\boldsymbol{\zeta}, \hat{\boldsymbol{\zeta}}$ and $\alpha \in [0,1]$).

- J_n is nonsmooth (its derivative is continuous). If ℓ is the quantile check loss, then f_n is also nonsmooth.

- If ℓ is the quadratic loss, then f_n has Lipschitz gradient, i.e., there exists $L_{f_n} \geq 0$ such that $\|\nabla f_n(\boldsymbol{\zeta}) - \nabla f_n(\hat{\boldsymbol{\zeta}})\| \leq L_{f_n} \|\boldsymbol{\zeta} - \hat{\boldsymbol{\zeta}}\|$ for all $\boldsymbol{\zeta}, \hat{\boldsymbol{\zeta}}$. While problem (4.5) with Lipschitz gradient functions can be solved efficiently by many optimization algorithms including FISTA (Beck and Teboulle (2009)), the nonsmooth check loss is more difficult for designing efficiently numerical methods.

- If ℓ is the check loss, then (4.5) is still convex, but fully nonsmooth (i.e., both f_n and J_n are nonsmooth).

4.3.2 Algorithmic design

Our algorithm mainly relies on the well-known accelerated proximal gradient method in Beck and Teboulle (2009); Nesterov (2013). However, the following steps are new.

- We incorporate the algorithm with a restart procedure recently studied in Fercoq and Qu (2016) to accelerate the performance of the algorithm. Our algorithm has theoretical guarantee even with restart.
- We design a proximal operator (see Definition below) for the regularizer J_n using an adaptive fast projected gradient methods with warm-start.
- For the check loss function, we apply smoothing technique to approximate this function by a smooth function depending on a parameter which can be adaptively updated in the algorithm.
- We design a new variant of the adaptive method proposed in Tran-Dinh (2016); Tran-Dinh et al. (2016) to solve (4.5) that has convergence rate guarantee while avoids any parameter tuning strategy.

The main ingredients of the algorithms consist of

- Evaluate the gradient vector of f_n or its smoothed approximation. Evaluate the Lipschitz constant of this gradient mapping.
- Compute the proximal operator of J_n which is defined as

  $$\mathrm{prox}_{\gamma J_n}(\boldsymbol{\zeta}) := \arg\min_{\hat{\boldsymbol{\zeta}}} \left\{ J_n(\hat{\boldsymbol{\zeta}}) + \tfrac{1}{2\gamma}\|\hat{\boldsymbol{\zeta}} - \boldsymbol{\zeta}\|_2^2 \right\}, \tag{4.6}$$

  for any $\boldsymbol{\zeta}$ and $\gamma > 0$.

We will describe these steps separately below. Now, we assume that these ingredients are given, we can present the main steps of our algorithm for solving (4.5) as follows.

We discuss in details the main steps of Algorithm 1. If f_n is a quadratic loss of the form $f_n(\boldsymbol{\zeta}) = \frac{1}{2}\|\tilde{X}\boldsymbol{\zeta} - y\|_2^2$. Then, $\nabla f_n(\boldsymbol{\zeta}) = \tilde{X}^T(\tilde{X}\boldsymbol{\zeta} - y)$, which is Lipschitz continuous with $L_{f_n} = \lambda_{\max}(\tilde{X}^T\tilde{X})$ (the maximum eigenvalue of $\tilde{X}^T\tilde{X}$). If f_n is a nonsmooth check loss, we

Algorithm 1 Adaptive fast Proximal Gradient algorithm (APG)

```
1. Choose an arbitrarily initial point $\boldsymbol{\zeta}^0 \in \mathbb{R}^{(n+1)d}$ and a desired tolerance $\varepsilon > 0$;
2. Evaluate $L_f := \lambda_{\max}(\tilde{X}^T\tilde{X})$. Set $\tau_0 = 1$, and $\hat{\boldsymbol{\zeta}}^0 := \boldsymbol{\zeta}^0$.
3. If the check loss is used, then input $\eta_1$.
4. For $t = 0, 1, \cdots, t_{\max}$, perform:
      Step 1: Set $L_{f_n} := L_f$ for the quadratic loss, and $L_{f_n} := \frac{L_f}{\eta_{t+1}}$ for the check loss. Then compute the step-size $\alpha_t = \frac{1}{L_{f_n}}$.
      Step 2: Compute approximately $\boldsymbol{\zeta}^{(t+1)} \approx \mathrm{prox}_{\alpha_t J_n}\left(\hat{\boldsymbol{\zeta}}^{(t)} - \alpha_t \nabla f_n(\hat{\boldsymbol{\zeta}}^{(t)})\right)$ up to the accuracy $\epsilon_t$.
      Step 3: If stopping-criterion is satisfied, terminate the algorithm.
      Step 4: If $f_n$ is the quadratic loss, update $\tau_{t+1} := \frac{1}{2}\left(1 + \sqrt{1 + 4\tau_t^2}\right)$.
            If $f_n$ is the check loss, update $\tau_{t+1}$ as the positive solution of $\tau^3 - \tau^2 - \tau_t^2\tau - \tau_t^2 = 0$.
      Step 5: Update the accelerated step $\hat{\zeta}^{(t+1)} := \boldsymbol{\zeta}^{(t+1)} + \frac{\tau_k - 1}{\tau_{k+1}}\left(\boldsymbol{\zeta}^{(t+1)} - \boldsymbol{\zeta}^{(t)}\right)$.
      Step 6: If $f_n$ is the check loss, the update $\eta_{t+2} := \left(\frac{\tau_{t+1}}{\tau_{t+1}+1}\right)\eta_{t+1}$.
      Step 7: Perform a restarting step if requested.
5. End of the main loop.
```

approximate it by a smooth function $f_n(\cdot;\eta)$ as in Subsection 4.3.2 below. The next step is to compute the proximal operator prox_g of g. We separate this step in Subsection 4.3.2 below.

Let ζ^* be an optimal solution of (4.5) with the optimal value $F_n(\boldsymbol{\zeta}^*)$. Then, for any $\boldsymbol{\zeta}$ we have $F_n(\boldsymbol{\zeta}) \geq F_n(\boldsymbol{\zeta}^*)$. We say that $\boldsymbol{\zeta}^{(t)}$ is an approximate solution to (4.5) with an accuracy $\varepsilon \geq 0$, if $F_n(\boldsymbol{\zeta}^{(t)}) - F(\boldsymbol{\zeta}^*) \leq \varepsilon$. In (4.5), we unfortunately cannot compute the proximal operator $\mathrm{prox}_{\gamma J_n}$ exactly, but rather up to a given accuracy $\epsilon_t > 0$ such that

$$0 \leq Q_n(\zeta^{(t+1)}; \hat{\boldsymbol{\zeta}}^{(t)}) - Q_n(\zeta_*^{(t+1)}; \hat{\boldsymbol{\zeta}}^{(t)}) \leq \epsilon_t, \tag{4.7}$$

as can be seen at Step a of Algorithm 1, where $Q_n(\boldsymbol{\zeta}; \hat{\boldsymbol{\zeta}}^{(t)})$ is a surrogate of F_n around $\hat{\boldsymbol{\zeta}}^{(t)}$

defined as

$$Q_n(\boldsymbol{\zeta}; \hat{\boldsymbol{\zeta}}^{(t)}) := f_n(\hat{\boldsymbol{\zeta}}^{(t)}) + \nabla f_n(\hat{\boldsymbol{\zeta}}^{(t)})^T(\boldsymbol{\zeta} - \hat{\boldsymbol{\zeta}}^{(t)}) + \frac{L_f}{2}\|\boldsymbol{\zeta} - \hat{\boldsymbol{\zeta}}^{(t)}\|^2 + J_n(\boldsymbol{\zeta}),$$

and $\boldsymbol{\zeta}_*^{(t+1)} := \operatorname{prox}_{\alpha_t J_n}\left(\hat{\boldsymbol{\zeta}}^{(t)} - \alpha_t \nabla f_n(\hat{\boldsymbol{\zeta}}^{(t)})\right)$. We note that $Q_n(\cdot; \hat{\zeta}^{(t)})$ is the objective function of the proximal operator problem at Step (a) of Algorithm 1. It is easy to show that if condition (4.7) holds, then we have $\|\zeta^{(t+1)} - \boldsymbol{\zeta}_*^{(t+1)}\| \equiv \left\|\zeta^{(t+1)} - \operatorname{prox}_{\alpha_t J_n}\left(\hat{\boldsymbol{\zeta}}^{(t)} - \alpha_t \nabla f_n(\hat{\boldsymbol{\zeta}}^{(t)})\right)\right\| \leq \sqrt{2\alpha_t \epsilon_t}$.

Now, we provide a general convergence result for Algorithm 1. This convergence result can be considered as slight modification of (Schmidt et al., 2011, Proposition 2). The proof sketch of this theorem can be found in the appendix.

Theorem 4.3.1. *Let f_n be a quadratic loss, and let $\{\boldsymbol{\zeta}^{(t)}\}$ be a sequence generated by Algorithm 1 where $\operatorname{prox}_{\gamma j_n}$ is computed approximately as (4.7) with the accuracy $\epsilon_t \geq 0$. Then, we have the following guarantee:*

$$F_n(\boldsymbol{\zeta}^{(t)}) - F_n(\boldsymbol{\zeta}^*) \leq \frac{2\lambda_{\max}(\tilde{X}^T\tilde{X})}{(t+1)^2}\left(\|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\| + R_t\right)^2, \qquad (4.8)$$

where $R_t = \frac{\sqrt{2}}{\sqrt{L_f}}\left(2\sum_{j=0}^{t-1}(j+1)\sqrt{\epsilon_j} + \sqrt{\sum_{j=0}^{t-1}(j+1)^2\epsilon_j}\right)$.

Consequently, for any accuracy $\varepsilon > 0$ and any positive constant $c \geq 1$, if the inner accuracy ϵ_t at each iteration t is chosen such that $\epsilon_t = \frac{c}{(t+1)^5}$, then the number of iterations needed to achieve an approximate solution $\boldsymbol{\zeta}^{(t)}$ of (4.5) with in the accuracy ε does not exceed $t_{\max} = \left\lfloor \frac{\sqrt{2\lambda_{\max}(\tilde{X}^T\tilde{X})}}{\sqrt{\varepsilon}}\|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\| + \frac{10\sqrt{2}c}{\sqrt{\varepsilon}}\right\rfloor$[1].

Smoothing technique for the check loss The check loss $\rho_\tau(r) = \tau r I(r \geq 0) - (1 - \tau)rI(r < 0) = (\tau - 0.5)r + 0.5|r|$ is nonsmooth. We can smooth this function by a smooth convex function $\rho_\tau(\cdot; \eta)$ depending on a given smoothness parameter $\eta > 0$. For any fixed

[1] Here, $\lfloor a \rfloor$ is the closest integer of a.

value of $\eta > 0$, the smoothed function $\rho_\tau(\cdot;\eta)$ needs to satisfies the following basic properties:

- First, $\rho_\tau(\cdot;\eta)$ is smooth and convex. Its gradient $\nabla_r \rho_\tau(\cdot;\eta)$ with respect to r is Lipschitz continuous with a Lipschitz constant L_ρ depending on η.

- Second, $\rho_\tau(\cdot;\eta)$ well approximates $\rho_\tau(\cdot)$. That is there exists a constant D_ρ independent of η such that $\rho_\tau(r;\eta) \le \rho_\tau(r) \le \rho_\tau(r;\eta) + \eta D_\rho$ for all r.

There are several smoothed functions for $\rho_\tau(\cdot)$. Here are two examples.

- *Huber loss:* The function $\rho_\tau(r;\eta) = \begin{cases} \frac{1}{2\eta}r^2 & \text{if } |r| \le \eta \\ |r| - \frac{\eta}{2} & \text{otherwise} \end{cases}$ is a smoothed approximation of $\rho_\tau(\cdot)$ with $L_\rho = \frac{1}{\eta}$ and $D_\rho = \frac{1}{2}$.

- *Logit-type loss:* The function $\rho_\tau(r;\eta) = \left(\tau - \frac{1}{2}\right) r + \frac{\eta}{2} \ln\left(e^{r/\eta} + e^{-r/\eta}\right)$ is a smoothed approximation of $\rho_\tau(\cdot)$ with $L_\rho = \frac{1}{\eta}$ and $D_\rho = \ln(2)$.

Now, we can prove the following properties of $\rho_\tau(\cdot;\eta)$, whose proof can found in the appendix.

Lemma 4.3.1. *Let us consider $f_n(\boldsymbol{\zeta};\eta) \overset{\triangle}{=} \sum_{i=1}^n \rho_\tau(y_i(\mathbf{x}_i^T\boldsymbol{\beta}_i + \mathbf{z}_i^T\boldsymbol{\gamma});\eta)$ as a smooth version of the check loss in* (4.4) *using either the Huber loss or the Logit-type loss. Then, this function is convex and differentiable. Its gradient $\nabla_{\boldsymbol{\zeta}} f_n(\cdot;\eta)$ is Lipschitz continuous with the Lipschitz constant $L_{f_n} = \frac{\lambda_{\max}(\tilde{X}^T\tilde{X})}{\eta}$. Moreover, we have*

$$f_n(\boldsymbol{\zeta};\eta) \le f_n(\boldsymbol{\zeta}) \le f_n(\boldsymbol{\zeta};\eta) + n\eta D_\rho, \tag{4.9}$$

for any $\boldsymbol{\zeta} \in \mathbb{R}^{(n+1)p}$ and $\eta > 0$.

Associated with $f_n(\boldsymbol{\zeta};\eta)$, we consider the smoothed problem of (4.5) as

$$\min_{\boldsymbol{\zeta}\in\mathbb{R}^{np+q}} \left\{ F_n(\boldsymbol{\zeta};\eta) \overset{\triangle}{=} f_n(\boldsymbol{\zeta};\eta) + J_n(\boldsymbol{\zeta}) \right\}. \tag{4.10}$$

Our goal is to compute an ε-approximation solution $\boldsymbol{\zeta}^{(t)}$ to the true solution $\boldsymbol{\zeta}^*$ of the original problem (4.5) as $F_n(\boldsymbol{\zeta}^{(t)}) - F_n(\boldsymbol{\zeta}^*) \leq \varepsilon$. The idea is to apply the fast proximal gradient method to approximately solve this smoothed problem (4.10) and then combines it with a homotopy scheme to decrease the smoothness parameter η at each iteration t as $\eta_{t+1} := \left(\frac{\tau_t}{\tau_t+1}\right)\eta_t$. The step-size parameter is updated by using the unique positive solution τ_{t+1} of the cubic equation $c_3(\tau) = \tau^3 + \tau^2 + \tau_t^2\tau - \tau_t^2 = 0$.

The following result tells us that if $\zeta^{(t)}$ is an approximate solution of (4.10), then it is also an approximate solution of the original problem (4.5). The following theorem shows the convergence of Algorithm 1 whose proof can be found in appendix.

Theorem 4.3.2. *Let $\{\boldsymbol{\zeta}^{(t)}\}$ be a sequence generated by Algorithm 1 where* $\mathrm{prox}_{\gamma J_n}$ *is computed approximately as* (4.7) *with the accuracy $\epsilon_t := \frac{c}{(t+1)^4}$ for some positive constant $c \geq 1$. Then we have the following guarantee:*

$$F_n(\boldsymbol{\zeta}^{(t+1)}) - F_n(\boldsymbol{\zeta}^*) \leq \frac{1}{(t+1)}[\frac{L_f}{2\eta_1}\|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\|^2 + 2n\eta_1 D_\rho + \frac{1.9\sqrt{cL_f}}{\sqrt{\eta_1}}\|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\| + \frac{35c}{2\eta_1} + \Gamma_t], \tag{4.11}$$

where $L_f := \lambda_{\max}(\tilde{X}^T\tilde{X})$, $\Gamma_t = 0$ for the Huber loss, and $\Gamma_t = \frac{D_\rho(1+4\eta_1\ln(t+1))}{2}$ for the Logit-type loss. Hence, for any accuracy $\varepsilon > 0$, the number of iterations needed to achieve an approximate solution $\boldsymbol{\zeta}^{(t)}$ of (4.5) *with in the accuracy ε does not exceed $t_{\max} = \mathcal{O}\left(\frac{1+\Gamma_\varepsilon}{\varepsilon}\right)$, where $\Gamma_\varepsilon = 0$ for the Huber loss, and $\Gamma_\epsilon = -\ln(\varepsilon)$ for the Logit-type loss.*

Evaluating the proximal operator for the regularizer We now describe how to approximately evaluate the proximal operator of J_n defined by (4.6) that satisfies the condition (4.7). Since computing $\boldsymbol{\zeta}^{(t+1)}$ requires to approximate the solution of the strongly convex problem (4.6), we instead consider its dual problem

$$\mathrm{prox}_{\gamma J_n}(u) = u - \gamma D^T\nu^*(u), \quad \text{where } \nu^*(u) := \arg\min_{\|\nu\|_\infty \leq 1}\left\{\frac{\gamma}{2}\|D_w^T\nu\|^2 - \nu^T D_w u\right\}. \tag{4.12}$$

Here, for a given $\gamma > 0$ and u, we need to solve the dual problem to obtain $\nu^*(u)$. Then, using the first formulation of (4.12) to compute $\mathrm{prox}_{\gamma J_n}(u)$. Since, we can only approximate $\nu^*(u)$, the proximal operator $\mathrm{prox}_{\gamma J_n}(u)$ can only be approximated within a given accuracy. We apply an accelerated projected gradient algorithm to approximate $\nu^*(u)$. The algorithm can be briefly presented in two lines as follows:

Accelerated projected gradient scheme to approximate $\mathrm{prox}_{\gamma J_n}(u)$**:** Given u, $\gamma > 0$ and an initial point $\nu^{(0)}$. Compute $L_D := \lambda_{\max}(DD^T)$. Set $\bar{\nu}^{(0)} = \hat{\nu}^{(0)} := \nu^{(0)}$, $s_0 := 1$ and $\Gamma_0 := 0$. At each iteration $j \geq 0$, we update

1. $\nu^{(j+1)} := \pi_{B_\infty}\Big(\hat{\nu}^{(j)} - \frac{1}{L_d} D_w(D_w^T \hat{\nu}^{(j)} - u)\Big)$.
2. $\hat{\nu}^{(j+1)} := \nu^{(j+1)} + \frac{s_j - 1}{s_{j+1}}(\nu^{(j+1)} - \nu^{(j)})$ where $s_{j+1} := \frac{1}{2}(1 + \sqrt{1 + 4s_j^2})$.
3. $\bar{\nu}^{j+1} := (1 - \omega_j)\bar{\nu}^j + \omega_j \nu^{(j+1)}$, where $\Gamma^{j+1} := \Gamma^j + s_{j+1}$, and $\omega_j := \frac{s_{j+1}}{\Gamma_{j+1}}$.

Here, $\pi_{B_\infty}(v) = \max\{\min\{v, 1\}, -1\}$ is the projection of v onto the ℓ_∞-unit ball $B_\infty := \{v \mid \|v\|_\infty \leq 1\}$. This algorithm is terminated after $j_{\max}$ iterations. The output of this routine is $\zeta^{(t+1)} := \hat{\boldsymbol{\zeta}}^{(t)} - \alpha_t \nabla f_n(\hat{\boldsymbol{\zeta}}^{(t)}) - \alpha_t D^T \bar{\nu}^{j_{\max}}$, which approximates $\boldsymbol{\zeta}_*^{(t+1)} = \mathrm{prox}_{\alpha_t J_n}\Big(\hat{\boldsymbol{\zeta}}^{(t)} - \alpha_t \nabla f_n(\hat{\boldsymbol{\zeta}}^{(t)})\Big)$.

Now, we analyze the computational effort to achieve the approximation point $\boldsymbol{\zeta}^{(t+1)}$ as in (4.7). By slightly adapting (Tran-Dinh et al., 2016, Theorem 2), we have the following bound $Q_n(\boldsymbol{\zeta}^{(t+1)}; \hat{\zeta}^{(t)}) - Q_n(\boldsymbol{\zeta}_*^{(t+1)}; \hat{\zeta}^{(t)}) \leq \frac{2L_D \|\nu^{(0)} - \nu_t^*\|^2}{(j+1)^2}$, where $\nu_t^* := \nu^*(\hat{\boldsymbol{\zeta}}^{(t)} - \alpha_t \nabla f_n(\hat{\boldsymbol{\zeta}}^{(t)}))$ defined in (4.12). In order to achieve an ϵ_t-approximate solution $\boldsymbol{\zeta}^{(t+1)}$ of $\boldsymbol{\zeta}_*^{(t+1)}$ at the t-th iteration, we requires $\frac{2L_D \|\nu^{(0)} - \nu_t^*\|^2}{(j+1)^2} \leq \epsilon_t$. Hence, the number of iteration $j_{\max}$ to achieve this goal is

$$j_{\max} := \left\lfloor \frac{\sqrt{2\lambda_{\max}(D_w D_w^T)}}{\sqrt{\varepsilon_t}} \|\nu^{(0)} - \nu_t^*\| \right\rfloor.$$

By exploiting a warm-start strategy using the previous approximate point $\nu^{(t-1)}$ for $\nu^{(0)}$, the distance $\|\nu^{(0)} - \nu_t^*\|$ becomes small. This implies that the maximum number of iterations

$j_{\max}$ is also small. In our implementation, we can fix this number to a given level such as $j_{\max} = 50$ to achieve an approximation $\boldsymbol{\zeta}^{(t+1)}$ in (4.7).

4.3.3 Initiating the Starting Points

Section 4.2.1 mentions that a proper selection of the starting values$\tilde{\boldsymbol{\beta}}$ can be important for the APG algorithm to converge to the global optimization point at a fast rate. Based on the assumption that each variable in X are independent of each included variable in Z, we introduce an ad-hoc method that can be easily applied to find a proper $\tilde{\boldsymbol{\beta}}$ in practice. We split the method into two steps as follows:

First, calculate the distance matrix of the data set based on $(X, \boldsymbol{y}^*)$, where $\boldsymbol{y}^*$ is the residual of the linear regression between $\boldsymbol{y}$ and Z. We use $\boldsymbol{y}^*$ instead of $\boldsymbol{y}$ in that the expectation of $\boldsymbol{y}^*$ is exactly $X\boldsymbol{\beta}$. This conclusion holds due to the fact that a linear regression produces unbiased coefficients estimate when the omitted variables in Z are all independent of those included in the model. In this way, we treat the response $\boldsymbol{y}^*$ as a new variable and calculate the distance matrix on $(X, \boldsymbol{y}^*)$. For example, the Manhattan distance between the ith and jth subjects is defined as $d(i,j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1 + \alpha\|y_i^* - y_j^*\|_1$ (Borg and Groenen (2005)). In this paper, we always choose $\alpha = 1$.

Second, denote $\tilde{\boldsymbol{\beta}} = (\tilde{\boldsymbol{\beta}}_1, \cdots, \tilde{\boldsymbol{\beta}}_n)$ and calculate each $\tilde{\boldsymbol{\beta}}_i$ based on the k-nearest neighbors of the ith subject with a linear regression model matching the loss function. The k-nearest neighbor set for the ith subject is defined as the k observations that: first, have the smallest distance $d(i,j)$ to the ith subject, and second whose response is within the neighbor of y_i as $O_\epsilon(y_i^*)$. The reason of the second restriction is to increase the chance that those neighbors come from the same latent groups as the ith subject does (Figure). The selection of the neighbor ball radius depends on the variation of the noise and $\boldsymbol{x}_i^T\boldsymbol{\beta}_i$ in the simulation studies. In particular, we vary ϵ from 0.5 to 6.

4.4 Numerical Analysis

In this section, we consider using examples to test the performance of latent supervised clustering under finite samples. In Section 4.4.1, we study the estimation accuracy, runtime, and the prediction performance using some synthetic datasets. As to the estimation accuracy, we focus on both the estimated coefficients and the detected number of subgroups. For the proposed method, we consider both the smooth approximate check loss and the quadratic loss as mentioned previously. For estimation accuracy comparison, we employ the method proposed in Ma and Huang (2016) with mannual extension to multivariate cases, which considers a general heterogeneous-effect vector $\boldsymbol{x}_i$ instead of the intercept only. For prediction performance comparison, we additionally include two standard methods: linear regression with all the two-way interactions of x_i's and random forest. In Section 4.4.2, we apply latent supervised clustering to analyze two real datasets and compare its prediction results with the models selected in Section 4.4.1.

In the numerical studies below, we pick the tuning parameters of latent supervised clustering, i.e. (λ_n, m, ϵ), as follows. The penalty tuning parameter λ_n varies in $\{2^{-3}, 2^{-2}, 2^{-1}, 2^0\}$; the number of neighbors is fixed to be $m = 10$; the radius of the response neighbor ϵ increases as $\boldsymbol{x}_i$ dimension goes up and is chosen from $\{2^{-1}, 2^0, 2, 2^2\}$.

4.4.1 Simulations

We conduct eight representative simulated examples to evaluate the performance of latent supervised clustering under various scenarios. Due to the linearity of the proposed model, we restrict the outcome-predictor relationship to be linear for all the examples. In summary, the first two examples are two subpopulation cases with linear boundaries and $\boldsymbol{x}_i$ has low dimension with no noisy component. The third example is modified from Wei and Kosorok (2013) by adding more covariates in $\boldsymbol{x}_i$. The fourth example discusses a case when $\boldsymbol{z}_i$ has a higher dimension while most of its covariates noisy. The fifth, sixth and eighth examples demonstrate the cases when the underlying subpopulation boundaries are nonlinear and $\boldsymbol{x}_i$ contains noisy variables. The seventh example covers the nonlinear boundary situation when number of underlying subgroups increases to five.

For each example, we first generate a training sample to estimate the parameters and obtain the fusion labels for each observation. We generate each covariate of $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ from an independent continuous uniform distribution $U(-2, 2)$ and the random noise ε_i from a normal distribution $N(0, 1)$. To select the best tuning parameter λ_n, we generate an equal size tuning sample and choose the λ_n that minimizes prediction error for the tuning set. To calculate the prediction error for each observation in the tuning set, we consider the estimated parameters for all the detected subpopulations in the training set and choose the one that produces the smallest mean square error. Then we calculate the average predicted means square error over all the tuning samples and treat it as the criterion for λ_n selection.

After we pick the best λ_n with all the estimated coefficients $\boldsymbol{\beta}_i$, we generate a test dataset that is ten times as large as the training one to assess the prediction performance. We divide prediction into two steps as described previously. First, we treat the clustering label of the training set as the underlying outcome and fit it with all the predictors using a classification model. In the simulation studies, we choose k-neatest-neighbor for most of the cases and use kernel discriminant analysis as an alternative when $\boldsymbol{x}_i$ has noisy variable. Second, for each observation from the testing set, we use the classification model to predict which cluster it comes from and plug in the corresponding estimated $\boldsymbol{\beta}_i$ to make predictions. The mean squared prediction error is reported for each model.

As to the comparison of estimation accuracy, we report the average and standard deviation of the square root of mean squared error (SMSE, Hastie et al. (2009)) of the estimated $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, which are defined as $\sqrt{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})/np}$ and $\sqrt{(\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma})^T(\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma})/q}$ respectively. In addition, we also compare the sample mean and standard deviation (in parenthesis) of the estimated number of subpopulations $\hat{K}$. For the runtime comparison, we record the average run time (in second) of each method under all examples. The simulations are repeated for 300 times. The details of the settings for the eight examples are listed as below.

Example 1 Univariate Linear Regression with Two Subgroups Suppose the underlying true model is a linear regression case written as,

$$y_i = \begin{cases} 1 - x_{i1} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1} \leq 1 \\ -1 + x_{i1} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1} > 1 \end{cases},$$

where $\boldsymbol{\gamma} = (1, -5, 2, 1, -3, 1, 3, 2, -4)^T$ and the random noise $\varepsilon_i \sim N(0, 1)$;

Example 2 Three Dimensional $\boldsymbol{x}_i$ with Noisy Variables in $\boldsymbol{z}_i$ We consider a case when $\boldsymbol{x}_i$ has three dimension and $\boldsymbol{z}_i$ includes noisy variables. In particular, the true model is written as,

$$y_i = \begin{cases} 1 - x_{i1} - 2x_{i2} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1} + x_{i2} \leq 0 \\ -1 + 2x_{i1} - x_{i2} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1} + x_{i2} > 0 \end{cases},$$

where $\boldsymbol{\gamma} = (1, -5, 3, 2, 1, 0, 0, 0, 0)^T$ and the random noise $\varepsilon_i \sim N(0, 1)$;

Example 3 Higher $\boldsymbol{x}_i$ Dimension with Noisy Variables We have $\boldsymbol{x}_i$ contain 25 variables with intercept included, and the latent subpopulation is determined by the first five of them. The true model is,

$$y_i = \begin{cases} -4 + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & 1 + x_{i1} + x_{i2} - 3x_{i3} + 2x_{i4} \leq 0 \\ 1 + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & 1 + x_{i1} + x_{i2} - 3x_{i3} + 2x_{i4} > 0 \end{cases},$$

where $\boldsymbol{\gamma} = (1, -5, 3, 2, 1, 0, 0, 0, 0)^T$ and the random noise $\varepsilon_i \sim N(0, 1)$;

Example 4 Higher $\boldsymbol{z}_i$ Dimension with Three Subgroups Consider a model that has 50 homogeneous variables $\boldsymbol{z}_i^T$ and three latent subpopulations as,

$$y_i = \begin{cases} 1 - x_{i1} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1} + x_{i2} + x_{i3} \leq -1 \\ -1 + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1} + x_{i2} + x_{i3} > 1 \\ 1 + x_{i1} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & \text{o.w.} \end{cases},$$

where $\boldsymbol{\gamma} = (1, -5, 3, 2, 1, \mathbf{0}_{45}^T)^T$, $\mathbf{0}_{45}^T$ represents a 45-dimensional zero vector and the random noise $\varepsilon_i \sim N(0, 1)$;

Example 5 Nonlinear Subgroups Boundary with Noisy Variables in $\boldsymbol{x}_i$ Consider a model in which x_i has 6 variables with intercept included and the two of them construct a nonlinear subpopulation boundary,

$$y_i = \begin{cases} 1 - x_{i1} - 3x_{i2} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1}^2 + x_{i2}^2 \leq 4 \\ 1 + 3x_{i1} + x_{i2} + 4x_{i3} - x_{i5} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1}^2 + x_{i2}^2 > 4 \end{cases},$$

where $\boldsymbol{\gamma} = (1, -5, 3, 2, 1)^T$ and the random noise $\varepsilon_i \sim N(0, 1)$;

Example 6 Complex Subgroups Boundary with Noisy Variables in $\boldsymbol{x}_i$ Consider a situation that has a more complex nonlinear subpopulation boundary:

$$y_i = \begin{cases} 1 + 5x_{i1} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1}^2 \sin x_{i2} + x_3^3 + \log(x_{i4} + 5) + x_{i5} \leq 5 \\ -1 - 3x_{i1} + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i, & x_{i1}^2 \sin x_{i2} + x_3^3 + \log(x_{i4} + 5) + x_{i5} > 5 \end{cases},$$

where $\gamma = (1, -5, 3, 2, 1)$ and the random noise $\varepsilon_i \sim N(0, 1)$;

Example 7 Nonlinear Subgroups Boundaries with Five Subgroups Consider a situation that there are 5 subpopulations with nonlinear boundaries $y_i = \sum_{k=1}^5 I((x_{i1}^2 + x_{i2}^2) \in (b_{k-1}, b_k]) \cdot (k-3)(1 + x_{i1} + \cdots + x_{i5}) + \boldsymbol{z}_i^T \boldsymbol{\gamma} + \varepsilon_i$ where $(b_0, b_1, b_2, b_3, b_4, b_5) = (-\infty, 2, 3.5, 5, 7, \infty)$, $\boldsymbol{\gamma} = (1, -5, 3, 2, 1)^T$, and the random noise $\varepsilon_i \sim N(0, 1)$;

Example 8 Complex Subgroups Boundaries with Three Subgroups and Noisy Variables Consider a model in which x_i has 11 variables with intercept included and 3 subpopulations as following,

$$
y_i = \begin{cases} 1 + x_{i1} + 3x_{i2} + 2x_{i3} + 3x_{i4} + 2x_{i5} + \boldsymbol{z}_i^T\boldsymbol{\gamma} + \varepsilon_i, & x_{i1}^2 + \exp(x_{i2}) \leq 2.5 \\ -1 + 3x_{i1} + x_{i2} - 5x_{i3} + 0x_{i4} - 2x_{i5} + \boldsymbol{z}_i^T\boldsymbol{\gamma} + \varepsilon_i, & x_{i1}^2 + \exp(x_{i2}) > 5.5 \\ 1 - x_{i1} - x_{i2} - x_{i3} + 5x_{i4} - 3x_{i5} + \boldsymbol{z}_i^T\boldsymbol{\gamma} + \varepsilon_i, & \text{o.w.} \end{cases},
$$

where $\boldsymbol{\gamma} = (1, -5, 3, 2, 1)^T$ and the random noise $\varepsilon_i \sim N(0, 1)$.

Table 4.1 presents the estimation accuracy of the manual extension of Ma and Huang (2016) (Concave), latent supervised clustering with quadratic loss (LSC-quad), and latent supervised clustering with check loss (LSC-check). From the results, latent supervised clustering with check loss almost always produces better mean square error results for both $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}$. In particular, LSC-quad and LSC-check both perform competitively in the first two examples when the boundaries are less complex and the dimension of $\boldsymbol{x}_i$ is small. The concave penalty method can also maintain relatively small average of mean square error while suffers a larger variability, which implies that the concave penalty can lead to unstable estimation results. When the underlying subpopulation structure becomes more complex, as the dimension of x_i increases as Example 3 or the boundraies become nonlinear as Example 5 and 6 show, the advantage of LSC-check becomes more clear due to its strong stability to the noises and outliers caused by being clustered in the wrong subpopulation. This demonstrates the advantage of using a more robust loss as well as adaptive fusion penalty. When there are over two subpopulations in the setting with even more complex boundaries as Example 7 and 8 show, none of the methods can produce satisfactory estimation accuracy while LSC-check still outperforms the others. This indicates that one should conduct additional data cleaning or variable selection steps before fitting latent supervised learning in practice, especially when the predictors can have complex relationships. As to the estimates of the subpopulation numbers, the two LSC methods perform better than Concave while all the methods tend to overestimate this number except for the case of $K = 5$.

Examples		**Concave**		**LSC-quad**		**LSC-check**	
#	(n,p,q,K)	$\hat{\boldsymbol{\beta}}$	$\hat{K}$	$\hat{\boldsymbol{\beta}}$	$\hat{K}$	$\hat{\boldsymbol{\beta}}$	$\hat{K}$
1	(100,2,9,2)	0.222	3.04	0.145	2.1	**0.115**	2.17
		(0.153)	(0.68)	(0.061)	(0.48)	(0.033)	(0.37)
3	(300,24,5,2)	0.408	2.32	0.315	2.21	**0.177**	2.11
		(0.22)	(0.68)	(0.180)	(0.49)	(0.122)	(0.41)
4	(300,4,50,3)	0.421	2.67	0.235	3.34	**0.179**	3.12
		(0.035)	(0.89)	(0.041)	(0.73)	(0.02)	(0.38)
6	(300,6,5,2)	0.616	2.73	0.348	2.15	**0.239**	2.13
		(0.161)	(0.52)	(0.152)	(0.22)	(0.134)	(0.18)
7	(600,6,5,5)	0.790	3.25	0.660	4.25	**0.469**	4.43
		(0.251)	(1.54)	(0.271)	(1.06)	(0.157)	(0.36)

Table 4.1: Simulation estimation accuracy: the average and standard deviation of the square root of mean squared error of the estimated $\boldsymbol{\beta}$ with the best results in bold. The $\hat{K}$ column provides the average and standard deviation of the detected number of clusters. Concave represents the method with concave fusion penalty, LSC-quad and LSC-check represent latent supervised clustering with quadratic loss and check loss respectively

As to the runtime comparison, Table 4.2 reports the average time that the selected methods take to return the results with one combination of tuning parameters. It is not surprising to see that latent supervised learning is faster than the concave penalty method with ADMM algorithm especially when the sample size n and dimension p become large. This is because the concave penalty method takes more iterations before convergence, and the ADMM algorithm needs to create $O(n^2p)$ additional intermediate parameters. In addition, the suggested specification of the weight vector $\boldsymbol{w}$ in latent supervised clustering reduces the number of the penalty terms tremendously. Within latent supervised learning methods, LSC-quad performs slightly faster than LSC-check, and this is due to the fact that the smooth check loss can slightly decrease the convergence speed of the proposed accelerated proximal gradient algorithm.

As to make predictions for the test data sets, we choose k-nearest-neighbor with $k = 10$ in most of the cases to predict the underlying label except for Example 3, 7, and 8 due to the affect of noisy predictors. For these three examples, we choose kernel discriminant analysis with the tuning parameters selected by the tuning datasets.

We report the mean squared error of prediction in Table 4.3. From the results, LSC

Examples					**Concave**	**LSC-quad**	**LSC-check**
#	n	p	q	K	Time(sec.)	Time(sec.)	Time(sec.)
1	100	2	9	2	18.5	**11.0**	12.1
2	300	3	9	2	692	**71.1**	107.3
3	200	24	5	2	16954	**2364.1**	2433.7
4	300	4	50	3	793.8	**161.5**	186.5
5	300	6	5	2	2225.1	**458.5**	537.1
6	300	6	5	2	2244.5	**436.7**	507.4
7	600	6	5	5	9513	**1255.8**	1398.8
8	600	11	5	3	40514.5	**4803.9**	5135.2

Table 4.2: Simulation runtime comparison: average run-time (in second) of the selected methods for each set of the tuning parameters with the best results in bold. Concave represents the method with concave fusion penalty, LSC-quad and LSC-check represent latent supervised clustering with quadratic loss and check loss respectively.

with check loss enjoys the best prediction performance with the minimal prediction error. For Examples 1 and 2, all the methods produce satisfactory results due to the less noises. Linear regression with two-way interactions perform as good as LSC in Example 3 because the underlying boundary fits the interaction assumption exactly. When the underlying boundaries become nonlinear, as Example 5-8 show, neither linear regression with interactions nor random forest can produce reliable prediction results because their model assumptions no longer hold. The performance of LSC is also significant better than the concave penalty method with lower values of both average prediction error and its variability. Similar to the estimation accuracy results, none of these methods can produce satisfactory prediction results when the latent subpopulation structure becomes chaotic as Example 8 shows.

4.4.2 Data Application

In this section, we apply the proposed method to two health care datasets from UCI Machine Learning Repository (Lichman (2013)) to evaluate its performance. We use the 5-fold cross-validation to divide the datasets into training and validation sets with 300 replications and then include the same methods as in the simulation examples with the same tuning parameter ranges. To predict the cluster labels of observations in the validation sets, we use k-nearest-neighbor with $k = 10$. For both the proposed method and the extension

Example	Concave	LSC-quad	LSC-check	Reg-Int	RF
1	0.284	0.083	**0.057**	0.388	0.499
	(0.05)	(0.004)	**(0.003)**	(0.025)	(0.081)
2	1.142	0.644	**0.486**	2.897	1.513
	(0.155)	(0.093)	**(0.085)**	(0.13)	(0.126)
3	10.531	6.936	**3.481**	3.599	9.759
	(0.308)	(0.202)	**(0.166)**	(0.113)	(1.505)
4	2.838	0.935	**0.776**	1.506	7.469
	(0.243)	(0.105)	**(0.106)**	(0.074)	(1.21)
5	12.453	10.379	**9.985**	30.325	46.881
	(1.893)	(1.778)	**(1.759)**	(1.017)	(2.835)
6	14.176	11.226	**10.305**	122.594	24.77
	(2.073)	(1.295)	**(1.26)**	(12.372)	(3.999)
7	14.639	12.379	**9.75**	20.829	16.787
	(1.954)	(0.91)	**(0.851)**	(0.675)	(2.041)
8	32.274	21.748	**16.92**	67.239	42.693
	(2.797)	(1.495)	**(1.351)**	(1.397)	(2.894)

Table 4.3: Simulation prediction accuracy: the mean squared prediction errors and standard deviations of the selected methods with the best results in bold. Concave represents the method with concave fusion penalty, LSC-quad and LSC-check represent latent supervised clustering with quadratic loss and check loss respectively, Reg-Int means linear regression with two-way interactions of x_i's, and RF represents random forest.

of the method with concave penalty, we follow the suggested “forward screening” idea in Section 2.1 to identify $\boldsymbol{x}_i$ from $\boldsymbol{z}_i$. In particular, we start with a parsimonious model that has only an intercept term in $\boldsymbol{x}_i$ and all the predictors in $\boldsymbol{z}_i$. Then we move variables from $\boldsymbol{z}_i$ to $\boldsymbol{x}_i$ one at a time, by choosing the one that boosts the average validation prediction accuracy the most. The process stops when the validation prediction performance is no longer improved. We report the mean squared prediction errors of all the methods for both the training and validation sets as a criterion and also briefly describe the pattern of the detected subpopulations by latent supervised clustering.

Cleveland Heart Disease This dataset records 303 heart disease patients in Cleverland with 14 attributes including the binary diagnosis variable. In the attributes, there is one variable named maximum-heart-rate-achieved which is correlated to presence status of the heart disease as well as cardiac mortality (Lauer et al., 1999). In this way, we treat this maximum-heart-rate as the outcome, the diagnosis variable as the underlying label, and all the other attributes as predictors. It is reasonable to assume that predictors may have heterogeneous effect between the disease group and the non-disease group. In this way, the learning goal is to detect such potential subpopulations and predict the outcome. We implement the proposed method with the “forward screening” idea and the order of the forward selection result is serum cholestoral (1), gender (2), resting blood pressure (3), the slope of the peak exercise ST segment (4), age (5), and exercise induced angina (6). Figure 4.1 presents the mean squared prediction errors for both training and validation sets of all the involved methods. In the figure, the six “LSC-c k” columns represent latent supervised learning with check loss and k variables in $\boldsymbol{x}_i$ that follows the “forward screening” order as described. One can see that the model achieves the best prediction accuracy for the validation sets when three variables, i.e. serum cholestoral, gender and resting blood pressure are put into $\boldsymbol{x}_i$ while all the others remain in $\boldsymbol{z}_i$. The training error becomes very small when one include five variable in x_i while the validation error is slightly worse, implying that there

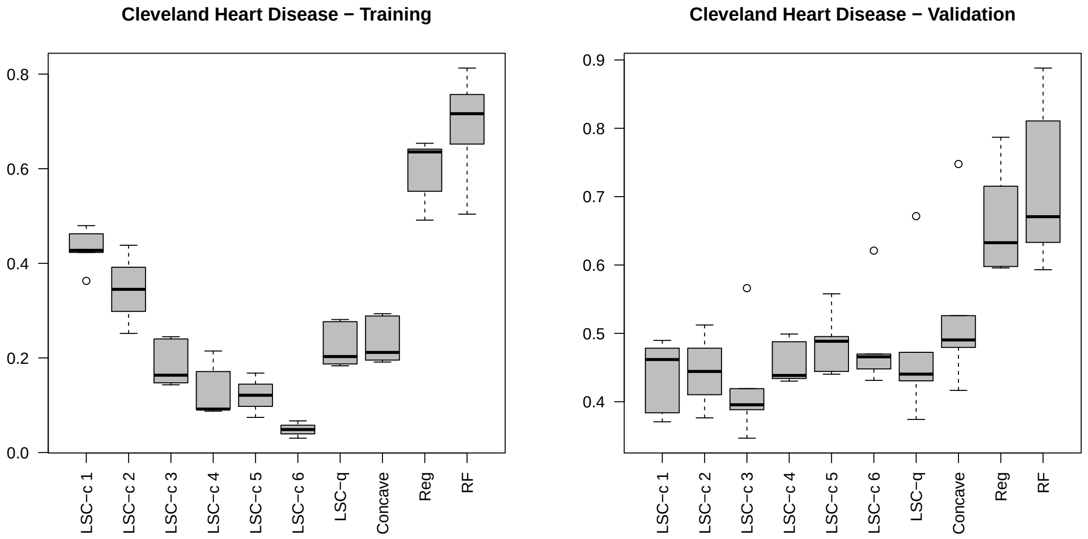

Figure 4.1: Cleveland Heart Disease: mean squared prediction errors. LSC-c1 - LSC-c6 presents latent supervised clustering with check loss that includes the corresponding number of variables in $\boldsymbol{x}_i$ by the "forward screening" idea, LSC-q represents latent supervised clustering with quadratic loss, Concave represents the method with concave fusion penalty, Reg means linear regression with two-way interactions of x_i's, and RF represents random forest.

might be overfitting issues.The column of LSC-q and Concave represent latent supervised clustering with quadratic loss and the method with concave fusion penalty that correspond to the optimal tuning results. Both of them perform worse than LSC-c with larger prediction error and variability. Reg and RF mean linear regression with two-way interactions and random forest. For this dataset, the two commonly used methods fail to achieve satisfactory prediction results, which might be due to the underlying subpopulation structure.

Other than the prediction accuracy, we are also interested in whether the detected subgroups in the sample can be correlated to the underlying heart disease status. We conduct a χ^2 test between the detected subgroup labels and the underlying diagnosis variable for each time of the cross validation. Figure 4.2 presents the p values of such χ^2 tests in one realization of the cross validation as well as the corresponding scatterplots of the observed and predicted outcome by LSC-c1, LSC-c3, and LSC-c5 respectively. From the results, the proposed method always suggests two subpopulations. It is not surprising to see that a larger dimension in x_i can produce a more significant p value because that provides more

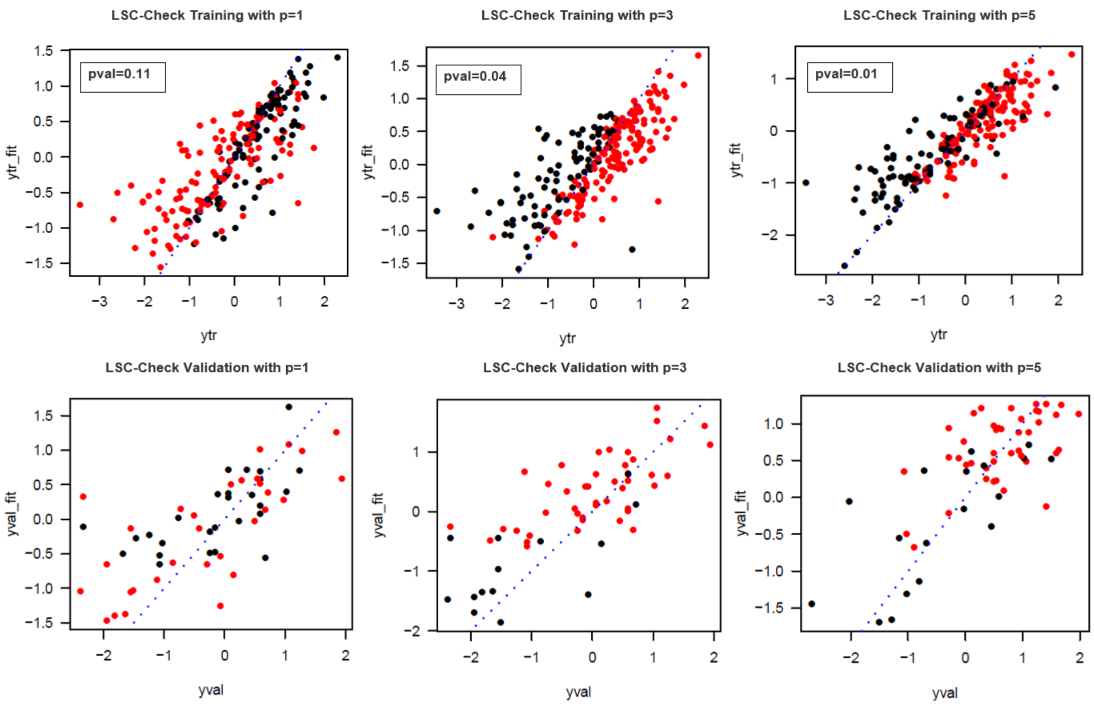

Figure 4.2: Cleveland Heart Disease: scatterplots of the observed and predicted outcome for one training and validation set with the corresponding p values of the χ^2 test. The detected subgroups are denoted by different colors and are clustered by latent supervised clustering with check loss and number of variables in $x_i$to be 1, 3 , and 5.

information to cluster the heterogeneity. One can also note that the difference between the outcome values between the two detected groups becomes clearer when the p values becomes more significant. This is also a reasonable finding because the underlying disease status is known to affect the maximum heart rate. Eventually, we would like to point out that the learning goals of clustering the subpopulation and making predictions may not be achieved by the same model. One can see from Figure 4.2 that LSC-c5 kinds of overfit the training set even through it may lead to a more convincing subgroup detection result with the smallest p value.

Pima Indian Diabetes The Pima Indian Diabetes dataset collects 768 females at least 21 years old of Pima Indian heritage with 8 attributes and a class variable indicating whether tested positive for diabetes. The 2-hour serum insulin is measured among the 8 attributes and it is considered as a proper surrogate outcome for the underlying diabetes test binary indicator. Therefore, we fit the 2-hour serum insulin using all the other attributes except for the diabetes test binary variable. We remove all the rows that contains missing values and 336 observations are left. Similar to the Cleveland Heart Disease dataset, we use a 5-fold cross validation to split the dataset and fit the training sets with the selected methods. We also apply the "forward screening" idea for variable selection in x_i and the order is diabetes pedigree function (1), diastolic blood pressure (2), body mass index (3), age (4), triceps skin fold thickness (5), and plasma glucose concentration (6). The mean squared prediction errors are presented in Figure 4.3 for all the methods. Similar to the finding in the Cleveland Heart Disease dataset, the proposed method with check loss and three variables in $\boldsymbol{x}_i$ achieves the best prediction performance. LSC-c4 and LSC-c5 can have competitive mean squared errors while the variances are slightly larger. LSC-c6 suffering overfiting problem with its prediction error larger than that of the proposed method with quadratic loss and the extension of method with concave penalty. The linear regression with interactions and random forest produce the worst prediction errors when compared with other methods. In addition, the proposed methods with optimal tuning parameters always suggested two latent subgroups, and the detected subgroup labels show significant relationship with the underlying diabetes test indicator according to the χ^2 test. The median of the p values is 0.031 and this can be treated as an evidence that the identified subgroup can be reasonable.

4.5 Discussions

In this paper, we proposed a novel machine learning method that aims to clustering the underlying subpopulation structure based on the heterogeneous relationship between outcome and predictors. Even though the main coverage is restricted to the scenarios of linear relationship between the outcome and predictors, the proposed method can be a very good exploratory tool in practice due to its weak assumptions on the underlying subpopulation structure. We proposed a very efficient algorithm with competitive convergence rate to solve the optimization problem, and also discuss the statistical consistency properties of the estimators for the coefficients. In numerical studies, the proposed method demonstrates

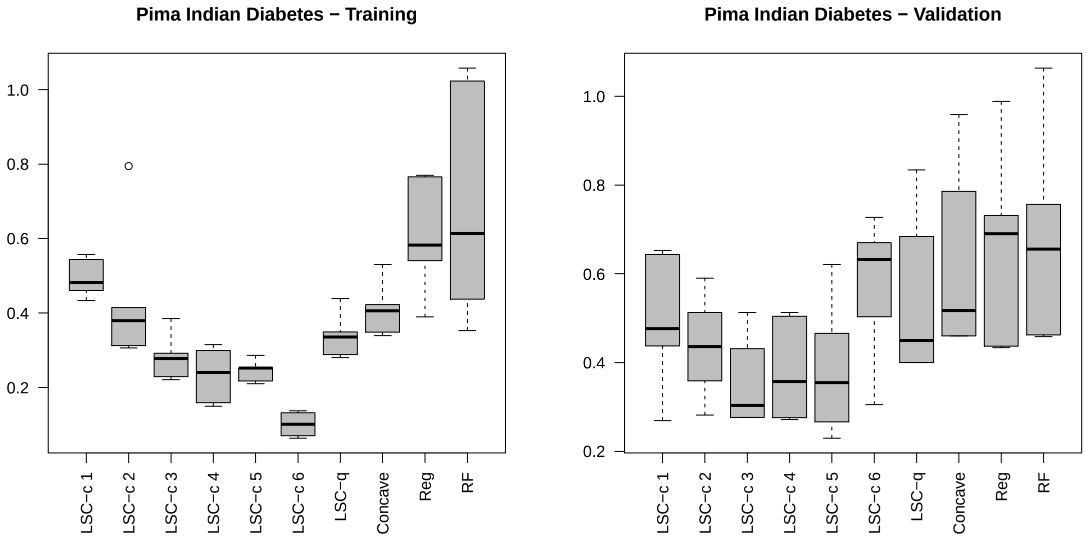

Figure 4.3: Pima Indian Diabetes: mean squared prediction errors. LSC-c1 - LSC-c6 presents latent supervised clustering with check loss that includes the corresponding number of variables in $\boldsymbol{x}_i$ by the "forward screening" idea, LSC-q represents latent supervised clustering with quadratic loss, Concave represents the method with concave fusion penalty, Reg means linear regression with two-way interactions of x_i's, and RF represents random forest.

strong capacity of both subpopulation detection and outcome prediction.

It still remains very interesting in how to encourage the clustering pattern in the estimated coefficients. For our work, this learning goal is pursued by making use of the convex clustering idea and adjusting its loss and penalty accordingly. In addition, there are open questions on whether one can extend the ideas of other clustering methods to achieve the same goal. For example, if the number of subpopulations is known, one might consider borrowing the idea of k-means. In particular, one can still start with an initial estimate $\tilde{\boldsymbol{\beta}}_i$ as discussed, and pick k centroids based on such n coefficient vectors. Then one can recursively implement the following three steps: 1. cluster these $\boldsymbol{\beta}_i$'s by their distances to the centroids; 2. update the centroids; 3. update the values of $\boldsymbol{\beta}_i$ by refitting the model within each cluster. Then after certain iterations, the clustering process might converge and the n coefficient vectors are to be divided in the k clusters.

CHAPTER 5: SUMMARY AND FUTURE RESEARCH

In this chapter, we summarize this dissertation and then discuss some open questions that can be potentially related to future work.

5.1 Summary

In the first chapter, we propose a new DOSK method in kernel learning that can perform variable selection and data extraction simultaneously. We show that under certain conditions, the new DOSK method can achieve selection consistency, and the estimated function can converge to the underlying function with a fast rate. We also develop an efficient algorithm to solve the corresponding optimization, which is guaranteed to converge to a local optimum. Numerical results show that our DOSK method is highly competitive among existing approaches. Note that the hinge loss used in the SVM can also encourage data sparsity because only the support vectors, a subset of the observations, contribute to the estimation results. We would like to point out that modeling data sparsity can be challenging for high-dimension data, especially when there are many noisy variables. One reason is that noisy variables can mislead the importance of each observation in the modeling process. As a consequence, the prediction performance of the SVM can also be affected as shown in the numerical studies.

In the second chapter, we use a modified loss function to improve the performance of OWL and then generalize OWL to solve the ordinal treatment problems. In particular, the proposed GOWL converts the optimal ordinal treatment finding problem into multiple optimal binary treatment finding subproblems under certain restrictions. The estimating process produces a group of estimated optimal treatment boundaries which would never cross and have monotonic intercepts.

In the third chapter, we proposed a novel machine learning method that aims to clustering the underlying subpopulation structure based on the heterogeneous relationship between outcome and predictors. Even though the main coverage is restricted to the scenarios of linear relationship between the outcome and predictors, the proposed method can be a very good exploratory tool in practice due to its weak assumptions on the underlying subpopulation structure. We proposed a very efficient algorithm with competitive convergence rate to solve the optimization problem, and also discuss the statistical consistency properties of the estimators for the coefficients. In numerical studies, the proposed method demonstrates strong capacity of both subpopulation detection and outcome prediction. The initial value of β_i can also be important in determining the time to convergence of the algorithm. Another possible way to obtain the initial value is to use a non-parametric Bayesian method with a proper prior distribution on β_i, including potentially priors with compact supports. This idea could be potentially more stable especially when observations of the same subpopulation also have large Euclidean distances between each other.

Next we discuss some open questions related to the three topics.

5.2 Future Research

5.2.1 Future Research for Double Sparsity Kernel Learning

It becomes a hot topic in the recent years about how to analyze big data with large sample size and high dimension. As a remark, our DOSK method can be generalized to alleviate the computational burden for applications with these massive data sets. Without loss of generality, take regression as an example. Suppose one needs to estimate a nonlinear underlying function, and the data set contains many observations and predictors. To perform kernel regression with such big data can be computationally inefficient. One way to circumvent this difficulty is to split the predictors into several parts or dividing the observations into several subsets, learn on each part individually, and then combine the results. This idea can be particularly useful when the datasets can be partitioned into certain segments such as time series. For each part, it can be reasonable that some subset of the variables and observations become important to represent the whole segment. In this way, each time one can perform our DOSK method on one piece of the data set. Because our DOSK method can have double sparsity in predictors and dual variables, for each sub-regression, it is possible to find a sparsely represented function that only involves a subset of observations and predictors. Then we can combine the selected observations and predictors to train for a global estimator. One can see that this approach can greatly reduce the computational time for problems with massive data sets. In addition, one possible future research direction is to examine how the data sparsity percentage changes when the variable dimension increases.

5.2.2 Future Research for Generalized Outcome Weighted Learning

There are various possible extensions for GOWL that could be considered. For example, one can incorporate a variable selection component into the objective function. In the literature, Xu et al. (2015) proposed variable selection in the linear case and Zhou et al. (2017) extended the idea for kernel learning. According to their ideas, one nature extension for GOWL is to include an l_1 penalty of the parameters into its optimization problem. In this way, variable sparsity could be achieved simultaneously when detecting the optimal ITR. The second possible extension that might improve the performance of GOWL is to modify the outcome in its optimization problem which is originally the reward R. Specifically, according to Fu et al. (2016) and Zhao et al. (2015a), one can consider fitting a model with R versus X and then put the residuals as the outcome in the optimization problem of GOWL instead. Such an adjustment is likely to further improve the ITR estimation results for some finite sample scenarios. Another potential extension is to apply GOWL to solve the dynamic treatment regime problem, i.e. how to maximize the clinical rewards when there are multiple stages of treatments. The idea of Zhao et al. (2015a) could possibly be adapted to such developments. In addition, Some further exploratory work can examine how to improve the performance of the proposed method when some adjacent treatments have very similar effects, which could lead to an unstable estimated ITR boundary between them. For example, suppose we have five treatments with Treatments 2 and 3 very similar. In this case, one can possibly perform a two stage analysis procedure. For the first stage, one can combine Treatments 2 and 3 as one treatment and perform GOWL. For the second stage, a binary

ITR model can be used to determine between 2 and 3.

5.2.3 Future Research for Latent Supervised Clustering

It still remains interesting in developing an efficient way to detect the subpopulation by clustering the estimated coefficients. For our work, this learning goal is pursued by making use of the convex clustering idea and adjusting its loss and penalty accordingly. In addition, there are open questions on whether one can extend the ideas of other clustering methods to achieve the same goal. For example, if the number of subpopulations is known, one might consider borrowing the idea of k-means. In particular, one can still start with an initial estimate $\tilde{\boldsymbol{\beta}}_i$ as discussed, and pick k centroids based on such n coefficient vectors. Then one can recursively implement the following three steps: 1. cluster these $\boldsymbol{\beta}_i$'s by their distances to the centroids; 2. update the centroids; 3. update the values of $\boldsymbol{\beta}_i$ by refitting the model within each cluster. Then after certain iterations, the clustering process might converge and the n coefficient vectors are to be divided in the k clusters.

APPENDIX A: DOUBLE SPARSITY KERNEL LEARNING

Proof of Theorem 2.2.1. Because the objective function ϕ is lower bounded by zero, to prove convergence, it suffices to prove that for each step of updating, the objective function value is non-increasing. To this end, we will show that ϕ is non-increasing for Steps 2-4 in Algorithm 2. First, notice that for fixed $\mathbf{w}$, the corresponding objective functions in the $\boldsymbol{\alpha}$ step and the b step are convex. Hence, ϕ is non-increasing for Steps 2 and 3. We will focus on Step 4 next.

Without loss of generality, suppose that $\nabla_{\mathbf{w}}\phi(\boldsymbol{\alpha}^{(t)}, b^{(t)}, \mathbf{w}^{(t-1)}) \neq \mathbf{0}$ (otherwise, the algorithm has already converged). We will prove that the directional derivative along $\Delta\mathbf{w}$ is negative, with which one can verify that after Step 4, the objective function value would decrease. To this end, observe that Step 4(a) can be regarded as to minimize $\psi(\mathbf{w}) = h\{g(\mathbf{w})\}$, where $h(\cdot)$ is a convex and continuously differentiable function and $g(\cdot)$ is a convex or concave and continuously differentiable function of $\mathbf{w}$. Since both h and g are continuously differentiable, they are locally Lipshcitz continuous, and so is ψ. Furthermore, because h and g are convex or concave, there exists an open neighborhood of $\mathbf{w}^{(t-1)}$, $\mathcal{N}(\mathbf{w}^{(t-1)})$, in which h and g are monotonic (Bertsekas et al., 2003). Therefore, in $\mathcal{N}(\mathbf{w}^{(t-1)})$, $\psi(\cdot)$ is monotonic.

Next, we prove that along the direction defined by $\Delta\mathbf{w}$, $\psi(\cdot)$ is monotonically deceasing in $\mathcal{N}(\mathbf{w}^{(t-1)})$. To this end, first notice that Step 4 computes a descent direction of $\tilde{\psi}_{\mathbf{w}^{(t-1)}}(\mathbf{w}) = h\{g(\mathbf{w}^{(t-1)}) + \nabla g(\mathbf{w}^{(t-1)})^T(\mathbf{w} - \mathbf{w}^{(t-1)})\}$. Because the objective function of $\mathbf{w}^{(QP)}$ is quadratic, thus strictly convex, $\tilde{\psi}_{\mathbf{w}^{(t-1)}}(\mathbf{w})$ is strictly decreasing along $\Delta\mathbf{w}$ within $\mathcal{N}(\mathbf{w}^{(t-1)})$. Next, by similar arguments as in the proof of Proposition 1 in Allen (2012), one can verify that $\psi(\cdot)$ is monotonically deceasing along $\Delta\mathbf{w}$ within $\mathcal{N}(\mathbf{w}^{(t-1)})$, and this completes the proof. ■

Proof of Theorem 2.3.1: Before we present our proof, we first give some lemmas.

Lemma A.0.1. *Suppose Assumptions 1-7 are valid. With λ_1, λ_2 and λ_3 as in Theorem 2.3.1, we have that $\|\hat{\boldsymbol{\alpha}}\|_1 = O_P\{\log(n)\}$ and $|\hat{b}| = O_P\{\log(n)\}$.*

Proof of Lemma A.0.1: With $\boldsymbol{\alpha} = \mathbf{0}$ and $b = 0$, we have $\phi(\mathbf{0}, 0, \mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, 0) \to E\{L(Y, 0)\}$ as $n \to \infty$, which is a constant. On the other hand, $\hat{\boldsymbol{\alpha}}$ and $\hat{b}$ are (part of) the solution to the objective function in (2.6). Hence,

$$\begin{aligned}\lambda_1\|\hat{\boldsymbol{\alpha}}\|_1 \leq & \frac{1}{n}\sum_{i=1}^{n} L\Big\{y_i, \sum_{j=1}^{n} K_{\hat{\mathbf{w}}}(x_i, x_j)\hat{\alpha}_j + \hat{b}\Big\} + \lambda_1\|\hat{\boldsymbol{\alpha}}\|_1 + \lambda_2\|\hat{\mathbf{w}}\|_1 + \lambda_3\hat{\boldsymbol{\alpha}}^T K_{\hat{\mathbf{w}}}\hat{\boldsymbol{\alpha}} \\ \leq & \phi(\mathbf{0}, 0, \mathbf{w}).\end{aligned}$$

Consequently, we have $\|\hat{\boldsymbol{\alpha}}\|_1 = O_P\{\log(n)\}$. For $|\hat{b}|$, in regression, because the fitted function $\hat{f}$ cannot be uniformly larger or smaller than the observed responses, we have that $|\hat{b}|$ is at most $O_P(\|\hat{\boldsymbol{\alpha}}\|_1)$, which is $O_P\{\log(n)\}$ (notice that we have assumed that the error term in regression are bounded for now). For classification problems, similar arguments hold ($\hat{f}$ cannot be uniformly positive or negative, otherwise the classification problem is of less interest), and $|\hat{b}| = O_P\{\log(n)\}$. This completes the proof. □

Lemma A.0.2. *Suppose Assumptions 1-7 are valid. We have that* $\|f_{\boldsymbol{\alpha}^*_n, b^*_n} - f_0\|_2 = O_P\{\log(n)/n\}$.

Proof of Lemma A.0.2: Notice that γ_j's are constants, and the kernel function $K_{\mathbf{w}^*}$ is Lipshcitz by Assumption 2. Hence, we have

$$\begin{aligned}&|f_{\boldsymbol{\alpha}^*_n, b^*_n}(\cdot) - f_0(\cdot)| \\ =&|\sum_{j=1}^{m}\gamma_j\{K_{\mathbf{w}^*}(\boldsymbol{x}_j, \cdot) - K_{\mathbf{w}^*}(\boldsymbol{z}_j, \cdot)\}| \\ =&O_P(\max_j \|\boldsymbol{x}_j - \boldsymbol{z}_j\|_2),\end{aligned}$$

and the goal is to prove that $\|\boldsymbol{x}_j - \boldsymbol{z}_j\|_2 = O_P\{\log(n)/n\}$ for all j. To this end, note that $\mathrm{pr}(\|\boldsymbol{x}_j - \boldsymbol{z}_j\|_2 > d) = (1 - P_d)^n$, where d is a small positive number, and $P_d = \mathrm{pr}(\|\boldsymbol{z} - \boldsymbol{z}_j\|_2 \leq d) = \int_{\|\boldsymbol{z} - \boldsymbol{z}_j\|_2 \leq d} dP$. Using Assumption 1, one can verify that we can choose $d = 2\log(n)/n$, such that $\mathrm{pr}(\|\boldsymbol{x}_j - \boldsymbol{z}_j\|_2 > d) = O_P(n^{-2})$. By the Borel–Cantelli Lemma, we have $\|\boldsymbol{x}_j - \boldsymbol{z}_j\|_2 = O_P\{\log(n)/n\}$ holds. This completes the proof. □

The next lemma generalizes a theoretical result from the margin-based classifier literature to broader ranges of learning problems. In particular, in Zhang and Liu (2013), it was shown that the convergence rate of excess risks for margin-based classifiers is related to the convergence rate of the estimated learning function. In Lemma A.0.3, we extend the discussion to more general situations, in which one uses differentiable loss functions to measure the goodness of fit of $\hat{f}$.

Lemma A.0.3. *Suppose Assumptions 1-7 are valid. Moreover, consider a loss function $\ell\{u(f, y)\}$ that is second order differentiable with respect to u, where $u(f, y)$ is a function of the response y and the learning function f. Assume that u has second order derivative with respect to f, and the two second order derivatives are both bounded. Then we have that, if the function f^* minimizes $E(\ell)$,*

$$|E[\ell\{u(Y, f)\}] - E[\ell\{u(Y, f^*)\}]| = O\{(\|f - f^*\|_2)^2\},$$

and if f^* is not the minimizer of $E(\ell)$,

$$|E[\ell\{u(Y, f)\}] - E[\ell\{u(Y, f^*)\}]| = O\{(\|f - f^*\|_2)\}.$$

Proof of Lemma A.0.3: This proof is analogous to that of Theorems 5 and 6 in Zhang and Liu (2013). Hence, for brevity, we only list the key steps. The first step is to introduce the idea of Bregman divergence. In particular, for a convex differentiable function $g(\cdot)$, its Bregman divergence d_g is defined as $d_g(f_1, f_2) = g(f_2) - g(f_1) - g'(f_1)(f_1 - f_2)$. Then, one can prove that the conditional excess risk $E[\ell\{u(Y, f)\}] - E[\ell\{u(Y, f^*)\}] \mid_{\boldsymbol{X}=\boldsymbol{x}}$ equals to the Bregman divergence $d_\ell\{f^*(\boldsymbol{x}), f(\boldsymbol{x})\}$. See the proof of Theorem 4 in Zhang and Liu (2013) for more details. Combining this result with Assumption 3, we can show, in a similar manner as in the proof of Theorems 5 and 6 in Zhang and Liu (2013), that the claim of Lemma A.0.3 holds. □

We are ready to prove Theorem 2.3.1. The proof follows a similar line as that of Theorem 1 in Zhang et al. (2015). Therefore, we only list out the key steps here. The first step is to decompose the excess risk into two parts, the estimation error and the approximation error. In particular, let $f_{\boldsymbol{\lambda}}$ be the best prediction function with respect to the penalized loss function for fixed $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$, i.e., $f_{\boldsymbol{\lambda}} = \operatorname{arginf}_f[E\{L(Y, f)\} + \lambda_1\|\boldsymbol{\alpha}\|_1 + \lambda_2\|\mathbf{w}\|_1 + \lambda_3\boldsymbol{\alpha}^T K_{\mathbf{w}}\boldsymbol{\alpha}]$. The estimation error is defined as $E\{L(Y, \hat{f})\} - E\{L(Y, f_{\boldsymbol{\lambda}})\}$, and the approximation error is defined to be $E\{L(Y, f_{\boldsymbol{\lambda}})\} - E\{L(Y, f_0)\}$.

Next, consider the function space $\hat{f}$ lies in, and denote it by $\mathcal{F}_{\boldsymbol{\lambda}}$. Define $g_f(\cdot) = s^{-1}\{L(\cdot, f) - L(\cdot, f_{\boldsymbol{\lambda}})\}$, where s is chosen such that the L_2 diameter of $\mathcal{G} = \{g_f : f \in \mathcal{F}_{\boldsymbol{\lambda}}\}$ is 1. Using Lemma A.0.1, one can verify that $s = O_P\{\log(n)\}$. From Lemma 2 in Zhang et al. (2015), we have that the upper bound of the L_2 entropy number of $\mathcal{G}$, $\log[N\{\eta, \mathcal{G}, L_2(T_X)\}]$, is of the order $O_P(\eta^{-2})$ (see, for example, Van der Vaart and Wellner, 2000, for introduction of the entropy numbers). Here T_X is the empirical measure of a training set, and the L_2 norm is $\|f\|_{L_2(T_X)} = \{n^{-1}\sum_{i=1}^n |f(\boldsymbol{x}_i, y_i)|^2\}^{1/2}$. Consequently, one can obtain that the estimation error is of the order $O_P\{\log(n)/\sqrt{n}\}$, by similar arguments as in the proof of Theorem 1 in Zhang et al. (2015). Therefore, by Lemma A.0.3, $\|\hat{f} - f_{\boldsymbol{\lambda}}\|_2 = O_P\{\log(n)/\sqrt{n}\}$.

On the other hand, to derive the bound for the approximation error, one can use Assumption 1, Lemmas A.0.2 and A.0.3. In particular, we have that $E[L\{Y, f_{\boldsymbol{\lambda}}(\boldsymbol{X})\}] - E[L\{Y, f_0(\boldsymbol{X})\}]$ converges at a rate faster than that of $\|f_{\boldsymbol{\alpha}_n^*, b_n^*} - f_0\|_2^2$ (recall the definition of $f_{\boldsymbol{\lambda}}$), which is $O_P[\{\log(n)/n\}^2] = O_P\{\log^2(n)/(n^2)\}$. Thus, by Lemma A.0.3, we have that $\|f_{\boldsymbol{\lambda}} - f_0\|_2 = O_P\{\log(n)/n\}$. Consequently, one has that $\|\hat{f} - f_0\|_2 \leq (\|\hat{f} - f_{\boldsymbol{\lambda}}\|_2 + \|f_{\boldsymbol{\lambda}} - f_0\|_2) = O_P\{\log(n)/\sqrt{n}\}$. This completes the proof. ■

Proof of Theorem 2.3.2: In the proof, we first assume that for regression problems, the distribution of the error has a bounded range. We will consider the more general case of sub-Gaussian distribution later.

The next lemma, Lemma A.0.4, is an important intermediate step to the proof of Theorem 2.3.2. With Lemma A.0.4, we can prove that the difference between $\hat{f}$ and the best function f_0, in terms of the difference in their expected partial derivatives with respect to w_j, is converging at the rate at least $O_P\{\log(n)/\sqrt{n}\}$. This further leads to the fact that the proposed λ_2 in Theorem 2.3.2 can correctly select the important variables $\boldsymbol{x}_{(1)}$ and discard the noise $\boldsymbol{x}_{(0)}$. Consequently, we can have the desired selection consistency for our DOSK method.

Lemma A.0.4. *Suppose Assumptions 1-7 are valid. With λ_1, λ_2 and λ_3 as in Theorem 2.3.2, we have that for any $j = 1, \ldots, p$,*

$$\left|\left[\frac{\partial E[L\{Y, \hat{f}(\boldsymbol{X})\}]}{\partial w_j} - \frac{\partial E[L\{Y, f_0(\boldsymbol{X})\}]}{\partial w_j}\right] |_{w_j=0,\ w_i=w_i^*,\ i\neq j}\right| = O_P\left\{\frac{\log(n)}{\sqrt{n}}\right\}.$$

Proof of Lemma A.0.4: The proof follows a similar line as that of Theorem 2.3.1 and Lemma A.0.3. □

We are ready to present the proof to Theorem 2.3.2.

First, we prove that for any j,

$$\begin{aligned}&\left|\left[\frac{\partial[\frac{1}{n}\sum_{i=1}^n L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]}{\partial w_j} - \frac{\partial E[L\{Y, f_0(\boldsymbol{X})\}]}{\partial w_j}\right] |_{w_j=0,\ w_i=w_i^*,\ i\neq j}\right| \\ =&O_P\left\{\frac{\log(n) \vee \log(p)}{\sqrt{n}}\right\}.\end{aligned} \tag{A.1}$$

To this end, observe that

$$\begin{aligned}&\left|\left[\frac{\partial[\frac{1}{n}\sum_{i=1}^n L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]}{\partial w_j} - \frac{\partial E[L\{Y, f_0(\boldsymbol{X})\}]}{\partial w_j}\right]\right| \\ \leq&\left|\left[\frac{\partial[\frac{1}{n}\sum_{i=1}^n L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]}{\partial w_j} - \frac{\partial E[L\{Y, \hat{f}(\boldsymbol{X})\}]}{\partial w_j}\right]\right| \\ +&\left|\left[\frac{\partial E[L\{Y, \hat{f}(\boldsymbol{X})\}]}{\partial w_j} - \frac{\partial E[L\{Y, f_0(\boldsymbol{X})\}]}{\partial w_j}\right]\right|.\end{aligned} \tag{A.2}$$

As Lemma A.0.4 bounds the second term on the RHS of (A.2), we proceed to show that the first term converges at the rate $O_P[\{\log(n) \vee \log(p)\}/\sqrt{n}]$. To this end, we need to introduce the Rademacher complexity (Mohri et al., 2012). In particular, let $\sigma_i;\ i = 1, \ldots, n$ be *i.i.d.* random variables, each taking the value 1 with probability 1/2, and −1 with probability 1/2. Let the set of training observations $(\boldsymbol{x}_i, y_i);\ i = 1, \ldots, n$, which are *i.i.d.* from P, be denoted by S. Define the function class $\mathcal{H}_n(\boldsymbol{\lambda})$ as $\mathcal{H}_n(\boldsymbol{\lambda}) = \{\hat{f} : \hat{f} = \text{argmin}_{\boldsymbol{\alpha},b,w}\, \phi(\boldsymbol{\lambda})\}$, where $\phi(\boldsymbol{\lambda})$ is the objective function in (2.6). With S fixed, we define the empirical Rademacher complexity of the function class $\mathcal{H}_n(\boldsymbol{\lambda})$ as

$$\hat{R}_n\{\mathcal{H}_n(\boldsymbol{\lambda})\} = E_{\boldsymbol{\sigma}}\{\sup_{f \in \mathcal{H}_n(\boldsymbol{\lambda})} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(\boldsymbol{x}_i)\},$$

where $E_{\boldsymbol{\sigma}}$ represents the expectation with respect to $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)$. Furthermore, denote the Rademacher complexity of $\mathcal{H}_n(\boldsymbol{\lambda})$ by

$$R_n\{\mathcal{H}_n(\boldsymbol{\lambda})\} = E_S \hat{R}_n\{\mathcal{H}_n(\boldsymbol{\lambda})\},$$

where E_S is the expectation with respect to the distribution of the sample S.

To bound the first term on the RHS of (A.2), we have the following lemma.

Lemma A.0.5. *Suppose Assumptions 1-7 are valid. With λ_1, λ_2 and λ_3 as in Theorem 2.3.2, we have that, for any $j = 1, \ldots, p$, with probability at least $1 - \delta$,*

$$\begin{aligned} \left| \left[\frac{\partial[\frac{1}{n}\sum_{i=1}^{n} L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]}{\partial w_j} - \frac{\partial E[L\{Y, \hat{f}(\boldsymbol{X})\}]}{\partial w_j} \right] \right| &\le C_1 R_n\{\mathcal{H}_n(\boldsymbol{\lambda})\} + T_n(\delta) \\ &\le C_1 \hat{R}_n\{\mathcal{H}_n(\boldsymbol{\lambda})\} + 3T_n(\delta/2), \end{aligned} \qquad \text{(A.3)}$$

where $T_n(\delta) = C_2\{n^{-1}\log(n)\log(1/\delta)\}^{1/2}$, and C_1, C_2 are universal constants that are independent of n.

The proof to Lemma A.0.5 is quite standard in the literature of Rademacher complexity. To bound the LHS of (A.3) by $C_1 R_n\{\mathcal{H}_n(\boldsymbol{\lambda})\} + T_n(\delta)$, one can use the McDiarmid inequality

(McDiarmid, 1989) and the symmetrization technique (Van der Vaart and Wellner, 2000). To bound $C_1 R_n\{\mathcal{H}_n(\boldsymbol{\lambda})\}$ by $C_1 \hat{R}_n\{\mathcal{H}_n(\boldsymbol{\lambda})\} + 2T_n(\delta/2)$, one can again use the McDiarmid inequality. See the proof of Lemma 3 in Zhang et al. (2015) for more details. Notice that there are two main differences between the proof of Lemma 3 in Zhang et al. (2015) and that of Lemma A.0.5. First, in Zhang et al. (2015), the Rademacher complexity was defined on the function class $\{L(\cdot, f) : f \in \mathcal{H}_n(\boldsymbol{\lambda})\}$. By Talagrand's Lemma (Lemma 4.2 in Mohri et al., 2012), the Rademacher complexity of $\{L(\cdot, f) : f \in \mathcal{H}_n(\boldsymbol{\lambda})\}$ can be further bounded by that of $\mathcal{H}_n(\boldsymbol{\lambda})$, if the loss function L is Lipshcitz. Second, the maximum change in the LHS of (A.3) if we replace one $\boldsymbol{x}_i$ or y_i can be bounded by $C_3 \log(n)/n$ (this is a direct result from Lemma A.0.1) with C_3 being another constant, instead of $O(n^{-1})$ as in Zhang et al. (2015). The rest of the proof is analogous to that of Lemma 3 in Zhang et al. (2015), and we omit the details here. □

The next step is to bound the empirical Rademacher complexity of $\mathcal{H}_n(\boldsymbol{\lambda})$. To this end, recall the definition of $\tilde{f}$, and notice that

$$E_{\boldsymbol{\sigma}}\{\sup_{f\in\mathcal{H}_n(\boldsymbol{\lambda})} \frac{1}{n}\sum_{i=1}^{n} \sigma_i f(\boldsymbol{x}_i)\} \leq E_{\boldsymbol{\sigma}}\{\sup_{f\in\mathcal{H}_n(\boldsymbol{\lambda})} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \tilde{f}(\boldsymbol{x}_i)\} + E_{\boldsymbol{\sigma}}\{\sup_{f\in\mathcal{H}_n(\boldsymbol{\lambda})} \frac{1}{n}\sum_{i=1}^{n} \sigma_i b\}. \qquad \text{(A.4)}$$

Hence, we proceed to bound the two terms on the RHS of (A.4). Notice that by Lemma A.0.1, the first term is equivalent to $E_{\boldsymbol{\sigma}}\{\sup_{\|\tilde{f}\|_{\mathcal{H}}=O_P\{\log(n)\}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \tilde{f}(\boldsymbol{x}_i)\}$, and the second term is equivalent to $E_{\boldsymbol{\sigma}}(\sup_{|b|=O_P\{\log(n)\}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i b)$. For the first term, one can use Theorem 5.5 in Mohri et al. (2012) to obtain that, with Assumption 2 valid, the corresponding empirical Rademacher complexity is of the order $O_P\{\log(n)/\sqrt{n}\}$. For the second term, notice that the distribution of Rademacher variables is similar to the binomial distribution. Therefore, we have that for large n, the distribution of $\sup_{|b|=O_P\{\log(n)\}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i b$ can be approximated by that of $|Z|$, where $\{C\sqrt{n}/\log(n)\}Z \sim N(0, 1)$, with C a universal constant. Hence, one can verify that

$$E_{\boldsymbol{\sigma}}\{\sup_{|b|=O_P\{\log(n)\}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i b\} = E(|Z|) = O_P\{\log(n)/\sqrt{n}\}.$$

Consequently, we have that $E_{\boldsymbol{\sigma}}\{\sup_{f\in\mathcal{H}_n(\boldsymbol{\lambda})} \frac{1}{n}\sum_{i=1}^{n}\sigma_i f(\boldsymbol{x}_i)\} = O_P\{\log(n)/\sqrt{n}\}$.

Next, choose $\delta = 2p^{-1}n^{-2}$. One has that $T_n(\delta/2) = O_P[n^{-1}\log(n)\{\log(p)\vee\log(n)\}]^{1/2}$. Consequently, with probability at least $2n^{-2}$, (A.4) holds true for all the predictors. Combining this with Lemma A.0.4 and the Borel–Cantelli Lemma, we have that (A.1) is proved.

We now need to show that $\frac{1}{n}\sum_{i=1}^{n} L\Big\{y_i, \sum_{j=1}^{n} K_{\mathbf{w}}(x_i,x_j)\alpha_j+b\Big\}$, as a function of $(\mathbf{w}^T,\boldsymbol{\alpha}^T,b)^T$, is strictly convex in a small neighborhood around $\Big((\mathbf{w}^*)^T,(\boldsymbol{\alpha}_n^*)^T,b_n^*\Big)^T$. Because we have shown that $f_{\boldsymbol{\alpha}_n^*,b_n^*}(\boldsymbol{x})$ converges to f_0 in a rate faster than that of $\hat{f}$ to f_0, this guarantees that once we arrive at a temporary point around $\Big((\mathbf{w}^*)^T,(\boldsymbol{\alpha}_n^*)^T,b_n^*\Big)^T$, the proposed algorithm in Section 2.2.3 would ensure that the solution $\hat{f}$ converges to the best function f_0. To this end, observe that in Assumption 5, we assume that $E\left[\frac{1}{n}\sum_{i=1}^{n} L\{Y_i, f(\boldsymbol{X}_i)\}\right]$ is strictly convex. Hence, it suffices to prove that

$$\sup_{(\mathbf{w}^T,\boldsymbol{\alpha}^T,b)^T\in\mathcal{N}} |\frac{1}{n}\sum_{i=1}^{n} L\Big\{y_i, \sum_{j=1}^{n} K_{\mathbf{w}}(x_i,x_j)\alpha_j+b\Big\} - E[\frac{1}{n}\sum_{i=1}^{n} L\{Y_i, f(\boldsymbol{X}_i)\}]| \to 0$$

almost surely. Note that when $\mathcal{N}$ is sufficiently small, we have $\sup_{f\in\mathcal{N}}|Pf| < \infty$. Moreover, by Lemma A.0.1 and similar arguments as in the proof of Theorem 1 in Zhang et al. (2015), one can have that the L_2 entropy of $\{f : f\in\mathcal{N}\}$ is $\log[N\{\epsilon,\mathcal{N},L_2(P_n)\}] = O[\log\{\log(n)\}]$, where P_n is the empirical measure of the training set. For any $M<\infty$, define $f_M = f\cdot I(f\leq M)$, and $\mathcal{N}_M = \{f_M : f\in\mathcal{N}\}$. One has that $\log[N\{\epsilon,\mathcal{N}_M,L_2(P_n)\}] = O[\log\{\log(n)\}]$. Therefore, by Theorem 6.2 in Wellner (2005), we have that $\mathcal{N}$ is a P-Glivenko-Cantelli class. One can then verify that this conclusion leads to that for n large, $\frac{1}{n}\sum_{i=1}^{n} L\Big\{y_i, \sum_{j=1}^{n} K_{\mathbf{w}}(x_i,x_j)\alpha_j+b\Big\}$ is convex.

Now we have that, by Assumption 6, the partial derivative of the empirical L loss with respect to each w_j is such that

$$\frac{\partial[\frac{1}{n}\sum_{i=1}^{n} L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]}{\partial w_j} \,|_{w_j=0,\ w_i=w_i^*,\ i\neq j} \preceq O_P\left\{\frac{\{\log(p)\vee\log(n)\}}{\sqrt{n}}\right\},$$

for $w_j \in \mathbf{w}_{(0)}$, and

$$\left[\frac{\partial[\frac{1}{n}\sum_{i=1}^{n} L\{y_i, \hat{f}(\boldsymbol{x}_i)\}]}{\partial w_j} - \frac{\partial E[L\{Y, f_0(\boldsymbol{X})\}]}{\partial w_j}\right] |_{w_j=0,\ w_i=w_i^*,\ i\neq j} \preceq O_P\left\{\frac{\{\log(p)\vee\log(n)\}}{\sqrt{n}}\right\},$$

for $w_j \in \mathbf{w}_{(1)}$. Because the objective function is locally convex, at the optimal point $(\hat{\mathbf{w}}, \hat{\boldsymbol{\alpha}}, \hat{b})$, selection consistency is equivalent to that $\lambda_2 \to 0$ at a rate no faster than $O_P\left\{\frac{\{\log(p)\vee\log(n)\}}{\sqrt{n}}\right\}$ (recall the soft thresholding rule in Tibshirani, 1996). Hence, we have proven the selection consistency for the DOSK method under the assumption that the distribution of the error has a bounded range.

Lastly, we need to finish the proof by considering the general case that the distribution of the error in regression is sub-Gaussian. This can be done by showing that with a high probability, the actual errors would be bounded in a range. Then we can prove that the corresponding partial derivatives etc. converge at the same rate, because the probability of sub-Gaussian random variables being significantly away from 0 converges to zero very fast, as the bound increases.

Without loss of generality, we assume that $\epsilon(\boldsymbol{X})$ follows a common sub-Gaussian distribution with c.d.f. Φ_ϵ. The generalization of this assumption to the heteroscedastic case is straightforward, because we are only concerned with the tail probability $\text{pr}(|\epsilon(\boldsymbol{X})| > t)$. Next, define $t^* = \Phi_\epsilon^{-1}\Big(0.5 + 0.5(1-\delta/2)^{1/n}\Big)$, where δ is a small positive number. It can be verified that with probability at least $1-\delta/2$, all the errors $\epsilon_i;\ i = 1, \ldots, n$ are in $[-t^*, t^*]$. Since Φ_ϵ is the c.d.f. of a sub-Gaussian distribution with a fixed parameter, t^* diverges at a rate slower than $O\{\log(n)\}$. One can check that the RHS of (A.2) can be bounded similarly as in the corresponding proofs, and this completes the proof. ■

Proof of Theorem 2.3.3: The proof of this theorem is analogous to that of Lemma A.0.5

and the second half of Theorem 2.3.2 (i.e., obtaining the bound on the empirical Rademacher complexity of $\mathcal{H}_n(\boldsymbol{\lambda})$, as well as the convergence rate of $T_n(\delta/2)$). Therefore we omit the details here. ■

APPENDIX B: ESTIMATING INDIVIDUAL TREATMENT RULES FOR ORDINAL TREATMENTS

B.1 Computational Algorithm for GOWL

Recall the main optimization problem

$$\sum_{i=1}^{n}\sum_{k=1}^{K-1}\frac{|r_i|}{P(a_i|x_i)}\left[I(r_i \geq 0)\left[1-a_i^{(k)}f(x_i^{(k)})\right]_+ + I(r_i<0)\left[1+a_i^{(k)}f(x_i^{(k)})\right]_+\right]+\lambda||f||^2. \quad \text{(B.1)}$$

We now introduce our algorithm to solve (B.1). Due to the convexity of the objective function in (B.1), we generalize the primal-dual method Vazirani (2013) used in SVM to estimate the classifier $f(x_i^{(k)})$. Starting from (B.1), by introducing a series of slack variable $\xi_i^{(k)}$ and $\psi_i^{(k)}$ for all observations $i = 1, \cdots, n$ and all duplicates $k = 1, \cdots, K-1$, we rewrite the minimization in (B.1) by minimizing the following objective function with respect to f and all slack variables,

$$\sum_{i=1}^{n}\sum_{k=1}^{K-1}\frac{\left|r_i^{(k)}\right|}{P(a_i|x_i)}\left[I(r_i^{(k)} \geq 0)\xi_i^{(k)} + I(r_i^{(k)}<0)\psi_i^{(k)}\right]+\lambda||f||^2, \quad \text{(B.2)}$$

with $\xi_i^{(k)} \geq 0, \psi_i^{(k)} \geq 0, \xi_i^{(k)} \geq 1-a_i^{(k)}f(x_i^{(k)})$, and $\psi_i^{(k)} \geq 1+a_i^{(k)}f(x_i^{(k)})$.

Next, we discuss how to solve (B.2) for the linear case in Section B.1.1 and the non-linear case in Section B.1.2.

B.1.1 Linear Decision Function Estimation

Suppose that the decision function $f(x_i^{(k)})$ above is a linear function of $x_i^{(k)}$ with the slope $\tilde{\beta}$ and an intercept $\tilde{b}$, i.e. $f(x_i^{(k)}) = \left[x_i^{(k)}\right]^T\tilde{\beta}+\tilde{b}$. Before introducing the algorithm, we express $f(x_i^{(k)}) = \left[x_i^{(k)}\right]^T\tilde{\beta}+\tilde{b} = x_i\beta + b_k$ by denoting $x_i^{(k)} = (x_i^T, e_k^T)^T$, where e_k^T is a $K-1$ dimensional row vector whose kth element is 1 while others are zeros. Note that $\tilde{\beta}^T = (\beta^T, b_1-\tilde{b}, \cdots, b_{K-1}-\tilde{b})$. In other words, the decision function on the duplicated covariate set $x_i^{(k)}$ can also be understood as a varying intercept function of x_i, i.e. $f(x_i^{(k)}) = g(x_i)+b_k$. On one hand, such a form of the decision function constructs $K-1$ parallel boundaries in the original sample space to avoid contradicting classifying results. On the other hand, for the ordinal treatment scenario, it is usually desirable to have the $K-1$ intercepts monotonic along the treatment group in terms of the interpretation, i.e. $b_i < (>) b_{i+1}$ for all $i = 1, \cdots, K-2$ when $K \geq 3$. We show in Section 4 that GOWL enjoys such a property under a reasonable condition. When the assumption of parallel linear boundaries becomes too strong, one can use nonlinear learning techniques to achieve more flexible boundaries as in Section 3.3.2.

To solve (B.2) with a linear decision function, we plug the expression of $f(x_i^{(k)})$ above back into (B.2) and reparamatrize the formula as:

$$\min_{\tilde{\beta},\xi,\psi} \left\{ \frac{1}{2}||\tilde{\beta}||^2 + C \sum_{i=1}^{K-1} \sum_{k=1}^{K-1} \frac{\left|r_i^{(k)}\right|}{P(a_i|x_i)} \left[I(r_i^{(k)} \geq 0)\xi_i^{(k)} + I(r_i^{(k)} < 0)\psi_i^{(k)} \right] \right\},$$

with $\xi_i^{(k)} \geq 0, \psi_i^{(k)} \geq 0, \xi_i^{(k)} \geq 1 - a_i^{(k)} f(x_i^{(k)}),\ \ \psi_i^{(k)} \geq 1 + a_i^{(k)} f(x_i^{(k)})$, and (ξ,ψ) denote all slack variables. By introducing the Lagrange multipliers, we can derive the Lagrange function for the primal problem as:

$$\begin{aligned} L_P &= \frac{1}{2}||\tilde{\beta}||^2 + C \sum_{i=1}^{n} \sum_{k=1}^{K-1} \frac{\left|r_i^{(k)}\right|}{P(a_i|x_i)} \left[I(r_i^{(k)} \geq 0)\xi_i^{(k)} + I(r_i^{(k)} < 0)\psi_i^{(k)} \right] \\ &- \sum_{i=1}^{n} \sum_{k=1}^{K-1} \mu_i^{(k)} \xi_i^{(k)} - \sum_{i=1}^{n} \sum_{k=1}^{K-1} \nu_i^{(k)} \psi_i^{(k)} - \sum_{i=1}^{n} \sum_{k=1}^{K-1} \alpha_i^{(k)} \left[a_i^{(k)} f(x_i^{(k)}) + \xi_i^{(k)} - 1 \right] \\ &- \sum_{i=1}^{n} \sum_{k=1}^{K-1} \eta_i^{(k)} \left[-a_i^{(k)} f(x_i^{(k)}) + \psi_i^{(k)} - 1 \right]. \end{aligned}$$

The corresponding dual problem can be derived by taking partial derivatives with respect to $(\tilde{\beta}, \xi, \psi)$ and simplifying the results using the Karush–Kuhn–Tucker conditions. Then, the dual problem becomes maximizing L_D with respect to the slack variables $\{\alpha_i^{(k)}, \eta_i^{(k)}; i = 1, \ldots, n; k = 1, \ldots, K-1\}$, where

$$\begin{aligned} L_D &= \sum_{i=1}^{n} \sum_{k=1}^{K-1} \alpha_i^{(k)} + \sum_{i=1}^{n} \sum_{k=1}^{K-1} \eta_i^{(k)} - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K-1} \sum_{j=1}^{n} \sum_{h=1}^{K-1} \alpha_i^{(k)} \alpha_j^{(h)} a_i^{(k)} a_j^{(k)} \left([x_i^{(k)}]^T \cdot [x_j^{(h)}] \right) \\ &- \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K-1} \sum_{j=1}^{n} \sum_{h=1}^{K-1} \eta_i^{(k)} \eta_j^{(h)} a_i^{(k)} a_j^{(k)} \left([x_i^{(k)}]^T \cdot [x_j^{(h)}] \right), \end{aligned}$$

with $0 \leq \alpha_i^{(k)} \leq \frac{C \cdot r_i^{(k)}}{P(a_i|x_i)} I(r_i^{(k)} \geq 0), 0 \leq \eta_i^{(k)} \leq \frac{C \cdot r_i^{(k)}}{P(a_i|x_i)} I(r_i^{(k)} < 0)$, and $\sum_{i=1}^n (\alpha_i^{(k)} - \eta_i^{(k)}) a_i^{(k)} = 0$. Note that the parameters in the dual problem above can be solved by applying standard quadratic programming with linear constrains. Furthermore, the slope estimate can be obtained via $\hat{\tilde{\beta}} = \sum_{i=1}^n \sum_{k=1}^{K-1} (\hat{\alpha}_i^{(k)} a_i^{(k)} \text{sign}(r_i^{(k)} \geq 0) x_i^{(k)})$. The intercept vector $\{b_1, \cdots, b_{K-1}\}$ can be estimated by plugging $\hat{\tilde{\beta}}$ back into the original maximization in (B.1) and solving a standard linear programming problem with linear constraints (Vazirani (2013)). Because there are $2n(K-1)$ parameters in the dual problem above, with a finite K, the computational complexity of (B.1) is the same as that of the standard primal-dual problem in the SVM.

B.1.2 Nonlinear Decision Function Estimation

The previous subsection solves (B.2) for the linear case. However, in practice, the linear assumption can be too strong for some problems. To make our model more flexible, we perform nonlinear learning by applying the kernel learning approach in Reproducing Kernel Hilbert Spaces (RKHS). Kernel learning in RKHS is flexible and has achieved great successes in many nonlinear learning studies (Kimeldorf and Wahba, 1970; Hastie et al., 2011).

Under the binary treatment case, we can show by the Representer Theorem (Kimeldorf and Wahba (1970)) that under some regularity conditions, the decision function on the data $(x_i^{(1)}, a_i^{(1)}, r_i^{(1)})$ can be written in the form that $f(x_i^{(1)}) = \sum_{j=1}^n k(x_i, x_j) c_j + \tilde{b}$, where $k(\cdot, \cdot)$ is the standard kernel function associated with the RKHS $\mathcal{H}$. When the treatment is extended into an ordinal variable, we need to define an extended version of the kernel function on the duplicated covariates $x_i^{(k)}$ to construct the decision function. In particular, we have $f(x_i^{(k)}) = \sum_{j=1}^n \sum_{h=1}^{K-1} \tilde{k}(x_i^{(k)}, x_j^{(h)}) \tilde{c}_j^{(h)} + \tilde{b}$, where $\tilde{k}(\cdot, \cdot)$ is the extended kernel function with the definition $\tilde{k}(x_i^{(k)}, x_j^{(h)}) = k(x_i, x_j) + e_k^T \cdot e_h$, and e_k is defined as in Section 3.3.1. Similar discussions were made in Ling and Lin (2006) and Cardoso and Pinto da Costa (2007). According to the newly defined extended kernel, $f(x_i^{(k)})$ can be rewritten as $\sum_{j=1}^n k(x_i, x_j) c_j + b_k$, where $c_j = \sum_{h=1}^{K-1} \tilde{c}_j^{(h)}$ and $b_k = \sum_{j=1}^n \tilde{c}_j^{(k)} + \tilde{b}$. One can tell from the new $f(x_i^{(k)})$ expression that due to the conversion of the ordinal problem into a big binary problem, the corresponding decision boundaries in the kernel-induced feature space are guaranteed not to cross with each other. Consequently, the sets $\{f(x^{(k)}) < 0\}$ for $1 \le k \le K-1$ produce more flexible noncrossing boundaries for the K ordinal treatments in the original space.

Given the expression of f with respect to the kernel representation, we can follow similar Lagrange optimizer steps as before to obtain the generalized primal-dual formula. We can derive the dual problem of maximizing L_D with respect to all slack variables, where

$$\begin{aligned} L_D &= \sum_{i=1}^{n}\sum_{k=1}^{K-1}\alpha_i^{(k)} + \sum_{i=1}^{n}\sum_{k=1}^{K-1}\eta_i^{(k)} - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K-1}\sum_{j=1}^{n}\sum_{h=1}^{K-1}\alpha_i^{(k)}\alpha_j^{(h)}a_i^{(k)}a_j^{(k)}\tilde{k}\left(x_i^{(k)}, x_j^{(h)}\right) \\ &- \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K-1}\sum_{j=1}^{n}\sum_{h=1}^{K-1}\eta_i^{(k)}\eta_j^{(h)}a_i^{(k)}a_j^{(k)}\tilde{k}\left(x_i^{(k)}, x_j^{(h)}\right), \end{aligned}$$

with $0 \le \alpha_i^{(k)} \le \frac{C\cdot r_i^{(k)}}{P(a_i|x_i)}I(r_i^{(k)} \ge 0), 0 \le \eta_i^{(k)} \le \frac{C\cdot r_i^{(k)}}{P(a_i|x_i)}I(r_i^{(k)} < 0)$, and $\sum_{i=1}^{n}(\alpha_i^{(k)} - \eta_i^{(k)})a_i^{(k)} = 0$. After the dual coefficients are estimated, the decision function can be written as $f(x_i^{(k)}) = \sum_{j=1}^{n}\sum_{h=1}^{K-1}\tilde{k}(x_i^{(k)}, x_j^{(h)})(\hat{\alpha}_j^{(h)}a_j^{(h)}\text{sign}(r_j^{(h)} \ge 0))$.

To implement the quadratic programming in the dual problems above, we use the open source package CVXOPT based on the Python programming in practice.

B.2 Proofs of Theorems

In this Section, we give the technical proofs for Lemma 3.4.1 and Theorem 3.4.1-3.4.5.

Proof of Lemma 3.4.1 We show the Fisher consistency property of GOWL for the binary treatment case in Lemma 3.4.1. Note that when $A \in \{1, 2\}$, the true optimal treatment rule $\mathcal{D}^*(x)$ can be rewritten as

$$I\left(E(R|X = x, A = 2) - E(R|X = x, A = 1) > 0\right) + 1.$$

Because X contains the intercept term, the duplicated covariate matrix $X^{(k)}$ is degenerated into X at this time with $A^{(k)} = A^{(1)} = \text{sign}(A - 1)$. Starting from the ϕ-risk $\mathcal{R}_\phi(f)$, we apply the total probability theorem to obtain

$$\begin{aligned} E[\frac{|R|}{P(A|X)}\phi\left(A^{(1)}f(X)), R|X\right)] &= \sum_a E\{\frac{|R|}{P(a|x)}\phi(A^{(1)}f(X)), R)|X, A^{(1)}=a\}P(a|x) \\ &= E\{|R|\phi(f(X)), R)|X, A^{(1)}=1\} \\ &\quad +E\{|R|\phi(-f(X)), R)|X, A^{(1)}=-1\}, \end{aligned}$$

where the summation is taken on $a \in \{-1, 1\}$, $\phi\left(Af(X)), R\right) = I(R \geq 0)\left[1 - Af(X)\right]_+ + I(R < 0)\left[1 + Af(X)\right]_+$ and the event $A^{(1)} = 1$ is equivalent to that of $A = 2$. To obtain an explicit minimizer f_ϕ^* for the objective function above, we need to discuss its range. In particular, when $f < -1$,

$$\begin{aligned} E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(1)}f(X)), R\right)\right] &= E\left[RI(R\geq 0)(1-f(X))|X, A=2\right] \\ &\quad -E\left[RI(R<0)(1-f(X))|X, A=1\right] \\ &= \{E\left[RI(R<0)|X, A=1\right] \\ &\quad -E\left[RI(R\geq 0)|X, A=2\right]\}f(X) \\ &\quad +E\left[RI(R<0)|X, A=1\right] \\ &\quad -E\left[RI(R\geq 0)|X, A=2\right]. \end{aligned}$$

In this case, one can tell that $E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(1)}f(X)), R\right)\right]$ is always non-negative. This is because for $R \geq 0$, $E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(1)}f(X)), R\right)\right] = -E\left[R|X, A=2\right]f(X) - E\left[R|X, A=2\right] \geq 0$ since $f < -1$. In addition, a similar argument can be made for $R < 0$. When $f > 1$, one can show that $E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(1)}f(X)), R\right)\right] \geq 0$ still holds based on the same derivation.

When $-1 \leq f \leq 1$, noting that $\phi\left(A^{(1)}f(X)), R\right) = I(R \geq 0)\left(1 - A^{(1)}f(X)\right) + I(R <$

0) $\Big(1 + A^{(1)} f(X)\Big)$, we have

$$\begin{aligned}
E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(1)}f(X)),R\right)\right] &= E[RI(R \geq 0)(1-f(X)) \\
&\quad -RI(R<0)(1+f(X))|X, A=2] \\
&\quad +E[RI(R \geq 0)(1+f(X)) \\
&\quad -RI(R<0)(1-f(X))|X, A=1] \\
&= \{E\left[R|A=1\right] - E\left[R|A=2\right]\} f(X) \\
&\quad +E\left[R|A=2\right] - E\left[R|A=1\right].
\end{aligned}$$

The right hand side of the equation above shows that, $E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(1)}f(X)),R\right)\right]$ becomes zero when $f(X) = 1$ and takes negative values as long as $E\left[R|A=2\right] - E\left[R|A=1\right] < 0$. Therefore, the minimizer of the $\phi-$risk, f^*_ϕ, should be within the interval $[-1, 1]$. More specifically, f^*_ϕ should satisfy $\text{sign}(f^*_\phi) = \text{sign}(E\left[R|A=2\right] - E\left[R|A=1\right])$, which indicates the surrogate ITR $\mathcal{D}^*_\phi(x) = I\left(f^*_\phi(x) > 0\right) + 1 = \mathcal{D}^*(x)$.

Proof of Theorem 3.4.1 We show that the Fisher consistency property of GOWL still holds for the ordinal treatment case in Theorem 3.4.1. By Lemma 3.4.1, for each $\mathcal{R}^{(k)}_\phi$ for which $k = 1, \cdots, K-1$, we note that the minimizer f^*_ϕ has a universal formula for all k so that $\text{sign}(f^*(x^{(k)})) = \text{sign}\Big(E(R|X^{(k)} = x^{(k)}, A^{(k)} = 1) - E(R|X^{(k)} = x^{(k)}, A^{(k)} = -1)\Big)$. Therefore, by definition, the surrogate ITR of x is

$$\begin{aligned}
\mathcal{D}^*_\phi(x) &= \sum_{k=1}^{K-1} I(E(R^{(k)}|X^{(k)} = x^{(k)}, A^{(k)} = 1) > E(R|X^{(k)} = x^{(k)}, A^{(k)} = -1)) + 1 \\
&= \sum_{k=1}^{K-1} I\left(E(R|X=x, A>k) > E(R|X=x, A \leq k)\right) + 1 \\
&= \sum_{k=1}^{K-1} I(\mathcal{D}^*(x) > k) + 1 \\
&= \mathcal{D}^*(x).
\end{aligned}$$

The second equation holds due to the definition of a duplicated data set that $X^{(k)} = (X, k)$ and $A^{(k)} = \text{sign}(A - k)$. The third equation holds due to the reward distribution assumption in Theorem 3.4.1. Note that the second and third equations always hold under the modified duplicate method which defines $r_i^{(k)} = r_i$ if $a_i \in \{k, k+1\}$.

Proof of Theorem 3.4.2 Recall the discussion that the $\phi-$risk can be rewritten as

$$\mathcal{R}_\phi = E\{\sum_{k=1}^{K-1}[\frac{|R|}{P(A|X)}(I(R \geq 0)\phi_1(A^{(k)}(g(x) + b_k)) + I(R < 0)\phi_2(A^{(k)}(g(x) + b_k)))]\},$$

where $\phi_1(u) = [1 - u]_+$ and $\phi_2(u) = [1 + u]_+$. Without loss of generality, we only need to show that under $E(R|A = k) > 0$ for $k = 1, \cdots, K$, the $\phi-$risk will not be decreased by swapping any two neighbors in the intercept vector $\boldsymbol{b}$ under $b_k > b_{k+1}$ for $k = 1, \cdots, K - 2$. Suppose that we swap b_m and b_{m+1} for any $m \in \{1, \cdots, K - 2\}$, then the new $\phi-$risk based on the swapped $\boldsymbol{b}$ can be written as

$$\begin{aligned} \mathcal{R}_\phi^s \;=\; & E\{\sum_{k \neq m, m+1}[\frac{|R|}{P(A|X)}(I(R \geq 0)\phi_1(A^{(k)}(g(x) + b_k)) \\ & + I(R < 0)\phi_2(A^{(k)}(g(x) + b_k)))]\} \\ & + E[\frac{|R|}{P(A|X)}(I(R \geq 0)\phi_1(A^{(m)}(g(x) + b_{m+1})) \\ & + I(R < 0)\phi_2(A^{(m)}(g(x) + b_{m+1})))] \\ & + E[\frac{|R|}{P(A|X)}(I(R \geq 0)\phi_1(A^{(m+1)}(g(x) + b_m)) \\ & + I(R < 0)\phi_2(A^{(m+1)}(g(x) + b_m)))]. \end{aligned}$$

Now we discuss how the two risks above are different based on the values of $A^{(m)}$ and $A^{(m+1)}$. One can note that $A^{(m)} \geq A^{(m+1)}$ always holds for any m by the definition that $A^{(m)} = \text{sign}\,(A > m)$. In this way, there are three possible situations for the values of $\left(A^{(m)}, A^{(m+1)}\right)$ to take: $(1, -1)$, $(-1, -1)$, and $(1, 1)$. We discuss each situation as follows.

First, given the event $\mathcal{A}_{10}^{(m)} = \left\{A^{(m)} = 1, A^{(m+1)} = -1\right\}$, we have that the difference

When	$\psi_1(x,b)$, $\psi_2(x,b)$
$g(x)+b_{m+1} < g(x)+b_m \leq -1$	$b_m - b_{m+1} > 0$
$g(x)+b_{m+1} \leq -1 < g(x)+b_m$	$2 + b_m - b_{m+1} - \phi_1(g(x)+b_m) > 0$
$-1 < g(x)+b_{m+1} < g(x)+b_m$	$b_m - b_{m+1} + \phi_1(g(x)+b_{m+1}) - \phi_1(g(x)+b_m) > 0$

Table B.1: All possible $\psi_1(x,b)$ and $\psi_2(x,b)$ results

between the swapped risk and original $\phi-$risk is

$$\begin{aligned}
E_{\mathcal{A}_{10}^{(m)}}\left(\mathcal{R}_\phi^s - \mathcal{R}_\phi\right) &= E_{\mathcal{A}_{10}^{(m)}}[\frac{|R|}{P(m+1|x)} I(R \geq 0)(\phi_1(g(x)+b_{m+1}) - \phi_1(g(x)+b_m))] \\
&+ E_{\mathcal{A}_{10}^{(m)}}[\frac{|R|}{P(m+1|x)} I(R \geq 0)(\phi_1(-(g(x)+b_m)) \\
&\quad -\phi_1(-(g(x)+b_{m+1})))] \\
&+ E_{\mathcal{A}_{10}^{(m)}}[\frac{|R|}{P(m+1|x)} I(R < 0)(\phi_2(g(x)+b_{m+1}) - \phi_2(g(x)+b_m))] \\
&+ E_{\mathcal{A}_{10}^{(m)}}[\frac{|R|}{P(m+1|x)} I(R < 0)(\phi_2(-(g(x)+b_m)) \\
&\quad -\phi_2(-(g(x)+b_{m+1})))] \\
&= E[RI(R \geq 0) \cdot \psi_1(x,b)|\mathcal{A}_{10}^{(m)}] + E[RI(R < 0) \cdot \psi_2(x,b)|\mathcal{A}_{10}^{(m)}],
\end{aligned}$$

where $E_{\mathcal{A}_{10}^{(m)}}\left(\mathcal{R}_\phi^s - \mathcal{R}_\phi\right)$ denotes the difference of the two risks under the event $\mathcal{A}_{10}^{(m)} = \left\{A^{(m)} = 1, A^{(m+1)} = -1\right\}$, $\psi_1(x,b) = \phi_1(g(x)+b_{m+1}) - \phi_1(g(x)+b_m) + \phi_1(-(g(x)+b_m)) - \phi_1(-(g(x)+b_{m+1}))$, and $\psi_2(x,b) = \phi_2(g(x)+b_m) - \phi_2(g(x)+b_{m+1}) + \phi_2(-(g(x)+b_{m+1})) - \phi_2(-(g(x)+b_m))$. The difference of such conditional expected rewards depends on whether $g(x)+b_m$ and $g(x)+b_{m+1}$ are greater than -1 or not. We summarize the result of each scenario in Table B.1. One can find that $\psi_1(x,b)$ is always equal to $\psi_2(x,b)$ and they are always non-negative. In this way, $E_{\mathcal{A}_{10}^{(m)}}\left(\mathcal{R}_\phi^s - \mathcal{R}_\phi\right) = \psi_1(x,b)\{E[RI(R \geq 0)|\mathcal{A}_{10}^{(m)}] + E[RI(R < 0)|\mathcal{A}_{10}^{(m)}]\} = \psi_1(x,b)E(R|\mathcal{A}_{10}^{(m)})$. Thus, one can see that when $b_m > b_{m+1}$, $E_{\mathcal{A}_{10}^{(m)}}(\mathcal{R}_\phi^s - \mathcal{R}_\phi) > 0$ will hold for arbitrary $m = 1, \cdots, K-2$ under the assumption that $E(R|\mathcal{A}_{10}^{(m)}) > 0$ where $\mathcal{A}_{10}^{(m)} = \{A^{(m)} = 1, A^{(m+1)} = -1\} = \{A = m+1\}$.

Under the second situation when the event $\mathcal{A}_{11}^{(m)} = \left\{A^{(m)} = 1, A^{(m+1)} = 1\right\}$ holds, the conditional difference of the two risks can be expressed as

$$\begin{aligned}
E_{\mathcal{A}_{11}^{(m)}}\left(\mathcal{R}_\phi^s-\mathcal{R}_\phi\right) &= E_{\mathcal{A}_{11}^{(m)}}[\frac{|R|}{P(A>m+1|x)}I(R\geq 0)(\phi_1(g(x)+b_{m+1}) \\
&\quad -\phi_1(g(x)+b_m))] \\
&+ E_{\mathcal{A}_{11}^{(m)}}[\frac{|R|}{P(A>m+1|x)}I(R\geq 0)(\phi_1(g(x)+b_m) \\
&\quad -\phi_1(g(x)+b_{m+1}))] \\
&+ E_{\mathcal{A}_{11}^{(m)}}[\frac{|R|}{P(A>m+1|x)}I(R<0)(\phi_2(g(x)+b_{m+1}) \\
&\quad -\phi_2(g(x)+b_m))] \\
&+ E_{\mathcal{A}_{11}^{(m)}}[\frac{|R|}{P(A>m+1|x)}I(R<0)(\phi_2(g(x)+b_m) \\
&\quad -\phi_2(g(x)+b_{m+1}))] \\
&= 0.
\end{aligned}$$

Lastly, when the event $\mathcal{A}_{00}^{(m)}=\left\{A^{(m)}=-1,A^{(m+1)}=-1\right\}$ holds, $E_{\mathcal{A}_{11}^{(m)}}\left(\mathcal{R}_\phi^s-\mathcal{R}_\phi\right)=0$ and the deductions are the same as that in the second scenario. Therefore, when $E_{\mathcal{A}_{10}^{(k)}}[R]>0$, we will have $E\left(\mathcal{R}_\phi^s-\mathcal{R}_\phi\right)\geq 0$, which means $b_k>b_{k+1}$ always holds for $k=1,\cdots,K-1$. The same deduction can be made for $b_k<b_{k+1}$ when the assumption $E_{\mathcal{A}_{10}^{(k)}}[R]<0$ holds.

Proof of Theorem 3.4.3 We first decompose the 0-1 risk based on its definition,

$$\mathcal{R}(f)=\sum_{k=1}^{K-1}\mathcal{R}^{(k)}(f)=\sum_{k=1}^{K-1}E\left[\frac{R}{P(A|X)}I\left(A^{(k)}\neq\text{sign}\left(f(X^{(k)})\right)\right)\right],$$

where $\mathcal{R}^{(k)}(f)=E\left[\frac{R}{P(A|X)}I\left(A^{(k)}\neq\text{sign}\left(f(X^{(k)})\right)\right)\right]$. Similarly, the $\phi-$risk could be decomposed as,

$$\mathcal{R}_\phi(f)=\sum_{k=1}^{K-1}\mathcal{R}_\phi^{(k)}(f)=\sum_{k=1}^{K-1}E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(k)}f(X^{(k)}),R\right)\right],$$

where $\mathcal{R}_\phi^{(k)}(f)=E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(k)}f(X^{(k)}),R\right)\right]$ and the $\phi(\cdot)$ function has the same definition as before. Next, we discuss the property of each $\mathcal{R}^{(k)}(f)$ piece following a similar idea found in Zhao et al. (2012) and then combine to draw the final conclusion.

Without loss of generality, we consider the case where the reward is a discrete variable and the derivation for the continuous case is analogous. To simplify notation, we let $\eta_r(x) = \Pr(A^{(k)} = 1|R = r, X^{(k)} = x)$ and $q_r(x) = |r|\Pr(R = r|X^{(k)} = x)$ for certain k. When the reward is discrete, the kth component of the Bayes risk is

$$\begin{aligned}\mathcal{R}^{(k)}(f) =& E\left[\sum_r |r| \Pr\left(R = r|X^{(k)}\right) E\left(\frac{I\left(A^{(k)} \neq \text{sign}\left(f(X^{(k)})\right)\right)}{P(A|X)}|R = r, X^{(k)}\right)\right] \\ =& E\left[\sum_r q_r(X^{(k)}) E\left(\frac{\eta_r(X^{(k)})}{P(A > k|X)} I\left(1 \neq \text{sign}(f(X^{(k)}))\right)\right.\right. \\ &+ \left.\left.\frac{1 - \eta_r(X^{(k)})}{P(A \leq k|X)} I(-1 \neq \text{sign}(f(X^{(k)})))\right)\right]. \qquad \text{(B.3)}\end{aligned}$$

To further simplify the expression, we define $h(x)$ and $\psi(x)$ such that given r and x, the following equations are satisfied:

$$\begin{aligned} h(x)\psi(x) &= \sum_r q_r(x) \frac{\eta_r(x)}{P(A > k|x)} \\ h(x)\left(1 - \psi(x)\right) &= \sum_r q_r(x) \frac{1 - \eta_r(x)}{P(A \leq k|x)}. \end{aligned}$$

It can be shown that $h(x) = \sum_r q_r(x)[\frac{\eta_r(x)}{P(A>k|x)} + \frac{1-\eta_r(x)}{P(A\leq k|x)}] > 0$ and $\psi(x) = [\sum_r q_r(x)[\frac{\eta_r(x)}{P(A>k|x)} + \frac{1-\eta_r(x)}{P(A\leq k|x)}]]^{-1}[\sum_r q_r(x)\frac{\eta_r(x)}{P(A>k|x)}]$. Therefore, (B.3) becomes

$$\mathcal{R}^{(k)}(f) = E[h(X^{(k)})[\psi(X^{(k)})I(\text{sign}(f(X^{(k)})) \neq 1) + (1 - \psi(X^{(k)}))I(\text{sign}(f(X^{(k)})) \neq 1)]]. \qquad \text{(B.4)}$$

We follow the same steps above and obtain that the kth component of the $\phi-$risk is

$$\mathcal{R}_\phi^{(k)}(f) = E\left\{h(X^{(k)})\left[\psi(X^{(k)})\phi\left(f(X^{(k)})\right) + \left(1 - \psi(X^{(k)})\right)\phi\left(-f(X^{(k)})\right)\right]\right\}.$$

We define the new function $C\left(\psi, \alpha\right) = \psi\phi(\alpha) + (1 - \psi)\phi(-\alpha)$ to rewrite the optimal $\phi-$risk as

$$\mathcal{R}_\phi(f^*) = \sum_{k=1}^{K-1} \mathcal{R}_\phi^{(k)}(f^*) = \inf_{\alpha\in\mathbb{R}} \sum_{k=1}^{K-1} E\left[h\left(X^{(k)}\right) C\left(\psi(X^{(k)}), \alpha\right)\right].$$

Then the excess $\phi-$risk is

$$\mathcal{R}_\phi(f) - \mathcal{R}_\phi(f_\phi^*) = \sum_{k=1}^{K-1} E\left[C\left(\psi(X^{(k)}), f(X^{(k)})\right) h(X^{(k)}) - \inf_{\alpha\in\mathbb{R}} C\left(\psi(X^{(k)}), \alpha\right) h(X^{(k)})\right].$$

According to the result of Bartlett et al. (2006) and the convexity of the loss $\phi(x)$, we have for an arbitrary element x in the duplicated sample space $\mathcal{X}^{(k)}$,

$$h(x)\left(2\psi - 1\right) = \inf_{\alpha:\alpha(2\psi-1)\leq 0} C\left(\psi, \alpha\right) h(x) - \inf_{\alpha\in\mathbb{R}} C\left(\psi, \alpha\right) h(x). \tag{B.5}$$

In this way, according to (B.4) and (B.5), we have for each $k = 1, \cdots, K-1$,

$$\begin{aligned}
\mathcal{R}^{(k)}(f) - \mathcal{R}^{(k)}(f^*) &\leq E\{I[\text{sign}(f(X^{(k)})) \neq \text{sign}(h(X^{(k)})(\psi(X^{(k)}) - \frac{1}{2}))] \\
&\quad \times |h(X^{(k)})(2\psi(X^{(k)}) - 1)|\} \\
&= E\{I[\text{sign}(f(X^{(k)})) \neq \text{sign}(h(X^{(k)})(\psi(X^{(k)}) - \frac{1}{2}))] \\
&\quad \times |\inf_{\alpha:\alpha(2\psi-1)\leq 0} C(\psi(X^{(k)}), \alpha) h(X^{(k)}) - \inf_{\alpha\in\mathbb{R}} C(\psi(X^{(k)}), \alpha) h(X^{(k)})|\}.
\end{aligned}$$

Because $C(\psi(X^{(k)}), f(X^{(k)}))h(X^{(k)}) \geq \inf_{\alpha:\alpha(2\psi-1)\leq 0} C(\psi(X^{(k)}), \alpha)h(X^{(k)})$ holds, when $\text{sign}(f(X^{(k)})) \neq \text{sign}(h(X^{(k)})(\psi(X^{(k)}) - \frac{1}{2}))$, the second term on the right side of the equal sign above is bounded by $C(\psi(X^{(k)}), f(X^{(k)}))h(X^{(k)}) - \inf_{\alpha\in\mathbb{R}} C(\psi(X^{(k)}), \alpha)h(X^{(k)})$. Therefore, when we sum the inequality through $k = 1, \cdots, K-1$ we will have

$$\begin{aligned}
\mathcal{R}(f) - \mathcal{R}(f^*) &= \sum_{k=1}^{K-1} \left\{\mathcal{R}^{(k)}(f) - \mathcal{R}^{(k)}(f^*)\right\} \\
&\leq \sum_{k=1}^{K-1} E\left[C\left(\psi(X^{(k)}), f(X^{(k)})\right) h(X^{(k)}) - \inf_{\alpha\in\mathbb{R}} C\left(\psi(X^{(k)}), \alpha\right) h(X^{(k)})\right] \\
&= \mathcal{R}_\phi(f) - \mathcal{R}_\phi(f_\phi^*).
\end{aligned}$$

Proof of Theorem 3.4.4 We consider the same decomposition idea used in the proof of Theorem 4.4 and express the $\phi-$risk as

$$\mathcal{R}_\phi(f) = \sum_{k=1}^{K-1} \mathcal{R}_\phi^{(k)}(f) = \sum_{k=1}^{K-1} E\left[\frac{|R|}{P(A|X)}\phi\left(A^{(k)}f(X^{(k)}), R\right)\right].$$

For the kth component of the $\phi-$risk, we define the loss part as $L_\phi^{(k)}(f) = \frac{|R|}{P(A|X)}\phi(A^{(k)}f, R)$. Then for any $f \in \mathcal{H}$ and any minimizer $\hat{f}_n$ of $\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi(f) + \lambda_n||f||^2\right)$, where $\mathbb{P}_n$ denotes the empirical mean, we have

$$\begin{aligned}\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n)\right) &\leq \mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n) + \lambda_n||\hat{f}_n||^2\right) \\ &\leq \mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(f) + \lambda_n||f||^2\right). \qquad \text{(B.6)}\end{aligned}$$

The second inequality holds because $\hat{f}_n$ minimizes $\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi(f) + \lambda_n||f||^2\right)$ given λ_n. By taking the limit superior on both sides of (B.6), we have that the following inequality holds for any $f \in \mathcal{H}$:

$$\limsup_{n\to\infty}\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n)\right) \leq \limsup_{n\to\infty}\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(f) + \lambda_n||f||^2\right) = \mathbb{P}\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(f)\right).$$

This yields the fact that

$$\limsup_{n\to\infty}\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n)\right) \leq \inf_{f\in\bar{\mathcal{H}}}\mathbb{P}\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(f)\right).$$

Furthermore, since $\lambda_n \to 0$ when $n \to \infty$, Theorem 4.5 will be proved if we can show $\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n)\right) - \mathbb{P}\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(f)\right) \to 0$ in probability.

To show the convergence condition above, we first prove that $||\hat{f}_n||^2$ can be bounded by some constant depending on n. By (B.6), if we let $f = 0$ then the inequality becomes

$$\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n)\right) + \lambda_n||\hat{f}_n||^2 \leq \mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(0)\right) = \mathbb{P}_n\left(\sum_{k=1}^{K-1} \frac{|R|}{P(A|X)}\phi\left(0, R\right)\right).$$

Based on the fact that $\mathbb{P}_n\left(\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n)\right) \geq 0$ and $\phi(0,R)$ is bounded by 2, if we denote $\pi_0 = \min\{P(a_i|x_i)\}$ for $i = 1, \cdots, n$ (i.e. the smallest prior probability among the K treatments), then

$$||\hat{f}_n||^2 \leq \frac{(K-1)\,\phi\,(0)}{\pi_0\lambda_n}\sum_{i=1}^{n}\frac{|r_i|}{n} \leq \frac{2\,(K-1)}{\pi_0\lambda_n}\sum_{i=1}^{n}\frac{|r_i|}{n}.$$

Due to the existence of $E|R|$, $\exists N \in \mathbb{N}^+$ so that for $\forall n > N$, there is an upper bound M such that $||\hat{f}_n||^2 < M$. Furthermore, since the class $\{\sqrt{\lambda_n}f : ||\lambda_n f|| \leq \sqrt{M}\}$ is included in a Donsker class and $\sum_{k=1}^{K-1} L_\phi^{(k)}(f)$ is Lipschitz continuous with respect to f, then $\{\sqrt{\lambda_n}\sum_{k=1}^{K-1} L_\phi^{(k)}(f) : ||\lambda_n f|| \leq \sqrt{M}\}$ is also a P-Donsker class. In this way, if we denote $\mathbb{P}$ as the population mean operator, then

$$\sqrt{n}\,(\mathbb{P}_n - \mathbb{P})\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n) = \sqrt{\lambda_n^{-1}}\sqrt{n}\,(\mathbb{P}_n - \mathbb{P})\left(\sqrt{\lambda_n}\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n)\right) = O_p(\sqrt{\lambda_n^{-1}}).$$

Eventually, moving the $\sqrt{n}$ from the left hand sides to the right hand side in the equation above and taking limits in probability on both sides, we have $\lim\limits_{n\to\infty}(\mathbb{P}_n - \mathbb{P})\sum_{k=1}^{K-1} L_\phi^{(k)}(\hat{f}_n) = \lim\limits_{n\to\infty} O_p\left(\sqrt{(n\lambda_n)^{-1}}\right) = 0$ in probability as $n\lambda_n \to \infty$.

Proof of Theorem 3.4.5 We apply the same technique used in Vert and Vert (2006), Steinwart and Scovel (2007b) and Zhao et al. (2012) to show the risk convergence property presented in Theorem 4.6. According to Theorem 4.4, Theorem 4.6 will be obtained immediately if we can show the same convergence results for $\mathcal{R}_\phi(\hat{f}_n) - \mathcal{R}_\phi(f_\phi^*)$. We decompose the upper bound of $\mathcal{R}_\phi(\hat{f}_n) - \mathcal{R}_\phi(f^*_\phi)$ using the decomposition idea discussed before, then

$$\begin{aligned}
\mathcal{R}_\phi(\hat{f}_n) - \mathcal{R}_\phi(f^*_\phi) &\leq \sum_{k=1}^{K-1} \left[\lambda_n ||\hat{f}_n||^2 + \mathcal{R}^{(k)}_\phi(\hat{f}_n) - \mathcal{R}^{(k)}_\phi(f^*_\phi)\right] \\
&= \sum_{k=1}^{K-1} \left[\lambda_n ||\hat{f}_n||^2 + \mathcal{R}^{(k)}_\phi(\hat{f}_n) - \inf_{f\in\mathcal{H}} \left(\lambda_n ||f||^2 + \mathcal{R}^{(k)}_\phi(f)\right)\right] \\
&\quad + \sum_{k=1}^{K-1} \left[\inf_{f\in\mathcal{H}} \left(\lambda_n ||f||^2 + \mathcal{R}^{(k)}_\phi(f)\right) - \mathcal{R}^{(k)}_\phi(f^*_\phi)\right] \\
&= \sum_{k=1}^{K-1} \left[\lambda_n ||\hat{f}_n||^2 + \mathcal{R}^{(k)}_\phi(\hat{f}_n) - \inf_{f\in\mathcal{H}} \left(\lambda_n ||f||^2 + \mathcal{R}^{(k)}_\phi(f)\right)\right] \\
&\quad + \sum_{k=1}^{K-1} \left[\inf_{f\in\mathcal{H}} \left(\lambda_n ||f||^2 + \mathcal{R}^{(k)}_\phi(f) - \mathcal{R}^{(k)}_\phi(f^*_\phi)\right)\right]. \qquad \text{(B.7)}
\end{aligned}$$

Now, we are to bound each of the $K-1$ pieces on the right hand side of (B.7) under the new loss function.

We first discuss how to bound the second term in (B.7). Because the distribution $\mathcal{P}_k$ has geometric noise exponent $0 < q_k < \infty$ with constant C_k for each $k = 1, \cdots, K-1$, then we can find $K-1$ pairs of q_k and C_k such that the following inequality holds for all k

$$E\left[\exp\left(-\frac{\Delta(X^{(k)})^2}{t}\right)\left|2\eta\left(X^{(k)}\right) - 1\right|\right] \leq C_k t^{q_k p/2}, t > 0.$$

By Theorem 2.7 in Steinwart and Scovel (2007b), we can show that there exists $K-1$ constants $c_{p,k}$ such that for arbitrary $\lambda_n > 0$, we have

$$\sum_{k=1}^{K-1} [\inf_{f\in\mathcal{H}} (\lambda_n ||f||^2 + \mathcal{R}^{(k)}_\phi(f) - \mathcal{R}^{(k)}_\phi(f^*_\phi))] \leq \sum_{k=1}^{K-1} c_{p,k}(\sigma_n^p \lambda_n + C_k(2p)^{q_k p/2}\sigma_n^{-q_k p}). \quad \text{(B.8)}$$

Noting that the kth item in the summation of (B.8) can be considered as $O(\lambda_n^{q_k/(q_k+1)})$ for $k = 1, \cdots, K-1$, we can further bound the summation as follows by defining $q = \arg\max_{q_k} \lambda_n^{q_k/(q_k+1)}$ and thus,

$$\sum_{k=1}^{K-1} \left[\inf_{f\in\mathcal{H}} \left(\lambda_n ||f||^2 + \mathcal{R}^{(k)}_\phi(f) - \mathcal{R}^{(k)}_\phi(f^*_\phi)\right)\right] \leq O(\lambda_n^{q/(q+1)}).$$

As to bounding the first term in (B.7), we choose to apply Theorem 5.6 of Steinwart and Scovel (2007b). To meet the assumptions, we first need to define the corresponding $\mathcal{F}$, Z, T, $\mathcal{G}$, $f_{T,\mathcal{F}}$ and $f_{P,\mathcal{F}}$ in Theorem 5.6 of Steinwart and Scovel (2007b) in our new framework.

We define Z as our sample space $\mathcal{X}$ in Section 2, T as the empirical measure P_n, $\mathcal{F}$ as $B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}})$, the subspace of $\mathcal{H}$ which is a ball of $\mathcal{H}$ of radius $\sqrt{\frac{M}{\lambda_n}}$ (where M is the upper bound of $||\lambda_n f||$ according to the proof of Theorem 4.5), $f_{P,\mathcal{F}}$ as the minimizer of the regularized $\phi-$risk under $\mathcal{F}$ and $f_{T,\mathcal{F}}$ as the empirical minimizer $\hat{f}_n$, i.e.,

$$f_{P,\mathcal{F}} = \underset{f \in B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}})}{\arg\min} \left(\sum_{k=1}^{K-1} \mathcal{R}_\phi^{(k)}(f) + \lambda_n ||f||^2 \right).$$

We define $\mathcal{G}$ as the function space considering the loss $L_\phi(f)+\lambda_n||f||^2$ where $\mathcal{R}_\phi(f) = EL_\phi(f)$. That is to say,

$$\mathcal{G}_{\phi,\lambda_n} = \left\{ \sum_{k=1}^{K-1} L_\phi^{(k)}(f) + \lambda_n||f||^2 - \sum_{k=1}^{K-1} L_\phi^{(k)}(f_{P,\mathcal{F}}) - \lambda_n||f_{P,\mathcal{F}}||^2 : f \in B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}}) \right\}.$$

Then the remaining work is to show the two conditions in Theorem 5.6 of Steinwart and Scovel (2007b): First, $\exists c \geq 0$, $0 < \alpha \leq 1$ and $B > 0$ such that $||g||_\infty \leq B$ and $E_P\left(g^2\right) \leq c\left(E_P g\right)^\alpha$ for $\forall g \in \mathcal{G}_{\phi,\lambda_n}$. Second, $\exists a \geq 1$ and $0 < b < 2$ such that $\sup_{P_n \in \mathcal{X}} \log \mathcal{N}\left(B^{-1}\mathcal{G}_{\phi,\lambda_n}, \epsilon, L_2\left(P_n\right)\right) \leq a\epsilon^{-b}$ for $\forall \epsilon > 0$.

For the first condition, because the new $\phi-$loss function in $L_\phi^{(k)}(f)$ is Lipschitz continuous for $k = 1, \cdots, K-1$ as discussed early, there exists constants C_k such that $\left|L_\phi^{(k)}(f) - L_\phi^{(k)}(f_{P,\mathcal{F}})\right| \leq C_k \left|f - f_{P,\mathcal{F}}\right|$, therefore

$$|L_\phi(f) - L_\phi(f_{P,\mathcal{F}})| \leq \sum_{k=1}^{K-1} \left|L_\phi^{(k)}(f) - L_\phi^{(k)}(f_{P,\mathcal{F}})\right| \leq C||f - f_{P,\mathcal{F}}||,$$

where $C = \sum_{k=1}^{K-1} C_k$. In this way,

$$\begin{aligned}
|g| &\leq \left|\sum_{k=1}^{K-1} L_\phi^{(k)}(f) - \sum_{k=1}^{K-1} L_\phi^{(k)}(f_{P,\mathcal{F}})\right| + \left|\lambda_n ||f||^2 - \lambda_n ||f_{P,\mathcal{F}}||^2\right| \\
&\leq C||f - f_{P,\mathcal{F}}|| + \lambda_n ||f||^2 - \lambda_n ||f_{P,\mathcal{F}}||^2 \qquad \text{(B.9)} \\
&\leq C||f - f_{P,\mathcal{F}}|| + \lambda_n ||f||^2.
\end{aligned}$$

Because we have $f \in B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}})$, then both f and $f_{P,\mathcal{F}}$ are bounded by $\sqrt{\frac{M}{\lambda_n}}$ so that

$$||g||_\infty \leq 2C\sqrt{\frac{M}{\lambda_n}} + M.$$

In other words, the constant B in the first condition can be taken as $2C\sqrt{\frac{M}{\lambda_n}} + M$. To reach the second part of the first condition, we take the second moment of (B.9) on both sides and obtain

$$\begin{aligned}
E\left(g^2\right) &\leq E\left(C||f - f_{P,\mathcal{F}}|| + \lambda_n ||f||^2 - \lambda_n ||f_{P,\mathcal{F}}||^2\right)^2 \\
&\leq E\left(C||f - f_{P,\mathcal{F}}|| + \lambda_n \left|||f + f_{P,\mathcal{F}}|| \cdot ||f - f_{P,\mathcal{F}}||\right|\right)^2 \\
&\leq \left(C + 2\lambda_n \sqrt{\frac{M}{\lambda_n}}\right)^2 ||f - f_{P,\mathcal{F}}||^2. \qquad \text{(B.10)}
\end{aligned}$$

To show $E_P\left(g^2\right) \leq c\left(E_P g\right)^\alpha$, we need to prove that the right hand side of (B.10) can be upper bounded by $c\left(E_P g\right)^\alpha$. Due to the convexity of $L_\phi(f)$, we have

$$\begin{aligned}
\frac{1}{2}\left[L_\phi(f) + L_\phi(f_{P,\mathcal{F}}) + \lambda_n ||f||^2 + \lambda_n ||f_{P,\mathcal{F}}||^2\right] &\geq L_\phi\left(\frac{f + f_{P,\mathcal{F}}}{2}\right) + \frac{1}{2}\left[\lambda_n ||f||^2 + \lambda_n ||f_{P,\mathcal{F}}||^2\right] \\
&= L_\phi\left(\frac{f + f_{P,\mathcal{F}}}{2}\right) + \lambda_n ||\frac{f + f_{P,\mathcal{F}}}{2}||^2 + \\
&\quad \lambda_n ||\frac{f - f_{P,\mathcal{F}}}{2}||^2.
\end{aligned}$$

Taking expectation on both sides and by the definition of $f_{P,\mathcal{F}}$ we have

$$\begin{aligned}\frac{1}{2}\Big[\mathcal{R}_\phi(f)+\mathcal{R}_\phi(f_{P,\mathcal{F}})+\lambda_n||f||^2+\lambda_n||f_{P,\mathcal{F}}||^2\Big] &\geq \mathcal{R}_\phi\left(\frac{f+f_{P,\mathcal{F}}}{2}\right)+\lambda_n||\frac{f+f_{P,\mathcal{F}}}{2}||^2+\\ &\quad \lambda_n||\frac{f-f_{P,\mathcal{F}}}{2}||^2\\ &\geq \mathcal{R}_\phi\left(f_{P,\mathcal{F}}\right)+\lambda_n||f_{P,\mathcal{F}}||^2+\lambda_n||\frac{f-f_{P,\mathcal{F}}}{2}||^2.\end{aligned}$$

Adjusting the inequality a bit and we obtain

$$\frac{1}{2}E_Pg=\frac{1}{2}\Big[\mathcal{R}_\phi(f)-\mathcal{R}_\phi(f_{P,\mathcal{F}})+\lambda_n||f||^2-\lambda_n||f_{P,\mathcal{F}}||^2\Big]\geq\lambda_n||\frac{f-f_{P,\mathcal{F}}}{2}||^2. \tag{B.11}$$

Then combining (B.10) and (B.11), we have

$$E\left(g^2\right)\leq 2\left(C+2\sqrt{\lambda_nM}\right)^2\lambda_n^{-1}E_Pg$$

Thus, $E_P\left(g^2\right)\leq c\left(E_Pg\right)^\alpha$ holds when $\alpha=1$ and $c=2\left(C+2\sqrt{\lambda_nM}\right)^2\lambda_n^{-1}$. The proof for the first condition is now completed. For the second condition, the entropy we are concerned about can be decomposed by the subadditivity property,

$$\begin{aligned}\log\mathcal{N}(B^{-1}\mathcal{G}_{\phi,\lambda_n},\epsilon,L_2(P_n)) &= \log\mathcal{N}(B^{-1}\{\sum_{k=1}^{K-1}L_\phi^{(k)}(f)+\lambda_n||f||^2\}:f\in B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}}),\epsilon,L_2(P_n))\\ &\leq \log\mathcal{N}(B^{-1}L_\phi(f):f\in B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}}),\epsilon,L_2(P_n))\\ &\quad +\log\mathcal{N}(B^{-1}\lambda_n||f||^2:f\in B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}}),\epsilon,L_2(P_n)).\end{aligned} \tag{B.12}$$

Since we have$|L_\phi(f_1)-L_\phi(f_2)|\leq C||f_1-f_2||$ for any f_1 and f_2, the corresponding $b_1=B^{-1}L_\phi\left(f_1\right)$ and $b_2=B^{-1}L_\phi\left(f_2\right)$ in $\left\{B^{-1}L_\phi\left(f\right):f\in B_{\mathcal{H}}\left(\sqrt{\frac{M}{\lambda_n}}\right)\right\}$ must also satisfy $||b_1-b_2||\leq B^{-1}C||f_1-f_2||$. In this way, the first term in (B.12) can be bounded as

$$\begin{aligned}\log\mathcal{N}(B^{-1}L_\phi(f):f\in B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}}),\epsilon,L_2(P_n)) &\leq \log\mathcal{N}(B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}}),\frac{B}{C}\epsilon,L_2(P_n))\\ &\leq \log\mathcal{N}(B_{\mathcal{H}}(1),\frac{B}{C}[\sqrt{\frac{M}{\lambda_n}}]^{-1}\epsilon,L_2(P_n)).\end{aligned}$$

If we apply Theorem 2.1 in Steinwart and Scovel (2007b), because $\frac{B}{C}\left[\sqrt{\frac{M}{\lambda_n}}\right]^{-1}$ is a constant, then for $\forall 0 < \nu \leq 2$ and $\forall \delta > 0$, there exists a constant c_1 such that for $\epsilon > 0$:

$$\log \mathcal{N}\left(B^{-1}L_\phi\left(f\right) : f \in B_{\mathcal{H}}\left(\sqrt{\frac{M}{\lambda_n}}\right), \epsilon, L_2\left(P_n\right)\right) \leq c_1 \sigma_n^{(1-\nu/2)(1+\delta)p}\epsilon^{-\nu}.$$

In this way, there exists a constant c_2 such that

$$\begin{aligned}\log \mathcal{N}(B^{-1}\mathcal{G}_{\phi,\lambda_n}, \epsilon, L_2(P_n)) &\leq c_1\sigma_n^{(1-\nu/2)(1+\delta)p}\epsilon^{-\nu} + \\ &\quad \log \mathcal{N}(B^{-1}\lambda_n||f||^2 : f \in B_{\mathcal{H}}(\sqrt{\frac{M}{\lambda_n}}), \epsilon, L_2(P_n)) \\ &\leq c_1\sigma_n^{(1-\nu/2)(1+\delta)p}\epsilon^{-\nu} + \log(\frac{M}{B\epsilon}) \\ &\leq c_2\sigma_n^{(1-\nu/2)(1+\delta)p}\epsilon^{-\nu}.\end{aligned}$$

The proof for the second condition is accomplished. Having established the two conditions above, we can apply Theorem 5.6 in Steinwart and Scovel (2007b) directly and reach the conclusion that there exists a $c_\nu > 0$ depending only on ν such that for $\forall n \geq 1$ and $\forall \tau \geq 1$,

$$\Pr{}^*\left(\mathcal{R}_\phi\left(\hat{f}_n\right) + \lambda_n||\hat{f}_n||^2 > \mathcal{R}_\phi\left(f_\phi^*\right) + \lambda_n||f_\phi^*||^2 + c_\nu\epsilon\left(n, a, B, c, \delta, x\right)\right) \leq e^{-\tau},$$

where

$$\begin{aligned}\epsilon\left(n, a, B, c, \delta, x\right) &= B^{2\nu/(4-2\alpha+\alpha\nu)}c^{(2-\nu)/(4-2\alpha+\alpha\nu)}\left(\frac{a}{n}\right)^{2/(4-2\alpha+\alpha\nu)} + B^{\nu/2}\delta^{(2-\nu)/4}\left(\frac{a}{n}\right)^{1/2} \\ &\quad + B\left(\frac{a}{n}\right)^{2/(2+\nu)} + \sqrt{\frac{\delta\tau}{n}} + \left(\frac{c\tau}{n}\right)^{1/(2-\alpha)} + \frac{B\tau}{n},\end{aligned}$$

and $\alpha = 1$, $c = c_2\sigma_n^{(1-\nu/2)(1+\delta)p}$, $\sigma_n = \lambda_n^{-1/(q+1)p}$. Once we obtain the convergence rate results of the surrogate risk, the same conclusion can be reached for the 0-1 loss immediately by applying our Theorem 3.4.3.

B.3 Numerical Study: Nonlinear Boundary Examples

For the nonlinear boundary examples, we consider the following four scenarios with $\mu(X)$ and $t(X, A)$ defined as,

1. $K = 2$: $\mu(X) = 1 + X_1^2 + X_2^2 - 2X_3 + 0.5X_4$ and $t(X, A) = 4(0.7 - X_1^2 - X_2^2)(2A - 3)$;

2. $K = 3$: $\mu(X) = 2 + 2X_1 + X_2 + 0.5X_3$ and $t(X, A) = 4\sum_{i=1}^{3} I\left(g(X) \in (b_{i-1}, b_i]\right)(2 - |A - i|)$, where $g(X) = -3 - X_1^2 + 2\exp\{X_2\} + (X_3 - 0.6X_4)^2 + X_5^3 + \exp\{X_6^2\}$, $b_0 = -\infty$, $b_1 = 0$, $b_2 = 1.3$ and $b_3 = \infty$;

3. $K = 5$: $\mu(X) = 2 + 2X_1 + X_2 + 0.5X_3$ and $t(X, A) = 4\sum_{i=1}^{5} I\left(g(X) \in (b_{i-1}, b_i]\right)(2 - |A - i|)$, where $g(X) = -3 - X_1^2 + 2\exp\{X_2\} + (X_3 - 0.6X_4)^2 + X_5^3 + \exp\{X_6^2\}$, $b_0 = -\infty$, $b_1 = -0.4$, $b_2 = 0.3$, $b_3 = 1.1$, $b_4 = 2.1$ and $b_5 = \infty$;

4. $K = 7$: $\mu(X) = 2 + 2X_1 + X_2 + 0.5X_3$ and $t(X, A) = 4\sum_{i=1}^{7} I\left(g(X) \in (b_{i-1}, b_i]\right)(2 - |A - i|)$, where $g(X) = -3 - X_1^2 + 2\exp\{X_2\} + (X_3 - 0.6X_4)^2 + X_5^3$, $b_0 = -\infty$, $b_1 = -0.7$, $b_2 = -0.2$, $b_3 = 0.4$, $b_4 = 1$, $b_5 = 1.8$, $b_6 = 2.8$ and $b_7 = \infty$.

Similar to the linear boundary cases, we have a symmetric reward-treatment curve in each scenario. We repeat the simulation 50 times with the tuning parameters ranging in the same domain.

From the results (see Table 2 in the original chapter), none of the method performs well when the sample size is small because the true boundary function has a complex structure. When n becomes large, GOWL with the Gaussian kernel outperforms PLS-l_1 in all cases due to PLS-l_1's wrong model specification. GOWL with the Gaussian kernel shows better performance than OWL with the same kernel in terms of both accuracy and value function error. For OWL, we find that the estimated optimal treatments are often the same as the actually assigned ones when σ_n takes large values. This situation becomes more severe when the treatment has seven categories. In addition, when $K = 7$, we find that obtaining a low value function MSE becomes challenging even for GOWL with the Gaussian kernel. This may be due to the difficulty of the ITR detection for the ordinal treatments under nonlinear learning. Finally, we would like to note that the monotonic property of the intercept vectors b holds in all simulated cases above.

So far, our focus has been on examples with parallel boundaries. We would like to point out that the proposed GOWL could also work well when the parallel assumption of the true boundaries does not hold. Under these circumstances, one should consider using nonlinear learning techniques hence the estimated boundaries would be flexible enough to approach the underlying true boundaries. To illustrate the idea with a 2-dimensional graph, we use a case with $n = 300$, $p = 2$ and $K = 3$ and follow the previous settings to simulate X and A. At this time, we have the Q-function generated by $Q(X, A, \mathcal{D}^*(X)) = 2 + X_1 + 0.5X_2 - 2|A - \mathcal{D}^*(X)|$ where $\mathcal{D}^*(\cdot)$, the optimal treatment rule, is defined as, $\mathcal{D}^*(X) = 1$ if $(X_1 + 1)^2 + (X_2 + 1)^2 < 1$; $\mathcal{D}^*(X) = 2$ if $X_1 + X_2 > 2/3$; $\mathcal{D}^*(X) = 3$ otherwise.

Different from what were discussed in the previous examples, the current boundary set consists of a straight line and a one-fourth of a circle. Using GOWL-Gaussian with the same tuning range as in Section 5.2, we plot the estimated boundaries (dashed curves) as well as the true boundaries (solid curves) in Figure B.1. The results show that the estimated ITR could still capture the underlying pattern of the optimal ITR well since the RKHS with the Gaussian kernel is very flexible. We repeat the simulations for 50 times and the average testing misclassification rate is 5.05%, which illustrates GOWL's competitive prediction ability under the cases of complex boundaries.

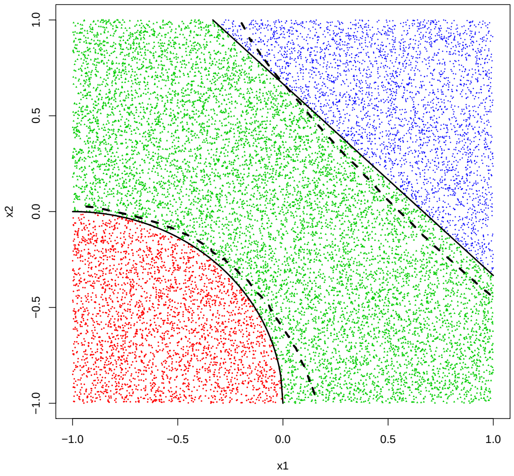

Figure B.1: Illustrating plot for the example with the true boundaries containing a linear line and a nonlinear curve. The solid curves indicate the true boundaries and the dashed curves represent the estimated boundaries by GOWL-Gaussian in one simulation. The points correspond to the observations in the test set with the color representing the optimal treatment: red-1, green-2 and blue-3.

APPENDIX C: IDENTIFYING HETEROGENEOUS EFFECT USING LATENT SUPERVISED CLUSTERING

Adjustment of Algorithm 1: Quadratic approximation

Algorithm 2 Line-search procedure to find an optimal step-size

The the iteration t of Algorithm 1, perform the following steps:

1. **Inputs:** $(\hat{\beta}^{(t)}, \hat{\gamma}^{(t)})$, and an estimate of L_f.
2. **Outputs:** $(\beta^{(t+1)}, \gamma^{(t+1)})$.
3. Set $L_t := \beta L_f$ for some $\beta \in (0, 1)$ (e.g., $\beta = 1/4$).
4. For $i_k = 1, 2, \cdots, i_{\max}$, perform:
   (a) If $L_t \geq L_f$, set $L_t = L_f$.
   (b) Compute a trial point $\tilde{\boldsymbol{\beta}}^{(t+1)}$ from
   $$\begin{cases} \boldsymbol{\tilde{\beta}^{(t+1)}} &= \mathrm{prox}_{(\lambda/L_t)\cdot g}\Big(\boldsymbol{\hat{\beta}^{(t)}} - \frac{1}{L_t}\nabla_\beta f(\boldsymbol{\hat{\beta}^{(t)}}, \hat{\gamma}^{(t)})\Big), \\ \boldsymbol{\tilde{\gamma}^{(t+1)}} &= \boldsymbol{\hat{\gamma}^{(t)}} - \frac{1}{L_t}\nabla_\gamma f(\boldsymbol{\hat{\beta}^{(t)}}, \hat{\gamma}^{(t)}). \end{cases}$$
   (c) Evaluate
   $$\begin{aligned} Q_{L_t}(\tilde{\boldsymbol{\beta}}^{(t+1)}, \tilde{\gamma}^{(t+1)}) &= f(\hat{\boldsymbol{\beta}}^{(t)}, \hat{\gamma}^{(t)}) + \nabla f(\hat{\boldsymbol{\beta}}^{(t)}, \hat{\gamma}^{(t)})^T\Big((\tilde{\boldsymbol{\beta}}^{(t+1)}, \tilde{\gamma}^{(t+1)}) \\ &\quad -(\hat{\boldsymbol{\beta}}^{(t)}, \hat{\gamma}^{(t)})\Big) + \tfrac{L_t}{2}\|(\tilde{\boldsymbol{\beta}}^{(t+1)}, \tilde{\gamma}^{(t+1)}) - (\hat{\boldsymbol{\beta}}^{(t)}, \hat{\gamma}^{(t)})\|_2^2. \end{aligned}$$
   (d) If $f(\tilde{\boldsymbol{\beta}}^{(t+1)}, \tilde{\gamma}^{(t+1)}) \leq Q_{L_t}(\tilde{\boldsymbol{\beta}}^{(t+1)}, \tilde{\gamma}^{(t+1)})$ or $L_t \geq L_f$, then terminate the line-search. Otherwise, set $L_t := 2L_t$ and repeat from (a).
5. End line-search and return $\boldsymbol{\beta}^{(t+1)} := \tilde{\boldsymbol{\beta}}^{(t+1)}$ and $\gamma^{(t+1)} := \tilde{\gamma}^{(t+1)}$.

Proof of Theorem 4.3.1 The proof of this theorem is slightly adapted the result in Schmidt et al. (2011). Using (Schmidt et al., 2011, Proposition 2) with $e_t = 0$, we have

$$F_n(\boldsymbol{\zeta}^{(t+1)}) - F^* \leq \frac{2L_f}{(t+1)^2}\left(\|\boldsymbol{\zeta}^0 - \boldsymbol{\zeta}^*\| + R_t\right)^2,$$

where $R_t = \frac{\sqrt{2}}{\sqrt{L_f}}\left(2\sum_{j=0}^{t-1} j\sqrt{\epsilon_j} + \sqrt{\sum_{j=0}^{t-1} j^2\epsilon_j}\right)$. This is indeed the bound (4.8). Now, we consider $\epsilon_t = \frac{c}{(t+1)^5}$. Using the well-known zeta function, we have $\sum_{j=0}^{t-1}(j+1)\sqrt{\epsilon_j} = \sum_{j=0}^{t-1}\frac{\sqrt{c}}{(j+1)^{1.5}} \leq$

$\sum_{j=0}^{\infty} \frac{\sqrt{c}}{(j+1)^{1.5}} \leq 2.62\sqrt{c}$. Similarly, $\sum_{j=0}^{t-1}(j+1)^2\epsilon_j = \sum_{j=0}^{t-1} \frac{c}{(j+1)^3} \leq \sum_{j=0}^{\infty} \frac{c}{(j+1)^3} \leq 1.203c$. Hence, we can upper bound R_t as $R_t \leq \frac{10c}{\sqrt{L_f}}$ as long as $c \geq 1$. By upper bounding (4.8) as $F_n(\boldsymbol{\zeta}^{(t+1)}) - F^* \leq \frac{2L_f}{(t+1)^2}\left(\|\boldsymbol{\zeta}^0 - \boldsymbol{\zeta}^*\| + \frac{10c}{\sqrt{L_f}}\right)^2 \leq \varepsilon$, we get $t+1 \geq \frac{\sqrt{2L_f}}{\sqrt{\varepsilon}}\|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\| + \frac{10\sqrt{2}c}{\sqrt{\varepsilon}}$, which gives a bound for $t_{\max}$ in the theorem by rounding the right-hand side. □

Proof of Lemma 4.3.1 It is obvious to write $\rho_\tau(r) = \tau r I(r \geq 0) - (1-\tau) r I(r < 0)$ as $\rho_\tau(r) = (\tau - 0.5)r + 0.5|r|$. Since the absolute function $s(r) = |r|$ can be written as $s(r) = \max\{r(u_1 - u_2) \mid u_1 + u_2 = 1, u_1 \geq 0, u_2 \geq 0\}$. We consider the function $p(u) = u_1\ln(u_1) + u_2\ln(u_2) + \ln(2)$ defined on a two dimensional standard simplex $\Delta_2 = \{u \in \mathbb{R}^2 \mid u_1 + u_2 = 1, u_1 \geq 0, u_2 \geq 0\}$. Now, we consider $s(r;\eta) = \max_u\{r(u_1 - u_2) - \eta p(u) \mid u \in \Delta\}$. Since $p(u) \geq 0$, it is clear that

$$s(r;\eta) \leq s(r) \leq s(r;\eta) + \eta\max\{p(u) \mid u \in \Delta_2\} = s(r;\eta) + \eta\ln(2).$$

Using this bound, we can show that the function $\rho_\tau(r) = (\tau - 0.5)r + 0.5s(r)$ can be approximated by $\rho_\tau(\tau;\eta) = (\tau - 0.5)r + 0.5s(r;\eta)$ as $\rho_\tau(\tau;\eta) \leq \rho_\tau(\tau) \leq \rho_\tau(\tau;\eta) + \eta\ln(2)$. Since $f_n(\boldsymbol{\zeta};\eta) \overset{\triangle}{=} \sum_{i=1}^n \rho_\tau(y_i(\mathbf{x}_i^T\boldsymbol{\beta}_i + \mathbf{z}_i^T\boldsymbol{\gamma});\eta)$, summing up the above inequality from $i = 1$ to n, we obtain the bound (4.9).

Next, we note that the function $s(\cdot;\eta)$ is the marginal of a strictly convex function for any $\eta > 0$ on Δ. Hence, by the classical Danskin theorem Boyd and Vandenberghe (2004), $\rho_\tau(\cdot;\eta)$ is convex and is differentiable. By solving this maximization problem with two variables u_1, u_2 directly, we obtain $s(r;\eta) = \eta\ln(e^{r/\eta} + e^{-r/\eta})$. By a few elementary calculations, we can show that $|\frac{d^2\rho_\tau(r;\eta)}{dr^2}| \leq \frac{1}{\eta}$. Using the definition $f_n(\boldsymbol{\zeta};\eta) \overset{\triangle}{=} \sum_{i=1}^n \rho_\tau(y_i(\mathbf{x}_i^T\boldsymbol{\beta}_i + \mathbf{z}_i^T\boldsymbol{\gamma});\eta)$ of f_n, we can show that $\|\nabla^2 f_n(\boldsymbol{\zeta};\eta)\| \leq \frac{\|\tilde{X}\|^2}{\eta} = \frac{\lambda_{\max}(\tilde{X}^T\tilde{X})}{\eta}$. Hence, f_n has the Lipschitz gradient with the Lipschitz constant $L_{f_n} = \frac{\lambda_{\max}(\tilde{X}^T\tilde{X})}{\eta}$. □

Proof of Theorem 4.3.2 Since $\nabla f_n(\cdot;\eta)$ is Lipschitz continuous with the Lipschitz constant $L_{f_n} = \frac{L_f}{\eta}$ and $L_f := \lambda_{\max}(\tilde{X}^T\tilde{X})$, similar to the proof of (Schmidt et al., 2011, Proposition 2), we obtain

$$
\begin{aligned}
F_n(\zeta^{(t+1)};\eta) \leq & F_n(\zeta;\eta) + \epsilon_t + L_{f_n}(\boldsymbol{\zeta}^{(t+1)} - \hat{\boldsymbol{\zeta}}^{(t)})^T(\boldsymbol{\zeta} - \hat{\boldsymbol{\zeta}}^{(t)}) + \frac{L_{f_n}}{2}\|\boldsymbol{\zeta}^{(t+1)} - \hat{\boldsymbol{\zeta}}^{(t)}\|^2 \\
& + L_{f_n} e_t^T(\boldsymbol{\zeta} - \hat{\boldsymbol{\zeta}}^{(t)}),
\end{aligned}
$$

where e_t is a vector satisfying $\|e_t\|^2 \leq \frac{2\epsilon_t}{L_{f_n}}$. Note that the proximity function $p(u) = u_1 \ln(u_1) + u_2 \ln(u_2) + \ln(2)$ is strongly convex with the parameter $\mu_p = 1$ in the ℓ_1-norm. Since $\|\cdot\|_1 \geq \|\cdot\|_2$, $p(\cdot)$ is also strongly convex in the ℓ_2-norm with the same parameter $\mu_p = 1$. Using this property, we can apply (Tran-Dinh, 2016, Lemma 2) to obtain the following estimate

$$
\begin{aligned}
F_n(\zeta^{(t+1)};\eta_{t+1}) \quad & \leq (1-\xi_t)F_n(\zeta^{(t)};\eta_t) + F_n(\boldsymbol{\zeta}^*) + \epsilon_t + \\
& \tfrac{\xi_t^2 L_f}{2\eta_{t+1}}[\|\tilde{\zeta}^{(t+1)} - \boldsymbol{\zeta}^*\|^2 - \|\tilde{\zeta}^{(t)} - \boldsymbol{\zeta}^*\|^2] + \tfrac{\xi_t\sqrt{2L_f\epsilon_t}}{\sqrt{\eta_{t+1}}}\|\tilde{\boldsymbol{\zeta}}^{(t)} - \boldsymbol{\zeta}^*\| + \Omega_t,
\end{aligned} \tag{C.1}
$$

where $\xi_t = \frac{1}{\tau_t}$ and $\tilde{\boldsymbol{\zeta}}^{(t)} = \frac{1}{\xi_t}(\hat{\boldsymbol{\zeta}}^{(t)} - (1-\xi_t)\boldsymbol{\zeta}^{(t)})$. Here, the quantity $\Omega_t = (1-\xi_t)(\eta_t - \eta_{t+1})D_\rho$ for the Logit-type loss and $\Omega_t = 0$ for the Huber loss. By the proof of (Schmidt et al., 2011, Proposition 2), we can bound $\|\tilde{\boldsymbol{\zeta}}^{(t)} - \boldsymbol{\zeta}^*\|$ as

$$
\|\tilde{\boldsymbol{\zeta}}^{(t)} - \boldsymbol{\zeta}^*\| \leq \|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\| + 2\sum_{j=0}^{t-1}\frac{\sqrt{2\epsilon_j}}{\xi_j\sqrt{L_{f_n}}} + \left(2\sum_{j=0}^{t-1}\frac{\epsilon_j}{L_{f_n}\xi_j^2}\right)^{1/2}.
$$

Using $L_{f_n} = \frac{L_f}{\eta_{t+1}}$, we define $\kappa_t := \frac{1}{\sqrt{L_f}}\sum_{j=0}^{t-1}\frac{\sqrt{2\eta_{j+1}\epsilon_j}}{\xi_j}$ and $\hat{\kappa}_t := \frac{1}{L_f}\sum_{j=0}^{t-1}\frac{\eta_{j+1}\epsilon_j}{\xi_j^2}$. Then, we have $\|\tilde{\boldsymbol{\zeta}}^{(t)} - \boldsymbol{\zeta}^*\| \leq \|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\| + 2\kappa_t + \sqrt{2\hat{\kappa}_t}$. Substituting this into (C.1) and rearranging the result, we get

$$
\begin{aligned}
\frac{\eta_{t+1}}{L_f\xi_t^2}\Big(F_n(\boldsymbol{\zeta}^{(t+1)};\eta_{t+1}) - F_n(\boldsymbol{\zeta}^*)\Big) + \frac{1}{2}\|\tilde{\zeta}^{(t+1)} - \boldsymbol{\zeta}^*\|^2 \leq & \frac{\eta_{t+1}(1-\xi_t)}{L_f\xi_t^2}\Big(F_n(\zeta^{(t)};\eta_t) - F_n(\boldsymbol{\zeta}^*)\Big) + \\
& \frac{1}{2}\|\tilde{\zeta}^{(t)} - \boldsymbol{\zeta}^*\|^2 + \frac{\eta_{t+1}\epsilon_t}{L_f\xi_t^2} + \\
& \frac{\sqrt{2\eta_{t+1}\epsilon_t}}{\xi_t\sqrt{L_f}}\Big(\|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\| + 2\kappa_t + \sqrt{2\hat{\kappa}_t}\Big) + \frac{\eta_{t+1}\Omega_t}{L_f\xi_t^2}.
\end{aligned}
$$

Now, using the definitions of κ_t and $\hat{\kappa}_t$, and noting that both sequences $\{\kappa_t\}$ and $\{\hat{\kappa}_t\}$ are increasing, we can show that $\sum_{j=0}^{t-1}\frac{\sqrt{2\eta_{j+1}\epsilon_j}}{\xi_j\sqrt{L_f}}(2\kappa_j + \sqrt{2\hat{\kappa}_j}) \leq (2\kappa_t + \sqrt{2\hat{\kappa}_t})\sum_{j=0}^{t-1}\frac{\sqrt{2\eta_{j+1}\epsilon_j}}{\xi_j\sqrt{L_f}} = 2\kappa_t^2 + \kappa_t\sqrt{2\hat{\kappa}_t}$. Summing up the above inequality from $j = 0$ to $j = t-1$, and using this estimate we obtain

$$\begin{aligned}\frac{\eta_t}{L_f\xi_{t-1}^2}\left(F_n(\boldsymbol{\zeta}^{(t)};\eta_t) - F_n(\boldsymbol{\zeta}^*)\right) + \frac{1}{2}\|\tilde{\zeta}^{(t)} - \boldsymbol{\zeta}^*\|^2 \leq& \frac{1}{2}\|\tilde{\zeta}^{(0)} - \boldsymbol{\zeta}^*\|^2 + \hat{\kappa}_t + \|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\|\kappa_t + \\ & 2\kappa_t^2 + \kappa_t\sqrt{2\hat{\kappa}_t} + \hat{\Omega}_t.\end{aligned}$$

Here, $\hat{\Omega}_t := \frac{\eta_1 D_\rho}{L_f} + \sum_{j=1}^{t-1}\frac{\eta_{j+1}\Omega_j}{L_f\xi_j^2} = \frac{\eta_1 D_\rho}{L_f} + \frac{D_\rho}{L_f}\sum_{j=1}^{t-1}\frac{\eta_{j+1}(1-\xi_j)(\eta_j-\eta_{j+1})}{\xi_j^2}$. Dropping the nonnegative term $\frac{1}{2}\|\tilde{\zeta}^{(t)} - \boldsymbol{\zeta}^*\|^2$, using $\tilde{\boldsymbol{\zeta}}^{(0)} = \boldsymbol{\zeta}^{(0)}$, and defining $R_t := \hat{\kappa}_t + \|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\|\kappa_t + 2\kappa_t^2 + \kappa_t\sqrt{2\hat{\kappa}_t}$, the last inequality leads to

$$F_n(\boldsymbol{\zeta}^{(t)};\eta_t) - F_n(\boldsymbol{\zeta}^*) \leq \frac{L_f}{\tau_{t-1}^2\eta_t}\left(\frac{1}{2}\|\zeta^{(0)} - \boldsymbol{\zeta}^*\|^2 + R_t\right). \tag{C.2}$$

Now, we consider the condition $\frac{\eta_{t+1}(1-\xi_t)}{\xi_t^2} = \frac{\eta_t}{\xi_{t-1}^2}$. Using this condition, $\xi_t = \frac{1}{\tau_t}$, and $\eta_{t+1} = \frac{\tau_t\eta_t}{\tau_t+1}$ we have $\frac{(\tau_t-1)\tau_t^2}{\tau_t+1} = \tau_{t-1}^2$, which leads to $\tau_t^3 - \tau_t^2 - \tau_{t-1}^2\tau_t - \tau_{t-1}^2 = 0$. Hence, τ_{t+1} is the solution of the cubic equation $\tau^3 - \tau^2 - \tau_t^2\tau - \tau_t^2 = 0$. Moreover, one can show that $\frac{t+2}{2} \leq \tau_t \leq t+1$, and $\eta_t \leq \frac{2\eta_1}{t+1}$. In this case, we have $\frac{1}{\tau_{t-1}^2\eta_t} \leq \frac{1}{t\eta_1}$ for $t \geq 1$. Next, let us choose $\epsilon_t = \frac{c}{(t+1)^4}$ for some positive constant $c \geq 1$, we bound κ_t and $\hat{\kappa}_t$ as follows:

$$\begin{aligned}\kappa_t &:= \tfrac{1}{\sqrt{L_f}}\textstyle\sum_{j=0}^{t-1}\tfrac{\sqrt{2\eta_{j+1}\epsilon_j}}{\xi_j} \leq \tfrac{\sqrt{2}}{\sqrt{L_f}}\textstyle\sum_{j=0}^{t-1}\sqrt{(j+1)\epsilon_j} = \tfrac{\sqrt{2c}}{\sqrt{L_f}}\textstyle\sum_{j=0}^{t-1}\tfrac{1}{(\sqrt{j+1})^3} \leq \tfrac{2.62\sqrt{2c}}{\sqrt{L_f}}\\ \hat{\kappa}_t &:= \tfrac{1}{L_f}\textstyle\sum_{j=0}^{t-1}\tfrac{\eta_{j+1}\epsilon_j}{\xi_j^2} \leq \tfrac{1}{L_f}\textstyle\sum_{j=0}^{t-1}(j+1)\epsilon_j \leq \tfrac{c}{L_f}\textstyle\sum_{j=0}^{t-1}\tfrac{1}{(j+1)^3} \leq \tfrac{1.203c}{L_f}.\end{aligned}$$

Hence, we can bound R_t as $R_t \leq \bar{R}_c := \frac{1.9\sqrt{L_f c}}{\eta_1}\|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\| + \frac{35c}{2\eta_1}$. One the other hand, we estimate $\hat{\Omega}_t$ as follows

$$\begin{aligned}\hat{\Omega}_t &= \frac{\eta_1 D_\rho}{L_f} + \frac{D_\rho}{L_f}\sum_{j=1}^{t-1}\frac{\eta_{j+1}(1-\xi_j)(\eta_j - \eta_{j+1})}{\xi_j^2} = \frac{\eta_1 D_\rho}{L_f} + \frac{D_\rho}{L_f}\sum_{j=1}^{t-1}\eta_j(\eta_j - \eta_{j+1})\tau_{j-1}^2\\ &= \frac{\eta_1 D_\rho}{L_f} + \frac{D_\rho}{L_f}\sum_{j=1}^{t-1}\frac{\eta_j^2\tau_{j-1}^2}{\tau_j + 1} \leq \frac{\eta_1 D_\rho}{L_f} + \frac{D_\rho}{L_f}\sum_{j=1}^{t-1}\frac{4\eta_1^2 j^2}{(j+1)^2(j+2)} \leq \frac{\eta_1 D_\rho(1+4\eta_1\ln(t+1))}{L_f}.\end{aligned}$$

Using the bound (4.9), we can show that $F_n(\boldsymbol{\zeta}^{(t)}) \leq F_n(\boldsymbol{\zeta}^{(t)};\eta_t) + \eta_t n D_\rho \leq F_n(\boldsymbol{\zeta}^{(t)};\eta_t) + \frac{2\eta_1 n D_\rho}{t+1}$. Combining these two estimates, (C.2), and then using $\frac{1}{\tau_{t-1}^2\eta_t} \leq \frac{1}{t\eta_1}$ for $t \geq 1$, we obtain

$$F_n(\boldsymbol{\zeta}^{(t+1)}) - F_n(\boldsymbol{\zeta}^*) \leq \frac{1}{(t+1)}\left(\frac{L_f}{2\eta_1}\|\boldsymbol{\zeta}^{(0)} - \boldsymbol{\zeta}^*\|^2 + \bar{R}_c + \Gamma_t\right) + \frac{2n\eta_1 D_\rho}{(t+2)},$$

which leads to the estimation (4.11), where $\Gamma_t = 0$ if we choose the Huber loss, and $\Gamma_t := \frac{D_\rho(1+4\eta_1\ln(t+1))}{2}$ if we choose the Logit-type loss. The remaining part of the theorem is a direct consequence of (4.11). □

REFERENCES

Agresti, A. (2014). *Wiley Series in Probability and Statistics : Categorical Data Analysis (3rd Edition).* Wiley, Somerset, NJ, USA.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19,** 716–723.

Allen, G. I. (2012). Automatic Feature Selection via Weighted Kernels and Regularization. *Journal of Computational and Graphical Statistics* **22,** 284–299.

Altstein, L. and Li, G. (2013). Latent subgroup analysis of a randomized clinical trial through a semiparametric accelerated failure time mixture model. *Biometrics* **69,** 52–61.

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

Aronszajn, N. (1950). Theory of Reproducing Kernels. *Transactions of the American Mathematical Society* **68,** 337–404.

Bache, K. and Lichman, M. (2015). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences.

Bartlett, P. L., Bousquet, O., and Mendelson, S. (2005). Local Rademacher Complexities. *Annals of Statistics* **33,** 1497–1537.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, Classification, and Risk Bounds. *Journal of the American Statistical Association* **101,** 138–156.

Beck, A. and Teboulle, M. (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences* **2,** 183–202.

Ben-David, S., Eiron, N., and Long, P. M. (2003). On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences* **66,** 496–514.

Bertsekas, D., Nedic, A., and Ozdaglar, A. E. (2003). *Convex Analysis and Optimization.* Athena Scientific.

Biesheuvel, E. and Hothorn, L. A. (2002). Many-to-one Comparisons in Stratified Designs. *Biometrical Journal* **44,** 101–116.

Blanchard, G., Bousquet, O., and Massart, P. (2008a). Statistical Performance of Support Vector Machines. *Annals of Statistics* **36,** 489–531.

Blanchard, G., Bousquet, O., and Massart, P. (2008b). Statistical Performance of Support Vector Machines. *The Annals of Statistics* **36,** 489–531.

Borg, I. and Groenen, P. J. F. (2005). *Modern Multidimensional Scaling: Theory and Applications.* Springer Science & Business Media.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge.

Breiman, L. and Friedman, J. H. (1985). Estimating Optimal Transformations for Multiple Regression and Correlation. *Journal of the American Statistical Association* **80,** 580–598.

Brookes, S. T., Whitely, E., Egger, M., Smith, G. D., Mulheran, P. A., and Peters, T. J. (2004). Subgroup analyses in randomized trials: risks of subgroup-specific analyses; power and sample size for the interaction test. *Journal of Clinical Epidemiology* **57,** 229–236.

Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* **2,** 121–167.

Cai, T., Tian, L., Wong, P. H., and Wei, L. J. (2011). Analysis of randomized comparative clinical trial data for personalized treatment selections. *Biostatistics* **12,** 270–282.

Candes, E. and Tao, T. (2005). Decoding by Linear Programming. *IEEE Transactions on Information Theory* **51,** 4203–4215.

Cardoso, J. S. and Pinto da Costa, J. F. (2007). Learning to Classify Ordinal Data: The Data Replication Method. *J. Mach. Learn. Res.* **8,** 1393–1429.

Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-supervised Learning.* MIT press Cambridge.

Chen, G., Zeng, D., and Kosorok, M. R. (2016). Personalized dose finding using outcome weighted learning. *Journal of the American Statistical Association* **Accepted,**.

Chen, Y.-C., Genovese, C. R., Tibshirani, R. J., and Wasserman, L. (2016). Nonparametric modal regression. *The Annals of Statistics* **44,** 489–514. arXiv: 1412.1716.

Chi, E. C., Allen, G. I., and Baraniuk, R. G. (2016). Convex biclustering. *Biometrics* pages n/a–n/a.

Chi, E. C. and Lange, K. (2015). Splitting Methods for Convex Clustering. *Journal of Computational and Graphical Statistics* **24,** 994–1013.

Cleveland, W. S. and Devlin, S. J. (1988). Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. *Journal of the American Statistical Association* **83,** 596–610.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning* **20,** 273–297.

De Boor, C. (2001). *A Practical Guide to Splines.* Springer New York.

Eilers, P. H. C. and Marx, B. D. (1996). Flexible Smoothing with B-splines and Penalties. *Statistical Science* **11,** 89–102.

Ein-Dor, P. and Feldmesser, J. (1987). Attributes of the Performance of Central Processing Units: A Relative Performance Prediction Model. *Communications of the ACM* **30,** 308–317.

Ellsworth, R. E., Decewicz, D. J., Shriver, C. D., and Ellsworth, D. L. (2010). Breast Cancer in the Personal Genomics Era. *Current Genomics* **11,** 146–161.

Fan, C., Lu, W., Song, R., and Zhou, Y. (2016). Concordance-assisted learning for estimating optimal individualized treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **Accepted,**.

Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association* **96,** 1348–1360.

Fan, J. and Lv, J. (2008). Sure Independence Screening for Ultrahigh Dimensional Feature Space. *Journal of the Royal Statistical Society: Series B* **70,** 849–911.

Fan, J. and Lv, J. (2010). A Selective Overview of Variable Selection in High Dimensional Feature Space. *Statistica Sinica* **20,** 101.

Fan, J. and Peng, H. (2004). Nonconcave Penalized Likelihood with a Diverging Number of Parameters. *Annals of Statistics* **32,** 928–961.

Feldman, V., Guruswami, V., Raghavendra, P., and Wu, Y. (2010). Agnostic Learning of Monomials by Halfspaces is Hard. *arXiv:1012.0729 [cs]* arXiv: 1012.0729.

Fercoq, O. and Qu, Z. (2016). Restarting accelerated gradient methods with a rough strong convexity estimate. *arXiv preprint arXiv:1609.07358* .

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics* **19,** 1–67.

Friedman, L. M., Furberg, C. D., and DeMets, D. (2010). *Fundamentals of Clinical Trials*. Springer Science & Business Media.

Fu, H., Zhou, J., and Faries, D. E. (2016). Estimating optimal treatment regimes via subgroup identification in randomized control trials and observational studies. *Statistics in Medicine* .

Goodman, V., Kuelbs, J., and Zinn, J. (1981). Some Results on the LIL in Banach Space with Applications to Weighted Empirical Processes. *The Annals of Probability* **9,** 713–752.

Gorski, J., Pfeuffer, F., and Klamroth, K. (2007). Biconvex Sets and Optimization with Biconvex Functions: A Survey and Extensions. *Mathematical Methods of Operations Research* **66,** 373–407.

Greenland, S. (2009). Interactions in epidemiology: relevance, identification, and estimation. *Epidemiology (Cambridge, Mass.)* **20,** 14–17.

Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2010). Pairwise variable selection for high-dimensional model-based clustering. *Biometrics* **66,** 793–804.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. N. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning* **46,** 389–422.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.* Springer Science & Business Media. Google-Books-ID: tVIjmNS3Ob8C.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2011). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.

Hennig, C. and Liao, T. F. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **62,** 309–369.

Herrett, E., Gallagher, A. M., Bhaskaran, K., Forbes, H., Mathur, R., van Staa, T., and Smeeth, L. (2015). Data Resource Profile: Clinical Practice Research Datalink (CPRD). *International Journal of Epidemiology* **44,** 827–836.

Hocking, T., Vert, J.-p., Joulin, A., and Bach, F. R. (2011). Clusterpath: an Algorithm for Clustering using Convex Fusion Penalties. pages 745–752.

Holman, R. R., Thorne, K. I., Farmer, A. J., Davies, M. J., Keenan, J. F., Paul, S., Levy, J. C., and 4-T Study Group (2007). Addition of biphasic, prandial, or basal insulin to oral therapy in type 2 diabetes. *The New England Journal of Medicine* **357,** 1716–1730.

Huber, P. J. (2004). *Robust Statistics.* John Wiley & Sons.

Kimeldorf, G. and Wahba, G. (1971). Some Results on Tchebycheffian Spline Functions. *Journal of Mathematical Analysis and Applications* **33,** 82–95.

Kimeldorf, G. S. and Wahba, G. (1970). A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines. *The Annals of Mathematical Statistics* **41,** 495–502.

Knight, K. and Fu, W. (2000). Asymptotics for Lasso-type Estimators. *The Annals of Statistics* **28,** 1356–1378.

Koenker, R. (2005). *Quantile Regression.* Cambridge University Press.

Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear Programming. The Regents of the University of California.

Laber, E. B. and Murphy, S. A. (2011). Adaptive Confidence Intervals for the Test Error in Classification. *Journal of the American Statistical Association* **106,** 904–913.

Laber, E. B. and Zhao, Y. Q. (2015). Tree-based methods for individualized treatment regimes. *Biometrika* **102,** 501–514.

Lagakos, S. W. (2006). The challenge of subgroup analyses–reporting without distorting. *The New England Journal of Medicine* **354,** 1667–1669.

Lauer, M. S., Francis, G. S., Okin, P. M., Pashkow, F. J., Snader, C. E., and Marwick, T. H. (1999). Impaired chronotropic response to exercise stress testing as a predictor of mortality. *JAMA* **281,** 524–529.

Lee, W., Du, Y., Sun, W., Hayes, D. N., and Liu, Y. (2012). Multiple Response Regression for Gaussian Mixture Models with Known Labels. *Statistical Analysis and Data Mining* **5,** 493–508.

Li, Y. and Zhu, J. (2008). L1-norm Quantile Regression. *Journal of Computational and Graphical Statistics* **17,** 163–185.

Lichman, M. (2013). UCI machine learning repository.

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R., and Klein, B. (2000). Smoothing Spline Anova Models for Large Data Sets with Bernoulli Observations and the Randomized GACV. *Annals of Statistics* **28,** 1570–1600.

Lin, Y. and Zhang, H. H. (2006). Component Selection and Smoothing in Multivariate Nonparametric Regression. *The Annals of Statistics* **34,** 2272–2297.

Ling, L. and Lin, H.-T. (2006). Ordinal regression by extended binary classification. *Advances in neural information processing systems* pages 865–872.

Lipkovich, I. and Dmitrienko, A. (2014). Strategies for Identifying Predictive Biomarkers and Subgroups with Enhanced Treatment Effect in Clinical Trials Using SIDES. *Journal of Biopharmaceutical Statistics* **24,** 130–153.

Liu, Y. (2007). Fisher Consistency of Multicategory Support Vector Machines. In *Eleventh International Conference on Artificial Intelligence and Statistics*, pages 289–296.

Ma, S. and Huang, J. (2016). A concave pairwise fusion approach to subgroup analysis. *Journal of the American Statistical Association* pages 1–42.

Mancinelli, L., Cronin, M., and Sadee, W. (2000). Pharmacogenomics: The promise of personalized medicine. *AAPS PharmSci* **2,** 29–41.

McDiarmid, C. (1989). On the Method of Bounded Differences. In *Surveys in Combinatorics*, pages 148–188. Cambridge University Press.

McLachlan, G. (2004). *Discriminant Analysis and Statistical Pattern Recognition.* John Wiley & Sons.

McLachlan, G. and Peel, D. (2004). *Finite Mixture Models.* John Wiley & Sons.

Minh, H. Q. (2010). Some Properties of Gaussian Reproducing Kernel Hilbert Spaces and Their Implications for Function Approximation and Learning Theory. *Constructive Approximation* **32,** 307–338.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning.* MIT press.

Nakai, K. and Kanehisa, M. (1991). Expert System for Predicting Protein Localization Sites in Gram-negative Bacteria. *Proteins* **11,** 95–110.

Nesterov, Y. (1998). Introductory Lectures on Convex Programming Volume I: Basic Course.

Nesterov, Y. (2013). Gradient methods for minimizing composite objective function. *Math. Program.* **140,** 125–161.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization.* Springer Science & Business Media.

others, A. D. A. a. (2014). Standards of Medical Care in Diabetes 2014. *Diabetes Care* **37,** S14–S80.

Peng, L., Xu, J., and Kutner, N. (2014). Shrinkage estimation of varying covariate effects based on quantile regression. *Statistics and Computing* **24,** 853–869.

Perou, C. M., SÃ¸rlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S. X., LÃ¸nning, P. E., BÃ¸rresen-Dale, A. L., Brown, P. O., and Botstein, D. (2000). Molecular portraits of human breast tumours. *Nature* **406,** 747–752.

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

Robins, J., Orellana, L., and Rotnitzky, A. (2008). Estimation and extrapolation of optimal treatment and testing strategies. *Statistics in Medicine* **27,** 4678–4721.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20,** 53–65.

Ruberg, S. J., Chen, L., and Wang, Y. (2010). The mean does not mean as much anymore: finding sub-groups for tailored therapeutics. *Clinical Trials (London, England)* **7,** 574–583.

Schmidt, M., Roux, N. L., and Bach, F. R. (2011). Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in neural information processing systems*, pages 1458–1466.

Scholkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning).* The MIT Press.

Shawe-Taylor, J. and Cristianini, N. (2004a). *Kernel Methods for Pattern Analysis.* Cambridge University Press.

Shawe-Taylor, J. and Cristianini, N. (2004b). *Kernel Methods for Pattern Analysis.* Cambridge University Press.

Shen, J. and He, X. (2015). Inference for Subgroup Analysis With a Structured Logistic-Normal Mixture Model. *Journal of the American Statistical Association* **110,** 303–312.

Simoncelli, T. (2014). Paving the Way for Personalized Medicine: FDA's Role in a New Era of Medical Product Development. Technical report, Federal Drug Administration (FDA).

Song, G. and Zhang, H. (2011). Reproducing Kernel Banach Spaces with the L1 Norm II: Error Analysis for Regularized Least Square Regression. *Neural computation* **23,** 2713–2729.

Steinwart, I. and Scovel, C. (2007a). Fast Rates for Support Vector Machines using Gaussian Kernels. *Annals of Statistics* **35,** 575–607.

Steinwart, I. and Scovel, C. (2007b). Fast rates for support vector machines using Gaussian kernels. *The Annals of Statistics* **35,** 575–607.

Street, W. N., Wolberg, W. H., and Mangasarian, O. L. (1993). Nuclear Feature Extraction for Breast Tumor Diagnosis. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 861–870.

Su, X., Meneses, K., McNees, P., and Johnson, W. O. (2011). Interaction trees: exploring the differential effects of an intervention programme for breast cancer survivors. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **60,** 457–474.

Su, X., Tsai, C.-L., Wang, H., Nickerson, D. M., and Li, B. (2009). Subgroup Analysis via Recursive Partitioning. *J. Mach. Learn. Res.* **10,** 141–158.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Tibshirani, R. (1994). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B* **58,** 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67,** 91–108.

Tibshirani, R. J. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society: Series B* **58,** 267–288.

Tran-Dinh, Q. (2016). Adaptive smoothing algorithms for nonsmooth composite convex minimization. *Computational Optimization and Applications* pages 1–27.

Tran-Dinh, Q., Fercoq, O., and Cevher, V. (2016). A smooth primal-dual optimization framework for nonsmooth composite convex minimization. *Tech. Report, STAT-OR-UNC* pages 1–29.

Tsevendorj, I. (2001). Piecewise-Convex Maximization Problems. *Journal of Global Optimization* **21,** 1–14.

vaart, A. W. v. d. and Wellner, J. (2000). *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer, New York.

Van der Vaart, A. W. and Wellner, J. A. (2000). *Weak Convergence and Empirical Processes with Application to Statistics*. Springer.

Vazirani, V. V. (2013). *Approximation Algorithms*. Springer Science & Business Media.

Vert, R. and Vert, J.-P. (2006). Consistency and Convergence Rates of One-Class SVMs and Related Algorithms. *J. Mach. Learn. Res.* **7,** 817–854.

Wager, S. and Athey, S. (2015). Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. *arXiv:1510.04342 [math, stat]* arXiv: 1510.04342.

Wahba, G. (1990). *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics.

Wang, L., Zhu, J., and Zou, H. (2007). Hybrid Huberized Support Vector Machines for Microarray Classification. In *Proceedings of the 24th International Conference on Machine Learning*, pages 983–990. ACM.

Wang, Z., Liu, H., and Zhang, T. (2014). Optimal Computational and Statistical Rates of Convergence for Sparse Nonconvex Learning Problems. *The Annals of Statistics* **42,** 2164–2201.

Wei, S. and Kosorok, M. R. (2013). Latent Supervised Learning. *Journal of the American Statistical Association* **108,** 957–970.

Wellner, J. A. (2005). Empirical Processes: Theory and Applications. *Notes for a Course Given at Delft University of Technology* .

Wu, C. F. J. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics* **11,** 95–103.

Wu, S., Shen, X., and Geyer, C. J. (2009). Adaptive Regularization using the Entire Solution Surface. *Biometrika* **96,** 513–527.

Wu, Y. and Liu, Y. (2012). Adaptively Weighted Large Margin Classifiers. *Journal of Computational and Graphical Statistics* **22,** 416–432.

Xu, Y., Yu, M., Zhao, Y.-Q., Li, Q., Wang, S., and Shao, J. (2015). Regularized outcome weighted subgroup identification for differential treatment effects. *Biometrics* **71,** 645–653.

Yao, W., Lindsay, B. G., and Li, R. (2012). Local Modal Regression. *Journal of nonparametric statistics* **24,** 647–663.

Ying, Z. (1993). A Large Sample Study of Rank Estimation for Censored Regression Data. *The Annals of Statistics* **21,** 76–99.

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

Zhang, C. and Liu, Y. (2013). Multicategory Large-margin Unified Machines. *Journal of Machine Learning Research* **14,** 1349–1386.

Zhang, C., Liu, Y., and Wu, Y. (2015). On Quantile Regression in Reproducing Kernel Hilbert Spaces with Data Sparsity Constraint. *Journal of Machine Learning Research* In press.

Zhang, C.-H. (2010). Nearly Unbiased Variable Selection under Minimax Concave Penalty. *The Annals of Statistics* **38,** 894–942.

Zhang, H. H., Cheng, G., and Liu, Y. (2011). Linear or Nonlinear? Automatic Structure Discovery for Partially Linear Models. *Journal of the American Statistical Association* **106,** 1099–1112.

Zhang, H. H., Lu, W., and Wang, H. (2010). On Sparse Estimation for Semiparametric Linear Transformation Models. *Journal of Multivariate Analysis* **101,** 1594–1606.

Zhang, T. (2004). Statistical Behavior and Consistency of Classification Methods Based on Convex Risk Minimization. *Annals of Statistics* **32,** 56–85.

Zhao, L., Tian, L., Cai, T., Claggett, B., and Wei, L. J. (2013). Effectively Selecting a Target Population for a Future Comparative Study. *Journal of the American Statistical Association* **108,** 527–539.

Zhao, P. and Yu, B. (2006). On Model Selection Consistency of Lasso. *Journal of Machine Learning Research* **7,** 2541–2563.

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

Zhao, Y.-Q., Zeng, D., Laber, E. B., and Kosorok, M. R. (2015a). New Statistical Learning Methods for Estimating Optimal Dynamic Treatment Regimes. *Journal of the American Statistical Association* **110,** 583–598.

Zhao, Y.-Q., Zeng, D., Laber, E. B., and Kosorok, M. R. (2015b). New Statistical Learning Methods for Estimating Optimal Dynamic Treatment Regimes. *Journal of the American Statistical Association* **110,** 583–598.

Zhao, Y. Q., Zeng, D., Laber, E. B., Song, R., Yuan, M., and Kosorok, M. R. (2014). Doubly robust learning for estimating individualized treatment with censored data. *Biometrika* page asu050.

Zhou, X., Mayer-Hamblett, N., Khan, U., and Kosorok, M. R. (2017). Residual Weighted Learning for Estimating Individualized Treatment Rules. *Journal of the American Statistical Association* **112,** 169–187.

Zou, H. (2006). The Adaptive Lasso and its Oracle Properties. *Journal of the American Statistical Association* **101,** 1418–1429.

Zou, H. and Hastie, T. J. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B* **67,** 301–320.

Zou, H. and Li, R. (2008). One-step Sparse Estimates in Nonconcave Penalized Likelihood Models. *The Annals of statistics* **36,** 1509.