UNIVERSITY OF CALIFORNIA
RIVERSIDE

Topics in Nonparametric Machine Learning: Subgroup Analysis and Deep Neural Networks Regression

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Mingming Liu

December 2021

Dissertation Committee:

Dr. Shujie Ma, Chairperson
Dr. Subir Ghosh
Dr. Esra Kurum

The Dissertation of Mingming Liu is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Shujie Ma, who guided me throughout my Ph.D. study. Without her guidance, patience, encouragement and support, I would not be able to complete this dissertation. Whenever I had questions, she was always ready to help me out. Her guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor for my Ph.D. study. I would like to thank you very much for all the help over these past four years.

Besides my advisor, I would like to thank my oral exam and thesis committee: Dr. Subir Ghosh, Dr. Esra Kurum, Dr. Zhiwei Zhang and Dr. Zhenyu Jia, for their constructive suggestions and insightful comments, and also their help during my study at UCR.

Last but not least, I would like to thank my parents and my husband for all the love, support and understanding. Without them, I would not have been here.

To my family for all the support.

ABSTRACT OF THE DISSERTATION

Topics in Nonparametric Machine Learning: Subgroup Analysis and Deep Neural Networks Regression

by

Mingming Liu

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, December 2021
Dr. Shujie Ma, Chairperson

In recent years, modern technology has facilitated the collection of large-scale data from medical records, health insurance databases, and other platforms. Due to the complex structure, the analysis of such data is very challenging. The dissertation focuses on the nonparametric machine learning techniques in subgroup analysis and deep neural networks regression.

The first part of the dissertation studies the heterogeneity in the disease progression, which is essential to the development of precision medicine that aims to tailor treatments to subgroups of patients with similar characteristics. Without a priori knowledge of grouping information, our goal is to identify subgroups of individuals who share a common disorder progress over time, i.e. longitudinal trajectory. We develop a subject-specific nonparametric regression model, where the heterogeneous trajectories are modeled through the subject-specific unknown functions and can be approximated by B-splines. We then apply the fusion penalized method that can automatically divide the individuals into different subgroups based on the B-spline coefficients as well as estimating the coefficients si-

multaneously. We also illustrate the performance of this method through simulation studies and a biomedical data application.

The second part of the dissertation considers a sparse deep ReLU network (SDRN) estimator obtained from empirical risk minimization with a Lipschitz loss function in the presence of a large number of features. Instead of utilizing full grids, the unknown target function is approximated by a deep ReLU network with sparse grids. Our framework can be applied to a variety of regression and classification problems. The unknown target function to estimate is assumed to be in a Sobolev space with mixed derivatives. Functions in this space only need to satisfy a smoothness condition rather than having a compositional structure. We develop non-asymptotic excess risk bounds for our SDRN estimator. We further derive that the SDRN estimator can achieve the same minimax rate of estimation (up to logarithmic factors) as one-dimensional nonparametric regression when the dimension of the features is fixed, and the estimator has a suboptimal rate when the dimension grows with the sample size. We show that the depth and the total number of nodes and weights of the ReLU network need to grow as the sample size increases to ensure a good performance, and also investigate how fast they should increase with the sample size. These results provide an important theoretical guidance and basis for empirical studies by deep neural networks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, advances in modern technologies have facilitated the collection of complex and large-scale data. To study the relationships among variables, regression analysis has been widely used. For instance, parametric regression models such as linear regression models are very convenient to study the relationships between the response variable and predictors. They are easy to understand and interpret. However, they may not be flexible enough to capture the hidden patterns in large-scale data. The linearity assumption (model assumption) can be easily violated due to the complex structure of the data in practice. The mis-specified models will lead to large bias in the estimators and false conclusions. In this dissertation, we focus on the nonparametric machine learning techniques used in subgroup analysis and deep neural networks regression, which can provide flexibility in modeling complex data without making restrictive structural assumptions and also have been shown to be very effective and powerful for estimating the unknown functions.

Chapter 2 reviews the basis functions of B-splines and sparse grids, which can be

used to approximate the nonparametric functions. In addition, to solve the high-dimensional problem, it introduces some penalized (regularized) regression models, such as Lasso, Ridge, MCP (minimax concave penalty) and SCAD (smoothly clipped absolute deviation penalty). Moreover, this chapter presents the alternating direction method of multipliers (ADMM) algorithm and Adam (adaptive moment estimation) algorithm that are well suited to the optimization problems.

Chapter 3 concentrates on the one-dimensional nonparametric regression in subgroup analysis. Subgroup analysis plays an important role in precision medicine. Uncovering the heterogeneity in the disease progression is a key factor to disease understanding and treatment development, so that interventions can be tailored to target the subgroups that will benefit most from the treatment, which is an important goal of precision medicine. However, in practice, one top methodological challenge hindering the heterogeneity investigation is that the true subgroup membership of each individual is often unknown. In this chapter, we aim to identify latent subgroups of individuals who share a common disorder progress over time, to predict latent subgroup memberships, and to estimate and infer the heterogeneous trajectories among the subgroups. To achieve these goals, we apply a concave fusion learning method proposed in [60, 61] to conduct subgroup analysis for longitudinal trajectories of the Alzheimer's disease data. The heterogeneous trajectories are represented by subject-specific unknown functions which are approximated by B-splines. The concave fusion method can simultaneously estimate the spline coefficients and merge them together for the subjects belonging to the same subgroup to automatically identify subgroups and recover the heterogeneous trajectories. The resulting estimator of the disease trajectory of

each subgroup is supported by an asymptotic distribution. It provides a sound theoretical basis for further conducting statistical inference in subgroup analysis. We also demonstrate the performance of this method through extensive simulation studies and a real data application.

Different from Chapter 3, which uses B-splines to approximate the unknown functions of one variable, Chapter 4 focuses on the nonparametric problems in high dimensions. If we have more variables, i.e. high dimensions, the approximation procedure will be more complicated. For instance, to predict an organism's phenotype (such as human disease, crop yield and drought resistance), which results from its genotype and environment, the researchers need to consider more factors such as genes, sunlight and nutrients. However, adding more factors will make the sample size you need grow exponentially and quickly become unmanageable. As a result, it will become more difficult to approximate the unknown predictive function and the computational cost is also very expensive. To address this problem, Chapter 4 considers a sparse deep ReLU network (SDRN) estimator obtained from empirical risk minimization with a Lipschitz loss function in the presence of a large number of features. More specifically, the estimator of the target function is built upon a network architecture of sparsely-connected deep neural networks with the rectified linear unit (ReLU) activation function. We consider the Sobolev spaces with square-integrable mixed second derivatives, which are commonly used for the sparse grids methods when dealing with the high-dimensional problems. Rather than requiring a compositional structure assumption, functions in this space only need to satisfy a smoothness condition, which is more flexible. In addition, regularization is used for preventing possible over fitting. Many

regression and classification problems can be solved by our framework. We also develop statistical properties of the proposed methodology. We derive non-asymptotic excess risk bounds for our SDRN estimator. We further show that our SDRN estimator can achieve the same optimal minimax estimation rate as one-dimensional nonparametric regression when the dimension of the features is fixed. Meanwhile, the SDRN estimator has a suboptimal rate when the dimension grows with the sample size. Moreover, to ensure a good performance, we show that the depth and the total number of nodes and weights of the ReLU network need to grow as the sample size increases. Simulation studies are conducted to evaluate the performance of the proposed method. We also illustrate the method through four real data applications.

The conclusions are given in Chapter 5. And the related technical proofs are included in the Appendix.

Chapter 2

Basis Functions and High-dimensional Regression

2.1 Basis Functions

Nonparametric regression is widely used when there is not a predetermined form to describe the relationship between the response variable and explanatory variables. To approximate the nonparametric components, we can use the basis functions, which consist of a particular basis for a function space. Functions in the function space can be uniquely represented by a linear combination of these basis functions. In this section, we introduce the basis functions of B-splines and sparse grids.

2.1.1 B-splines

In mathematics, a B-spline (basis spline) is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition. Any spline

function of a given degree can be represented by a linear combination of B-splines of that degree. A spline function is a piecewise polynomial function. B-spline curves are determined by the order q and the number of interior knots N. A spline of order q is a piecewise polynomial function of degree $q-1$. We start with the definition of knots. Let

$$a_0 = t_0 \leq t_1 \leq \cdots \leq t_N \leq t_{N+1} = b_0$$

be a knot sequence, where t_0 and t_{N+1} are the two end points, and $\{t_j\}_{j=1}^{N}$ is the interior knots sequence. The B-spline basis functions [20] are defined recursively as

$$B_{i,1}(x) = \begin{cases} 1, & \text{if } t_i \leq x \leq t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$B_{i,k+1}(x) = \omega_{i,k}(x) B_{i,k}(x) + [1 - \omega_{i+1,k}(x)] B_{i+1,k}(x),$$

where

$$\omega_{i,k}(x) = \begin{cases} \frac{x - t_i}{t_{i+k} - t_i}, & \text{if } t_{i+k} \neq t_i \\ 0. & \text{otherwise} \end{cases}$$

Note that $B_{i,1}(x)$ are the B-splines of order 1, which satisfy $\sum_i B_{i,1}(x) = 1$, and $B_{i,k+1}(x)$ are the higher order B-splines.

Let $G = G^{(q-2)}$ be the space spanned by the B-splines with order q. For any function f in this space, it can be expressed uniquely by the linear combination of B-spline basis functions, i.e.

$$f(x) = \sum_{i=1}^{N+q} B_{i,q}(x) \beta_i,$$

where $N + q$ is the number of basis functions and β_i's are B-spline coefficients. When $f(\cdot)$ is a function of multiple variables, it can be estimated through the tensor product of B-spline basis functions for each variable (full grids method). However, in high-dimensional

problems, it is very complex to implement that. Therefore, in the following, we present the sparse grids basis.

2.1.2 Sparse Grids

We first introduce a hierarchical basis of piecewise linear functions. To approximate functions of one variable x on $[0,1]$, a simple choice of a basis function is the standard hat function $\phi(x)$:

$$\phi(x) = \begin{cases} 1 - |x|, & \text{if } x \in [-1,1] \\ 0, & \text{otherwise.} \end{cases}$$

To generate a one-dimensional hierarchical basis, we consider a family of grids Ω_ℓ of level ℓ characterized by a grid size $h_\ell = 2^{-\ell}$ and $2^\ell + 1$ points $x_{\ell,s} = sh_\ell$ for $0 \le s \le 2^\ell$. On each Ω_ℓ, the piecewise linear basis functions $\phi_{\ell,s}$ are given as

$$\phi_{\ell,s}(x) = \phi(\frac{x - x_{\ell,s}}{h_\ell}),\ 0 \le s \le 2^\ell,$$

on the support $[x_{\ell,s} - h_\ell, x_{\ell,s} + h_\ell] \cap [0,1]$. The hierarchical increment spaces W_ℓ on each Ω_ℓ are given by

$$W_\ell = \text{span}\{\phi_{\ell,s} : s \in I_\ell\},$$

where $I_\ell = \{s \in \mathbb{N} : 0 \le s \le 2^\ell;\ s \text{ are odd numbers for } \ell \ge 1\}$ and $\mathbb{N} = \{0, 1, 2, \dots, \}$. We can see that for each $\ell \ge 1$, the supports of all basis functions $\phi_{\ell,s}$ spanning W_ℓ are mutually disjoint. Then the hierarchical space of functions up to level L is

$$V_L = \bigoplus_{0 \le \ell \le L} W_\ell = \text{span}\{\phi_{\ell,s} : s \in I_\ell, 0 \le \ell \le L\}.$$

To approximate functions of d-dimensional variables $\boldsymbol{x} = (x_1, \dots, x_d)^\top$ on $\mathcal{X} = [0,1]^d$, we employ a tensor product construction of the basis functions. We consider a

family of grids $\Omega_{\boldsymbol{\ell}}$ of level $\boldsymbol{\ell} = (\ell_1, ..., \ell_d)^\top$ with interior points $\boldsymbol{x}_{\boldsymbol{\ell},\boldsymbol{s}} = \boldsymbol{s} \cdot \boldsymbol{h}_{\boldsymbol{\ell}}$, where $\boldsymbol{h}_{\boldsymbol{\ell}} = (h_{\ell_1}, ..., h_{\ell_d})^\top$ with $h_{\ell_j} = 2^{-\ell_j}$ and $\boldsymbol{s} = (s_1, ..., s_d)^\top$ for $0 \leq s_j \leq 2^{\ell_j}$ and $j = 1, ..., d$. On each $\Omega_{\boldsymbol{\ell}}$, the basis functions $\phi_{\boldsymbol{\ell},\boldsymbol{s}}$ are given as

$$\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = \prod_{j=1}^{d} \phi_{\ell_j,s_j}(x_j),\ \mathbf{0}_d \leq \boldsymbol{s} \leq 2^{\boldsymbol{\ell}},$$

The hierarchical increment spaces $W_{\boldsymbol{\ell}}$ are given by

$$W_{\boldsymbol{\ell}} = \text{span}\{\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) : \boldsymbol{s} \in I_{\boldsymbol{\ell}}\},$$

where $I_{\boldsymbol{\ell}} = I_{\ell_1} \times \cdots \times I_{\ell_d}$, and $I_{\ell j} = \{s_j \in \mathbb{N} : 0 \leq s_j \leq 2^{\ell_j}, s_j \text{ are odd numbers for } \ell_j \geq 1\}$. Then the hierarchical space of functions up to level $\boldsymbol{L} = (L_1, ..., L_d)^\top$ is

$$V_{\boldsymbol{L}} = \bigoplus_{\mathbf{0} \leq \boldsymbol{\ell} \leq \boldsymbol{L}} W_{\boldsymbol{\ell}} = \text{span}\{\phi_{\boldsymbol{\ell},\boldsymbol{s}} : \boldsymbol{s} \in I_{\boldsymbol{\ell}}, \mathbf{0}_d \leq \boldsymbol{\ell} \leq \boldsymbol{L}\}.$$

For any function f in the space $V_{\boldsymbol{L}}$, it can be represented by the hierarchical basis:

$$f(\boldsymbol{x}) = \sum_{\mathbf{0}_d \leq \boldsymbol{\ell} \leq \boldsymbol{\infty}} \sum_{s \in I_{\boldsymbol{\ell}}} \gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} \phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = \sum_{\mathbf{0}_d \leq \boldsymbol{\ell} \leq \boldsymbol{\infty}} g_{\boldsymbol{\ell}}(\boldsymbol{x}), \tag{2.1}$$

where $\gamma^0_{\boldsymbol{\ell},\boldsymbol{s}}$ are the hierarchical coefficients and $g_{\boldsymbol{\ell}}(\boldsymbol{x}) = \sum_{s \in I_{\boldsymbol{\ell}}} \gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} \phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) \in W_{\boldsymbol{\ell}}$.

In practice, one can use a truncated version to approximate the function $f(\cdot)$ given in (2.1), so that

$$f(\boldsymbol{x}) \approx \sum\nolimits_{0 \leq |\boldsymbol{\ell}|_\infty \leq m} \sum\nolimits_{s \in I_{\boldsymbol{\ell}}} \gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} \phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = \sum\nolimits_{0 \leq |\boldsymbol{\ell}|_\infty \leq m} g_{\boldsymbol{\ell}}(\boldsymbol{x}),$$

which is constructed based on the space with full grids: $V_m^{(\infty)} = \bigoplus_{0 \leq |\boldsymbol{\ell}|_\infty \leq m} W_{\boldsymbol{\ell}} = \text{span}\{\phi_{\boldsymbol{\ell},\boldsymbol{s}} : s \in I_{\boldsymbol{\ell}}, 0 \leq |\boldsymbol{\ell}|_\infty \leq m\}$. The dimension of the space $V_m^{(\infty)}$ is $|V_m^{(\infty)}| = (2^m + 1)^d$, which increases with d in an exponential order.

For dimension reduction, we consider the hierarchical space with sparse grids:

$$V_m^{(1)} = \bigoplus_{|\boldsymbol{\ell}|_1 \leq m} W_{\boldsymbol{\ell}} = \text{span}\{\phi_{\boldsymbol{\ell},\boldsymbol{s}} : s \in I_{\boldsymbol{\ell}}, |\boldsymbol{\ell}|_1 \leq m\}.$$

The function $f(\cdot)$ given in (2.1) can be approximated by

$$f_m(\boldsymbol{x}) = \sum\nolimits_{|\boldsymbol{\ell}|_1 \leq m} \sum\nolimits_{s \in I_{\boldsymbol{\ell}}} \gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} \phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = \sum\nolimits_{|\boldsymbol{\ell}|_1 \leq m} g_{\boldsymbol{\ell}}(\boldsymbol{x}).$$

Clearly, when $d = 1$, the dimension of the hierarchical space with sparse grids is the same as that of the space with full grids. The dimensionality issue does not exist. Table 2.1 provides the number of basis functions for the hierarchical space with sparse grids $V_m^{(1)}$ and the space with full grids $V_m^{(\infty)}$ when the dimension of the covariates d increases from 2 to 8 and the m value increases from 0 to 4. We see that the number of basis functions for the space with sparse grids is dramatically reduced compared to the space with full grids, when the dimension d or m value become larger, so that the dimensionality problem can be lessened.

	Sparse grids					Full grids				
	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$
$d=2$	4	8	17	37	81	4	9	25	81	289
$d=3$	8	20	50	123	297	8	27	125	729	4913
$d=4$	16	48	136	368	961	16	81	625	6561	83521
$d=5$	32	112	352	1032	2882	32	243	3125	59049	1419857
$d=6$	64	256	880	2768	8204	64	729	15625	531441	24137569
$d=7$	128	576	2144	7184	22472	128	2187	78125	4782969	410338673
$d=8$	256	1280	5120	18176	59744	256	6561	390625	43046721	6975757441

Table 2.1: The number of basis functions for the space with sparse grids and the space with full grids.

2.2 High-dimensional Regression

We assume that the relationship between the response variable and covariates can be described as

$$y_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \ldots, n, \tag{2.2}$$

where y_i is the response variable, $\boldsymbol{x}_i$ is a p-vector of covariates, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is the vector of the unknown regression coefficients, and ε_i are i.i.d random errors with mean 0 and constant variance. The first entry in each $\boldsymbol{x}_i$ is 1 so that the intercept is included in $\boldsymbol{\beta}$. Let $\boldsymbol{Y} = (y_1, \ldots, y_n)^T, \boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^T$. Model (2.2) can be rewritten as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad i = 1, \ldots, n. \tag{2.3}$$

Define $L(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \boldsymbol{x}_i^T \boldsymbol{\beta})^2 = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})$. When $p < n$, the ordinary least square estimator can be obtained through

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}). \tag{2.4}$$

(2.4) has a closed-form solution with $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}$, in which it is assumed that $(\boldsymbol{X}^T \boldsymbol{X})^{-1}$ is well defined. However, in high-dimensional problems, the number of covariates can be larger than the number of observations in practice, i.e. $p > n$. In this situation, model (2.2) can not be identified since $\boldsymbol{X}^T \boldsymbol{X}$ is not invertible due to the multicollinearity. Even in low-dimensional problems ($p < n$), the predictors variables (covariates) can also be highly correlated. To tackle this problem, we consider the penalized regression methods, in which all the predictor variables are kept in the model but regularize the regression coefficients $\boldsymbol{\beta}$ by shrinking them toward 0.

2.2.1 Penalized Regression

The commonly used penalized regression methods include ridge regression and lasso regression, in which the ridge penalty (L_2 penalty) or lasso penalty (L_1 penalty) is added in the minimization criterion. The penalty term will control the size of $\boldsymbol{\beta}$. Additionally, we introduce MCP and SCAD penalties. In penalized regression, to obtain the coefficients, we minimize

$$L_1(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) + \sum_{j=1}^{p} p(\beta_j, \lambda),$$

where $p(\cdot\,, \lambda)$ is a penalty function and $\lambda \geq 0$ is a penalization tuning parameter controlling the strength of the penalty term.

Ridge

In ridge regression [35], we minimize

$$L_1(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} \beta_j^2.$$

We then get the ridge estimator $\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I}_p)^{-1}\boldsymbol{X}^T\boldsymbol{Y}$, where $\boldsymbol{I}_p$ is a $p \times p$ identity matrix. In this case, $\boldsymbol{X}^T\boldsymbol{X} + \lambda \boldsymbol{I}_p$ is always invertible for $\lambda > 0$. Clearly, when $\lambda = 0$, the ridge estimator is the same as the ordinary least square estimator. As λ increases, the bias in the estimator increases but the variance decreases. Therefore, the ridge regression works well to avoid over fitting issue. When λ is very large, the ridge regression shrinks the estimator $\hat{\boldsymbol{\beta}}_{\text{ridge}}$ toward 0, but not to be exactly 0. As a result, ridge regression are usually used to deal with multicollinearity issue instead of performing variable selection.

Lasso

[86] first introduced the lasso penalty $p(t, \lambda) = \lambda |t|$. In lasso regression, we minimize

$$L_1(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_i| .$$

It can be seen that lasso regression uses a L_1 penalty, while ridge regression considers a L_2 penalty. Although the problems seems similar, their solutions behave very differently. Compared to ridge penalty, lasso penalty can shrink some coefficients to be 0 exactly when λ increases, which leads to a sparse estimator. Based on this, lasso regression can remove the insignificant predictors from the model. In other words, lasso regression additionally performs variable selection. However, as lasso penalty applies the same penalization to each coefficient, it tends to over-shrink the large coefficients, and thus results in biased estimates. In the following, we introduce two concave penalties, MCP (minimax concave penalty [98]) and SCAD (smoothly clipped absolute deviation penalty [27]), which not only induce the nearly unbiased estimates, but also enjoy the sparsity property.

MCP and SCAD

For MCP, it has the form

$$p_\tau(t, \lambda) = \lambda \int_0^{|t|} (1 - x/(\tau\lambda))_+ dx, \quad \tau > 1,$$

and the SCAD penalty has the form

$$p_\tau(t, \lambda) = \lambda \int_0^{|t|} \min \{1, (\tau - x/\lambda)_+/(\tau - 1)\}, \quad \tau > 2,$$

where τ is a parameter that controls the concavity of the penalty function, and $(x)_+ = x$, if $x > 0$; $(x)_+ = 0$, otherwise. In particular, when $\tau \to \infty$, both penalties converge to

the lasso penalty (L_1 penalty). Similar to lasso penalty, these two concave penalties enjoy the sparsity as well. That is, as λ increases, they can shrink some coefficients to 0 exactly. In practice, we only want to shrink small coefficients, and do not want to shrink the large coefficients. This can be achieved by employing MCP and SCAD penalties. Therefore, both of them produce nearly unbiased estimates. To estimate the coefficients here, we minimize

$$L_1(\boldsymbol{\beta}) = L(\boldsymbol{\beta}) + \sum_{j=1}^{p} p_\tau(\beta_j, \lambda),$$

where $p_\tau(\cdot\ , \lambda)$ represents the MCP or SCAD penalty.

2.2.2 ADMM Algorithm

Recently, statistics and machine learning with large-scale data is a very popular topic of widespread interest, such as in medicine, artificial intelligence, computational biology, etc. Many such problems can be posed in the framework of convex optimization. The alternating direction method of multipliers (ADMM), as a simple but powerful algorithm, has been widely used for solving the structured convex optimization problems. From the discussion of a number of examples, [9] showed that ADMM is well suited for the large-scale distributed problems arising in applied statistics and machine learning.

Following [9], the ADMM algorithm solves problems in the form

$$\begin{aligned} &\text{minimize} \quad f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ &\text{subject to} \quad \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} = \boldsymbol{c} \end{aligned} \tag{2.5}$$

where f and g are convex functions, $\boldsymbol{x} \in \boldsymbol{R}^n, \boldsymbol{z} \in \boldsymbol{R}^m, \boldsymbol{A} \in \boldsymbol{R}^{p\times n}, \boldsymbol{B} \in \boldsymbol{R}^{p\times m}$ and $\boldsymbol{c} \in \boldsymbol{R}^p$. The augmented Lagrangian for this problem is

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + g(\boldsymbol{z}) + \boldsymbol{\mu}^T(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c}) + (\rho/2)\left\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c}\right\|_2^2,$$

where $\|\cdot\|_2$ is the L_2 norm with $\|\boldsymbol{a}\|_2 = (\sum |a_i|^2)^{1/2}$, $\boldsymbol{\mu}$ is the dual variable or Lagrange multiplier and $\rho > 0$ is a penalty parameter. Then ADMM consists of the iterations

$$\boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}} L_\rho(\boldsymbol{x}, \boldsymbol{z}^k, \boldsymbol{\mu}^k) \tag{2.6}$$

$$\boldsymbol{z}^{k+1} = \arg\min_{\boldsymbol{z}} L_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{z}, \boldsymbol{\mu}^k) \tag{2.7}$$

$$\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}^k + \rho(\boldsymbol{A}\boldsymbol{x}^{k+1} + \boldsymbol{B}\boldsymbol{z}^{k+1} - \boldsymbol{c}) \tag{2.8}$$

It can be seen that ADMM updates $\boldsymbol{x}$ and $\boldsymbol{z}$ in an alternating or sequential way, which accounts for the term alternating direction. At step $k+1$, we have the primal residual $\boldsymbol{r}^{k+1} = \boldsymbol{A}\boldsymbol{x}^{k+1} + \boldsymbol{B}\boldsymbol{z}^{k+1} - \boldsymbol{c}$ and dual residual $\boldsymbol{s}^{k+1} = \rho\boldsymbol{A}^T\boldsymbol{B}(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)$. As ADMM proceeds, the primal residual and dual residual converge to zero.

2.2.3 Adam Algorithm

Adam (adaptive moment estimation) algorithm [44] is another popular optimization algorithm, which is of great importance in deep learning. Even though ADMM presents promising performance in many conventional machine learning applications and can be applied to deep learning, there still exist some challenges. For instance, it converges slowly to high accuracy. Moreover, it is very time-consuming to implement when there are a large number of features and a big sample size, which also needs a big memory. In contrast, Adam algorithm considers first-order gradient-based optimization with little memory requirement. It is straightforward to implement and also computationally efficient.

Referring to [44], let $f(\theta)$ be a stochastic scalar function that is differentiable with respect to the parameters θ. To minimize the objective function, the Adam algorithm proposed in [44] is given as follows.

Algorithm 1 Adam algorithm

Require: θ_0: Initial parameter vector

$m_0 \longleftarrow 0$ (Initialize 1^{st} moment vector)

$v_0 \longleftarrow 0$ (Initialize 2^{nd} moment vector)

$t \longleftarrow 0$ (Initialize timestep)

while θ_t not converged **do**

$t \longleftarrow t + 1$

$g_t \longleftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradient w.r.t. stochastic objective at timestep t)

$m_t \longleftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$ (Update biased first moment estimate)

$v_t \longleftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ (Update biased second raw moment estimate)

$\widehat{m}_t \longleftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)

$\widehat{v}_t \longleftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)

$\theta_t \longleftarrow \theta_{t-1} - \alpha \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)

end while

return θ_t

Note that, $g_t = \nabla_\theta f_t(\theta)$ represents the gradient, i.e. the vector of partial derivatives of f_t with respect to θ evaluated at t, and g_t^2 indicates the elementwise square $g_t \bigodot g_t$. α is the step size, and $\beta_1, \beta_2 \in [0, 1]$ are the exponential decay rates for the moment estimates. The good default choices are $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

Chapter 3

A Fusion Learning Method to Subgroup Analysis of Alzheimer's Disease

3.1 Introduction

Alzheimer's disease (AD) is the leading cause of dementia for adults. It is a progressive disease that worsens over time. Patients with AD show symptoms of memory loss, mental decline, delusion and so forth as the disease progresses. The progression of AD varies from person to person, and patients with AD have experienced it in different ways. The lack of a good understanding of the heterogeneity in the disease progression through the population is a key reason for failures of disease-modifying treatments for AD. As a result, very little progress has been made for the AD treatment development since 2003 [96]. To overcome

this difficulty, one has to first understand the heterogeneity in the disease trajectories, so that interventions can be tailored to target the subgroups that will benefit most from the treatment, which is an important goal of precision medicine. The progression of AD is often measured by cognitive scores at multiple time points, resulting in a collection of longitudinal data. One major methodological challenge hindering the heterogeneity investigation is that the true subgroup membership of each individual is often unknown.

The growth mixture modeling (GMM) method [28, 82, 42, 65] has been popularly used for the identification and prediction of latent subpopulations for longitudinal data. This method requires to specify the underlying distribution of the data, which is often hard to obtain for longitudinal data, because of their complex structure. The k-means algorithm [34] is another popular clustering method. It divides the data into subgroups based on the distances between measurement vectors of subjects. It is difficult to apply this method to cluster functional curves, especially arising from longitudinal data with missing measurements. Moreover, both GMM and k-means methods need to pre-specify the number of subgroups, which is often unknown in practice, and thus introduces additional complications in the estimation procedure.

To overcome these challenges, we apply the concave fusion learning method proposed in [60, 61] to conduct subgroup analysis for longitudinal trajectories of the AD data. This semi-supervised machine learning method applies concave penalty functions to pairwise differences of clinical outcomes or unknown treatment coefficients in a regression model. It can automatically identify memberships from latent subgroups and estimate the number of subgroups simultaneously without specifying the underlying distribution. Although the fu-

sion learning method was originally considered in [60, 61] for the cross-sectional data setting with independent observations, it also has a great potential for subgroup analysis of other data settings such as longitudinal data and survival data. In this article, we extend this method to the longitudinal AD data, and investigate its numerical performance through extensive simulation studies with both balanced and unbalanced correlated repeated measures designs. Moreover, we propose two different data-driven methods based on the modified Bayes Information Criterion BIC and the Calinski-Harabasz (CH) index, respectively, for selecting the optimal tuning parameter involved in the concave fusion penalization method, while the CH method was not considered in [60, 61]. We also thoroughly investigate the performance of these two data-driven methods through numerical studies.

To cluster the AD patients based on their cognitive scores observed over time, we consider a subject-specific nonparametric regression model, in which the heterogeneity can be driven by observed or unobserved latent covariates. More specifically, we model each patient's cognitive scores through an unknown functional curve of time. We approximate each curve by B-splines [20, 51, 93, 58], and then apply pairwise fusion penalties to the spline coefficients, so that patients with similar disease trajectories can be automatically clustered into the same homogeneous subgroup. As a result, patients in the same identified subgroup share the same disease progressive curve. We use an alternating direction method of multipliers (ADMM) algorithm [9] that has a good convergence property to solve the optimization problem. Different from the GMM method, our method does not require to pre-specify the number of subgroup, nor does it need to provide the underlying distribution of the data. Instead, our estimation procedure only involves a working correlation matrix

[49, 90, 57, 62] for the repeated measures of each subject. We show that the resulting estimator of the functional curve for each subgroup is robust to the specification of the correlation matrix, i.e., it is still a consistent estimator even if the working correlation matrix is mis-specified. Moreover, we establish point-wise asymptotic normality of the functional curve estimator for each subgroup, so that statistical inference can be further conducted based on our clustering and estimation results.

The rest is organized as follows. Section 3.2 describes the proposed model. Section 3.3 introduces the model estimation procedure using concave fusion penalization method. In Section 3.4, we establish the theoretical properties of the proposed estimators. Simulation studies are presented in Section 3.5. Section 3.6 illustrates the application of the proposed method to Alzheimer's disease data. Discussions are provided in Section 3.7. The related technical proofs are included in the Appendix A.

3.2 Model

In a longitudinal study, subjects are usually measured repeatedly over a time period. Suppose the data consist of $(Y_i(t_{ij}), t_{ij}), i = 1, \ldots, n, j = 1, \ldots, m_i$, where $\{t_{ij}, j = 1, \ldots, m_i\}$ are the distinct time points that the measurements of the ith subject are taken, and $Y_i(t_{ij})$ is the observed response for the ith subject at time t_{ij}. Our goal of this article is to understand how the change of trajectories may differ across individual subjects. To study the longitudinal trajectories of the ith subject, we consider the subject-specific nonparametric regression model:

$$Y_i(t_{ij}) = \beta_i(t_{ij}) + \varepsilon_i(t_{ij}), \tag{3.1}$$

where $\beta_i(t)$'s are the unknown smooth functions of t, and the errors $\varepsilon_i(t)$'s satisfy $E(\varepsilon_i(t)) = 0$ and $\text{Cov}(\varepsilon_i(t), \varepsilon_{i'}(t')) = \delta(t,t')I\{i = i'\}$ with $I\{\cdot\}$ being an indicator function. For simplicity, we denote $Y_{ij} = Y_i(t_{ij})$ and $\varepsilon_{ij} = \varepsilon_i(t_{ij})$. Model (3.1) can be rewritten as

$$Y_{ij} = \beta_i(t_{ij}) + \varepsilon_{ij}. \tag{3.2}$$

In this model, the trajectory of the ith subject over time is represented by the subject-specific unknown function $\beta_i(t)$. Due to the heterogeneity of the trajectories, we assume $\beta_i(t)$'s arise from K different groups with $K \geq 1$. To be specific, we have $\beta_i(t) = \alpha_k(t)$ for all $i \in \mathcal{G}_k$, where $\mathcal{G} = (\mathcal{G}_1, ..., \mathcal{G}_K)$ is a mutually exclusive partition of $\{1, ..., n\}$ and $\alpha_k(t)$ is the common function for all the $\beta_i(t)$'s from group $\mathcal{G}_k$. In practice, the number of subgroups K can be much smaller than the sample size n, and it is often unknown.

3.3 Estimation

In order to identify the subgroups of the heterogeneous trajectories, we first approximate the nonparametric functions $\beta_i(\cdot)$'s in (3.2) using B-splines. Referring to [59], let $a_0 = \zeta_0 < \zeta_1 < \cdots < \zeta_J < \zeta_{J+1} = b_0$ be a partition of $[a_0, b_0]$ into $J+1$ subintervals $I_l = [\zeta_l, \zeta_{l+1}), l = 0, \cdots, J-1$ and $I_J = [\zeta_J, b_0]$, where $\{\zeta_l\}_{l=1}^J$ is a sequence of interior knots. Denote the rth order normalized B-spline basis as $\{B_1(t), \ldots, B_S(t)\}^T$ (see [20]), in which $S = J + r$ is the number of basis functions. Then, $\beta_i(t_{ij})$ in (3.2) can be approximated by a linear combination of the B-spline functions,

$$\beta_i(t_{ij}) \approx \sum_{d=1}^{S} \gamma_{id} B_d(t_{ij}) = \boldsymbol{B}(t_{ij})^T \boldsymbol{\gamma}_i, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m_i, \tag{3.3}$$

where $\boldsymbol{B}(t_{ij}) = (B_1(t_{ij}), \ldots, B_S(t_{ij}))^T$ and $\boldsymbol{\gamma}_i = (\gamma_{i1}, \ldots, \gamma_{iS})^T$. In this case, the trajectory heterogeneity represented by $\beta_i(t)$ is reflected on the B-spline coefficient $\boldsymbol{\gamma}_i$. Therefore, our goal can be transformed into identifying the subgroups based on the $\boldsymbol{\gamma}_i$'s.

Let $\boldsymbol{Y}_i = (Y_{i1}, \ldots, Y_{im_i})^T, \boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \ldots, \varepsilon_{im_i})^T$ and $\boldsymbol{X}_i = (\boldsymbol{B}_{i1}, \ldots, \boldsymbol{B}_{im_i})^T$, where $\boldsymbol{B}_{ij} = \boldsymbol{B}(t_{ij})$. Given (3.3), for each i, model (3.2) can be written in matrix notation as

$$\boldsymbol{Y}_i \approx \boldsymbol{X}_i\boldsymbol{\gamma}_i + \boldsymbol{\varepsilon}_i, \quad i = 1, \cdots, n. \tag{3.4}$$

As in [49, 90, 62], we let $\boldsymbol{\Sigma}_i$ and $\boldsymbol{V}_i$ be the true and assumed working covariance of $\boldsymbol{Y}_i$, where $\boldsymbol{\Sigma}_i = \mathrm{Var}(\boldsymbol{Y}_i)$ and $\boldsymbol{V}_i = \boldsymbol{A}_i^{1/2}\boldsymbol{R}_i\boldsymbol{A}_i^{1/2}$, $\boldsymbol{A}_i$ represents a $m_i \times m_i$ diagonal matrix containing the marginal variances of Y_{ij}, and $\boldsymbol{R}_i$ is an invertible working correlation matrix. The true covariance $\boldsymbol{\Sigma}_i$ is often unknown in practice, so we use a working covariance $\boldsymbol{V}_i$ to replace $\boldsymbol{\Sigma}_i$ in the estimation procedure. The structure of the working correlation $\boldsymbol{R}_i$ is pre-specified. Throughout, we assume that $\boldsymbol{V}_i$ depends on a nuisance finite dimensional parameter vector $\boldsymbol{\eta}$.

Following [61], we utilize a fusion learning approach with concave penalty to estimate model (3.4). For any vector $\boldsymbol{a}$, define its L_2 norm as $\|\boldsymbol{a}\|_2 = (\sum a_i^2)^{1/2}$. The objective function is constructed as

$$Q_n(\boldsymbol{\gamma}; \lambda) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i)^T\boldsymbol{V}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i) + \sum_{1\le i<j\le n} p\left(\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|_2, \lambda\right), \tag{3.5}$$

where $\boldsymbol{\gamma} = \left(\boldsymbol{\gamma}_1^T, \ldots, \boldsymbol{\gamma}_n^T\right)^T$ and $p(\cdot, \lambda)$ is a concave penalty function with a tuning parameter $\lambda \ge 0$. For a given $\lambda > 0$, define

$$\hat{\boldsymbol{\gamma}}(\lambda) = \arg\min_{\boldsymbol{\gamma}} Q_n(\boldsymbol{\gamma}; \lambda). \tag{3.6}$$

When λ is large enough, the penalty shrinks some pairs of $\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|_2$ to zero. For two subjects with $\|\hat{\boldsymbol{\gamma}}_i(\lambda) - \hat{\boldsymbol{\gamma}}_j(\lambda)\|_2 = 0$, they are clustered into the same group. Based on this fact, we can partition the heterogeneous trajectories into subgroups. For convenience, we write $\hat{\boldsymbol{\gamma}}(\hat{\lambda}) \equiv \hat{\boldsymbol{\gamma}}$. Let $\left\{\hat{\boldsymbol{\theta}}_1, \ldots, \hat{\boldsymbol{\theta}}_{\hat{K}}\right\}$ be the unique values of $\hat{\boldsymbol{\gamma}}$, where $\hat{K}$ is the number of these distinct values. In the kth subgroup, we denote the set of the corresponding indices by $\hat{\mathcal{G}}_k = \left\{i : \hat{\boldsymbol{\gamma}}_i = \hat{\boldsymbol{\theta}}_k,\ 1 \leq i \leq n\right\}$ with $1 \leq k \leq \hat{K}$. To select the optimal tuning parameter λ, a data-driven procedure such as BIC or the Calinski-Harabasz index is considered. It is noteworthy that our method can also be applied to the case that the true number of subgroups K is known. In this scenario, we will choose a λ value that corresponds to the estimated number of subgroups $\hat{K}$ which is equal to or the closest one to the true number of subgroups K. If two $\hat{K}$ values are equally distant from K, we use the larger one to determine the λ value.

An appropriate selection of the penalty is very critical to the model estimation. Instead of choosing lasso penalty $p_\tau(t, \lambda) = \lambda |t|$ [86], which results in biased estimates due to the over-shrinkage of large coefficients, we use the minimax concave penalty (MCP) [98] by inducing nearly unbiased estimators with the form

$$p_\tau(t, \lambda) = \lambda \int_0^{|t|} (1 - x/(\tau\lambda))_+ dx, \quad \tau > 1,$$

where τ is a parameter controlling the concavity of the penalty function, and $(a)_+ = a$, if $a > 0$ and $(a)_+ = 0$, otherwise. Moreover, it is more aggressive in enforcing a sparser solution. Consequently, MCP is a more desirable choice.

Another problem is how to choose the working covariance matrix $\boldsymbol{V}_i$. Here we consider an unequally spaced AR(1) structure for the working covariance matrix $\boldsymbol{V}_i$, such

that $V_i(t,s) = \sigma^2 \rho^{\kappa|t-s|}$, where $\kappa = \frac{1}{|t_{(1)} - t_{(2)}|}$ with $t_{(1)}$, $t_{(2)}$ being the first two time points. Note that our estimator of the functional curve for each subgroup is consistent even if the working covariance matrix is mis-specified, i.e., $\boldsymbol{V}_i \neq \boldsymbol{\Sigma}_i$. First, we estimate σ^2 by taking the mean of the estimated variance $\hat{\sigma}_i^2$, $i = 1, \ldots, n$, where $\hat{\sigma}_i^2$ is calculated within subject by using ordinary least squares (OLS) residuals. Due to the fact that these residuals may be small and thus underestimate the true errors, we modify these residuals by replacing $\hat{\varepsilon}_{ij}$ with $\hat{\varepsilon}_{ij}^* = \hat{\varepsilon}_{ij}/(1 - h_{ij})$, where h_{ij} is the jth diagonal element of the projection matrix $\boldsymbol{H}_i$ for subject i. This modification is suggested by [63]. Given (3.4), we have $\boldsymbol{H}_i = \boldsymbol{X}_i(\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T$. Next, we estimate correlation ρ by taking the average of the estimated correlation between the two adjacent time points, in which we only consider the adjacent time points having the scaled distance equalling 1, i.e. $\kappa\,|t-s| = 1$. Accordingly, $\boldsymbol{V}_i$ can be obtained.

Computation Using ADMM Algorithm

It is worth noting that the penalty function in (3.5) is not separable in $\boldsymbol{\gamma}_i$'s. To obtain the solution of (3.6), following [61], we derive an ADMM algorithm to minimize the objective function (3.5). By introducing a new set of parameters $\boldsymbol{\delta}_{ij} = \boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j$, the problem can be reformulated as the following constrained optimization:

$$\min \ \frac{1}{2} \sum_{i=1}^{n} (\boldsymbol{Y}_i - \boldsymbol{X}_i \boldsymbol{\gamma}_i)^T \boldsymbol{V}_i^{-1} (\boldsymbol{Y}_i - \boldsymbol{X}_i \boldsymbol{\gamma}_i) + \sum_{1 \leq i < j \leq n} p_\tau \left(\|\boldsymbol{\delta}_{ij}\|_2, \lambda \right),$$
$$\text{subject to } \boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j - \boldsymbol{\delta}_{ij} = \boldsymbol{0}. \tag{3.7}$$

Denote by $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \boldsymbol{a}^T \boldsymbol{b}$ the inner product of two vectors. The above constrained optimization can be transformed into its augmented Lagrangian optimization problem, i.e,

minimize:

$$L(\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\upsilon}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i)^T\boldsymbol{V}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i) + \sum_{1\le i<j\le n} p_\tau\left(\|\boldsymbol{\delta}_{ij}\|_2, \lambda\right)$$
$$+ \sum_{i<j}\langle \boldsymbol{\upsilon}_{ij}, \boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j - \boldsymbol{\delta}_{ij}\rangle + \frac{\vartheta}{2}\sum_{i<j}\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j - \boldsymbol{\delta}_{ij}\|_2^2, \tag{3.8}$$

where $\boldsymbol{\delta} = \left\{\boldsymbol{\delta}_{ij}^T, i<j\right\}^T$, the dual variables $\boldsymbol{\upsilon} = \left\{\boldsymbol{\upsilon}_{ij}^T, i<j\right\}^T$ are the Lagrange multipliers and ϑ is the penalty parameter. Then we can compute the estimates of $(\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\upsilon})$ through iterations using the ADMM algorithm.

Given the value of $\boldsymbol{\delta}^m, \boldsymbol{\upsilon}^m$ at step m, we update the estimates at step $m+1$ as follows:

$$\boldsymbol{\gamma}^{m+1} = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} L\left(\boldsymbol{\gamma}, \boldsymbol{\delta}^m, \boldsymbol{\upsilon}^m\right), \tag{3.9}$$

$$\boldsymbol{\delta}^{m+1} = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} L\left(\boldsymbol{\gamma}^{m+1}, \boldsymbol{\delta}, \boldsymbol{\upsilon}^m\right), \tag{3.10}$$

$$\boldsymbol{\upsilon}_{ij}^{m+1} = \boldsymbol{\upsilon}_{ij}^m + \vartheta\left(\boldsymbol{\gamma}_i^{m+1} - \boldsymbol{\gamma}_j^{m+1} - \boldsymbol{\delta}_{ij}^{m+1}\right). \tag{3.11}$$

Notice that the problem in (3.9) is equivalent to minimizing the function

$$f(\boldsymbol{\gamma}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i)^T\boldsymbol{V}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i) + \frac{\vartheta}{2}\sum_{i<j}\left\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j - \boldsymbol{\delta}_{ij}^m + \vartheta^{-1}\boldsymbol{\upsilon}_{ij}^m\right\|_2^2 + C_0$$
$$= \frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\gamma})^T\boldsymbol{V}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\gamma}) + \frac{\vartheta}{2}\left\|\boldsymbol{A}\boldsymbol{\gamma} - \boldsymbol{\delta}^m + \vartheta^{-1}\boldsymbol{\upsilon}^m\right\|_2^2 + C_0,$$

where $\boldsymbol{Y} = \left(\boldsymbol{Y}_1^T, ..., \boldsymbol{Y}_n^T\right)^T$, $\boldsymbol{X} = \operatorname{diag}(\boldsymbol{X}_1, ..., \boldsymbol{X}_n)$, $\boldsymbol{V} = \operatorname{diag}(\boldsymbol{V}_1, ..., \boldsymbol{V}_n)$, $\boldsymbol{A} = \boldsymbol{D} \otimes \boldsymbol{I}_S$ (Kronecker product) and C_0 is a constant independent of $\boldsymbol{\gamma}$. Here $\boldsymbol{D} = \{(\boldsymbol{e}_i - \boldsymbol{e}_j), i<j\}^T$, in which $\boldsymbol{e}_i$ is a $n \times 1$ vector with the i th element being 1 and the remaining ones being 0, and $\boldsymbol{I}_S$ is a $S \times S$ identity matrix. Thus, we can update $\boldsymbol{\gamma}^{m+1}$ by

$$\boldsymbol{\gamma}^{m+1} = \left(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X} + \vartheta\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\left[\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{Y} + \vartheta\boldsymbol{A}^T\left(\boldsymbol{\delta}^m - \vartheta^{-1}\boldsymbol{\upsilon}^m\right)\right]. \tag{3.12}$$

In (3.10), given $\boldsymbol{\gamma}^{m+1}$ and $\boldsymbol{\upsilon}^m$, the minimization problem is the same as minimizing

$$\frac{\vartheta}{2}\left\|\boldsymbol{\zeta}_{ij}^m-\boldsymbol{\delta}_{ij}\right\|_2^2+p_\tau\left(\left\|\boldsymbol{\delta}_{ij}\right\|_2,\lambda\right)$$

with respect to $\boldsymbol{\delta}_{ij}$, where $\boldsymbol{\zeta}_{ij}^m=\boldsymbol{\gamma}_i^{m+1}-\boldsymbol{\gamma}_j^{m+1}+\vartheta^{-1}\boldsymbol{\upsilon}_{ij}^m$. Consequently, for MCP penalty with $\tau>1/\vartheta$, we have:

$$\boldsymbol{\delta}_{ij}^{m+1}=\begin{cases}\frac{\mathrm{ST}(\boldsymbol{\zeta}_{ij}^m,\lambda/\vartheta)}{1-1/(\tau\vartheta)} & \text{if } \|\boldsymbol{\zeta}_{ij}^m\|_2\le\tau\lambda,\\ \boldsymbol{\zeta}_{ij}^m & \text{if } \|\boldsymbol{\zeta}_{ij}^m\|_2>\tau\lambda,\end{cases}\tag{3.13}$$

where $\mathrm{ST}(\boldsymbol{z},t)=(1-t/\left\|\boldsymbol{z}\right\|_2)_+\boldsymbol{z}$ is the groupwise soft thresholding operator.

Given the discussion above, we summarize the detailed ADMM algorithm as follows:

Algorithm 2 ADMM algorithm

1: Initialize $\boldsymbol{\delta}^0$, $\boldsymbol{\upsilon}^0$.
2: **for** $m=0,1,2,\cdots$ **do**
3: Update $\boldsymbol{\gamma}^{m+1}$ using (3.12)
4: Update $\boldsymbol{\delta}^{m+1}$ using (3.13)
5: Update $\boldsymbol{\upsilon}^{m+1}$ using (3.11)
6: **if** the convergence criterion is met, **then**
7: Stop and denote the last iteration by $\hat{\boldsymbol{\gamma}}(\lambda)$,
8: **else**
9: $m=m+1$.
10: **end if**
11: **end for**

Ensure: Output

We stop the ADMM algorithm when the primal residual $\boldsymbol{r}^{m+1}=\boldsymbol{A}\boldsymbol{\gamma}^{m+1}-\boldsymbol{\delta}^{m+1}$ is close to zero such that $\left\|\boldsymbol{r}^{m+1}\right\|_2<\varepsilon$ for some small value $\varepsilon>0$.

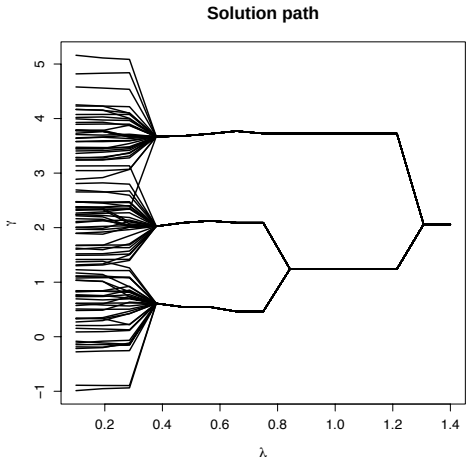


Figure 3.1: Solution path for $(\hat{\gamma}_{31}(\lambda), \ldots, \hat{\gamma}_{3n}(\lambda))$ against λ with $n = 100, T = 20$ for balanced data of Middle case from Three Subgroup Example in Section 3.5.

Remark 3.1 *To start ADMM algorithm, an appropriate initial value is very important. First, given model (3.4), we use the ordinary least squares estimate of each subject as the initial estimate* $\boldsymbol{\gamma}^0$*, i.e.* $\boldsymbol{\gamma}_i^0 = (\boldsymbol{X}_i^T \boldsymbol{X}_i)^{-1} \boldsymbol{X}_i^T \boldsymbol{Y}_i, i = 1, \cdots, n$*, which is a consistent estimate. Then, let initial estimates* $\boldsymbol{\delta}_{ij}^0 = \boldsymbol{\gamma}_i^0 - \boldsymbol{\gamma}_j^0$ *in* $\boldsymbol{\delta}^0$ *and* $\boldsymbol{\upsilon}^0 = \mathbf{0}$*.*

Remark 3.2 *To compute the solution path of* $\boldsymbol{\gamma}$ *against* λ*, we consider a grid of* λ *values with* $\lambda_{min} = \lambda_0 < \lambda_1 < \cdots < \lambda_K = \lambda_{max}$*, where* $0 \leq \lambda_{min} < \lambda_{max} < \infty$*. Given a* λ *value in* $[\lambda_{min}, \lambda_{max}]$*, we can compute* $\hat{\boldsymbol{\gamma}}(\lambda)$ *given in (3.6) by using ADMM algorithm. Referring to [61], a warm start and continuation strategy is used for updating the solutions. Specifically, we compute* $\hat{\boldsymbol{\gamma}}(\lambda_0)$ *by using* $\boldsymbol{\gamma}^0$ *as the initial value, then* $\hat{\boldsymbol{\gamma}}(\lambda_k)$ *by using* $\hat{\boldsymbol{\gamma}}(\lambda_{k-1})$ *as the initial value* $(k = 1, \cdots, K)$*.*

Figure 3.1 illustrates the solution path for the estimates of B-spline coefficients $(\hat{\gamma}_{31}(\lambda), \ldots, \hat{\gamma}_{3n}(\lambda))$ against λ. It is computed on a grid of λ values in interval $[\lambda_{\min}, \lambda_{\max}]$. From Figure 3.1, we observe that when λ is very small, too many subgroups are identified.

With λ value increasing, the estimated number of subgroups decreases, then becomes to 1 for a large λ value. If the actual number of subgroups is given ($K = 3$), based on the solution path, we can select a λ between 0.6 and 0.8 as the tuning parameter, where $\hat{K}$ equals the true number of subgroups; otherwise, BIC or the Calinski-Harabasz index is used to decide the optimal tuning parameter λ.

3.4 Theoretical Properties

In this section, we establish the theoretical properties of the proposed estimators. We first introduce some notations. Let $\boldsymbol{\beta}(t) = (\beta_1(t), \ldots, \beta_n(t))^T$ with $\beta_i(t)$ being the function of the ith subject, and $\boldsymbol{\alpha}(t) = (\alpha_1(t), \ldots, \alpha_K(t))^T$ with $\alpha_k(t)$ being the common function for the kth subgroup. For any square integrable function $g(t)$ on the compact support $\mathbb{T}$, denote its L_2 norm by $\|g\|_2 = \left\{\int_{\mathbb{T}} g(t)^2 dt\right\}^{1/2}$ and squared L_2 norm by $\|g\|_2^2 = \int_{\mathbb{T}} g(t)^2 dt$. Then, for a vector valued function $\mathbf{g}(t) = (g_1(t), \ldots, g_L(t))^T$, its squared L_2 norm is defined as $\|\mathbf{g}\|_2^2 = \sum_{l=1}^{L} \|g_l\|_2^2$. Let $b = \min_{k \neq k'} \|\alpha_k - \alpha_{k'}\|_2$ be the minimum distance between smoothing functions α_k and $\alpha_{k'}$ from any two clusters.

We also give the definitions for notations $O(\cdot)$ and $O_p(\cdot)$ as follows. If $\{x_n\}_1^\infty$ is any real sequence, $\{b_n\}_1^\infty$ is a sequence of positive real numbers, and there exists a constant $C_* < \infty$ such that $|x_n|/b_n \leq C_*$ for all n, we say that x_n is at most of the order of magnitude of b_n, and write $x_n = O(b_n)$. If, for any $\varepsilon > 0$, there exists $C_\varepsilon < \infty$ such that the stochastic sequence $\{X_n\}_1^\infty$ satisfies $\sup_n P(|X_n| > C_\varepsilon) < \varepsilon$, we write $X_n = O_p(1)$. If $\{Y_n\}_1^\infty$ is another sequence, either stochastic or nonstochastic, and $X_n/Y_n = O_p(1)$, we say that $X_n = O_p(Y_n)$, or in words, X_n is at most of order Y_n in probability.

Asymptotic properties

Definition. A random sequence $\{\xi_k, k \geq 1\}$ is said to be α-mixing if the α-mixing coefficient

$$\alpha(s) \stackrel{\text{def}}{=} \sup_{k \geq 1} \sup\{|P(A \cap B) - P(A)P(B)| : A \in \mathcal{F}_{s+k}^{\infty}, B \in \mathcal{F}_1^k\}$$

converges to 0 as $s \to \infty$, where $\mathcal{F}_a^b$ is the σ algebra generated by $\xi_a, \xi_{a+1}, \ldots, \xi_b$.

Among various mixing conditions used in the literature, the α-mixing is reasonably weak and is known to be fulfilled by many stochastic processes including many time series models. For instance, [31] derived the conditions under which a linear process is α-mixing. The linear autoregressive and the bilinear time series models are strongly mixing with mixing coefficients decaying exponentially under very mild assumptions, see the page 99 of [21] for more details. We refer to [45, 11] and references therein for more discussions on the α-mixing condition.

We denote by $C^{(r)} = \left\{\phi | \phi^{(r)} \in C(\mathbb{T})\right\}$ the space of the rth order smooth functions on the compact support $\mathbb{T}$ such that their rth order derivatives belong to $C(\mathbb{T})$, which is the class of all continuous functions on $\mathbb{T}$.

Regularity Conditions:

(C1) The observation time points t_{ij}, $i = 1, \ldots, n$, $j = 1, \ldots, m_i$, are chosen independently from a distribution $F(\cdot)$ with the density $f(\cdot)$. Moreover, the density function $f(t)$ is uniformly bounded away from 0 and infinity on its compact support $\mathbb{T}$. Without loss of generality, we assume $\mathbb{T} = [a_0, b_0]$.

(C2) There exists a positive constant M such that $E(\varepsilon(t)^4) \leq M$ for all $t \in \mathbb{T}$. In addition,

the random sequence $\{\varepsilon_{ij}\}$ for each i satisfies α-mixing condition with the α-mixing coefficient satisfying $\alpha(s) \leq C^* s^{-\alpha}$ for $\alpha > \frac{2+\kappa_0}{1-\kappa_0}$, where $0 < \kappa_0 < 1$, and C^* is a positive constant with $0 < C^* < \infty$.

(C3) The functions $\beta_i(\cdot) \in C^{(r)}$, for $i = 1, \ldots, n$.

(C4) The spline knot sequences $\{\zeta_l\}_{l=0}^{J+1}$ have bounded mesh ratio. That is, for some positive constant C_{01},

$$\frac{\max_{0 \leq l \leq J} |\zeta_{l+1} - \zeta_l|}{\min_{0 \leq l \leq J} |\zeta_{l+1} - \zeta_l|} \leq C_{01}.$$

(C5) There exist positive constants $0 < C_1 < C_2 < \infty$ such that the eigenvalues of $\boldsymbol{\Sigma}$=diag$(\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_n)$ and $\boldsymbol{V}$=diag$(\boldsymbol{V}_1, \ldots, \boldsymbol{V}_n)$ lie between C_1 and C_2.

Condition (C1) is identical to condition (C1) in [40] and assumption (A1) in [70]. This condition ensures that the observation time points are randomly scattered and it can be modified or weakened according to Remarks 3.1 and 3.2 in [40]. Condition (C2) is a standard requirement for moments and the mixing coefficient for an α-mixing process as assumed in [45] and [11]. This condition allows the errors to be weakly dependent. Many linear and nonlinear time series models like the linear autoregressive and the bilinear time series models are strongly mixing with the mixing coefficients decaying exponentially, see [21] (page 99) for more details. Conditions (C3)-(C4) are frequently assumed in the spline approximation literature; see for example [100, 89, 58]. The smoothness condition on $\beta_i(\cdot)$ given by Condition (C3) determines the rate of the approximation error of the spline estimator $\hat{\beta}_i(\cdot)$. Condition (C4) ensures that the knot sequence has a bounded mesh ratio; that is, the knots are quasi-uniform. Condition (C5) is commonly used in the literature related to longitudinal data, such as in [41, 62] and the references therein.

Let the nonparametric function subspace $M_{\mathcal{G}}^{\beta}$ corresponding to the group partition be $M_{\mathcal{G}}^{\beta} = \{\beta(\cdot) : \beta_i(\cdot) = \alpha_k(\cdot),\ \beta_i(\cdot) \in C^{(r)}, \text{for } i \in \mathcal{G}_k, 1 \leq k \leq K\}$, while the subspace $M_{\mathcal{G}}^{\gamma}$ of B-spline coefficients corresponding to the group partition is denoted by $M_{\mathcal{G}}^{\gamma} = \{\boldsymbol{\gamma} : \boldsymbol{\gamma}_i = \boldsymbol{\theta}_k,\ \boldsymbol{\gamma}_i \in R^S,\ \text{for } i \in \mathcal{G}_k, 1 \leq k \leq K\}$, where $\boldsymbol{\theta}_k$ is the common B-spline coefficients in the kth subgroup. By using the proposed method, we have $\hat{\boldsymbol{\gamma}} = (\hat{\boldsymbol{\gamma}}_1^T, \ldots, \hat{\boldsymbol{\gamma}}_n^T)^T$, where $\hat{\boldsymbol{\gamma}}_i$ is the estimated B-spline coefficient for subject i with $\hat{\boldsymbol{\gamma}}_i = \hat{\boldsymbol{\theta}}_k$ for all $i \in \hat{\mathcal{G}}_k$. Then, the estimated function for each i is

$$\hat{\beta}_i(t) = \boldsymbol{B}(t)^T \hat{\boldsymbol{\gamma}}_i, \tag{3.14}$$

for any $t \in \mathbb{T}$. Let $\hat{\boldsymbol{\alpha}}^{or}(t) = (\hat{\alpha}_1^{or}(t), \ldots, \hat{\alpha}_K^{or}(t))$, where $\hat{\alpha}_k^{or}(t)$ is the estimated common function for group $\mathcal{G}_k$ by assuming that the true memberships are known.

Theorem 3.1 *Suppose conditions (C1)-(C5) hold, and for any fixed K, if $J = O(N_0^{\varsigma})$ with $0 < \varsigma < 1$, the oracle estimator $\hat{\boldsymbol{\alpha}}^{or}$ satisfies $\|\hat{\boldsymbol{\alpha}}^{or} - \boldsymbol{\alpha}\|_2^2 = O_p(J/N_0 + J^{-2r})$, where $N_0 = \min_{1 \leq k \leq K} N_k$ and $N_k = \sum_{i \in \mathcal{G}_k} m_i$.*

It is worth noting that the convergence rate given in Theorem 3.1 consists of two parts, which are the approximation error of order J^{-2r} and the estimation error of order J/N_0. We can see that the increase of J leads to smaller approximation error but larger estimation error, whereas the decrease of J leads to larger approximation error but smaller estimation error, i.e., there is a trade-off between the bias and variance. By letting $J/N_0 = J^{-2r}$, we can obtain the optimal order of J which is $N_0^{1/(2r+1)}$. Plugging it into the convergence rate, it follows that $\|\hat{\boldsymbol{\alpha}}^{or} - \boldsymbol{\alpha}\|_2^2 = O_p(J/N_0 + J^{-2r}) = O_p\left(N_0^{-2r/(2r+1)}\right)$, which reaches the minimax convergence rate for spline regression.

The following Theorem 3.2 gives the convergence rate of the estimated function $\hat{\beta}_i(t)$ in (3.14) for each i.

Theorem 3.2 *Suppose conditions (C1)-(C5) hold, if there exists a constant $C > 0$ such that $Cb \geq \tau\lambda$ and $J = O(m_{(n)}^{\varsigma})$ with $0 < \varsigma < 1$, then, for each i, $\|\hat{\beta}_i - \beta_i\|_2^2 = O_p(J/m_{(n)} + J^{-2r})$, where $m_{(n)} = \min_{1 \leq i \leq n} m_i$.*

Theorem 3.3 *Assume $\hat{\mathcal{G}}$ and $\mathcal{G}_0$ respectively be the estimated and true subgroup membership. Under the same conditions in Theorem 3.2, we have $P\left(\hat{\mathcal{G}} = \mathcal{G}_0\right) \to 1$ as $m_{(n)} \to \infty$.*

Theorem 3.3 gives the model selection consistency result for the penalized method. Thus, given the estimated subgroup membership, we may write $\hat{\boldsymbol{\alpha}}(t) = (\hat{\alpha}_1(t), \ldots, \hat{\alpha}_K(t))^T$ for any given $t \in \mathbb{T}$, and the following theorem holds.

Theorem 3.4 *Under the same conditions in Theorem 3.3. If $J/m_{(n)}^{1/(2r+1)} \to \infty$, we have*

$$\mathrm{Var}\left(\hat{\boldsymbol{\alpha}}(t)\right)^{-1/2} \left(\hat{\boldsymbol{\alpha}}(t) - \boldsymbol{\alpha}(t)\right) \xrightarrow{d} N(\mathbf{0}, \boldsymbol{I}_K),$$

where $\boldsymbol{I}_K$ is a K-dimensional identity matrix and $\mathrm{Var}\left(\hat{\boldsymbol{\alpha}}(t)\right)$ is given in (A.15) of Appendix A. In particular,

$$\mathrm{Var}\left(\hat{\alpha}_k(t)\right)^{-1/2} \left(\hat{\alpha}_k(t) - \alpha_k(t)\right) \xrightarrow{d} N(0, 1)$$

for $k = 1, \ldots, K$, where $\mathrm{Var}\left(\hat{\alpha}_k(t)\right) = \boldsymbol{e}_k^T \mathrm{Var}\left(\hat{\boldsymbol{\alpha}}(t)\right) \boldsymbol{e}_k$, and $\boldsymbol{e}_k$ is the K-dimensional vector with the kth element taken to be 1 and 0 elsewhere.

We can use the asymptotic distribution established in Theorem 3.4 to construct pointwise confidence intervals of the functional curve for each subgroup.

3.5 Simulation Studies

In this section, we investigate the performance of our proposed approach by conducting simulation studies. Balanced and unbalanced data are both considered.

Two different criteria are used to select the optimal tuning parameter. One is the modified Bayes Information Criterion (BIC) [88] for high-dimensional data settings by minimizing

$$\text{BIC}(\lambda) = \log\left[\sum_{i=1}^{n}(\boldsymbol{Y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\gamma}}_i(\lambda))^T\boldsymbol{R}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\hat{\boldsymbol{\gamma}}_i(\lambda))/N\right] + C_n\frac{\log N}{N}(\hat{K}(\lambda)S), \quad (3.15)$$

where C_n is a positive number which can depend on n and $N = \sum_{i=1}^{n} m_i$. Following [60], we let $C_n = c\ \log(\log(nS))$, where c is a positive constant, and we choose $c = 0.6$. The other criterion is the Calinski-Harabasz index [12] by maximizing

$$\text{CH}(\lambda) = \frac{B_{\hat{K}(\lambda)}/(\hat{K}(\lambda) - 1)}{W_{\hat{K}(\lambda)}/(n - \hat{K}(\lambda))}, \quad (3.16)$$

where $B_{\hat{K}(\lambda)}$ and $W_{\hat{K}(\lambda)}$ are the between and within group sum of square errors of the estimated subgroups given a λ value. We apply this index to the initial value $\boldsymbol{\gamma}_i^0$'s, which are the ordinary least squares estimates of (3.4) given in Remark 3.1. Note that $\text{CH}(\lambda)$ is not defined for $\hat{K}(\lambda) = 1$. Based on these criteria, we can select the optimal λ and obtain the corresponding group membership. Here we use fixed values for ϑ and τ in ADMM algorithm: $\vartheta = 1$ and $\tau = 3$.

To evaluate the accuracy of the clustering results, we provide three measures: Rand Index (RI) [72], Normalized Mutual Information (NMI) [87] and accuracy percentage (%). The accuracy percentage (%) is defined as the proportion of subjects that are correctly identified. These three values are between 0 and 1, with higher values indicating better

performance.

3.5.1 Two Subgroups Example

We simulate data from the heterogeneous model with two subgroups

$$Y_{ij} = \beta_i(t_{ij}) + \varepsilon_{ij}, \quad i = 1, \ldots, n, \ j = 1, \ldots, m_i,$$

where $\beta_i(t) = \alpha_1(t)$ if $i \in \mathcal{G}_1$ and $\beta_i(t) = \alpha_2(t)$ if $i \in \mathcal{G}_2$.

We first consider balanced data. In this situation, we have $m_i = T$ for all i's. The time points t_{ij}'s are chosen equally spaced on $[0, 1.2]$. The error term $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \ldots, \varepsilon_{iT})^T$ is generated from $N(\mathbf{0}, \boldsymbol{\Sigma}_E)$, in which $\boldsymbol{\Sigma}_E$ has AR(1) covariance structure with $\rho = 0.3$ and $\sigma = 0.5$. Four setups of (n, T) are considered: $\{n = 100, T = 20\}$, $\{n = 100, T = 50\}$, $\{n = 150, T = 20\}$ and $\{n = 150, T = 50\}$. Moreover, to choose $\{\alpha_1(t), \alpha_2(t)\}$, we also consider three different cases by increasing the distance between the two functions from close to middle, then to far, which are shown below:

$$\text{Close} \begin{cases} \alpha_1(t) = -0.5t^2 + 1.25t, \\ \alpha_2(t) = -t^2 + 2.5t, \end{cases} \qquad \text{Middle} \begin{cases} \alpha_1(t) = -0.5t^2 + 1.25t, \\ \alpha_2(t) = -1.3t^2 + 3.25t, \end{cases}$$

$$\text{Far} \begin{cases} \alpha_1(t) = -0.5t^2 + 1.25t, \\ \alpha_2(t) = -2.5t^2 + 6.25t. \end{cases}$$

Figure 3.2 shows the true functions (black line) and simulated trajectories (blue line and red line) of the three distance cases, respectively, based on one sample with $n = 100$, $T = 20$ for balanced data. We can see that there are a lot of overlaps in Close and Middle cases, especially in Close case, it looks more like one group.

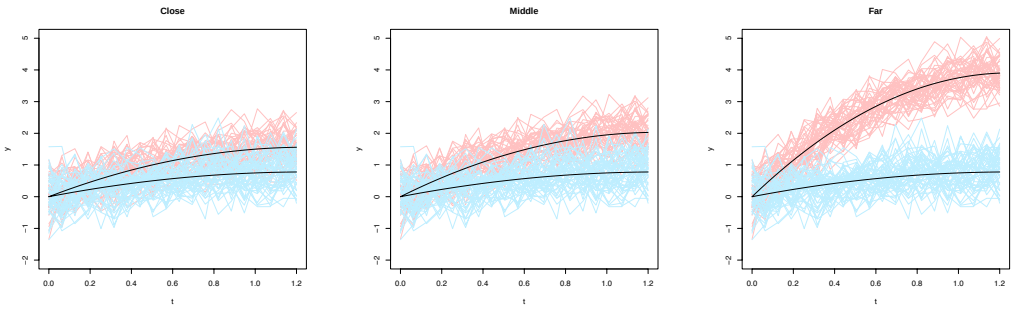


Figure 3.2: The black lines represent the true functions, while the red and blue lines represent the simulated trajectories of the corresponding subgroups under one replication when $n = 100,\ T = 20$ for balanced data in Two Subgroups Example. The distance between the true functions increases from close, to middle, to far.

The unbalanced data is based on the balanced data setting. However, we randomly allow 50% of the subjects to miss either 30% or 40% or 50% of time points. Next, we conduct simulations to illustrate the performance of our proposed method. 100 replications are taken here. Quadratic splines with one interior knot are used to approximate the nonparametric components. The quadratic splines are B-splines with order $r = 3$. As the order of the B-splines increases, the estimated curve becomes smoother. The quadratic splines can yield smooth enough curves while preventing over-smoothing. Based on the convergence rate given in Theorem 4.2 on page 8, the optimal order of the number of interior knots J is $m_{(n)}^{\frac{1}{2r+1}}$ by letting $J/m_{(n)} = J^{-2r}$. We choose $J = \left\lfloor m_{(n)}^{\frac{1}{2r+1}} \right\rfloor = \left\lfloor m_{(n)}^{\frac{1}{7}} \right\rfloor$, where $\lfloor a \rfloor$ denotes the largest integer no bigger than a. Then $J = \left\lfloor m_{(n)}^{\frac{1}{7}} \right\rfloor = 1$ when $m_{(n)} = 10, 20, 25, 50$ in our simulation settings.

Table 3.1 not only reports the summary measurements of the estimated number of subgroups $\hat{K}$ (sample mean, median, per, where per is the percentage of $\hat{K}$ equaling to the true number of subgroups), but also the summary measurements of the clustering

accuracy (average values of RI, NMI, %) by using different model selection criteria (BIC, CH) under different setups of $\{n,\ T\}$ when the distance between functions increases (Close, Middle, Far). Balanced and unbalanced data are both included. Note that when calculating RI, NMI and %, we only include the replications with $\hat{K}$ equaling to the true number of subgroup ($\hat{K} = 2$).

From Table 3.1, we can see that both BIC and CH criteria perform well and give the similar results for most of the cases. When T increases, the summary measurements of $\hat{K}$ (mean, median, per) and accuracy measurements (RI, NMI, %) both increase. In details, the mean of $\hat{K}$ gets close to 2 and median $\hat{K}$ becomes to 2, where 2 is the true number of subgroups, while the accuracy measurements (RI, NMI, %) are close to 1 or even become to 1 for both balanced and unbalanced data, which indicates good clustering results. What's more, with the distance between the true functions getting larger, it is much easier to correctly identify the subgroups. Accordingly, we observe that the mean and median of $\hat{K}$ become to 2, while the RI, NMI and % become to 1 when the distance is sufficiently large (Far case). On the contrary, in Close case, since the trajectories of the two subgroups in Figure 3.2 show a lot of overlaps, it is more difficult to identify the subgroups, which results in the low percentage (per) of correctly selecting the number of subgroups when $T = 20$. Under this case, if we can cluster the subjects into two subgroups, BIC criterion presents higher accuracy performance in group membership. However, if T increases to 50, all the measurements become much better and it is more likely to correctly identify the subgroups. Compared with unbalanced data, balanced data shows slightly better results.

			Balanced						Unbalanced					
Functions	setting	criterion	mean	median	per	RI	NMI	%	mean	median	per	RI	NMI	%
Close	n=100, T=20	BIC	1.34	1.00	0.20	0.9089	0.7459	0.9515	1.43	1.00	0.08	0.9015	0.7289	0.9475
		CH	1.55	1.00	0.35	0.8729	0.6818	0.9223	1.73	1.50	0.28	0.7652	0.4861	0.8304
	n=100, T=50	BIC	1.98	2.00	0.98	0.9953	0.9836	0.9977	1.97	2.00	0.97	0.9855	0.9510	0.9927
		CH	1.98	2.00	0.98	0.9953	0.9834	0.9977	1.97	2.00	0.97	0.9855	0.9510	0.9927
	n=150, T=20	BIC	1.45	1.00	0.23	0.9271	0.7820	0.9620	1.50	1.00	0.08	0.8868	0.6855	0.9400
		CH	1.57	1.00	0.31	0.8746	0.6876	0.9178	1.72	1.00	0.24	0.6563	0.2940	0.7131
	n=150, T=50	BIC	2.00	2.00	1.00	0.9923	0.9719	0.9961	2.00	2.00	1.00	0.9855	0.9484	0.9927
		CH	2.00	2.00	1.00	0.9922	0.9717	0.9961	1.98	2.00	0.98	0.9852	0.9472	0.9925
Middle	n=100, T=20	BIC	2.00	2.00	1.00	0.9960	0.9859	0.9980	2.00	2.00	1.00	0.9903	0.9664	0.9951
		CH	2.00	2.00	1.00	0.9952	0.9830	0.9976	2.00	2.00	1.00	0.9901	0.9655	0.9950
	n=100, T=50	BIC	2.00	2.00	1.00	0.9998	0.9993	0.9999	2.00	2.00	1.00	0.9996	0.9985	0.9998
		CH	2.00	2.00	1.00	0.9998	0.9993	0.9999	2.00	2.00	1.00	0.9996	0.9985	0.9998
	n=150, T=20	BIC	2.00	2.00	1.00	0.9967	0.9870	0.9983	2.00	2.00	1.00	0.9874	0.9535	0.9937
		CH	2.00	2.00	1.00	0.9967	0.9870	0.9983	2.00	2.00	1.00	0.9865	0.9503	0.9932
	n=150, T=50	BIC	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	0.9999	0.9995	0.9999
		CH	2.00	2.00	1.00	0.9999	0.9995	0.9999	2.00	2.00	1.00	0.9997	0.9990	0.9999
Far	n=100, T=20	BIC	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	1.0000	1.0000	1.0000
		CH	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	1.0000	1.0000	1.0000
	n=100, T=50	BIC	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	1.0000	1.0000	1.0000
		CH	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	1.0000	1.0000	1.0000
	n=150, T=20	BIC	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	1.0000	1.0000	1.0000
		CH	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	1.0000	1.0000	1.0000
	n=150, T=50	BIC	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	1.0000	1.0000	1.0000
		CH	2.00	2.00	1.00	1.0000	1.0000	1.0000	2.00	2.00	1.00	1.0000	1.0000	1.0000

Table 3.1: The sample mean and median of $\hat{K}$, the percentage (per) of $\hat{K}$ equaling to the true number of subgroups, the Rand Index (RI), Normalized mutual information (NMI), and accuracy percentage (%) equaling the proportion of subjects that are identified correctly under BIC and CH criteria based on 100 realizations in Two Subgroups Example. Balanced and unbalanced data are both included under different $\{n,\ T\}$ setups and function distances.

Furthermore, to study the estimation accuracy, we calculate the square root of the mean squared error (RMSE) of the estimated function in each subgroup only when $\hat{K}$

equals the true number of subgroups K. In the kth subgroup, we use the formula below to find the corresponding RMSE of the estimated function $\hat{\alpha}_k(t)$ (RMSE$_k$):

$$\text{RMSE}_k = \sqrt{\frac{1}{H}\sum_{h=1}^{H}[\hat{\alpha}_k(t_h) - \alpha_k(t_h)]^2} = \sqrt{\frac{1}{H}\sum_{h=1}^{H}[\boldsymbol{B}_{(k)}(t_h)^T\hat{\boldsymbol{\gamma}}_{(k)} - \alpha_k(t_h)]^2},\ k = 1, \ldots, K,$$

where $\boldsymbol{B}_{(k)}(t)$ is the B-spline basis vector of the kth subgroup, $\hat{\boldsymbol{\gamma}}_{(k)}$ is the corresponding estimated B-spline coefficient after refitting model (3.4) and $\{t_1, \cdots, t_H\}$ is a grid of equally spaced points spanning the original time range $[0, 1.2]$ with $H = 50$.

For oracle (Oracle) method, we use the true group membership to calculate RMSE. As shown in Table 3.2, the RMSE values under different model selection criteria (BIC, CH) and $\{n,\ T\}$ setups are comparable to those of the oracle ones for almost all cases.

Lastly, the estimated nonparametric curves $\hat{\alpha}_k(t)$ (blue, red lines) and true curves $\alpha_k(t)$ (black line) of the two subgroups for balanced data among the 100 replications are plotted in Figure 3.3. Notice that we only plot the replications when the estimated number of subgroups equals the true number of subgroups. On each row, from left to right, it represents the Close, Middle, and Far cases with same setting of $\{n,\ T\}$ respectively. Then either n or T is increased compared to the first row. Given each column, it is obvious that the bands consisting by red or blue lines becomes narrower as T or n increases. Besides, no matter for which setups of $\{n,\ T\}$, the estimated curves are very close to the true ones for all distance cases.

	Close				Middle				Far			
	Balanced		Unbalanced		Balanced		Unbalanced		Balanced		Unbalanced	
n=100, T=20	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$
Oracle	0.0363	0.0369	0.0385	0.0400	0.0363	0.0369	0.0385	0.0400	0.0363	0.0369	0.0385	0.0400
BIC	0.0512	0.0467	0.0570	0.0461	0.0365	0.0377	0.0394	0.0408	0.0363	0.0369	0.0385	0.0400
CH	0.0626	0.0610	0.1075	0.1128	0.0364	0.0375	0.0395	0.0407	0.0363	0.0369	0.0385	0.0400
n=100, T=50	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$
Oracle	0.0247	0.0235	0.0261	0.0243	0.0247	0.0235	0.0261	0.0243	0.0247	0.0235	0.0261	0.0243
BIC	0.0253	0.0236	0.0276	0.0252	0.0248	0.0234	0.0262	0.0243	0.0247	0.0235	0.0261	0.0243
CH	0.0253	0.0237	0.0276	0.0253	0.0248	0.0234	0.0262	0.0243	0.0247	0.0235	0.0261	0.0243
n=150, T=20	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$
Oracle	0.0314	0.0281	0.0337	0.0302	0.0314	0.0281	0.0337	0.0302	0.0314	0.0281	0.0337	0.0302
BIC	0.0387	0.0403	0.0432	0.0435	0.0317	0.0282	0.0344	0.0302	0.0314	0.0281	0.0337	0.0302
CH	0.0602	0.0606	0.1762	0.1698	0.0317	0.0281	0.0347	0.0303	0.0314	0.0281	0.0337	0.0302
n=150, T=50	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$
Oracle	0.0199	0.0212	0.0214	0.0221	0.0199	0.0212	0.0214	0.0221	0.0199	0.0212	0.0214	0.0221
BIC	0.0204	0.0217	0.0221	0.0228	0.0199	0.0212	0.0214	0.0221	0.0199	0.0212	0.0214	0.0221
CH	0.0204	0.0217	0.0220	0.0227	0.0199	0.0213	0.0214	0.0221	0.0199	0.0212	0.0214	0.0221

Table 3.2: The mean of square root of the MSE (RMSE) for the estimated functions $\hat{\alpha}_1(t), \hat{\alpha}_2(t)$ under BIC, CH and Oracle methods in Two Subgroups Example.

To further illustrate the performance of our proposed method in unbalanced data, we generate data with $m_i \sim \text{Uniform}\{5, 6, ..., 20\}, i = 1, ..., n,\ n = 100, 1000$, and keep other simulation settings the same as before. We report the numerical results for Middle and Far cases in Table 3.3 and 3.4, as the curves from different subgroups in the Close case are too close to be separated based on the previous simulation results; see Table 3.1. Table 3.3 shows that the median value of $\hat{K}$ equals to the true number of subgroups, which is 2. As the mean functions of the subgroups become more separated (from Middle to Far case),

the mean value of $\hat{K}$ gets closer to 2, and the average values of RI, NMI and the accuracy percentage (%) approach to 1. Moreover, Table 3.4 shows that the RMSE values of the estimated functions by our method are comparable to those of the oracle ones obtained by assuming that the true memberships are known. These results demonstrate that our proposed method performs well for clustering heterogeneous trajectories from unbalanced data.

Functions	setting	criterion	mean	median	per	RI	NMI	%
Middle	n=100	BIC	2.11	2.00	0.91	0.9526	0.8526	0.9756
		CH	2.07	2.00	0.94	0.9542	0.8563	0.9765
	n=1000	BIC	2.02	2.00	0.98	0.9483	0.8293	0.9734
		CH	2.01	2.00	0.99	0.9491	0.8318	0.9738
Far	n=100	BIC	2.00	2.00	1.00	0.9962	0.9865	0.9981
		CH	2.00	2.00	1.00	0.9960	0.9857	0.9980
	n=1000	BIC	2.00	2.00	1.00	0.9963	0.9830	0.9982
		CH	2.00	2.00	1.00	0.9962	0.9828	0.9981

Table 3.3: The sample mean and median of $\hat{K}$, the percentage (per) of $\hat{K}$ equaling to the true number of subgroups, the Rand Index (RI), Normalized mutual information (NMI), and accuracy percentage (%) equaling the proportion of subjects that are identified correctly under BIC and CH criteria based on 100 realizations with $m_i \sim \text{Uniform}\{5, 6, \ldots, 20\}$ in Two Subgroups Example.

3.5.2 Three Subgroups Example

We simulate data from the heterogeneous model with three subgroups

$$Y_{ij} = \beta_i(t_{ij}) + \varepsilon_{ij}, \quad i = 1, \ldots, n, \ j = 1, \ldots, m_i,$$

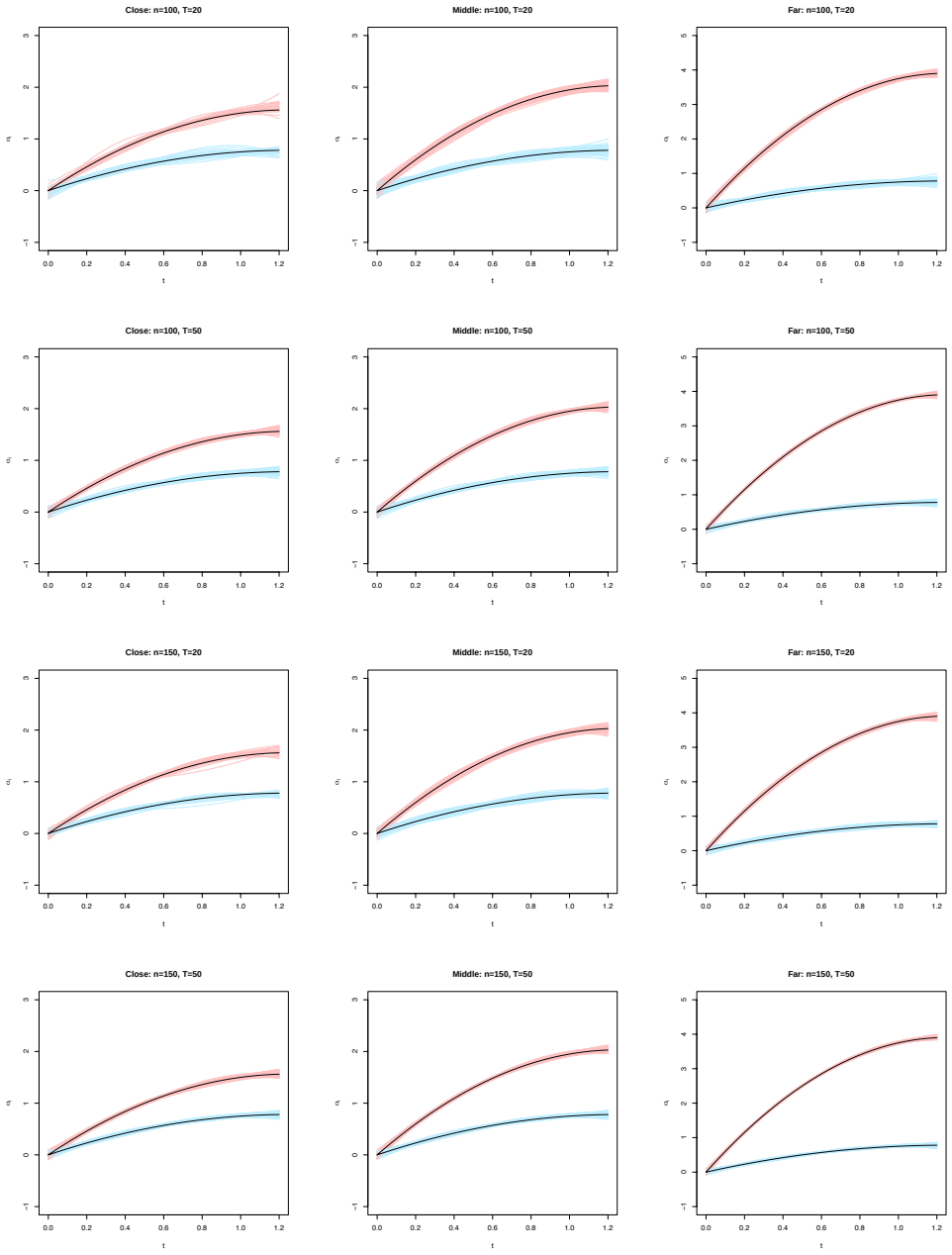


Figure 3.3: The black lines represent the true functions, while the red and blue lines are the corresponding fitted curves for the estimated subgroups by using BIC criterion when $\hat{K} = 2$ among the 100 replications for balanced data in Two Subgroups Example. On each row, from left to right, it corresponds to close, middle, and far cases with the same setting of $\{n,\ T\}$.

	Middle				**Far**			
	n=100		n=1000		n=100		n=1000	
	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$
Oracle	0.0427	0.0399	0.0127	0.0131	0.0427	0.0399	0.0127	0.0131
BIC	0.0449	0.0434	0.0165	0.0154	0.0424	0.0401	0.0131	0.0131
CH	0.0446	0.0443	0.0163	0.0155	0.0424	0.0401	0.0131	0.0132

Table 3.4: The mean of square root of the MSE (RMSE) for the estimated functions $\hat{\alpha}_1(t), \hat{\alpha}_2(t)$ under BIC, CH and Oracle methods with $m_i \sim$ Uniform $\{5, 6, \ldots, 20\}$ in Two Subgroups Example.

where $\beta_i(t) = \alpha_1(t)$ if $i \in \mathcal{G}_1$, $\beta_i(t) = \alpha_2(t)$ if $i \in \mathcal{G}_2$ and $\beta_i(t) = \alpha_3(t)$ if $i \in \mathcal{G}_3$. We generate data in the same way as that in Two Subgroups Example. The three functions for Close, Middle and Far cases are chosen as:

$$\text{Close} \begin{cases} \alpha_1(t) = -0.6t^2 + 1.5t, \\ \alpha_2(t) = -1.3t^2 + 3.25t + 0.2, \\ \alpha_3(t) = -2.2t^2 + 5.5t + 0.1, \end{cases} \qquad \text{Middle} \begin{cases} \alpha_1(t) = -0.4t^2 + t, \\ \alpha_2(t) = -1.3t^2 + 3.25t + 0.2, \\ \alpha_3(t) = -2.4t^2 + 6t + 0.1, \end{cases}$$

$$\text{Far} \begin{cases} \alpha_1(t) = -0.3t^2 + 0.75t, \\ \alpha_2(t) = -4t^2 + 10t + 0.2, \\ \alpha_3(t) = -8.5t^2 + 21.25t + 0.3. \end{cases}$$

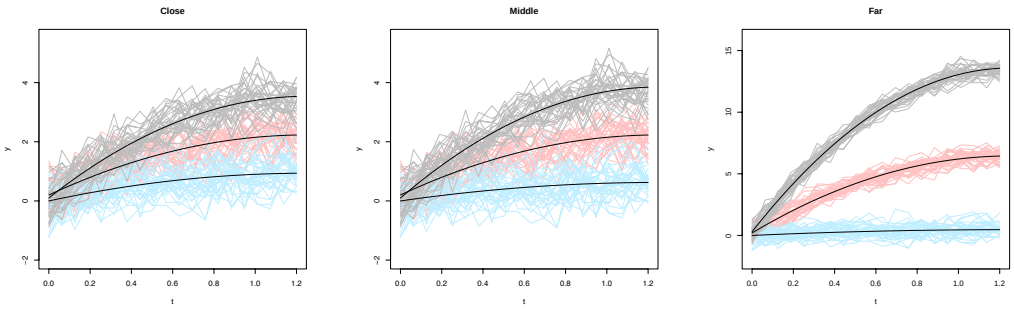


Figure 3.4: The black lines represent the true functions, while the grey, red and blue lines represent the simulated trajectories of the corresponding subgroups under one replication when $n = 100,\ T = 20$ for balanced data. The distance between the true functions increases from close, to middle, to far.

Figure 3.4 displays the true functions and corresponding trajectories of the three subgroups under one sample with $n = 100,\ T = 20$ for balanced data. From left to right, the distance between true functions gets larger. We next conduct simulations to do subgroup analysis by using our method. Table 3.5, based on 100 realizations, presents the mean, median, per of $\hat{K}$ and the average values of RI, NMI, % for all setups under BIC and CH criteria, respectively. In this table, we observe that the performance for balanced data is better than the corresponding unbalanced data. BIC and CH criteria are consistent due to similar results. When T or the distance between true functions increases, the values of RI, NMI, and % become larger. Moreover, to demonstrate the estimation accuracy, Table 3.6 lists the average values of RMSE for the estimated functions $\hat{\alpha}_k(t)$ $(k = 1, 2, 3)$ when $\hat{K}$ equals 3, while Figure 3.5 shows the estimated nonparametric curves (grey, red, blue lines) and true curves (black lines). From Table 3.6, it can be seen that the RMSE of $\hat{\alpha}_k(t)$'s are close to those of the oracle estimators. In Figure 3.5, we also observe that the estimated curves are very close to the true curves. And the bands formed by the corresponding

estimated curves become narrower as n or T increases.

			Balanced						**Unbalanced**					
Functions	setting	criterion	mean	median	per	RI	NMI	%	mean	median	per	RI	NMI	%
Close	n=100, T=20	BIC	3.00	3.00	1.00	0.9962	0.9882	0.9971	3.00	3.00	1.00	0.9870	0.9603	0.9899
		CH	2.79	3.00	0.79	0.9965	0.9891	0.9973	2.52	3.00	0.52	0.9887	0.9665	0.9915
	n=100, T=50	BIC	2.98	3.00	0.98	1.0000	1.0000	1.0000	2.98	3.00	0.98	0.9998	0.9993	0.9998
		CH	2.98	3.00	0.98	1.0000	1.0000	1.0000	2.98	3.00	0.98	0.9996	0.9988	0.9997
	n=150, T=20	BIC	3.00	3.00	1.00	0.9982	0.9939	0.9987	3.00	3.00	1.00	0.9910	0.9709	0.9931
		CH	2.91	3.00	0.91	0.9982	0.9940	0.9987	2.69	3.00	0.69	0.9903	0.9691	0.9928
	n=150, T=50	BIC	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	0.9999	0.9997	0.9999
		CH	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	0.9999	0.9997	0.9999
Middle	n=100, T=20	BIC	3.00	3.00	1.00	0.9996	0.9988	0.9997	3.00	3.00	1.00	0.9965	0.9889	0.9973
		CH	3.00	3.00	1.00	0.9995	0.9983	0.9996	2.98	3.00	0.98	0.9962	0.9880	0.9970
	n=100, T=50	BIC	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
		CH	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
	n=150, T=20	BIC	3.00	3.00	1.00	0.9998	0.9994	0.9999	3.00	3.00	1.00	0.9978	0.9925	0.9983
		CH	3.00	3.00	1.00	0.9999	0.9997	0.9999	3.00	3.00	1.00	0.9978	0.9929	0.9984
	n=150, T=50	BIC	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
		CH	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
Far	n=100, T=20	BIC	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
		CH	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
	n=100, T=50	BIC	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
		CH	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
	n=150, T=20	BIC	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
		CH	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
	n=150, T=50	BIC	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000
		CH	3.00	3.00	1.00	1.0000	1.0000	1.0000	3.00	3.00	1.00	1.0000	1.0000	1.0000

Table 3.5: The sample mean and median of $\hat{K}$, the percentage (per) of $\hat{K}$ equaling to the true number of subgroups, the Rand Index (RI), Normalized mutual information (NMI), and accuracy percentage (%) equaling the proportion of subjects that are identified correctly under BIC and CH criteria based on 100 realizations in Three Subgroups Example. Balanced and unbalanced data are both considered under different $\{n,\ T\}$ setups and function distances.

	Close						Middle						Far					
	Balanced			Unbalanced			Balanced			Unbalanced			Balanced			Unbalanced		
n=100, T=20	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$
Oracle	0.0472	0.0430	0.0465	0.0508	0.0462	0.0498	0.0472	0.0430	0.0465	0.0508	0.0462	0.0498	0.0472	0.0430	0.0465	0.0508	0.0462	0.0498
BIC	0.0466	0.0447	0.0470	0.0513	0.0489	0.0508	0.0469	0.0432	0.0467	0.0505	0.0468	0.0500	0.0472	0.0430	0.0465	0.0508	0.0462	0.0498
CH	0.0466	0.0425	0.0474	0.0543	0.0438	0.0506	0.0469	0.0433	0.0467	0.0504	0.0462	0.0501	0.0472	0.0430	0.0465	0.0508	0.0462	0.0498
n=100, T=50	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$
Oracle	0.0300	0.0295	0.0316	0.0312	0.0318	0.0332	0.0300	0.0295	0.0316	0.0312	0.0318	0.0332	0.0300	0.0295	0.0316	0.0312	0.0318	0.0332
BIC	0.0299	0.0297	0.0317	0.0310	0.0322	0.0332	0.0300	0.0295	0.0316	0.0312	0.0318	0.0332	0.0300	0.0295	0.0316	0.0312	0.0318	0.0332
CH	0.0299	0.0297	0.0317	0.0310	0.0322	0.0332	0.0300	0.0295	0.0316	0.0312	0.0318	0.0332	0.0300	0.0295	0.0316	0.0312	0.0318	0.0332
n=150, T=20	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$
Oracle	0.0353	0.0373	0.0355	0.0382	0.0394	0.0377	0.0353	0.0373	0.0355	0.0382	0.0394	0.0377	0.0353	0.0373	0.0355	0.0382	0.0394	0.0377
BIC	0.0353	0.0376	0.0357	0.0393	0.0414	0.0384	0.0353	0.0374	0.0356	0.0383	0.0401	0.0379	0.0353	0.0373	0.0355	0.0382	0.0394	0.0377
CH	0.0345	0.0360	0.0356	0.0381	0.0405	0.0389	0.0353	0.0374	0.0355	0.0383	0.0401	0.0379	0.0353	0.0373	0.0355	0.0382	0.0394	0.0377
n=150, T=50	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$	$\hat{\alpha}_1(t)$	$\hat{\alpha}_2(t)$	$\hat{\alpha}_3(t)$
Oracle	0.0226	0.0255	0.0250	0.0251	0.0273	0.0266	0.0226	0.0255	0.0250	0.0251	0.0273	0.0266	0.0226	0.0255	0.0250	0.0251	0.0273	0.0266
BIC	0.0226	0.0255	0.0250	0.0251	0.0272	0.0266	0.0226	0.0255	0.0250	0.0251	0.0273	0.0266	0.0226	0.0255	0.0250	0.0251	0.0273	0.0266
CH	0.0226	0.0255	0.0250	0.0251	0.0273	0.0266	0.0226	0.0255	0.0250	0.0251	0.0273	0.0266	0.0226	0.0255	0.0250	0.0251	0.0273	0.0266

Table 3.6: The mean of square root of the MSE (RMSE) for the estimated functions $\hat{\alpha}_1(t), \hat{\alpha}_2(t), \hat{\alpha}_3(t)$ under BIC, CH and Oracle methods in Three Subgroups Example.

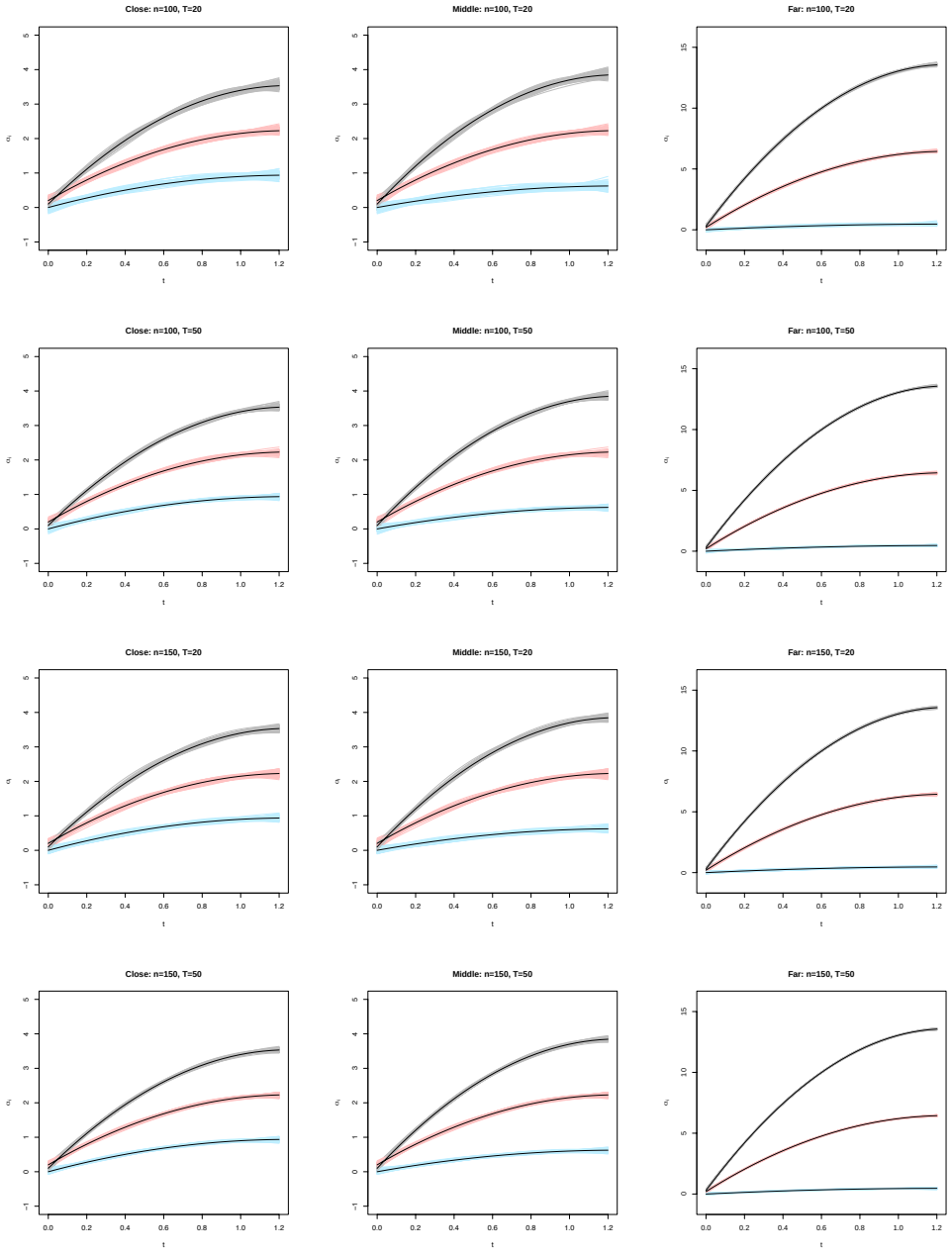


Figure 3.5: The black lines represent the true functions, while the grey, red and blue lines are the corresponding fitted curves for the estimated subgroups by using BIC criterion when $\hat{K} = 3$ among the 100 replications for balanced data in Three Subgroups Example. On each row, from left to right, it corresponds to close, middle, and far cases with the same setting of $\{n,\ T\}$.

3.6 Real Data Application

In this section, we apply our method to Alzheimer's disease (AD) data, which can be obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (`adni.loni.usc.edu`). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI is to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). For up-to-date information, see `www.adni-info.org`.

We consider two steps in our analytic procedure. The first step is to use the proposed method to identify the latent subgroups and recover the memberships in each subgroup using the observed data. The second step is to use the information from the identified subgroups and the baseline covariates to classify future patients into the identified subgroups.

In the first step, to conduct latent subgroup analysis, we use the longitudinal data of ADASCOG13 (Alzheimer's Disease Assessment Scale-Cognitive Subscale) for each patient from ADNI1, ADNIGO and ADNI2 at different time points (0, 6, 12, 18, 24, 36, 48, 60, 72, 84, 96, 108, 120 months). The data are unbalanced due to the fact that patients may have missing measurements at some time points. Thus, the number of observed measurements for all patients ranges from 1 to 13. ADASCOG13 is widely used as a test of cognitive functions, consisting of thirteen tests, with the values ranging from 0 to 85 to assess the severity of the dementia. Higher values indicate more severe of the dementia due to more

cognitive errors. To apply our subgroup analysis method, we delete patients with less than 4 measurements. As a result, there are 1253 patients used in our analysis.

We take ADASCOG13 as the response to fit the heterogeneous model (3.2). The values of ADASCOG13 are standardized to apply the fusion penalized method. Following the guidance from our simulation studies, we use quadratic splines with one interior knot to approximate the nonparametric functions. As a result, we identify two subgroups, one subgroup with 892 patients and the other one with 361 patients. Figure 3.6 displays the trajectories of individual patients within each subgroup and the estimated mean curve for each subgroup. Clearly, the subgroup depicted in red can be viewed as a non-progression group as the values of the estimated mean curve for this subgroup remain constant over time. In contrast, the subgroup shown in blue can be viewed as a progression group, as we can observe a clear increasing trend of the estimated mean curve for this subgroup over time. Note that the increasing value of ADASCOG13 indicates cognitive decline. Therefore, the progression group is potentially of interest to be recruited in clinical trials when testing whether a drug can slow down the cognitive decline. By our proposed fusion learning method, we can successfully identify two subgroups with their memberships recovered.

In the second step, we are interested in classifying future patients into the two identified subgroups using information from baseline covariates. We collect information of several baseline covariates, including ADASCOG13, mmseTOT (Mini-Mental State Examination total score), FAQTOTAL (functional activities questionnaires total score), cdrSB (clinical dementia rating sum of boxes), ApoE4 (Apolipoprotein E4) status and Education. Among them, ADASCOG13, mmseTOT, FAQTOTAL and cdrSB are the baseline mea-

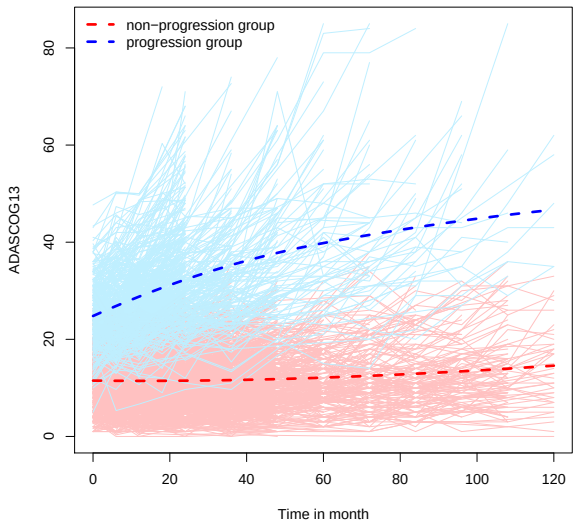


Figure 3.6: The trajectories of individual patients within each identified subgroup (blue, red solid lines) and the estimated mean curve (dashed lines) for each subgroup based on ADASCOG13. The blue group is the progression group, with higher values of ADASCOG13, indicating faster cognition decline.

surements of cognition or functional activities. We exclude the 8 patients whose covariates are not observed at the baseline in the classification step. Thus, there are 889 patients in the non-progression group and 356 patients in the progression group. To understand which covariates that contribute to the group difference, we conduct a two-sample test to compare the means between the two subgroups for each covariate. The P-values are reported in Table 3.7, and they are very small for all covariates. Compared with the non-progression group, patients in the progression group clearly have more severe dementia symptoms given that they have higher ADASCOG13, FAQTOTAL and cdrSB and lower mmseTOT at baseline, as well as more AopE4 carriers. Moreover, they also have less education. These findings corroborate the results given in the literature. In general, the cognition tends to decline more quickly if the disease of a patient is more severe at baseline. ApoE4 is known as one

	Non-progression group	Progression group	
Baseline Covariates	**Mean (SD)**	**Mean (SD)**	**P-value**
ADASCOG13	11.61 (5.15)	24.91 (6.42)	< 0.001
mmseTOT	28.46 (1.62)	25.31 (2.41)	< 0.001
FAQTOTAL	1.46 (3.05)	7.67 (6.71)	< 0.001
cdrSB	0.81 (0.96)	2.70 (1.66)	< 0.001
ApoE4 carrier (%)	35% (0.02)	69% (0.02)	< 0.001
Education	16.21 (2.71)	15.36 (3.05)	< 0.001

Table 3.7: Mean and standard deviation (SD) for each baseline covariate; P-value shows the significant difference existing in the two subgroups. ApoE4 is tested by two proportion z-test, while other covariates are tested by two sample t-test.

important risk factor for AD onset, and ApoE4 carriers tend to show earlier cognitive decline onset than the non-carriers [75]. Additionally, some studies have shown that patients with lower education are more likely to develop AD [43]. Based on the results in Table 3.7, we include all baseline covariates in the classification step.

Next, we use the two identified subgroups obtained from our fusion learning method, and the six baseline covariates given in Table 3.7 to perform classification. Binary variables created from the memberships of the progression group and the non-progression group are used as the responses, and the six baseline covariates are used as the predictors in the classification task. We randomly split the dataset into 80% training data and 20% test data. The training data is used to fit a predictive model, while the test data is used to examine the prediction performance. We apply four popular supervised methods (predictive models) for classification, including the logistic regression, random forest, boosting (gradient boosting machines) and support vector machine (SVM) with linear kernel. The four methods are implemented using the R packages "stats", "randomForest", "gbm" and

"e1071", respectively. For the methods involving hyper parameters, we apply 5-fold cross validation (CV) based on a grid search to select the optimal hyper parameters that maximize the accuracy. Table 3.8 reports the accuracy, specificity, precision, recall, F1 score and AUC (area under the ROC curve) obtained from the test data for the four predictive models. They are commonly used metrics to evaluate the classification performance. Accuracy is the percentage of correct predictions. Specificity is the proportion of true negatives out of the total actual negatives, and it measures how well a method can identify the true negatives. Precision is the ratio of true positives to all positives, while recall refers to the ratio of true positives to the size of the actual positive class. Precision measures the ability of a classification (predictive) model to identify the true positives, and recall assesses its ability to find all the positive cases. F1 score is the weighted average between precision and recall. AUC measures the ability of a classifier to distinguish between classes. From Table 3.8, we observe that the values of accuracy, specificity and AUC are all above 0.9 for the four classification methods (predictive models). The values of recall and F1 score for the boosting method also exceed 0.9. In general, boosting outperforms the other three methods based on all metrics, and therefore it is recommended for the classification task. In conclusion, our two-step procedure is useful for identifying latent subgroups and then further classifying future patients into the identified subgroups based on their baseline characteristics.

3.7 Discussion

In this chapter, we consider the subgroup analysis for longitudinal trajectories of the AD data based on a heterogeneous nonparametric regression model. We use B-splines to

Predictive Models	accuracy	specificity	precision	recall	F1 score	AUC
logistic	0.924	0.944	0.859	0.871	0.865	0.908
random forest	0.920	0.939	0.847	0.871	0.859	0.905
boosting	0.944	0.955	0.889	0.914	0.901	0.935
SVM	0.932	0.950	0.873	0.886	0.879	0.918

Table 3.8: Accuracy, specificity, precision, recall, F1 score and AUC obtained from the test data. The progression group is defined as the positive class.

approximate the nonparametric functional curves, and cluster the subjects into subgroups by applying concave pairwise fusion penalties on the spline coefficients. Our method can automatically identify the latent memberships, and recover the disease trajectory curves of subgroups simultaneously without a prior knowledge of the number of the subgroups. Different from the GMM method that requires to specify an underlying distribution of the data, our method only needs a working correlation matrix of the repeated measures within each subject. Moreover, the resulting estimators of the functional curves are robust to the specification of the working correlation matrix. Simulation studies indicate promising performance of our proposed method. It has been demonstrated as an effective tool for subgroup analysis of the AD data considered in this chapter. As a future work, we plan to extend the proposed method to the joint modeling of survival and longitudinal data, which commonly occur in clinical studies. However, further investigations are needed to develop the computational algorithm and theoretical properties.

Chapter 4

Sparse Deep Neural Networks Regression

4.1 Introduction

Advances in modern technologies have facilitated the collection of large-scale data that are growing in both sample size and the number of variables. Although the conventional parametric models such as generalized linear models are convenient for studying the relationships between variables, they may not be flexible enough to capture the complex patterns in large-scale data. With a large sample size, the bias due to model misspecification becomes more prominent compared to sampling variability, and may lead to false conclusions. The problem of model mis-specification can be solved by nonparametric regression methods that are capable of approximating the unknown target function well without a restrictive structural assumption. Theoretically, we hope that both the bias and the vari-

ance of the functional estimator decrease as the sample size increases. Moreover, the bias is reduced by increasing the variance and vice versa, so that a tradeoff between bias and variance can be achieved for an accurate prediction.

In the classical multivariate regression context with a smoothness condition imposed on the target function, the conventional nonparametric smoothing methods such as local kernels and splines [85, 16, 24, 74, 91, 56] suffer from the so-called "curse of dimensionality" [6], i.e., the convergence rate of the resulting functional estimator deteriorates sharply as the dimension of the predictors increases. As such it is desirable to develop analytic tools that can alleviate the curse of dimensionality while preserving sufficient flexibility, to accommodate the large volume as well as the high dimensionality of the modern data.

In recent years, deep neural networks with multiple hidden layers have been demonstrated to be powerful and effective for approximating multivariate functions, and have been successfully applied to many fields, including computer vision, language processing, speech recognition and biomedical studies [47, 76, 26]. Curiosity has been aroused among researchers about why deep neural networks are so effective in prediction and classification, and thus investigation of their theoretical properties has received increasing attention. One immediate and important research problem would be to figure out under what circumstances and how the deep neural networks can circumvent the curse of dimensionality when estimating a multivariate function. It is worth noting that the alleviation of the dimensionality effect happens at the cost of sacrificing flexibility and generality. To this end, approximation theory using deep neural networks has been established for different classes

of functions [50, 71, 68], which are more restrictive than the traditional smoothness spaces such as Hölder and Sobolev spaces. An alternative way to handle the dimensionality problem is to assume that the target function is defined on a low-dimensional manifold, so that dimensionality reduction can be achieved [14, 77, 81].

Although deep learning has already been widely applied in analysis of modern data because of its impressive empirical performance, the investigation of statistical properties of the estimators from deep learning is still in an early stage, and needs a great deal of efforts. In regression analysis, some inspirational works have shown that the least squares estimator based on deep neural networks can achieve an optimal rate of convergence [83], when the regression function has a compositional structure [5, 78], or the covariates lie on a low-dimensional manifold [17, 14, 77, 69]. The structures considered in [71], [5] and [78] cover several nonparametric and semiparametric models, such as the additive models [29, 84, 37, 36, 53, 99, 97], single-index models and their variants [92, 48, 55].

In this chapter, we consider the Sobolev spaces of functions with square-integrable mixed second derivatives (also called Korobov spaces), which are commonly assumed for the sparse grid approaches addressing the high dimensional problems [32, 68]. Functions in the Korobov spaces only need to satisfy a smoothness condition rather than having a compositional structure, and thus can be more flexible and general for exploring the hidden patterns between the response and the predictors. Moreover, instead of using the least-squares method considered in most works [5, 69, 78], we estimate the target function through empirical risk minimization (ERM) with a Lipschitz loss function satisfying mild conditions. Regularization is also employed for preventing possible over fitting. The family of loss func-

tions that we consider is a general class, and it includes the quadratic, Huber, quantile and logistic loss functions as special cases, so that many regression and classification problems can be solved by our framework. Classification is a crucial task of supervised learning, and robust regression is an important tool for analyzing data with heavy tails. Our estimator of the target function is built upon a network architecture of sparsely-connected deep neural networks with the rectified linear unit (ReLU) activation function. ReLU has been shown to have computational advantage over the sigmoid functions used mainly in shallow networks [19]. Although shallow networks enjoy the Universal Approximation property [19] and can achieve fast convergence rates for functions with structural assumptions [4], their computational complexity can be exponential and they may need to be converted to incremental convex programming [8]. To alleviate the computational burden, we can consider deeper networks that often require fewer parameters [7, 23, 67, 66].

We develop statistical properties of our proposed methodology. The statistical theory is essential for a better understanding of the analytic procedure. We derive non-asymptotic excess risk bounds for our sparse deep ReLU network (SDRN) estimator obtained from empirical risk minimization with the Lipschitz loss function. Specifically, we provide an explicit bound, as a function of the dimension of the feature space, network complexity and sample size, for both of the approximation error and the estimation error of our SDRN estimator, while [68] uses an accuracy value $\epsilon > 0$ for the approximation error without data fitting. Moreover, we derive a non-asymptotic bound for the network complexity, from which we can see more clearly how the network increases with the dimension, and how large the dimension is allowed to be. This bound has not been provided in

[68]. These newly established bounds provide an important theoretical guidance on how the network complexity should be related to the sample size, so that a tradeoff between the two errors can be achieved to secure an optimal fitting from the dataset.

It is of interest to find out that in our framework, the dimension of the feature space is allowed to increase with the sample size n with a rate slightly slower than $\log(n)$, while most existing theories on neural network estimators focus on the scenario with a fixed dimension. We further show that that our SDRN estimator can achieve the same optimal minimax estimation rate (up to logarithmic factors) as one-dimensional nonparametric regression when the dimension is fixed; the effect of the dimension is passed on to a logarithmic factor, so the curse of dimensionality is alleviated. The SDRN estimator has a suboptimal (slightly slower than the optimal rate) when the dimension increases with the sample size. To ensure a good performance, the depth and the total number of nodes and weights of the network, which are used to measure the network complexity [2], need to grow as the sample size n increases. We establish that when the depth increases with n at a logarithmic rate, the number of nodes and weights only need to grow with n at a polynomial rate. These results provide a theoretical basis for empirical studies by deep neural networks, and are also demonstrated from our numerical analysis.

This chapter is organized as follows. Section 4.2 provides the basic setup. Section 4.3 discusses approximation of the target function by the ReLU networks. Section 4.4 introduces the sparse deep ReLU network estimator obtained from empirical risk minimization and establishes the theoretical properties. Section 4.5 further discusses the conditions imposed on the loss function. Section 4.6 reports results from simulation studies, and Sec-

tion 4.7 illustrates the proposed method through real data applications. Some concluding remarks are given in Section 4.8. All the technical proofs are provided in the Appendix B.

4.2 Basic Setup

Notations: Let $\mathbf{a}_d = (a, ..., a)^\top$ be a d-dimensional vector of a's. let $|A|$ be the cardinality of a set A. Denote $|\boldsymbol{a}|_p = (\sum_{i=1}^m |a_i|^p)^{1/p}$ as the L^p-norm of a vector $\boldsymbol{a}= (a_1, \ldots, a_m)^\top$, and $|\boldsymbol{a}|_\infty = \max_{i=1,\ldots,m} |a_i|$. For two vectors $\boldsymbol{a} = (a_1, \ldots, a_m)^\top$ and $\boldsymbol{b} = (b_1, \ldots, b_m)^\top$, denote $\boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^m a_i b_i$. Moreover, for any arithmetic operations involving vectors, they are performed element-by-element. For any two values a and b, denote $a \vee b = \max(a, b)$. For two sequences of positive numbers a_n and b_n, $a_n \ll b_n$ means that $b_n^{-1} a_n = o(1)$, $a_n \lesssim b_n$ means that there exists a constant $C \in (0, \infty)$ and $n_0 \geq 1$ such that $a_n \leq C b_n$ for $n \geq n_0$, and $a_n \asymp b_n$ means that there exist constants $C, C' \in (0, \infty)$ and $n_0 \geq 1$ such that $a_n \leq C b_n$ and $b_n \leq C' a_n$ for $n \geq n_0$.

We consider a general setting of many supervised learning problems. Let $Y \in \mathcal{Y} \subset \mathbb{R}$ be a real-valued response variable and $\boldsymbol{X}= (X_1, \ldots, X_d)^\top$ be d-dimensional independent variables with values in a compact support $\mathcal{X} \subset \mathbb{R}^d$. Without loss of generality, we let $\mathcal{X} = [0, 1]^d$. Let $(\boldsymbol{X}_i^\top, Y_i)^\top$, $i = 1, ..., n$ be i.i.d. samples (a training set of n examples) drawn from the distribution of $(\boldsymbol{X}^\top, Y)^\top$. We consider the mapping $f : \mathcal{X} \to \mathbb{R}$. Our goal is to estimate the unknown target function $f(\boldsymbol{x})$ using sparse deep neural networks from the training set.

Let $\mu : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ be a Borel probability measure of $(\boldsymbol{X}^\top, Y)^\top$. For every $\boldsymbol{x} \in \mathcal{X}$, let $\mu(y|\boldsymbol{x})$ be the conditional (w.r.t. $\boldsymbol{x}$) probability measure of Y. Let μ_X be the

marginal probability measure of $\boldsymbol{X}$. For any $1 \leq p \leq \infty$, let $\mathcal{L}^p(\mathcal{X}) = \{f : \mathcal{X} \to \mathbb{R},\ f$ is lebesgue measurable on $\mathcal{X}$ and $||f||_{L^p} < \infty\}$, where $||f||_{L^p} = (\int_{\boldsymbol{x}\in\mathcal{X}} |f(\boldsymbol{x})|^p d\boldsymbol{x})^{1/p}$ for $1 \leq p < \infty$, and $||f||_{L^\infty} = ||f||_\infty = \sup_{\boldsymbol{x}\in\mathcal{X}} |f(\boldsymbol{x})|$. For $1 \leq p < \infty$, denote $||f||_p = (\int_{\boldsymbol{x}\in\mathcal{X}} |f(\boldsymbol{x})|^p d\mu_X(\boldsymbol{x}))^{1/p}$ and $||f||_{p,n} = (n^{-1}\sum_{i=1}^n |f(\boldsymbol{X}_i)|^p)^{1/p}$. Let $\rho : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. The true target function f_0 is defined as

$$f_0 = \arg\min_{f\in\mathcal{L}^p(\mathcal{X})} \mathcal{E}(f), \text{ where } \mathcal{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \rho(f(\boldsymbol{x}), y) d\mu(\boldsymbol{x}, y). \tag{4.1}$$

Next we introduce the Korobov spaces, in which the functions need to satisfy a certain smoothness condition. The partial derivatives of f with multi-index $\boldsymbol{k} = (k_1, ..., k_d)^\top \in \mathbb{N}^d$ is given as $D^{\boldsymbol{k}} f = \frac{\partial^{|\boldsymbol{k}|_1} f}{\partial x_1^{k_1} \cdots \partial x_d^{k_d}}$, where $\mathbb{N} = \{0, 1, 2, ..., \}$ and $|\boldsymbol{k}|_1 = \sum_{j=1}^d k_j$.

Definition. For $2 \leq p \leq \infty$, the Sobolev spaces of mixed second derivatives (also called Korobov spaces) $W^{2,p}(\mathcal{X})$ are define by

$$W^{2,p}(\mathcal{X}) = \{f \in \mathcal{L}^p(\mathcal{X}) : D^{\boldsymbol{k}} f \in \mathcal{L}^p(\mathcal{X}), |\boldsymbol{k}|_\infty \leq 2\}, \text{ where } |\boldsymbol{k}|_\infty = \max_{j=1,...,d} k_j.$$

Assumption 4.1 *We assume that $f_0 \in W^{2,p}(\mathcal{X})$, for a given $2 \leq p \leq \infty$.*

Remark 4.1 *Assumption 4.1 imposes a smoothness condition on the target function [10, 32, 68]. The Korobov spaces $W^{2,p}(\mathcal{X})$ are subsets of the regular Sobolev spaces defined as $S^{2,p}(\mathcal{X}) = \{f \in \mathcal{L}^p(\mathcal{X}) : D^{\boldsymbol{k}} f \in \mathcal{L}^p(\mathcal{X}), |\boldsymbol{k}|_1 \leq 2\}$ assumed in the traditional nonparametric regression setting [91]. For instance, when $d = 2$, $|\boldsymbol{k}|_\infty = \max(k_1, k_2) \leq 2$ implies $|\boldsymbol{k}|_1 = k_1 + k_2 \leq 4$. If $f \in W^{2,p}(\mathcal{X})$, it needs to satisfy*

$$\frac{\partial f}{\partial x_j}, \frac{\partial^2 f}{\partial x_j^2}, \frac{\partial^2 f}{\partial x_1 \partial x_2}, \frac{\partial^3 f}{\partial x_1^2 \partial x_2}, \frac{\partial^3 f}{\partial x_1 \partial x_2^2}, \frac{\partial^4 f}{\partial x_1^2 \partial x_2^2} \in \mathcal{L}^p(\mathcal{X}).$$

If $f \in S^{2,p}(\mathcal{X})$, it needs to satisfy $\frac{\partial f}{\partial x_j}, \frac{\partial^2 f}{\partial x_j^2}, \frac{\partial^2 f}{\partial x_1 \partial x_2} \in \mathcal{L}^p(\mathcal{X})$. The nonparametric regression methods built upon the regular Sobolev spaces often suffer from the "curse of dimensionality".

The Korobov spaces are commonly assumed for the sparse grid approaches addressing the high dimensional problems, and many popular structured nonparametric models satisfy this condition; see [32]. Note that when $d = 1$ (one-dimensional nonparametric regression), the Korobove and the Sobolev spaces are the same, i.e., if $f \in W^{2,p}(\mathcal{X})$ or $f \in S^{2,p}(\mathcal{X})$, it needs to satisfy $\frac{\partial f}{\partial x_1}, \frac{\partial^2 f}{\partial x_1^2} \in \mathcal{L}^p(\mathcal{X})$.

Assumption 4.2 *For any $y \in \mathcal{Y}$, the loss function $\rho(\cdot, y)$ is convex and it satisfies the Lipschitz property such that there exists a constant $0 < C_\rho < \infty$, for almost every $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$, $|\rho(f_1(\boldsymbol{x}), y) - \rho(f_2(\boldsymbol{x}), y)| \leq C_\rho |f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})|$, for any $f_1, f_2 \in \mathcal{F}$, where $\mathcal{F}$ is a neural network space given in Section 4.4.*

Remark 4.2 *The above Lipschitz assumption is satisfied by many commonly used loss functions. Several examples are provided below.*

Example 1 ***Quadratic loss*** *used in mean regression is given as $\rho(f(\boldsymbol{x}), y) = (y - f(\boldsymbol{x}))^2$. Assuming that $f : \mathcal{X} \to \mathbb{R}$ is M-bounded such that $\sup_{f \in \mathcal{F}} |f(\boldsymbol{x}) - y| \leq M$ holds for almost every $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$, where M is a positive constant, the quadratic loss satisfies Assumption 4.2 with $C_\rho = 2M$.*

Example 2 ***Huber loss*** *is popularly used for robust regression, and it is defined as*

$$\rho(f(\boldsymbol{x}), y) = \begin{cases} 2^{-1}(y - f(\boldsymbol{x}))^2 & \text{if } |f(\boldsymbol{x}) - y| \leq \delta \\ \delta |y - f(\boldsymbol{x})| - \delta^2/2 & \text{if } |f(\boldsymbol{x}) - y| > \delta \end{cases}. \tag{4.2}$$

It satisfies Assumption 4.2 with $C_\rho = \delta$.

Example 3 ***Quantile loss*** *is another popular loss function for robust regression, and it*

is defined as

$$\rho\left(f(\boldsymbol{x}), y\right) = (y - f(\boldsymbol{x}))(\tau - I\{y - f(\boldsymbol{x}) \leq 0\}) \tag{4.3}$$

for $\tau \in (0, 1)$. *It satisfies Assumption 4.2 with* $C_\rho = 1$.

Example 4 ***Logistic loss*** *is used in logistic regression for binary responses as well as for classification. The loss function is* $\rho\left(f(\boldsymbol{x}), y\right) = \log(1 + e^{f(\boldsymbol{x})}) - yf(\boldsymbol{x})$ *for* $y \in \{0, 1\}$. *It satisfies Assumption 4.2 with* $C_\rho = 2$.

4.3 Approximation of The Target Function by ReLU Networks

We consider feedforward neural networks which consist of a collection of input variables, one output unit and a number of computational units (nodes) in different hidden layers. In our setting, the d-dimensional covariates $\boldsymbol{X}$ are the input variables, and the approximated function is the output unit. Each computational unit is obtained from the units in the previous layer by using the form:

$$z = \sigma\left(\sum\nolimits_{j=1}^{N} w_j \widetilde{z}_j + b\right),$$

where $\{\widetilde{z}_j, 1 \leq j \leq N\}$ are the computational units in the previous layer, and $\{w_j, 1 \leq j \leq N\}$ are the weights. Following [2], we measure the network complexity by using the depth of the network defined as the number of layers, the total number of units (nodes), and the total number of weights, which is the sum of the number of connections and the number of units. Moreover, $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function which is chosen by practitioners. We use the rectified linear unit (ReLU) function given as $\sigma(x) = \max(0, x)$.

For any function $f \in W^{2,p}(\mathcal{X})$, it has a unique expression in a hierarchical basis ([10]) such that $f(\boldsymbol{x}) = \sum_{\mathbf{0}_d \leq \boldsymbol{\ell} \leq \boldsymbol{\infty}} \sum_{s \in I_{\boldsymbol{\ell}}} \gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} \phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})$, where $\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = \prod_{j=1}^{d} \phi_{\ell_j,s_j}(x_j)$ are the tensor product piecewise linear basis functions defined on the grids $\Omega_{\boldsymbol{\ell}}$ of level $\boldsymbol{\ell} = (\ell_1, ..., \ell_d)^\top$ and $I_{\boldsymbol{\ell}}$ are the index sets of level $\boldsymbol{\ell}$. We refer to Section 2.1.2 for a detailed discussion on the hierarchical basis functions. The hierarchical coefficients $\gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} \in \mathbb{R}$ are given as (Lemma 3.2 of [10]):

$$\gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} = \int_{\mathcal{X}} \prod_{j=1}^{d} \left(-2^{-(\boldsymbol{\ell}_j+1)} \phi_{\ell_j,s_j}(x_j) \right) D^{\mathbf{2}} f(\boldsymbol{x}) d\boldsymbol{x} \tag{4.4}$$

where $\mathbf{2} = 2\mathbf{1}_d$, and satisfy (Lemma 3.3 of [10])

$$|\gamma^0_{\boldsymbol{\ell},\boldsymbol{s}}| \leq 6^{-d/2} 2^{-(3/2)|\boldsymbol{\ell}|_1} \Big(\int_{\boldsymbol{x}_{\boldsymbol{\ell},\boldsymbol{s}} - \boldsymbol{h}_{\boldsymbol{\ell}}}^{\boldsymbol{x}_{\boldsymbol{\ell},\boldsymbol{s}} + \boldsymbol{h}_{\boldsymbol{\ell}}} |D^{\mathbf{2}} f(\boldsymbol{x})|^2 d\boldsymbol{x} \Big)^{1/2} \leq 6^{-d/2} 2^{-(3/2)|\boldsymbol{\ell}|_1} ||D^{\mathbf{2}} f||_{L^2}. \tag{4.5}$$

Moreover, the above result leads to (Lemma 3.4 of [10])

$$||g_{\boldsymbol{\ell}}||_{L^2} \leq 3^{-d} 2^{-2|\boldsymbol{\ell}|_1} \Big(\int_{\mathcal{X}} |D^{\mathbf{2}} f(\boldsymbol{x})|^2 d\boldsymbol{x} \Big)^{1/2} = 3^{-d} 2^{-2|\boldsymbol{\ell}|_1} ||D^{\mathbf{2}} f||_{L^2}. \tag{4.6}$$

Assumption 4.3 *Assume that for all $\boldsymbol{x} \in \mathcal{X}$, $0 \leq \mu'_X(\boldsymbol{x}) \leq c_\mu$ for some constant $c_\mu \in (0, \infty)$.*

Assumption 4.3 and (4.6) imply that

$$||g_{\boldsymbol{\ell}}||_2 \leq c_\mu ||g_{\boldsymbol{\ell}}||_{L^2} \leq c_\mu 3^{-d} 2^{-2|\boldsymbol{\ell}|_1} ||D^{\mathbf{2}} f||_{L^2}. \tag{4.7}$$

For any $f \in W^{2,p}(\mathcal{X})$, Section 2.1.2 shows that it can be well approximated by the hierarchical basis functions with sparse grids such that

$$f(\boldsymbol{x}) \approx f_m(\boldsymbol{x}) = \sum_{|\boldsymbol{\ell}|_1 \leq m} \sum_{s \in I_{\boldsymbol{\ell}}} \gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} \phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}).$$

Then the hierarchical space with sparse grids is given as

$$V_m^{(1)} = \text{span}\{\phi_{\boldsymbol{\ell},\boldsymbol{s}} : s \in I_{\boldsymbol{\ell}}, |\boldsymbol{\ell}|_1 \leq m\}.$$

The hierarchical space with sparse grids achieves great dimension reduction compared to the space with full grids as shown in Table (2.1). In the following proposition, we provide an upper and a lower bounds for the dimension (cardinality) of the space $V_m^{(1)}$.

Proposition 4.1 *The dimension of the space $V_m^{(1)}$ satisfies*

$$2^{d-1}(2^m + 1) \leq |V_m^{(1)}| \leq 2\sqrt{\frac{2}{\pi}}\frac{\sqrt{d-1}}{(m+d)}2^m \left(4e\frac{m+d}{d-1}\right)^{d-1}.$$

Remark 4.3 *[10] gives an asymptotic order for the cardinality of $V_m^{(1)}$ which is $|V_m^{(1)}| = \mathcal{O}(c(d)2^m m^{d-1})$, where $c(d)$ is a constant depending on d, and the form of $c(d)$ has not been provided. [68] numerically demonstrated how quickly $c(d)$ can increase with d. In Proposion 4.1, we give an explicit form for the upper bound of $|V_m^{(1)}|$ that has not been derived in the literature. From this explict form, we can more clearly see how the dimension of $V_m^{(1)}$ increases with d. This established bound is important for studying the tradeoff between the bias and variance of the estimator obtained from ERM.*

The following proposition provides the approximation error of the approximator $f_m(\cdot)$ obtained from the sparse grids to the true unknown function $f \in W^{2,p}(\mathcal{X})$.

Proposition 4.2 *For any $f \in W^{2,p}(\mathcal{X})$, $2 \leq p \leq \infty$, under Assumption 4.3, one has that for $d = 2$, $||f_m - f||_2 \leq 18^{-1}c_\mu 2^{-2m}(m+3)||D^{\mathbf{2}}f||_{L^2}$; for $d \geq 3$,*

$$||f_m - f||_2 \leq \widetilde{c}2^{-2m}\sqrt{d-2}\left(\frac{e}{3}\frac{m+d}{d-2}\right)^{d-1}||D^{\mathbf{2}}f||_{L^2}, \tag{4.8}$$

where $\widetilde{c} = 2^{-1}c_\mu(3\sqrt{2\pi}e)^{-1}$.

Proposition 4.2 shows that the approximator error decreases as the m value increases.

Moreover, it is interesting to see that there is a mathematical connection between those hierarchical basis functions and the ReLU networks [50, 94, 68]. In the following, we will present several results given in [94], and discuss how to approximate the basis functions $\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})$ using the ReLU networks. Consider the "tooth" function $g : [0,1] \to [0,1]$ given as $g(x) = 2x$ for $x < 1/2$ and $g(x) = 2(1-x)$ for $x \geq 1/2$, and the iterated functions $g_r(x) = \underbrace{g \circ g \circ \cdots \circ g}_{r}(x)$. Let

$$f_R(x) = x - \sum\nolimits_{r=1}^{R} \frac{g_r(x)}{2^{2r}}.$$

It is clear that $f_R(0) = 0$. It is shown in [94] that for the function $f(x) = x^2$ with $x \in [0,1]$, it can be approximated by $f_R(x)$ such that

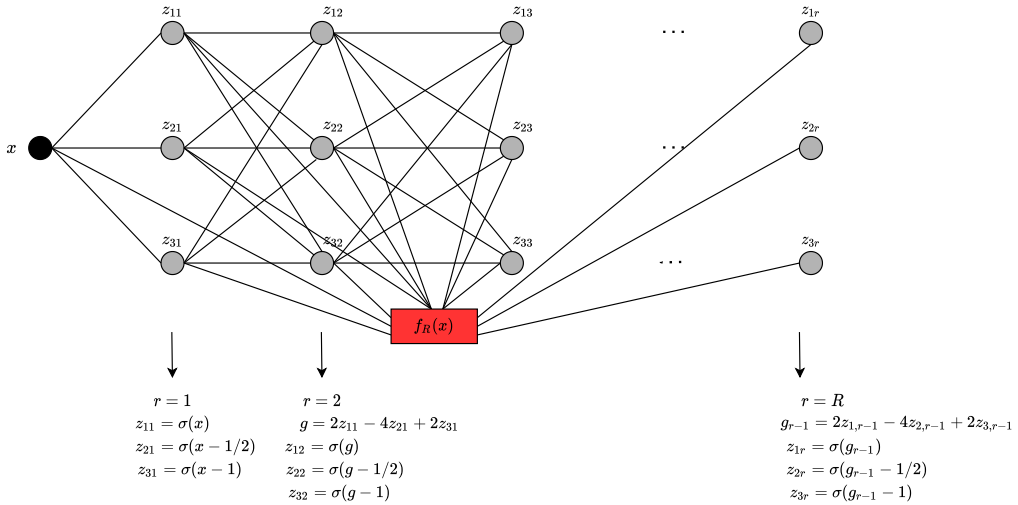
$$||f - f_R||_\infty \leq 2^{-2R-2}.$$

Moreover, the tooth function g can be implemented by a ReLU network: $g(x) = 2\sigma(x) - 4\sigma(x-1/2) + 2\sigma(x-1)$ which has 1 hidden layer and 3 computational units. Therefore, $f_R(x)$ can be constructed by a ReLU network with the depth $R+2$, the computational units $3R+1$, and the number of weights $12R - 5 + 3R + 1 = 15R - 4$. The plot in Figure 4.1 shows the construction of the function $f_R(\cdot)$ by a ReLU network (denoted as Sub1).

Next, we approximate the function $xy = 2^{-1}((x+y)^2 - x^2 - y^2)$ for $x \in [0,1]$ and $y \in [0,1]$ by a ReLU network as follows. By the above results, we have $|f_R(\frac{x+y}{2}) - (\frac{x+y}{2})^2| \leq 2^{-2R-2}$, $|2^{-2}f_R(x) - 2^{-2}x^2| \leq 2^{-2}2^{-2R-2}$ and $|2^{-2}f_R(y) - 2^{-2}y^2| \leq 2^{-2}2^{-2R-2}$. Let

$$\widetilde{f}_R(x,y) = 2\left\{ f_R(\frac{x+y}{2}) - \frac{f_R(x)}{2^2} - \frac{f_R(y)}{2^2} \right\},$$

Figure 4.1: The construction of the function $f_R(\cdot)$ by a ReLU network, denoted as sub network 1 (Sub1).



and $\widetilde{f}_R(x, y)$ can be implemented by a ReLU network having the depth, the computational units and the number of weights being $c_1 R + c_2$, where the constants c_1 and c_2 can be different for these three measures. Moreover, $\widetilde{f}_R(x, y) = 0$ if $xy = 0$. For all $x \in [0, 1]$ and $y \in [0, 1]$,

$$\left|\widetilde{f}_R(x, y) - xy\right| = 2\left|\left\{f_R(\frac{x+y}{2}) - \frac{f_R(x)}{2^2} - \frac{f_R(y)}{2^2}\right\} - \left\{(\frac{x+y}{2})^2 - \frac{x^2}{2^2} - \frac{y^2}{2^2}\right\}\right|$$

$$\leq 2\left(2^{-2R-2} + 2^{-2}2^{-2R-2} + 2^{-2}2^{-2R-2}\right) = 3 \cdot 2^{-2R-2}. \tag{4.9}$$

Figure 4.2 depicts the construction of $\widetilde{f}_R(x, y)$ from the Sub1's, and we denote it as sub network 2 (Sub2).

If there are two covariates $\boldsymbol{X} = (X_1, X_2)^\top$, then $\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = \phi_{\ell_1,s_1}(x_1)\phi_{\ell_2,s_2}(x_2)$ can be approximated by $\widetilde{f}_R(\phi_{\ell_1,s_1}(x_1), \phi_{\ell_2,s_2}(x_2))$. Next we approximate $\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})$ by a ReLU network from a binary tree structure for the general setting with d-dimensional covariates $\boldsymbol{X} = (X_1, \ldots, X_d)^\top$. For notational simplicity, we denote $F_{j_1,\ldots,j_q} = \phi_{\ell_{j_1},s_{j_1}}(x_{j_1}) \times \cdots \times \phi_{\ell_{j_q},s_{j_q}}(x_{j_q})$ and $\widetilde{F}_{j_1,\ldots,j_q j'_1,\ldots,j'_p} = \widetilde{f}_R(F_{j_1,\ldots,j_q}, F_{j'_1,\ldots,j'_p})$. Then for any $1 \leq j_1 \neq j_2 \leq d$,

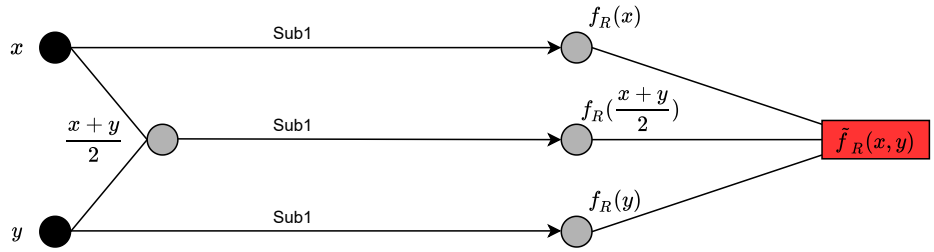


Figure 4.2: The construction of $\widetilde{f}_R(x,y)$ from the Sub1's, we denote it as subnetwork 2 (Sub2).

$F_{j_1}F_{j_2} = \phi_{\ell_{j_1},s_{j_1}}(x_{j_1})\phi_{\ell_{j_2},s_{j_2}}(x_{j_2})$ can be approximated by $\widetilde{F}_{j_1j_2} = \widetilde{f}_R(F_{j_1},F_{j_2})$, and (4.9) leads to $|\widetilde{F}_{j_1j_2} - F_{j_1}F_{j_2}| \le 3\cdot 2^{-2R-2}$. Next, we approximate $\widetilde{F}_{j_1j_2}\widetilde{F}_{j_3j_4}$ with $\widetilde{F}_{j_1j_2j_3j_4} = \widetilde{f}_R(\widetilde{F}_{j_1j_2},\widetilde{F}_{j_3j_4})$, and

$$|\widetilde{F}_{j_1j_2j_3j_4} - \widetilde{F}_{j_1,j_2}\widetilde{F}_{j_3,j_4}| \le 3\cdot 2^{-2R-2}.$$

These results lead to

$$\begin{aligned}
|\widetilde{F}_{j_1j_2}\widetilde{F}_{j_3j_4} - F_{j_1}F_{j_2}F_{j_3}F_{j_4}| &= |\widetilde{F}_{j_1j_2}\widetilde{F}_{j_3j_4} - F_{j_1}F_{j_2}\widetilde{F}_{j_3j_4} + F_{j_1}F_{j_2}\widetilde{F}_{j_3j_4} - F_{j_1}F_{j_2}F_{j_3}F_{j_4}| \\
&\le |\widetilde{F}_{j_1j_2} - F_{j_1}F_{j_2}||\widetilde{F}_{j_3j_4}| + |F_{j_1}F_{j_2}||\widetilde{F}_{j_3j_4} - F_{j_3}F_{j_4}| \\
&\le 2\cdot 3\cdot 2^{-2R-2}.
\end{aligned}$$

Thus

$$\begin{aligned}
|\widetilde{F}_{j_1j_2j_3j_4} - F_{j_1}F_{j_2}F_{j_3}F_{j_4}| &\le |\widetilde{F}_{j_1j_2j_3j_4} - \widetilde{F}_{j_1j_2}\widetilde{F}_{j_3j_4}| + |\widetilde{F}_{j_1j_2}\widetilde{F}_{j_3j_4} - F_{j_1}F_{j_2}F_{j_3}F_{j_4}| \\
&\le (1+2)\cdot 3\cdot 2^{-2R-2}.
\end{aligned}$$

By following the same argument, we have

$$\begin{aligned}
&|\widetilde{F}_{j_1j_2j_3j_4j_5j_6j_7j_8} - F_{j_1}F_{j_2}F_{j_3}F_{j_4}F_{j_5}F_{j_6}F_{j_7}F_{j_8}| \\
\le\ &|\widetilde{F}_{j_1\cdots j_8} - \widetilde{F}_{j_1j_2j_3j_4}\widetilde{F}_{j_5j_6j_7j_8}| + |\widetilde{F}_{j_1j_2j_3j_4} - F_{j_1}F_{j_2}F_{j_3}F_{j_4}||\widetilde{F}_{j_5j_6j_7j_8}|
\end{aligned}$$

$$
\begin{aligned}
&+|F_{j_1}F_{j_2}F_{j_3}F_{j_4}||\widetilde{F}_{j_5j_6j_7j_8} - F_{j_5}F_{j_6}F_{j_7}F_{j_8}| \\
\leq\ & (1+2(1+2))\cdot 3\cdot 2^{-2R-2} = (1+2+2^2)\cdot 3\cdot 2^{-2R-2}.
\end{aligned}
$$

Define

$$
\widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = \widetilde{F}_{1\cdots d}.
$$

We continue the above process and it can be shown from mathematical induction [68] that for all $\boldsymbol{x}\in\mathcal{X}$,

$$
\begin{aligned}
&|\widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) - \phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})| = |\widetilde{F}_{1\cdots d} - F_1\times\cdots\times F_d| \\
\leq\ & (1+2+2^2+\cdots+2^{\lfloor\log_2 d\rfloor-1})\cdot 3\cdot 2^{-2R-2} \leq 3\cdot 2^{-2R-2}(d-1), \qquad (4.10)
\end{aligned}
$$

where $\lfloor a\rfloor$ is the largest integer no greater than a. Moreover, $\widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = 0$ if $\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = 0$. The ReLU network used to approximate $\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})$ has depth $\mathcal{O}(R)\times\log_2 d = \mathcal{O}(R\log_2 d)$, the computational units $\mathcal{O}(R)\times(d+2^{-1}d+\cdots+2^{-\lfloor\log_2 d\rfloor+1}d) = \mathcal{O}(Rd)$, and the number of weights $\mathcal{O}(Rd)$. Figure 4.3 shows the construction of each approximated basis function approximator $\widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})$ from the Sub2's, and we denote it as sub network 3 (Sub3).

Then the ReLU network approximator of the unknown function $f(\boldsymbol{x})$ is

$$
\widetilde{f}_R(\boldsymbol{x}) = \sum_{|\boldsymbol{\ell}|_1\leq m}\sum_{s\in I_{\boldsymbol{\ell}}}\gamma^0_{\boldsymbol{\ell},\boldsymbol{s}}\widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) = \sum_{|\boldsymbol{\ell}|_1\leq m}\widetilde{g}_{\boldsymbol{\ell}}(\boldsymbol{x}). \qquad (4.11)
$$

The following proposition provides the approximation error of the approximator $\widetilde{f}_R(\cdot)$ obtained from the ReLU network to the true unknown function $f(\cdot)$.

Proposition 4.3 *For any $f\in W^{2,p}(\mathcal{X})$, $2\leq p\leq\infty$, under Assumption 4.3, one has for $d=2$,*

$$
||\widetilde{f}_R - f||_2 \leq \left\{\sqrt{\frac{3}{8}}2^{-2R}(d-1)(\sqrt{\frac{2}{3}})^{d-1} + 9^{-1}c_\mu 2^{-2m}(m+3)\right\}||D^{\mathbf{2}}f||_{L^2};
$$

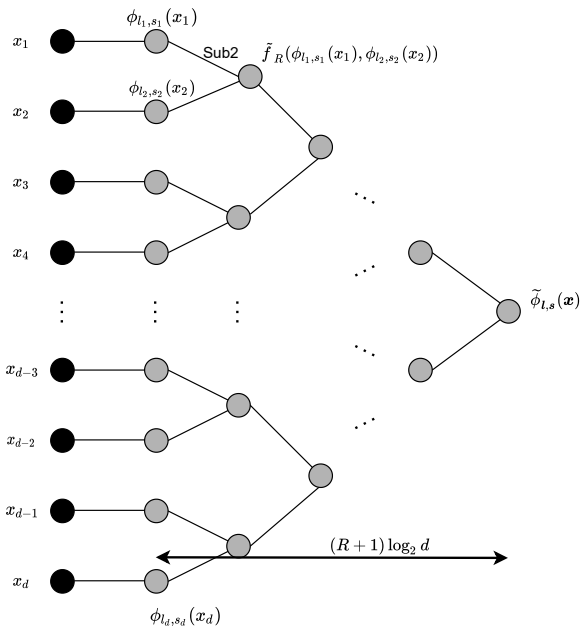


Figure 4.3: The construction of $\tilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})$ from the Sub2's, we denote it as subnetwork 3 (Sub3).

for $d \geq 3$,

$$||\widetilde{f}_R - f||_2 \leq \left\{ \sqrt{\frac{3}{8}} 2^{-2R}(d-1)(\sqrt{\frac{2}{3}})^{d-1} + \widetilde{c} 2^{-2m}\sqrt{d-2} \left(\frac{e}{3} \frac{m+d}{d-2} \right)^{d-1} \right\} ||D^{\mathbf{2}} f||_{L^2},$$

where $\widetilde{c} = 2^{-1} c_\mu (3\sqrt{2\pi}e)^{-1}$. *The ReLU network that is used to construct the approximator* $\widetilde{f}_R$ *has depth* $\mathcal{O}(R \log_2 d)$, *the number of computational units* $\mathcal{O}(Rd) \times |V_m^{(1)}| = \mathcal{O}\left(2^m d^{3/2} R \, (m+d)^{-1} \left(4e \frac{m+d}{d-1} \right)^{d-1} \right)$, *and the number of weights* $\mathcal{O}(Rd) \times |V_m^{(1)}|$
$= \mathcal{O}\left(2^m d^{3/2} R \, (m+d)^{-1} \left(4e \frac{m+d}{d-1} \right)^{d-1} \right)$.

Remark 4.4 *From Figure 4.3 and the mathematical expression (4.11), we see that the approximator* $\widetilde{f}_R(\cdot)$ *of the unknown function* $f(\cdot)$ *is constructed from a sparse deep ReLU network, as the nodes on each layer are not fully connected with the nodes from the previous layer, and the depth of the network has the order of* $R \log_2 d$ *which increases with* R.

Remark 4.5 *[68] showed that the approximation error of the deep ReLU network can achieve accuracy* $\epsilon > 0$. *We further derive an explicit form of the bound to see how it depends on the dimension and the network complexity. In Theorem 4.3, we will show that* m *and* R *need to grow with the sample size* n *slowly at a logarithmic rate to achieve tradeoff between bias and variance, so the depth of the ReLU network grow with* n *at a logarithmic rate, and the number of computational units increase with* n *at a polynomial rate.*

4.4 Sparse Deep ReLU Network Estimator

In this section, we will introduce the sparse deep ReLU network (SDRN) estimator of the unknown function f_0 obtained from (4.1). As discussed in Section 4.3, for $f_0 \in W^{2,p}(\mathcal{X})$, there exists a sparse deep ReLU approximator $\widetilde{f}_R(\boldsymbol{x}) = \sum_{|\boldsymbol{\ell}|_1 \leq m} \sum_{s \in I_{\boldsymbol{\ell}}} \gamma^0_{\boldsymbol{\ell},\boldsymbol{s}} \widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})$,

where the expression of $\gamma^0_{\boldsymbol{\ell},\boldsymbol{s}}$ is given in (4.4), which has the approximation error given in Theorem 4.3 with $f(\cdot)$ replaced by the true target function $f_0(\cdot)$ in (4.1).

Define the ReLU network class as

$$\mathcal{F}(\widetilde{\phi}, m, B) = \{f_{RL} : \mathcal{X} \to \mathbb{R}, f_{RL}(\boldsymbol{x}) = \sum_{|\boldsymbol{\ell}|_1 \leq m} \sum_{s \in I_{\boldsymbol{\ell}}} \gamma_{\boldsymbol{\ell},\boldsymbol{s}} \widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}), \eta_{\boldsymbol{\ell},\boldsymbol{s}} \in \mathbb{R}, ||f_{RL}||_\infty \leq B\}, \tag{4.12}$$

with $B \geq \max(||f_0||_\infty, ||\widetilde{f}_R||_\infty)$. Then $\widetilde{f}_R \in \mathcal{F}(\widetilde{\phi}, m, B)$. Denote $\boldsymbol{\gamma} = \{\gamma_{\boldsymbol{\ell},\boldsymbol{s}} : s \in I_{\boldsymbol{\ell}}, |\boldsymbol{\ell}|_1 \leq m\}^\top$ and $\widetilde{\phi}(\boldsymbol{x}) = \{\widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) : s \in I_{\boldsymbol{\ell}}, |\boldsymbol{\ell}|_1 \leq m\}^\top$, $f_{RL}(\boldsymbol{x})$ can be written as $f_{RL}(\boldsymbol{x}) = \widetilde{\phi}(\boldsymbol{x})^\top \boldsymbol{\gamma}$. We obtain the unpenalized SDRN estimator $\widehat{f}^U_{RL}$ of f_0 from minimizing the following empirical risk:

$$\widehat{f}^U_{RL} = \arg \min_{f_{RL} \in \mathcal{F}(\widetilde{\phi}, m, B)} \mathcal{E}_n(f_{RL}), \text{ where } \mathcal{E}_n(f_{RL}) = n^{-1} \sum_{i=1}^n \rho(f_{RL}(\boldsymbol{X}_i), Y_i). \tag{4.13}$$

Similarly, we can also obtain the penalized SDRN estimator $\widehat{f}^P_{RL}$ of f_0 from minimizing

$$\widehat{f}^P_{RL} = \arg \min_{f_{RL} \in \mathcal{F}(\widetilde{\phi}, m, B)} \{\mathcal{E}_n(f_{RL}) + 2^{-1}\lambda ||f_{RL}||^2_\Psi\},$$

where $\lambda > 0$ is a tuning parameter for the L_2 penalty, and $||f_{RL}||^2_\Psi = \boldsymbol{\gamma}^\top \left[\int \{\widetilde{\phi}(\boldsymbol{x})\widetilde{\phi}(\boldsymbol{x})^\top\} d\boldsymbol{x}\right] \Psi \boldsymbol{\gamma}$. The L_2 penalty is often used to prevent overfitting. Here, we let $\Psi = \left[\int \{\widetilde{\phi}(\boldsymbol{x})\widetilde{\phi}(\boldsymbol{x})^\top\} d\boldsymbol{x}\right]^{-1}$, so that $||f_{RL}||^2_\Psi = \boldsymbol{\gamma}^\top \boldsymbol{\gamma}$.

We use $\widehat{f}_{RL}$ as a generic notation for a SDRN estimator; it can be either the unpenalized or the penalized estimator. For a given estimator $\widehat{f}_{RL}$, we define the overall error as $\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_0)$, which is used to measure how close the estimator $\widehat{f}_{RL}$ to the true target function f_0. Let

$$f^0_{RL} = \arg \min_{f_{RL} \in \mathcal{F}(\widetilde{\phi}, m, B)} \mathcal{E}(f_{RL}), \text{ where } \mathcal{E}(f_{RL}) = \int_{\mathcal{X} \times \mathcal{Y}} \rho(f_{RL}(\boldsymbol{x}), y) d\mu(\boldsymbol{x}, y). \tag{4.14}$$

Then the overall error of the estimator $\widehat{f}_{RL}$ can be splitted into the approximation error $\mathcal{E}(f_{RL}^0) - \mathcal{E}(f_0)$ and the sampling error $\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_{RL}^0)$ such that

$$\underbrace{\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_0)}_{\text{overall error}} = \underbrace{\mathcal{E}(f_{RL}^0) - \mathcal{E}(f_0)}_{\text{approximation error}} + \underbrace{\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_{RL}^0)}_{\text{estimation error}}.$$

We will establish the upper bounds for the approximation error and the estimation error, respectively, as follows.

We introduce the following Bernstein condition that is required for obtaining the probability bound for the estimation error of our SDRN estimator.

Assumption 4.4 *There exists a constant* $0 < a_\rho < \infty$ *such that*

$$a_\rho ||f - f_{RL}^0||_2^2 \leq \mathcal{E}(f) - \mathcal{E}(f_{RL}^0) \tag{4.15}$$

for any $f \in \mathcal{F}(\widetilde{\phi}, m, B)$.

Remark 4.6 *The Bernstein condition given in (4.15) for Lipschitz loss functions is used in the literature in order to establish probability bounds of estimators obtained from empirical risk minimization [1]. A more general form is* $a_\rho ||f - f_{RL}^0||_2^{2\kappa} \leq \mathcal{E}(f) - \mathcal{E}(f_{RL}^0)$ *for some* $\kappa \geq 1$. *The parameter* κ *can affect the estimator's rate of convergence. For proof convenience, we let* $\kappa = 1$ *which is satisfied by many commonly used loss functions. We will give a detailed discussion on this Bernstein condition, and will present different examples in Section 4.5 of the Appendix.*

Remark 4.7 *From the Lipschitz condition given in Assumption 4.2, we have that there exists a constant* $0 < M_\rho < \infty$ *such that* $|\rho(f(\boldsymbol{x}), y)| \leq M_\rho$, *for almost every* $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ *and any* $f \in \mathcal{F}(\widetilde{\phi}, m, B)$.

Another condition is given below and it is used for controlling the approximation error from the ReLU networks.

Assumption 4.5 *There exists a constant* $0 < b_\rho < \infty$ *such that*

$$\mathcal{E}(f) - \mathcal{E}(f_0) \leq b_\rho ||f - f_0||_2^2 \tag{4.16}$$

for any $f \in \mathcal{F}(\widetilde{\phi}, m, B)$.

Remark 4.8 *Assumption 4.5 is introduced for controlling the approximation error* $\mathcal{E}(f_{RL}^0) - \mathcal{E}(f_0)$*, but it is not required for establishing the upper bound of the sampling error* $\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_{RL}^0)$*. The approximation error* $\mathcal{E}(f_{RL}^0) - \mathcal{E}(f_0)$ *can be well controlled based on the result from Proposition 4.3 together with Assumption 4.5. Without this assumption, the approximation error will have a slower rate. Assumption 4.5 is satisfied by the quadratic, logistic, quantile and Huber loss functions under mild conditions. More discussions on this assumption will be provided in Section 4.5 of the Appendix.*

Under Condition (4.16) given in Assumption 4.5, by the definition of f_{RN}^0 given in (4.14) and Proposition 4.3, the approximation error

$$\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) \leq \mathcal{E}(\widetilde{f}_R) - \mathcal{E}(f_0) \leq b_\rho ||\widetilde{f}_R - f_0||_2^2.$$

Since f_0 satisfies Assumption 4.1, then $||D^{\mathbf{2}} f_0||_{L^2} \leq C_f$ for some constant $C_f \in (0, \infty)$. Next proposition presents an upper bound for the approximation error when the unknown function f_0 is approximated by the SDRN obtained from the ERM in (4.14).

Proposition 4.4 *Under Assumptions 4.1, 4.3 and 4.5, and* $m^{-1} = o(1)$ *and* $m \lesssim R$*, one has*

$$\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) \leq \zeta_{R,m,d},$$

where

$$\begin{aligned}\zeta_{R,m,d} &= 4b_\rho C_f^2 \widetilde{c}^2 2^{-4m} d \left(\frac{e}{3}\frac{m+d}{d-2}\right)^{2(d-1)}, \text{ for } d \geq 3 \\ \zeta_{R,m,d} &= 81^{-1} b_\rho C_f^2 c_\mu^2 2^{-4m}(m+3)^2, \text{ for } d = 2. \end{aligned} \tag{4.17}$$

Note that without Assumption 4.5, we obtain a looser bound for $\mathcal{E}(f_{RL}^0) - \mathcal{E}(f_0) = \mathcal{O}(\zeta_{R,m,d}^{1/2})$ based on the result $\mathcal{E}(\widetilde{f}_R) - \mathcal{E}(f_0) \leq C_\rho ||\widetilde{f}_R - f_0||_2$ which is directly implied from Assumption 4.2.

Next we establish the bound for the sampling error $\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_{RL}^0)$. Let $\mathcal{N}(\delta, \mathcal{F}, ||\cdot||_\infty)$ be the covering number, that is, the minimal number of $||\cdot||_\infty$- balls with radius δ that covers $\mathcal{F}$ and whose centers reside in $\mathcal{F}$. Let $\lambda_{\min,\widetilde{\phi}} = \lambda_{\min}\left\{\int \widetilde{\phi}(\boldsymbol{x})\widetilde{\phi}(\boldsymbol{x})^\top d\mu_X(\boldsymbol{x})\right\}$. In the theorem below, we provide an upper bound for the estimation error $\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_{RL}^0)$.

Theorem 4.1 *Under Assumptions 4.1-4.4, we have that for any $\epsilon > 0$, i)*

$$P\left(\mathcal{E}(\widehat{f}_{RL}^U) - \mathcal{E}(f_{RL}^0) > \epsilon\right) \leq \mathcal{N}(\sqrt{2}C_\rho^{-1}\epsilon/8, \mathcal{F}(\widetilde{\phi}, m, B), ||\cdot||_\infty) \exp\left(-n\epsilon/C^*\right),$$

and ii) with $\lambda\lambda_{\min,\widetilde{\phi}}^{-1} \leq 5^{-1}a_\rho^{1/2}\min(a_\rho^{1/2}, B\sqrt{\epsilon/2})$,

$$P\left\{\mathcal{E}(\widehat{f}_{RL}^P) - \mathcal{E}(f_{RL}^0) > 2\epsilon\right\} \leq \mathcal{N}(\sqrt{2}C_\rho^{-1}\epsilon/8, \mathcal{F}(\widetilde{\phi}, m, B), ||\cdot||_\infty) \exp\left(-n\epsilon/C^*\right),$$

where $C^* = 64(C_\rho^2 a_\rho^{-1} + 4M_\rho/3)$, in which C_ρ, a_ρ and M_ρ are constants given in Assumptions 4.2 and 4.4 and Remark 4.7.

Remark 4.9 *Let $c' = 1$ if $\widehat{f}_{RL} = \widehat{f}_{RL}^U$ and $c' = 2$ if $\widehat{f}_{RL} = \widehat{f}_{RL}^P$. The two probability bounds established in Theorem 4.1 can be summarized as*

$$P\left\{\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_{RL}^0) > c'\epsilon\right\} \leq \mathcal{N}(\sqrt{2}C_\rho^{-1}\epsilon/8, \mathcal{F}(\widetilde{\phi}, m, B), ||\cdot||_\infty) \exp\left(-n\epsilon/C^*\right),$$

where $\widehat{f}_{RL}$ can be both the unpenalized and penalized estimators.

Theorem 4.2 *Under the same assumptions as given in Theorem 4.1,*

$$P\left(\mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_{RL}^0) > c' \frac{C^* |V_m^{(1)}|}{n} \max(1, \log \frac{C^{**} n}{|V_m^{(1)}| \varsigma})\right) \leq \varsigma,$$

where $c' = 1$ *if* $\widehat{f}_{RL} = \widehat{f}_{RL}^U$ *and* $c' = 2$ *if* $\widehat{f}_{RL} = \widehat{f}_{RL}^P$, C^* *is given in Theorem 4.1,* $C^{**} = 12 C_\rho B / C^*$ *and* $\varsigma = (\frac{C^{**} C^*}{\epsilon})^{|V_m^{(1)}|} \exp(-n\epsilon / C^*)$, *for any* $\epsilon \in (0, C_\rho B/2)$.

Based on the upper bound for the estimation error given in Theorem 4.2, and the bound for the approximation error given in (4.17), we can further obtain the risk rate of the SDRN estimator $\widehat{f}_{RL}$ presented in the following theorems.

Theorem 4.3 *Under Assumptions 4.1-4.5,* $2^m \asymp n^{1/5}$ *and* $m \lesssim R$, *if (i)* $d \asymp (\log_2 n)^\kappa$ *for some constant* $\kappa \in (0, 1)$, *then* $\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) = o(n^{-4/5+\varpi} (\log_2 n)^{3\kappa - 2})$ *and* $\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_{RN}^0) = \mathcal{O}_p(n^{-4/5+\varpi/2} (\log_2 n)^{3\kappa/2})$, *for an arbitrarily small* $\varpi > 0$. *Thus*

$$\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_0) = o_p(n^{-4/5+\varpi} (\log_2 n)^{3\kappa - 2}).$$

The above rate is satisfied by both $\widehat{f}_{RN}^U$ *and* $\widehat{f}_{RN}^P$ *with* $\lambda = \mathcal{O}(\lambda_{\min, \tilde{\phi}} n^{-2/5+\varpi/4} (\log_2 n)^{3\kappa/4})$.

If $R \asymp \log_2 n$, *the ReLU network that is used to construct the estimator* $\widehat{f}_{RN}$ *has depth* $\mathcal{O}[\log_2 n \{\log_2 (\log_2 n)\}]$, *the number of computational units* $\mathcal{O}\{(\log_2 n)^{5\kappa/2-1} n^{1/5+\varpi/2}\}$, *and the number of weights* $\mathcal{O}\{(\log_2 n)^{5\kappa/2-1} n^{1/5+\varpi/2}\}$.

If (ii) $d \asymp 1$, *then* $\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) = \mathcal{O}(n^{-4/5} (d^{-1} \log_2 n)^{2d-2})$ *and* $\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_{RN}^0) = \mathcal{O}_p(n^{-4/5} (d^{-1} \log_2 n)^d)$. *Thus*

$$\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_0) = \mathcal{O}_p(n^{-4/5} (d^{-1} \log_2 n)^{2d-2}).$$

The above rate is satisfied by both $\widehat{f}_{RN}^U$ *and* $\widehat{f}_{RN}^P$ *with* $\lambda = \mathcal{O}(\lambda_{\min, \tilde{\phi}} n^{-2/5} (d^{-1} \log_2 n)^{d/2})$.

If $R \asymp \log_2 n$, the ReLU network that is used to construct the estimator $\widehat{f}_{RN}$ has depth $\mathcal{O}(\log_2 n)$, the number of computational units $\mathcal{O}\{(\log_2 n)^{d-1} n^{1/5}\}$, and the number of weights $\mathcal{O}\{(\log_2 n)^{d-1} n^{1/5}\}$.

Remark 4.10 *Note that Assumption 4.5 is not required for obtaining the convergence rate of the sampling error $\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f^0_{RN})$, it is only needed for the rate of the approximation error $\mathcal{E}(f^0_{RN}) - \mathcal{E}(f_0)$. Without this assumption, the rate of $\mathcal{E}(f^0_{RN}) - \mathcal{E}(f_0)$ is slower.*

Remark 4.11 *The risk rates in Theorem 4.3 are summarized as $\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_0)$ $= o_p(n^{-4/5+\varpi}(\log_2 n)^{3\kappa-2})$ if $d \asymp (\log_2 n)^{\kappa}$ and $\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_0) = \mathcal{O}_p(n^{-4/5}(d^{-1}\log_2 n)^{2d-2})$ if $d \asymp 1$.*

Remark 4.12 *We focus on deriving the optimal risk rate for the SDRN estimator of the unknown function f_0 when it belongs to the Korobov space of mixed derivatives of order $\beta = 2$. Then the derived rate can be written as $n^{-2\beta/(2\beta+1)}(d^{-1}\log_2 n)^{2d-2}$ when d is fixed. It is possible to derive a similar estimator for a smoother regression function that has mixed derivatives of order $\beta > 2$ when Jacobi-weighted Korobov spaces [79] are considered. This can be an interesting topic for future work.*

Remark 4.13 *It is worth noting that for the classical nonparametric regression estimators such as spline estimators [83], the optimal minimax risk rate is $n^{-4/(4+d)}$, if the regression function belongs to the Sobolev spaces $S^{2,p}(\mathcal{X})$. This rate suffers from the curse of dimensionality as d increases. For one-dimensional nonparametric regression with $d = 1$, the optimal rate becomes $n^{-4/5}$.*

[5] showed that their least squares neural network estimator can achieve the rate $n^{-2\beta/(2\beta+d^*)}$ (up to a log factor), if the regression function satisfies a β-smooth generalized

hierarchical interaction model of order d^*. When $\beta = 2$, the rate is $n^{-4/(4+d^*)}$. The rates mentioned above require d to be fixed. [5] consider a smooth activation function, while [78] established a similar optimal rate for ReLU activation function.

Theorem 4.3 shows that when f_0 belongs to the Korobov spaces $W^{2,p}(\mathcal{X})$, our SDRN estimator has the risk rate $n^{-4/5}(d^{-1}\log_2 n)^{2d-2}$ and it achieves the optimal minimax rate (up to a log factor) as one-dimensional nonparametric regression, if the dimension d is fixed. The effect of d is passed on to a logarithm order, so the curse of dimensionality can be alleviated. When d increases with n with an order $(\log_2 n)^\kappa$, the risk rate is slightly slower than $n^{-4/5}$.

ADAM Algorithm

To estimate the coefficients in the penalized SDRN estimator of the target function, we use the Adam algorithm [44]. Let $\Phi = \{\widetilde{\phi}(\boldsymbol{X}_1), ..., \widetilde{\phi}(\boldsymbol{X}_n)\}^\top$. Then the estimate of the coefficient vector $\boldsymbol{\gamma}$ in the penalized SDRN estimate of the target function solves the following optimization:

$$g(\boldsymbol{\gamma}) = \min_{\gamma} \sum_{i=1}^{n} \rho(\widetilde{\phi}(\boldsymbol{X}_i)^\top \boldsymbol{\gamma}, Y_i) + 2^{-1}\lambda^* \boldsymbol{\gamma}^\top \boldsymbol{\gamma},$$

where $\lambda^* = n\lambda$. We adopt the Adam algorithm studied in [44] for obtaining the estimate of $\boldsymbol{\gamma}$. This algorithm considers first-order gradient-based optimization, and it is straightforward to implement and has little memory requirements. It is well suited for optimization with large number of parameters and sample size. The algorithm is given as follows.

Require: $\boldsymbol{\gamma}_0$: Initial parameter vector

$m_0 \longleftarrow 0$ (Initialize 1^{st} moment vector)

$v_0 \longleftarrow 0$ (Initialize 2^{nd} moment vector)

$t \longleftarrow 0$ (Initialize timestep)

while

$t \longleftarrow t + 1$

$h_t \longleftarrow \nabla_{\boldsymbol{\gamma}} g_t(\boldsymbol{\gamma}_{t-1})$ (Get gradient w.r.t. stochastic objective at timestep t)

$m_t \longleftarrow \beta_1 m_{t-1} + (1 - \beta_1) h_t$ (Update biased first moment estimate)

$v_t \longleftarrow \beta_2 v_{t-1} + (1 - \beta_2) h_t^2$ (Update biased second raw moment estimate)

$\widehat{m}_t \longleftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)

$\widehat{v}_t \longleftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)

$\boldsymbol{\gamma}_t \longleftarrow \boldsymbol{\gamma}_{t-1} - \alpha \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)

end while until convergence

return $\boldsymbol{\gamma}_t$

We set the step size $\alpha = 0.1$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ as suggested in the literature.

4.5 Discussions on Assumptions 4.4 and 4.5

We first state a general condition given in Assumption 4.6 presented below. We will show that if a loss function satisfies this condition, then it will satisfy Assumption 4.4 (Bernstein condition) and Assumption 4.5.

Assumption 4.6 *For all $y \in \mathcal{Y}$, the loss function $\rho(\cdot, y)$ is strictly convex and it has a bounded second derivative such that $\rho''(\cdot, y) \in [2a_\rho, 2b_\rho]$ almost everywhere, for some constants $0 < a_\rho \leq b_\rho < \infty$.*

Assumption 4.6 is satisfied by a variety of classical loss functions such as quadratic loss and logistic loss. For example, for the quadratic loss $\rho(f(\boldsymbol{x}), y) = (y - f(\boldsymbol{x}))^2$, clearly $\rho''(\cdot, y) = 2$, so $a_\rho = b_\rho = 1$.

Let f_0 solve $\int_{\mathcal{Y}} \rho'(f_0(\boldsymbol{x}), y) d\mu(y|\boldsymbol{x}) = 0$ and $f_0 \in W^{2,p}(\mathcal{X})$. Then f_0 is the target function that minimizes the expected risk given in (4.1). Lemma 4.1 given below will show that Assumptions 4.4 and 4.5 are implied from Assumption 4.6.

Lemma 4.1 *Under Assumption 4.6, for any $f \in \mathcal{F}(\widetilde{\phi}, m, B)$, one has $a_\rho ||f - f^0_{RL}||_2^2 \leq \mathcal{E}(f) - \mathcal{E}(f^0_{RL})$ and $\mathcal{E}(f) - \mathcal{E}(f_0) \leq b_\rho ||f - f_0||_2^2$.*

It is easy to see that the quantile and Huber loss functions do not satisfy Assumption 4.6. In the lemmas below we will show that under mild conditions, Assumptions 4.4 and 4.5 are satisfied by the quantile and Huber loss functions.

Lemma 4.2 *Assume that for all $\boldsymbol{x} \in \mathcal{X}$, it is possible to define a conditional density function of $Y|\boldsymbol{X} = \boldsymbol{x}$ such that $1/C_1 \leq \mu'(u|\boldsymbol{x}) \leq 1/C_2$ for some $C_1 \geq C_2 > 0$ for all $u \in \{u \in \mathbb{R}$: $|u - f^0_{RL}(\boldsymbol{x})| \leq 2B$ or $|u - f_0(\boldsymbol{x})| \leq 2B\}$. Then for any $f \in \mathcal{F}(\widetilde{\phi}, m, B)$, the quantile loss given in (4.3) satisfies $a_\rho ||f - f^0_{RL}||_2^2 \leq \mathcal{E}(f) - \mathcal{E}(f^0_{RL})$ and $\mathcal{E}(f) - \mathcal{E}(f_0) \leq b_\rho ||f - f_0||_2^2$ with $a_\rho = (2C_1)^{-1}$ and $b_\rho = (2C_2)^{-1}$.*

Lemma 4.3 *Assume that for all $\boldsymbol{x} \in \mathcal{X}$, $1/c_1 \leq \mu(u + \delta|\boldsymbol{x}) - \mu(u - \delta|\boldsymbol{x}) \leq 1/c_2$ for some $c_1 \geq c_2 > 0$ for all $u \in \{u \in \mathbb{R}$: $|u - f^0_{RL}(\boldsymbol{x})| \leq 2B$ or $|u - f_0(\boldsymbol{x})| \leq 2B\}$, where $\mu(u|\boldsymbol{x})$ is the conditonal cumulative function of Y given $Y|\boldsymbol{X} = \boldsymbol{x}$. Then for any $f \in \mathcal{F}(\widetilde{\phi}, m, B)$, the Huber loss given in (4.2) satisfies $a_\rho ||f - f^0_{RL}||_2^2 \leq \mathcal{E}(f) - \mathcal{E}(f^0_{RL})$ and $\mathcal{E}(f) - \mathcal{E}(f_0) \leq b_\rho ||f - f_0||_2^2$ with $a_\rho = (2c_1)^{-1}$ and $b_\rho = (2c_2)^{-1}$.*

The proofs of Lemmas 4.1-4.3 are provided in Section B.7 of the Appendix.

4.6 Simulation Studies

In this section, we conduct simulation studies to assess the finite-sample performance of the proposed methods.

Date generating process

For illustration of the methods, we generate data from the following nonlinear models:

$$\text{Model 1}: \mathbb{E}(Y_i|X_i) = X_{i1}^2 + X_{i2}^2 + 1.5\sin(\sqrt{1.5}\pi(X_{i1}+X_{i2})) + \frac{X_{i3}}{X_{i1}^2+X_{i2}^2+1} + 1;\ X_i \sim \mathcal{U}\left([0,1]^5\right);$$

$$\text{Model 2}: \mathbb{E}(Y_i|X_i) = X_{i1}X_{i2} + \frac{e^{\sin(2\pi(X_{i3}+X_{i4}))}}{1+e^{\cos(2\pi X_{i5})}} + \tan(\frac{X_{i1}}{X_{i2}^2+X_{i4}^4+2});\ X_i \sim \mathcal{U}\left([0,1]^7\right);$$

$$\text{Model 3}: \mathbb{E}(Y_i|X_i) = 1.5X_{i5}\cos(X_{i1}X_{i2}+X_{i3}+X_{i4}) + X_{i3}^2X_{i7}\sqrt{X_{i6}X_{i8}+X_{i9}+0.1} + \frac{2X_{i7}}{2+X_{i5}^2+X_{i10}^4} + 1;$$

$$X_i \sim \mathcal{U}\left([0,1]^{10}\right);$$

$$\text{Model 4}: \mathbb{E}(Y_i|X_i) = \mathbb{P}(Y_i=1|X_i) = \frac{e^{\mu_i}}{1+e^{\mu_i}};$$

$$\mu_i = X_{i5}\cos(X_{i1}X_{i2}+X_{i3}+X_{i4}) + X_{i3}^2X_{i7}\sqrt{X_{i6}X_{i8}+X_{i9}+0.1} + \frac{X_{i7}}{2+X_{i5}^2+X_{i10}^4} - 3X_{i5} + 1;$$

$$X_i \sim \mathcal{U}\left([0,1]^{10}\right);$$

For Models 1-3, we generate the responses from $Y_i = E(Y_i \mid X_i) + \epsilon_i$, and ϵ_i are independently generated from the standard normal distribution and Laplace distribution, respectively, for $1 \le i \le n$. For each setting, we run $n_{rep} = 100$ replications. For the SDRN estimator, we use $m = \max(\lfloor 0.2\log_2 n \rfloor + c, 0)$ and $R = 3\max(\lfloor 0.2\log_2 n \rfloor, m)$, which satisfy the conditions in Theorem 4.3, for different choices of c. Let the tuning parameter for the ridge penalty be $\lambda = \kappa n^{-1}$.

For Models 1-3, we evaluate the estimated function based on the same set of the covariate values $x_i^*(1 \le i \le n)$, which are independently generated from $\mathcal{U}([0,1]^p)$. Let

$\widehat{f}(x_i^*)$ be the estimate of the target function $f(x_i^*)$. We report

$$\begin{aligned}
\text{average bias}^2 &= \frac{1}{n}\sum\nolimits_{i=1}^{n}\{\frac{1}{n_{rep}}\sum\nolimits_{j=1}^{n_{rep}}\widehat{f}(x_i^*) - f(x_i^*)\}^2, \\
\text{average variance} &= \frac{1}{n}\sum\nolimits_{i=1}^{n}\{\frac{1}{n_{rep}}\sum\nolimits_{j=1}^{n_{rep}}\widehat{f}(x_i^*)^2 - (\frac{1}{n_{rep}}\sum\nolimits_{j=1}^{n_{rep}}\widehat{f}(x_i^*))^2\}, \\
\text{average mse} &= \frac{1}{n}\sum\nolimits_{i=1}^{n}\frac{1}{n_{rep}}\sum\nolimits_{j=1}^{n_{rep}}\{\widehat{f}(x_i^*) - f(x_i^*)\}^2.
\end{aligned}$$

Tables 4.1 and 4.2 report the average mean squared error (MSE), average bias^2 and average variance the SDRN estimators obtained from the quadratic and quantile ($\tau = 0.5,\ 0.25$) loss functions for Model 1 when $n = 2000, 5000$. We let κ equal 0.1, 0.5, 1, 2, 4 and c equal -2, -1, 0, 1, 2, respectively. From Table 4.1 with $n = 2000$, we observe that when the value of κ is fixed, the increase of the c value results in an overall trend of decreasing bias^2 and increasing variance. When c is too small (for example, $c = -2$), the estimator can have a large bias due to possible underfitting. For a larger value of c, it correspondingly needs a larger value of κ for the ridge penalty to prevent overfitting. A good choice of $(\kappa,\ c)$ leads to an optimal fitting with the smallest MSE. The smallest MSE value for each case is highlighted in bold and red, corresponding to the optimal fitting. We see that the estimate with the smallest MSE for each case achieves a good balance between the bias^2 and variance. Moreover, when the error is generated from the Laplace distribution, the estimate from the quantile ($\tau = 0.5$) loss, which is a robust estimate, has smaller MSE compared to that obtained from the quadratic loss. Table 4.2 shows the results for $n = 5000$. We observe similar patterns as Table 4.1. Clearly, when n increases, the MSE values become smaller. This corroborates our convergence results in Theorem 4.3. Tables 4.3 and 4.4 show the average MSE, average bias^2 and average variance for Model 2 when $n = 2000, 5000$. The results in Model 2 show similar pattens as those observed from Table 4.1 for Model 1.

			quadratic					**quantile** ($\tau = 0.5$)					**quantile** ($\tau = 0.25$)				
	κ		$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$	$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$	$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$
Normal error	0.1	bias2	0.4970	0.0879	0.0206	0.0096	0.0146	0.5036	0.0932	0.0258	0.0120	0.0143	0.5723	0.0949	0.0265	0.0190	0.0681
		var	0.0235	0.0504	0.1331	0.3420	0.7154	0.0301	0.0530	0.1300	0.3031	0.6141	0.0367	0.0604	0.1415	0.3069	0.5728
		mse	0.5205	0.1383	0.1538	0.3516	0.7300	0.5336	0.1463	0.1558	0.3151	0.6283	0.6090	0.1553	0.1680	0.3259	0.6408
	0.5	bias2	0.5019	0.1046	0.0341	0.0178	0.0194	0.5303	0.1563	0.0582	0.0299	0.0253	0.6317	0.1714	0.0607	0.0291	0.0321
		var	0.0181	0.0328	0.0746	0.1594	0.2975	0.0179	0.0304	0.0633	0.1285	0.2450	0.0192	0.0331	0.0667	0.1297	0.2334
		mse	0.5199	0.1375	0.1087	0.1772	0.3169	0.5483	0.1867	**0.1215**	0.1584	0.2703	0.6509	0.2046	**0.1274**	0.1588	0.2655
	1	bias2	0.5100	0.1347	0.0498	0.0273	0.0262	0.5718	0.2449	0.0998	0.0520	0.0405	0.6922	0.2805	0.1078	0.0522	0.0402
		var	0.0146	0.0257	0.0544	0.1101	0.1997	0.0126	0.0229	0.0439	0.0848	0.1540	0.0126	0.0238	0.0452	0.0850	0.1506
		mse	0.5246	0.1604	**0.1042**	0.1374	0.2259	0.5843	0.2678	0.1436	0.1369	0.1945	0.7048	0.3044	0.1530	0.1372	0.1907
	2	bias2	0.5301	0.1980	0.0811	0.0448	0.0387	0.6567	0.3917	0.1792	0.0949	0.0695	0.7878	0.4587	0.2013	0.1007	0.0685
		var	0.0110	0.0195	0.0384	0.0741	0.1314	0.0080	0.0166	0.0296	0.0545	0.0964	0.0075	0.0160	0.0298	0.0541	0.0934
		mse	0.5411	0.2175	0.1195	0.1189	0.1701	0.6647	0.4083	0.2088	0.1495	0.1658	0.7954	0.4747	0.2311	0.1547	0.1620
	4	bias2	0.5766	0.3063	0.1395	0.0769	0.0610	0.8157	0.5864	0.3171	0.1776	0.1245	0.9401	0.6628	0.3666	0.1969	0.1290
		var	0.0076	0.0144	0.0264	0.0487	0.0845	0.0046	0.0111	0.0194	0.0343	0.0590	0.0040	0.0098	0.0186	0.0333	0.0577
		mse	0.5843	0.3207	0.1659	0.1256	0.1456	0.8203	0.5975	0.3365	0.2119	0.1834	0.9441	0.6727	0.3851	0.2303	0.1867
Laplace error	0.1	bias2	0.4989	0.0894	0.0225	0.0131	0.0220	0.5079	0.0940	0.0273	0.0136	0.0176	0.5950	0.1035	0.0308	0.0136	0.0369
		var	0.0370	0.0948	0.2634	0.6866	1.4377	0.0327	0.0560	0.1388	0.3628	0.8235	0.0443	0.0851	0.2007	0.4450	0.8621
		mse	0.5359	0.1842	0.2859	0.6998	1.4597	0.5406	0.1501	0.1661	0.3764	0.8411	0.6393	0.1886	0.2315	0.4586	0.8990
	0.5	bias2	0.5037	0.1068	0.0362	0.0195	0.0230	0.5370	0.1600	0.0613	0.0328	0.0287	0.6609	0.2065	0.0767	0.0361	0.0290
		var	0.0285	0.0611	0.1446	0.3141	0.5887	0.0192	0.0329	0.0646	0.1356	0.2757	0.0226	0.0470	0.0926	0.1785	0.3214
		mse	0.5322	0.1679	0.1807	0.3337	0.6118	0.5562	0.1929	**0.1259**	0.1684	0.3044	0.6834	0.2535	**0.1693**	0.2147	0.3504
	1	bias2	0.5117	0.1373	0.0521	0.0290	0.0291	0.5809	0.2552	0.1050	0.0559	0.0444	0.7295	0.3370	0.1361	0.0668	0.0462
		var	0.0231	0.0471	0.1040	0.2144	0.3914	0.0133	0.0255	0.0448	0.0866	0.1641	0.0146	0.0323	0.0622	0.1160	0.2030
		mse	0.5348	0.1844	0.1561	0.2434	0.4205	0.5942	0.2806	0.1498	0.1425	0.2085	0.7440	0.3693	0.1982	0.1828	0.2492
	2	bias2	0.5316	0.2012	0.0839	0.0468	0.0410	0.6705	0.4119	0.1883	0.1006	0.0745	0.8384	0.5216	0.2493	0.1281	0.0850
		var	0.0174	0.0349	0.0720	0.1422	0.2546	0.0085	0.0185	0.0309	0.0551	0.0987	0.0087	0.0206	0.0396	0.0731	0.1258
		mse	0.5490	0.2361	**0.1558**	0.1890	0.2956	0.6790	0.4305	0.2191	0.1557	0.1733	0.8471	0.5423	0.2888	0.2012	0.2108
	4	bias2	0.5777	0.3097	0.1428	0.0794	0.0632	0.8425	0.6109	0.3352	0.1875	0.1325	1.0066	0.7201	0.4291	0.2448	0.1616
		var	0.0121	0.0249	0.0483	0.0917	0.1615	0.0048	0.0121	0.0208	0.0352	0.0594	0.0046	0.0123	0.0238	0.0438	0.0775
		mse	0.5898	0.3347	0.1911	0.1711	0.2246	0.8473	0.6230	0.3560	0.2227	0.1919	1.0111	0.7324	0.4528	0.2886	0.2391

Table 4.1: The average MSE, bias^2 and variance of the SDRN estimators obtained from the quadratic and quantile ($\tau = 0.5,\ 0.25$) loss functions based on the 100 simulation replications when $n = 2000$ for Model 1.

			quadratic					**quantile** ($\tau = 0.5$)					**quantile** ($\tau = 0.25$)				
	κ		$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$	$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$	$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$
Normal error	0.1	bias2	0.4761	0.0921	0.0163	0.0049	0.0070	0.4789	0.0930	0.0185	0.0063	0.0073	0.5261	0.0944	0.0191	0.0075	0.0250
		var	0.0099	0.0222	0.0608	0.1666	0.4315	0.0138	0.0273	0.0690	0.1735	0.4026	0.0171	0.0307	0.0774	0.1854	0.4014
		mse	0.4860	0.1143	0.0770	0.1715	0.4384	0.4927	0.1203	0.0875	0.1798	0.4099	0.5432	0.1251	0.0965	0.1929	0.4263
	0.5	bias2	0.4765	0.0957	0.0222	0.0087	0.0087	0.4830	0.1105	0.0324	0.0146	0.0120	0.5446	0.1151	0.0338	0.0143	0.0140
		var	0.0086	0.0170	0.0407	0.0944	0.2018	0.0102	0.0179	0.0398	0.0861	0.1780	0.0112	0.0195	0.0431	0.0899	0.1794
		mse	0.4851	0.1127	0.0630	0.1031	0.2105	0.4933	0.1284	**0.0723**	0.1007	0.1900	0.5558	0.1346	**0.0770**	0.1042	0.1934
	1	bias2	0.4775	0.1041	0.0290	0.0129	0.0118	0.4920	0.1422	0.0486	0.0241	0.0188	0.5677	0.1531	0.0514	0.0234	0.0179
		var	0.0076	0.0141	0.0318	0.0693	0.1400	0.0081	0.0139	0.0290	0.0599	0.1166	0.0083	0.0148	0.0309	0.0618	0.1178
		mse	0.4851	0.1182	**0.0608**	0.0822	0.1517	0.5001	0.1561	0.0776	0.0839	0.1354	0.5760	0.1680	0.0823	0.0852	0.1358
	2	bias2	0.4807	0.1259	0.0414	0.0207	0.0175	0.5161	0.2106	0.0807	0.0417	0.0311	0.6098	0.2347	0.0871	0.0418	0.0298
		var	0.0063	0.0112	0.0238	0.0490	0.0947	0.0060	0.0104	0.0204	0.0402	0.0748	0.0057	0.0108	0.0211	0.0409	0.0757
		mse	0.4870	0.1371	0.0652	0.0697	0.1122	0.5221	0.2210	0.1011	0.0820	0.1059	0.6155	0.2455	0.1082	0.0828	0.1055
	4	bias2	0.4907	0.1744	0.0656	0.0346	0.0275	0.5721	0.3294	0.1421	0.0744	0.0537	0.6815	0.3751	0.1575	0.0781	0.0531
		var	0.0049	0.0086	0.0171	0.0336	0.0624	0.0041	0.0076	0.0138	0.0262	0.0472	0.0035	0.0076	0.0139	0.0261	0.0469
		mse	0.4956	0.1830	0.0827	0.0682	0.0899	0.5761	0.3370	0.1559	0.1006	0.1010	0.6851	0.3827	0.1714	0.1042	0.1000
Laplace error	0.1	bias2	0.4759	0.0920	0.0168	0.0062	0.0112	0.4811	0.0928	0.0184	0.0066	0.0089	0.5514	0.0976	0.0213	0.0077	0.0118
		var	0.0162	0.0418	0.1194	0.3319	0.8607	0.0166	0.0274	0.0680	0.1889	0.4871	0.0237	0.0445	0.1130	0.2686	0.5857
		mse	0.4920	0.1338	0.1362	0.3381	0.8719	0.4976	0.1202	0.0864	0.1955	0.4960	0.5751	0.1421	0.1343	0.2763	0.5976
	0.5	bias2	0.4763	0.0953	0.0223	0.0093	0.0103	0.4863	0.1086	0.0316	0.0145	0.0127	0.5737	0.1257	0.0392	0.0166	0.0128
		var	0.0143	0.0318	0.0797	0.1868	0.4003	0.0121	0.0181	0.0386	0.0864	0.1917	0.0153	0.0283	0.0615	0.1269	0.2455
		mse	0.4906	0.1272	0.1020	0.1961	0.4106	0.4985	0.1267	**0.0702**	0.1009	0.2043	0.5889	0.1541	**0.1006**	0.1435	0.2583
	1	bias2	0.4774	0.1035	0.0288	0.0133	0.0127	0.4967	0.1402	0.0472	0.0236	0.0191	0.6008	0.1764	0.0613	0.0283	0.0199
		var	0.0127	0.0264	0.0621	0.1368	0.2767	0.0096	0.0145	0.0284	0.0588	0.1235	0.0111	0.0216	0.0435	0.0862	0.1603
		mse	0.4901	0.1299	0.0909	0.1501	0.2894	0.5063	0.1547	0.0756	0.0824	0.1426	0.6120	0.1980	0.1049	0.1145	0.1802
	2	bias2	0.4808	0.1251	0.0411	0.0208	0.0179	0.5235	0.2118	0.0792	0.0409	0.0313	0.6499	0.2778	0.1067	0.0514	0.0349
		var	0.0106	0.0209	0.0461	0.0963	0.1863	0.0070	0.0114	0.0203	0.0390	0.0742	0.0074	0.0155	0.0296	0.0565	0.1019
		mse	0.4914	0.1459	**0.0872**	0.1171	0.2042	0.5305	0.2232	0.0995	0.0799	0.1055	0.6572	0.2933	0.1363	0.1079	0.1368
	4	bias2	0.4911	0.1734	0.0651	0.0344	0.0275	0.5847	0.3397	0.1421	0.0734	0.0536	0.7315	0.4324	0.1941	0.0968	0.0644
		var	0.0083	0.0159	0.0328	0.0654	0.1222	0.0046	0.0088	0.0144	0.0256	0.0461	0.0045	0.0104	0.0194	0.0357	0.0638
		mse	0.4994	0.1893	0.0979	0.0999	0.1496	0.5893	0.3484	0.1565	0.0990	0.0996	0.7360	0.4428	0.2135	0.1324	0.1282

Table 4.2: The average MSE, bias^2 and variance of the SDRN estimators obtained from the quadratic and quantile ($\tau = 0.5,\ 0.25$) loss functions based on the 100 simulation replications when $n = 5000$ for Model 1.

			quadratic					**quantile** ($\tau = 0.5$)					**quantile** ($\tau = 0.25$)				
	κ		$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$	$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$	$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$
Normal error	0.1	bias2	0.2400	0.1005	0.0936	0.0462	0.0392	0.2423	0.1098	0.0971	0.0512	0.0392	0.2500	0.1163	0.1058	0.1105	0.1629
		var	0.0414	0.1149	0.2878	0.5325	0.6728	0.0357	0.0949	0.2274	0.4545	0.6739	0.0358	0.0943	0.2189	0.4127	0.6366
		mse	0.2814	0.2154	0.3814	0.5787	0.7119	0.2780	0.2047	0.3246	0.5058	0.7131	0.2858	0.2106	0.3247	0.5232	0.7995
	0.5	bias2	0.2417	0.1222	0.1033	0.0644	0.0476	0.2568	0.1532	0.1189	0.0791	0.0538	0.2625	0.1649	0.1216	0.0870	0.0918
		var	0.0184	0.0479	0.1096	0.2110	0.3344	0.0120	0.0349	0.0794	0.1617	0.3090	0.0111	0.0334	0.0757	0.1534	0.3129
		mse	0.2601	0.1701	0.2129	0.2754	0.3820	0.2688	0.1881	0.1983	0.2408	0.3628	0.2737	0.1983	0.1973	0.2404	0.4048
	1	bias2	0.2470	0.1426	0.1138	0.0772	0.0563	0.2877	0.1822	0.1407	0.0990	0.0690	0.2814	0.1934	0.1438	0.1003	0.0802
		var	0.0113	0.0310	0.0697	0.1363	0.2282	0.0063	0.0210	0.0482	0.0985	0.1842	0.0056	0.0196	0.0453	0.0925	0.2465
		mse	0.2583	0.1735	0.1834	0.2135	0.2845	0.2940	0.2031	0.1888	0.1975	0.2532	0.2871	0.2130	0.1892	0.1928	0.3267
	2	bias2	0.2639	0.1687	0.1315	0.0941	0.0695	0.3649	0.2154	0.1720	0.1271	0.0922	0.3156	0.2226	0.1731	0.1260	0.0926
		var	0.0063	0.0192	0.0432	0.0855	0.1487	0.0030	0.0118	0.0282	0.0580	0.1100	0.0025	0.0108	0.0260	0.0562	0.1547
		mse	0.2702	0.1879	0.1747	0.1796	0.2182	0.3679	0.2272	0.2002	**0.1850**	0.2023	0.3181	0.2334	0.1991	**0.1822**	0.2473
	4	bias2	0.3108	0.1982	0.1582	0.1177	0.0884	0.5242	0.2598	0.2112	0.1657	0.1246	0.3629	0.2551	0.2051	0.1583	0.1209
		var	0.0032	0.0113	0.0260	0.0518	0.0926	0.0013	0.0062	0.0157	0.0329	0.0696	0.0009	0.0054	0.0141	0.0339	0.1028
		mse	0.3140	0.2095	0.1842	**0.1696**	0.1810	0.5255	0.2660	0.2269	0.1986	0.1942	0.3639	0.2606	0.2193	0.1922	0.2237
Laplace error	0.1	bias2	0.2410	0.1021	0.0946	0.0521	0.0448	0.2431	0.1104	0.0966	0.0535	0.0449	0.2559	0.1258	0.1019	0.0822	0.1605
		var	0.0774	0.2202	0.5546	1.0523	1.3398	0.0382	0.1015	0.2661	0.5975	1.0618	0.0466	0.1294	0.3051	0.6110	0.9585
		mse	0.3184	0.3224	0.6492	1.1044	1.3846	0.2813	0.2118	0.3627	0.6510	1.1067	0.3025	0.2552	0.4070	0.6932	1.1189
	0.5	bias2	0.2416	0.1231	0.1033	0.0661	0.0501	0.2595	0.1557	0.1196	0.0803	0.0580	0.2693	0.1785	0.1307	0.0869	0.0661
		var	0.0350	0.0917	0.2105	0.4144	0.6626	0.0124	0.0362	0.0847	0.1821	0.3908	0.0139	0.0440	0.1018	0.2093	0.4346
		mse	0.2766	0.2148	0.3139	0.4805	0.7127	0.2719	0.1919	0.2043	0.2624	0.4488	0.2832	0.2224	0.2325	0.2962	0.5006
	1	bias2	0.2461	0.1431	0.1137	0.0779	0.0578	0.2915	0.1855	0.1424	0.1002	0.0719	0.2912	0.2061	0.1562	0.1075	0.0775
		var	0.0217	0.0592	0.1336	0.2664	0.4505	0.0065	0.0218	0.0506	0.1063	0.2153	0.0069	0.0253	0.0601	0.1257	0.3346
		mse	0.2678	**0.2023**	0.2473	0.3443	0.5083	0.2980	0.2073	0.1930	0.2066	0.2872	0.2982	0.2314	0.2162	0.2332	0.4121
	2	bias2	0.2619	0.1688	0.1315	0.0943	0.0702	0.3702	0.2192	0.1750	0.1291	0.0938	0.3284	0.2345	0.1863	0.1380	0.0985
		var	0.0122	0.0366	0.0825	0.1660	0.2923	0.0031	0.0123	0.0295	0.0612	0.1209	0.0031	0.0135	0.0338	0.0749	0.1876
		mse	0.2742	0.2054	0.2139	0.2603	0.3625	0.3733	0.2316	0.2046	**0.1903**	0.2147	0.3314	0.2481	0.2202	**0.2130**	0.2861
	4	bias2	0.3076	0.1976	0.1580	0.1177	0.0886	0.5337	0.2647	0.2149	0.1693	0.1273	0.3748	0.2680	0.2178	0.1752	0.1332
		var	0.0063	0.0216	0.0494	0.0998	0.1808	0.0013	0.0064	0.0165	0.0344	0.0712	0.0012	0.0067	0.0180	0.0436	0.0918
		mse	0.3139	0.2191	0.2074	0.2175	0.2694	0.5351	0.2711	0.2315	0.2036	0.1985	0.3759	0.2747	0.2358	0.2188	0.2250

Table 4.3: The average MSE, bias^2 and variance of the SDRN estimators obtained from the quadratic and quantile ($\tau = 0.5,\ 0.25$) loss functions based on the 100 simulation replications when $n = 2000$ for Model 2.

			quadratic					**quantile** ($\tau = 0.5$)					**quantile** ($\tau = 0.25$)				
	κ		$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$	$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$	$c=-2$	$c=-1$	$c=0$	$c=1$	$c=2$
Normal error	0.1	bias2	0.2328	0.0938	0.0851	0.0282	0.0210	0.2343	0.0970	0.0863	0.0316	0.0211	0.2432	0.1011	0.0900	0.0628	0.1419
		var	0.0226	0.0677	0.1927	0.4270	0.7133	0.0235	0.0636	0.1660	0.3637	0.6593	0.0244	0.0652	0.1670	0.3485	0.5866
		mse	0.2554	0.1614	0.2777	0.4552	0.7343	0.2578	0.1605	0.2524	0.3952	0.6804	0.2676	0.1663	0.2570	0.4114	0.7285
	0.5	bias2	0.2335	0.1020	0.0886	0.0389	0.0240	0.2389	0.1198	0.0961	0.0517	0.0289	0.2476	0.1293	0.0989	0.0591	0.0645
		var	0.0130	0.0330	0.0816	0.1686	0.3035	0.0104	0.0273	0.0638	0.1341	0.2846	0.0104	0.0272	0.0631	0.1323	0.3053
		mse	0.2465	0.1350	0.1702	0.2075	0.3275	0.2493	0.1471	0.1599	0.1858	0.3135	0.2579	0.1565	0.1620	0.1914	0.3698
	1	bias2	0.2350	0.1130	0.0933	0.0489	0.0292	0.2476	0.1412	0.1070	0.0664	0.0396	0.2547	0.1528	0.1108	0.0700	0.0478
		var	0.0090	0.0226	0.0537	0.1102	0.2014	0.0063	0.0177	0.0403	0.0840	0.1811	0.0061	0.0173	0.0394	0.0824	0.2271
		mse	0.2440	0.1357	0.1470	0.1591	0.2305	0.2538	0.1589	0.1473	0.1504	0.2207	0.2608	0.1701	0.1503	0.1524	0.2749
	2	bias2	0.2395	0.1310	0.1018	0.0620	0.0380	0.2709	0.1686	0.1259	0.0849	0.0546	0.2705	0.1806	0.1310	0.0872	0.0577
		var	0.0057	0.0149	0.0343	0.0703	0.1300	0.0034	0.0109	0.0248	0.0510	0.0982	0.0031	0.0104	0.0239	0.0506	0.1100
		mse	0.2451	0.1459	0.1362	0.1323	0.1679	0.2743	0.1795	0.1507	0.1360	0.1528	0.2736	0.1910	0.1549	**0.1378**	0.1677
	4	bias2	0.2524	0.1554	0.1168	0.0782	0.0512	0.3296	0.2001	0.1547	0.1102	0.0763	0.3004	0.2095	0.1589	0.1117	0.0770
		var	0.0032	0.0094	0.0214	0.0437	0.0813	0.0017	0.0063	0.0148	0.0302	0.0575	0.0014	0.0058	0.0139	0.0301	0.0719
		mse	0.2556	0.1648	0.1382	**0.1219**	0.1325	0.3313	0.2064	0.1695	0.1404	**0.1337**	0.3018	0.2153	0.1728	0.1418	0.1489
Laplace error	0.1	bias2	0.2329	0.0937	0.0862	0.0326	0.0281	0.2356	0.0963	0.0860	0.0334	0.0248	0.2495	0.1072	0.0920	0.0467	0.0884
		var	0.0407	0.1280	0.3696	0.8463	1.4189	0.0244	0.0647	0.1831	0.4419	0.9316	0.0301	0.0883	0.2283	0.4951	0.8929
		mse	0.2736	0.2218	0.4558	0.8790	1.4470	0.2600	0.1610	0.2691	0.4753	0.9564	0.2796	0.1955	0.3203	0.5417	0.9813
	0.5	bias2	0.2336	0.1019	0.0890	0.0405	0.0270	0.2413	0.1195	0.0956	0.0522	0.0308	0.2545	0.1409	0.1057	0.0613	0.0432
		var	0.0233	0.0620	0.1562	0.3317	0.5988	0.0105	0.0266	0.0650	0.1426	0.3189	0.0121	0.0350	0.0824	0.1769	0.3833
		mse	0.2569	0.1638	0.2452	0.3722	0.6257	0.2519	0.1461	0.1605	0.1948	0.3498	0.2666	0.1759	0.1881	0.2382	0.4264
	1	bias2	0.2352	0.1129	0.0935	0.0499	0.0311	0.2513	0.1417	0.1064	0.0664	0.0408	0.2633	0.1663	0.1203	0.0765	0.0488
		var	0.0162	0.0422	0.1024	0.2157	0.3966	0.0063	0.0171	0.0402	0.0858	0.2053	0.0070	0.0218	0.0509	0.1080	0.2877
		mse	0.2514	**0.1551**	0.1959	0.2656	0.4277	0.2576	0.1588	0.1466	0.1522	0.2461	0.2703	0.1881	0.1711	0.1845	0.3365
	2	bias2	0.2398	0.1310	0.1019	0.0626	0.0392	0.2766	0.1704	0.1257	0.0846	0.0552	0.2823	0.1945	0.1434	0.0972	0.0636
		var	0.0103	0.0277	0.0651	0.1367	0.2551	0.0034	0.0106	0.0243	0.0509	0.1027	0.0036	0.0128	0.0304	0.0642	0.1811
		mse	0.2501	0.1586	0.1669	0.1993	0.2943	0.2800	0.1810	0.1500	0.1354	0.1579	0.2859	0.2073	0.1738	**0.1613**	0.2447
	4	bias2	0.2530	0.1556	0.1168	0.0786	0.0519	0.3377	0.2036	0.1554	0.1098	0.0763	0.3157	0.2235	0.1731	0.1251	0.0866
		var	0.0059	0.0173	0.0401	0.0841	0.1588	0.0017	0.0062	0.0144	0.0297	0.0586	0.0017	0.0070	0.0174	0.0386	0.0943
		mse	0.2589	0.1729	0.1569	0.1626	0.2108	0.3394	0.2099	0.1698	0.1395	**0.1349**	0.3174	0.2306	0.1905	0.1637	0.1809

Table 4.4: The average MSE, bias^2 and variance of the SDRN estimators obtained from the quadratic and quantile ($\tau = 0.5,\ 0.25$) loss functions based on the 100 simulation replications when $n = 5000$ for Model 2.

	Quadratic (Normal)						**Quantile** (Laplace)					
	SDRN	FNN	Kernel	GAM	GBM	RF	**SDRN**	FNN	Kernel	GAM	GBM	RF
bias2	**0.0188**	0.0164	0.0359	0.0438	0.0308	0.0339	**0.0225**	0.0164	0.0434	0.0440	0.0362	0.0375
var	**0.0275**	0.0335	0.0413	0.0139	0.0260	0.0188	**0.0289**	0.0366	0.0437	0.0112	0.0262	0.0229
mse	**0.0462**	0.0499	0.0773	0.0578	0.0568	0.0527	**0.0514**	0.0530	0.0871	0.0552	0.0623	0.0604

Table 4.5: The average MSE, bias^2 and variance of the six methods obtained from the quadratic loss for normal error and quantile ($\tau = 0.5$) loss for Laplace error based on the 100 simulation replications when $n = 2000$ for Model 3.

Next, we use Model 3 to compare the performance of our proposed SDRN estimator with that of four other popular nonparametric methods, including the fully-connected feedforward neural networks (FNN), the local linear kernel regression (Kernel), the generalized additive models (GAM), the gradient boosted machines (GBM) and the random forests (RF). For FNN, ReLU is used as the activation function. For GAM, we use a cubic regression spline basis. For all methods, we report the results from the optimal fitting with the optimal tuning parameters that minimize the MSE value based on a grid search. Table 4.5 reports the average MSE, bias^2 and variance for the six methods based on the 100 replicates when $n = 2000$. The quadratic loss and the quantile ($\tau = 0.5$) loss are used, respectively, for the normal and Laplace errors for all methods. We observe that our SDRN has the smallest average MSEs under both settings. Among all methods, the GAM method has the largest bias due to model misspecification, and the Kernel has the largest variance due to the dimensionality problem.

For Model 4, we use the metrics, accuracy, sensitivity, specificity, precision, recall and F1 score, to evaluate the classification performance of different methods. The estimates of parameters are obtained from the training dataset, and the evaluation is performed on the

n=2000	**SDRN**	FNN	GAM	GBM	RF
Accuracy	**0.9328**	0.9301	0.9254	0.9287	0.9214
Sensitivity/Recall	**0.9309**	0.9333	0.9202	0.9260	0.9232
Specificity	**0.9347**	0.9268	0.9308	0.9314	0.9197
Precision	**0.9374**	0.9294	0.9312	0.9335	0.9234
F1	**0.9327**	0.9303	0.9253	0.9287	0.9217
n=5000	**SDRN**	FNN	GAM	GBM	RF
Accuracy	**0.9568**	0.9472	0.9508	0.9442	0.9379
Sensitivity/Recall	**0.9578**	0.9573	0.9492	0.9400	0.9327
Specificity	**0.9558**	0.9366	0.9526	0.9488	0.9434
Precision	**0.9593**	0.9425	0.9554	0.9519	0.9470
F1	**0.9580**	0.9492	0.9521	0.9455	0.9392

Table 4.6: The average of accuracy, sensitivity, precision, recall and F1 score of the five methods based on the 100 simulation replications for Model 4.

test dataset. The training and test datasets are generated independently from Model 4 with the same sample size. For all methods, we report the results from the optimal fitting with the optimal tuning parameters that minimize the prediction error based on a grid search. Table 4.6 shows the average value of accuracy, sensitivity, specificity, precision, recall and F1 score based on 100 replications for the SDRN, FNN, GAM, GBM and RF methods using logistic loss. We observe that SDRN outperforms other methods in terms of accuracy and F1 score. The F1 score conveys the balance between the precision and the recall. When the sample size n is increased from 2000 to 5000, the performance of all methods is improved.

4.7 Real Data Application

In this section, we illustrate our proposed method by using two datasets with continuous response variables (Boston housing data and Abalone data) and two datasets with binary responses (Haberman's survival data and BUPA data). Each dataset is randomly split into 75% training data and 25% test data. The training data is used to fit the model, whereas the test data is used to examine the prediction accuracy. Then, we compare our SDRN with five methods, including LM/GLM (linear model/generalized linear model), FNN, GBM, RF and GAM. For all methods, the tuning parameters are selected by 5-fold cross validations based on a grid search.

4.7.1 Boston Housing Data

The Boston housing data set [33] is available in the R package (mlbench). It contains 506 census tracts of Boston from the 1970 census. Each census tract represents one observation. Thus, there are 506 observations and 14 attributes in the dataset, where MEDV (the median value of owner-occupied homes) is the response variable. Following [25], seven explanatory variables are considered: CRIM (per capita crime rate by town), RM (average number of rooms per dwelling), tax (full-value property-tax rate per USD 10,000), NOX (nitric oxides concentration in parts per 10 million), PTRATIO (pupil-teacher ratio by town), AGE (proportion of owner-occupied units built prior to 1940) and LSTAT (percentage of lower status of the population). Since the value of the MEDV variable is censored at 50.0 (corresponding to a median price of $50,000), we remove the 16 censored observations and use the remaining 490 observations for analysis.

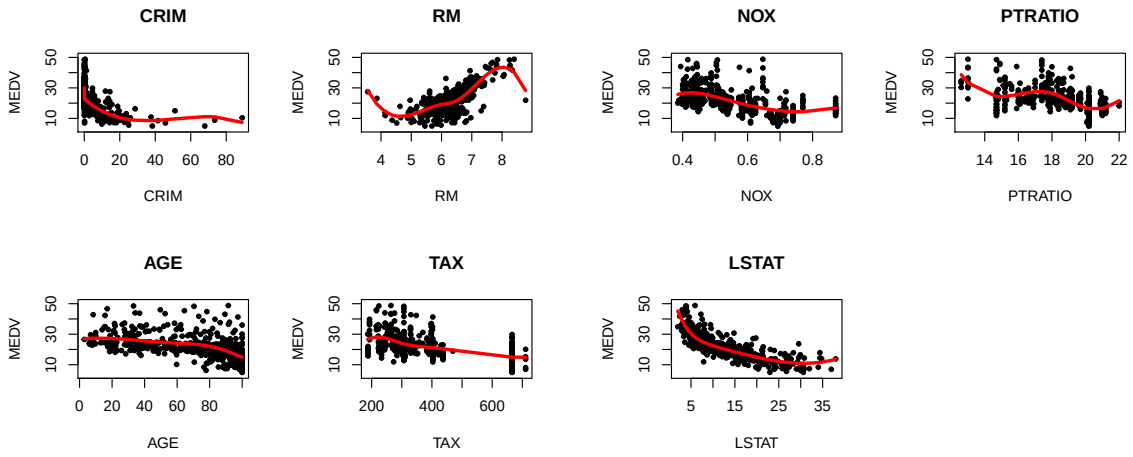


Figure 4.4: Scatter plot of MEDV versus each covariate, where the red line represents the fitted mean curve by using cubic B-splines.

For preliminary analysis of nonlinear patterns, Figure 4.4 shows the scatter plots of the response MEDV against each covariate with the red lines representing the fitted mean curves by using cubic B-splines. We observe that the MEDV value has a clear nonlinear changing pattern with these covariates. The MEDV value has an overall increasing pattern with RM, whereas it decreases as CRIM, NOX, PTRATIO, TAX and LSTAT increase. The MEDV value starts decreasing slowly as AGE increases. However, when the AGE passes 60, it starts dropping dramatically.

Next, we use our SDRN method with quadratic loss to fit a mean regression of this data, and compare it with LM, FNN, GBM, RF and GAM methods. Table 4.7 shows the mean squared prediction error (MSPE) from the six methods. We observe that SDRN outperforms other methods with the smallest MSPE. The LM method has the largest MSPE, as it cannot capture the nonlinear relationships between MEDV and the covariates. GAM has the second largest MSPE due to its restrictive additive structure without allowing

interaction effects. The coefficient of determination R^2 for SDRN is 0.953, while it is 0.743 for LM.

	SDRN	LM	FNN	GBM	RF	GAM
MSPE	**7.316**	15.815	7.655	7.351	7.617	9.297

Table 4.7: The mean squared prediction error (MSPE) from six different methods using quadratic loss for the Boston housing data.

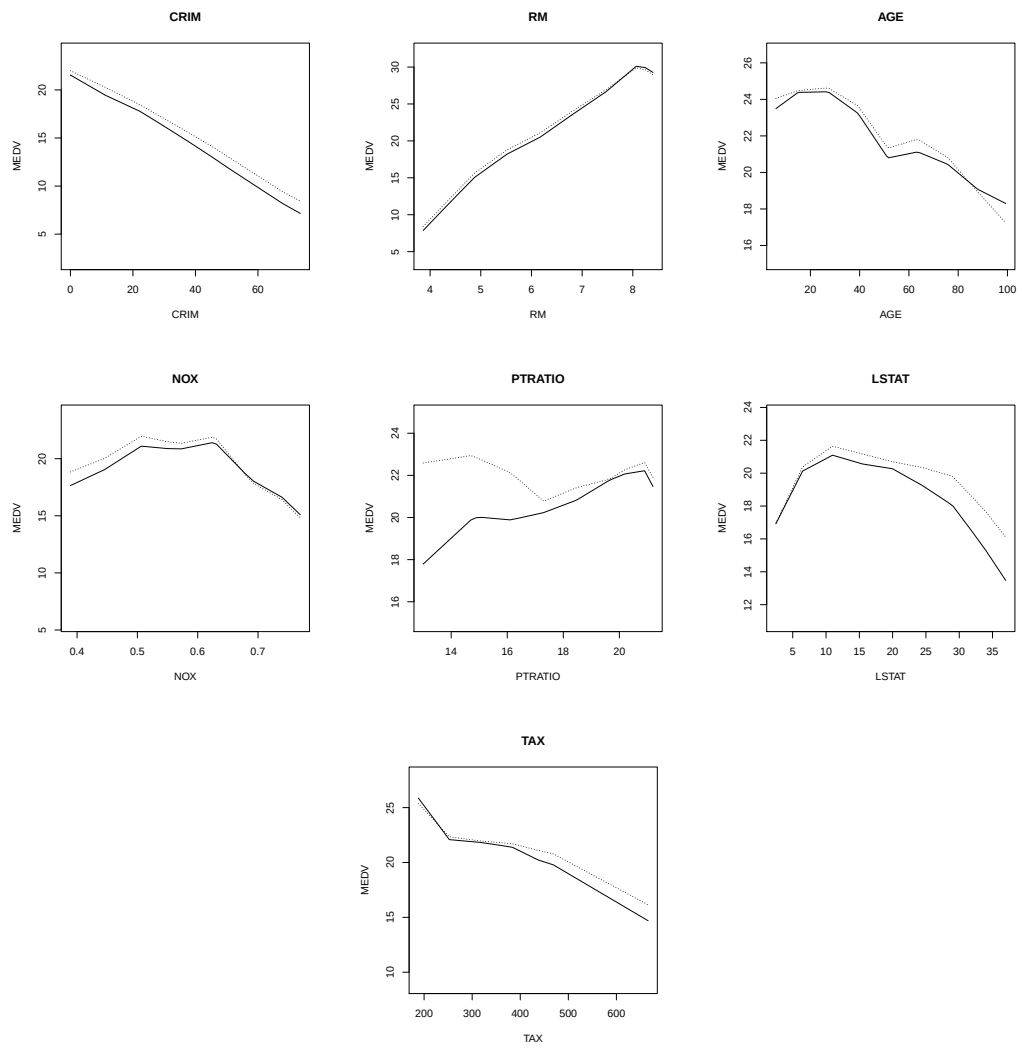


Figure 4.5: The estimated mean (solid lines) and median (dashed lines) curves of MEDV against each covariate, while other covariates are fixed at their mean values for Boston housing data.

To explore the nonlinear patterns between MEDV and each covariate, in Figure 4.5 we plot the estimated conditional mean function of MEDV versus each covariate (solid lines), and the estimated conditional median function of MEDV versus each covariate (dashed lines), obtained from our SDRD method with the quadratic loss and the quantile ($\tau = 0.5$) loss, respectively, while other covariates are fixed at their mean values. We see that the

fitted MEDV has a clear decreasing trend with CRIM, AGE and TAX, while it increases with RM. The estimated MEDV value drops steadily as CRIM is climbing while the values of other covariates are controlled, indicating that crime rates can significant impact the house prices. The relationship between MEDV and AGE is more nonlinear, although it has an overall declining pattern. The estimated MEDV maintains a relatively stable value when AGE is between 0-30, and then it begins to drop progressively after the AGE passes 30. When AGE is 50-70, it becomes stable again, and then declines after AGE passes 70. The estimated MEDV value increases a bit as NOX level increases. However, it drops sharply after the NOX value is greater than 0.6. The increasing pattern in the beginning can be explained by the fact that a higher NOX implies that a region can be more industrialized and thus has a higher home price. When the NOX value passes a certain value, the air pollution is more severe and becomes a major concern, the home prices will go down quickly. For CRIM, RM, AGE, NOX and TAX, the conditional mean and median curves are similar to each other. For PTRATIO, the median curve has a more stable value, whereas the mean curve has an increasing pattern. There is a visible difference between the two curves when the PTRATIO value is small. After PTRATIO is larger than 17, the two curves become similar to each other. The difference at the small value of PTRATIO can be caused by a few outliers, as the mean curve fitting can be more sensitive to outliers. For LSTAT, after its value is greater than 5, we can see a steady decreasing trend of MEDV as LSTAT increases. And the decreasing pattern is more dramatic as the LSTAT value becomes larger.

4.7.2 Abalone Data

The abalone dataset is available at the UCI Machine Learning Repository [22], which contains 4177 observations and 9 attributes. The attributes are: Sex (male, female and infant), Length (longest shell measurement), Diameter (perpendicular to length), Height (with meat in shell), Whole weight (whole abalone), Shucked weight (weight of meat), Viscera weight (gut weight after bleeding), Shell weight (after being dried) and Rings (+1.5 gives the age in years). The goal is to predict the age of abalone based on these physical measurements. Since the age depends on the Rings, we take the Rings as the response variable. Since Length and Diameter are highly correlated with the correlation coefficient 0.9868 and infant has no gender, we delete Length and Sex and use the remaining six covariates, Diameter, Height, Whole weight, Shucked weight, Viscera weight and Shell weight, in our analysis. In the dataset, there are two observations with zero value for Height, and two other observations are outliers, so we delete these four observations and use the remaining 4173 observations in our analysis.

For exploratory analysis, Figure 4.6 depicts the scatter plots of the response variable Rings versus Diameter, Height, Whole weight and Shell weight, and the fitted mean curves using cubic B-splines. Clearly, the response has an increasing pattern with these covariates. It has a stronger nonlinear relationship with Whole weight and Shell weight. Table 4.8 presents the MSPE values in the test data for six different methods using the quadratic loss. We observe that SDRN has a slightly smaller MSPE value than other methods. The coefficient of determination R^2 obtained from SDRN is 0.587. It is larger than the R^2 from LM, which is 0.533, due to a clear nonlinear pattern between the response and some of the

covariates. Additionally, Figure 4.7 depicts the fitted mean curves (solid lines) and median curves (dashed lines) of the response Rings versus each of the four covariates obtained from our SDRN method with the quadratic loss and quantile ($\tau = 0.5$) loss, respectively, while other covariates are fixed at their mean values. We see an overall increasing trend of the fitted lines for the four covariates. For Diameter, Whole.weight and Shell.weight, the fitted value of Rings increases steadily as the covariate value increases. However, after the covariate value is beyond a certain point, the estimated value of Rings becomes stable. For Height, the estimated value of Rings increases with Height in the beginning stage, it becomes stable when Height is from 0.7-0.13, and then it increases again. Moreover, the estimated conditional mean function is similar to the estimated conditional median function in general for this dataset.

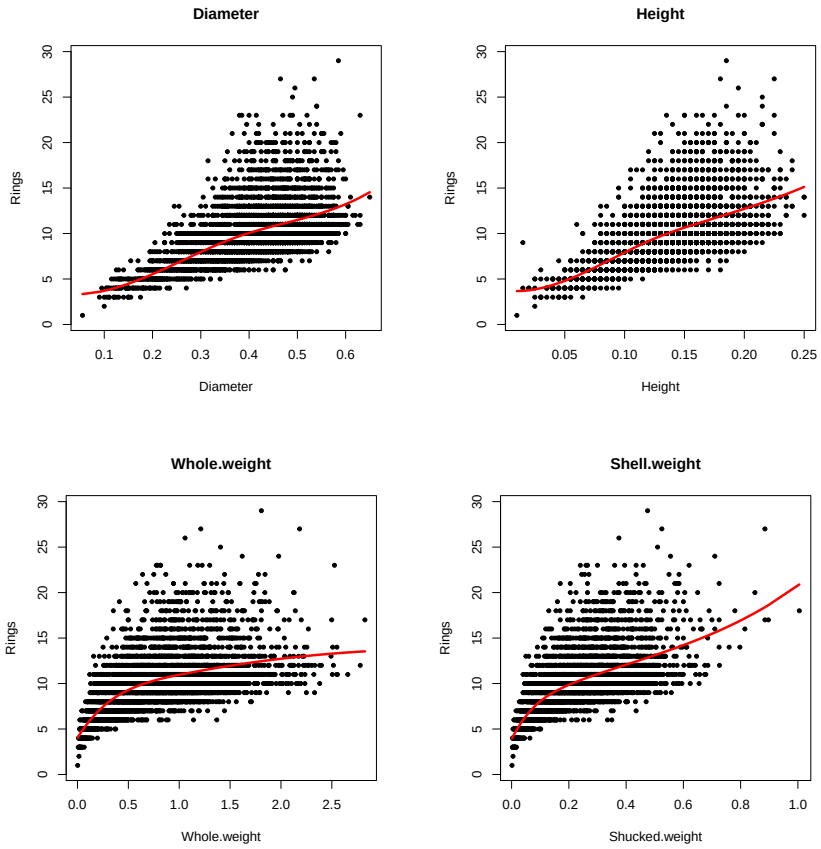


Figure 4.6: Scatter plots of the response Rings versus four covariates and the fitted mean curve using cubic B-splines.

	SDRN	LM	FNN	GBM	RF	GAM
MSPE	**4.414**	4.957	4.482	4.636	4.564	4.560

Table 4.8: The mean squared prediction error (MSPE) from the six different methods for the abalone data.

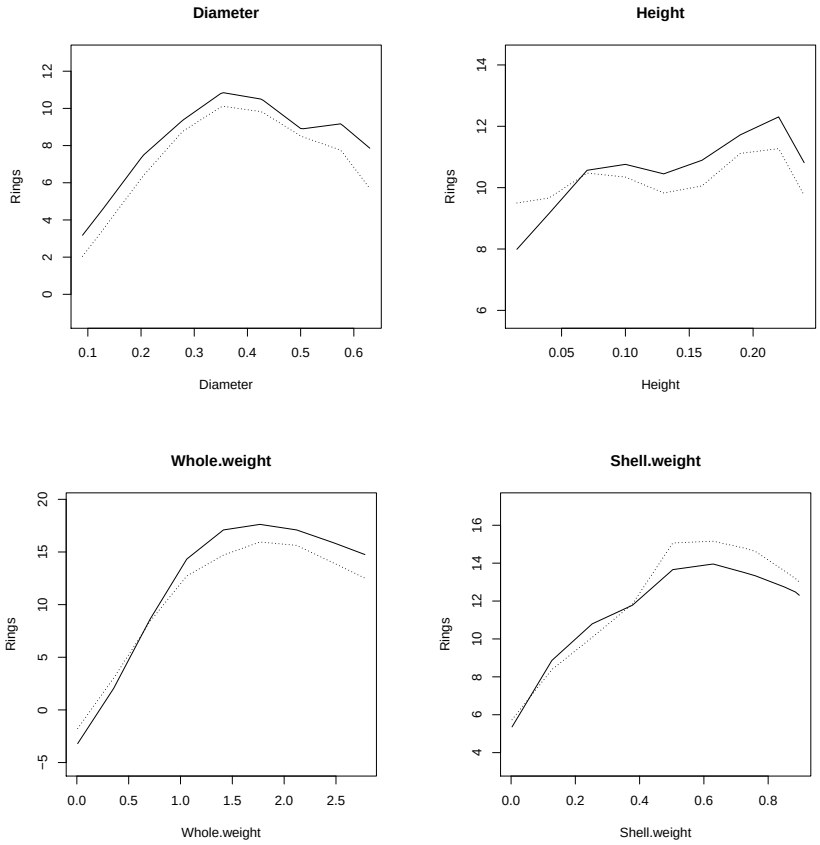


Figure 4.7: The estimated mean (solid lines) and median (dashed lines) curves of Rings against each covariate, while other covariates are fixed at their mean values for Boston housing data for Abalone data.

4.7.3 Haberman's Survival Data

The Haberman's Survival data is available at the UCI Machine Learning Repository [22]. The dataset contains cases from a study conducted at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. It has 306 observations and 4 attributes, which are age of patient at time of operation, patient's year of operation (minus 1900), the number of positive axillary nodes detected and survival status (survived 5 years or longer or died within 5 years). Based on the survival

status column, we define $Y_i = 1$ if the ith patient survived 5 years or longer, otherwise $Y_i = 0$. Then we apply different machine learning methods to this dataset for classification.

Table 4.9 presents the accuracy, precision, recall, F1 and AUC (area under the ROC curve) obtained from the test data for the survival group. We observe that SDRN outperforms other methods with the highest accuracy, precision, F1 score and AUC. Figure 4.8 shows the estimated log-odds functions versus Age and the number of positive axillary nodes, respectively, while other covariates are fixed at their mean values. With age increasing, the estimated log-odds value decreases, indicating decreasing survival probabilities. For the number of positive axillary nodes, the estimated log-odds function drops quickly to a small value when the number of positive axillary nodes increases from 0 to 12, and then it becomes stable and remains at a low point. This result indicates that when the number of positive axillary nodes is within a threshold value, it has a strong adverse effect on the survival probability. However, when it passes the threshold value, the survival probability remains at a very small value. In summary, we can clearly observe a nonlinear pattern of the estimated function in both plots. [46] also mentioned that the GLM could have a poor performance for this dataset because of the nonlinearity. Moreover, we use McFadden's pseudo $R^2 = 1 - \frac{\log\hat{L}(M_{full})}{\log\hat{L}(M_{null})}$ to further evaluate the model fitting, where $\hat{L}(M_{full})$ is the estimated likelihood with all predictors and $\hat{L}(M_{null})$ is the estimated likelihood without any predictors. The higher value of the pseudo R^2 indicates better model fitting. The pseudo R^2 from SDRN is 0.2012, and it is larger than the pseudo $R^2 = 0.1056$ from GLM.

	SDRN	GLM	FNN	GBM	RF	GAM
Accuracy	**0.714**	0.701	0.701	0.688	0.688	0.688
Precision	**0.754**	0.735	0.750	0.738	0.746	0.738
Recall	**0.891**	0.909	0.872	0.873	0.845	0.873
F1	**0.817**	0.813	0.807	0.800	0.797	0.800
AUC	**0.677**	0.633	0.635	0.641	0.641	0.667

Table 4.9: Accuracy, Precision, Recall, F1 and AUC for the survival group obtained by different methods with logistic loss for Haberman's Survival Data.

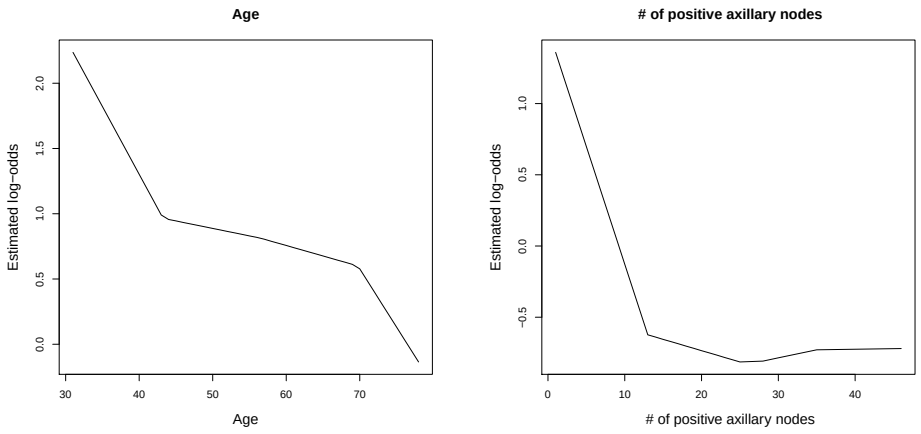


Figure 4.8: The estimated log-odds functions versus Age and the number of positive axillary nodes, respectively, while other covariates are fixed at their mean values.

4.7.4 BUPA Data

The BUPA Liver Disorders dataset is available at the UCI Machine Learning Repository [22]. It has 345 rows and 7 columns, with each row constituting the record of a single male individual. The first 5 variables are blood tests that are considered to be sensitive to liver disorders due to excessive alcohol consumption; they are mean corpuscular

volume (mcv), alkaline phosphotase (alkphos), alanine aminotransferase (sgpt), aspartate aminotransferase (sgot) and gamma-glutamyl transpeptidase (gammagt). We use them as covariates. The 6th variable is the number of half-point equivalents of alcoholic beverages drunk per day. Following [64], we dichotomize it to a binary response by letting $Y_i = 1$ if the number of drinks is greater than 3, otherwise $Y_i = 0$. The 7th column in the dataset was created by BUOA researchers for traing and test data selection.

We first calculate the McFadden-pseudo R^2 for GLM and SDRN with logistic loss, respectively. The pseudo R^2 for GLM is 0.2355, while it is 0.2584 for SDRN, indicating that the SDRN method yields a better prediction. In addition, Table 4.10 shows the accuracy, precision, recall, F1 and AUC for the group with the number of drinks greater than 3 obtained from the six methods with logistic loss. We see that SDRN has the largest accuracy, recall, F1 and AUC. The accuracy from GLM and GAM is smaller than other methods due to possible model misspecification of these two methods. To explore the nonlinear patterns, Figure 4.9 depicts the estimated log-odds functions versus the mcv, alkphos, sgot and gammagt predictors, respectively, while other covariates are fixed at their mean values. We can see that the estimated log-odds has a clear increaing pattern with mcv and sgot, indicating that the mcv and sgot levels can be strong indicators for alcohol consumption. For gammagt, the estimated log-odds increases quickly as the the level of gammagt is elevated. Its value remains to be positive as gammagt passes a certain value. The estimated log-odds has a quadratic nonlinear relationship with alkphos. Abnormal (either low or high) levels of alkphos is connected to a few health problems. Low levels of alkphos indicate a deficiency in zinc and magnesium, or a rare genetic disease called hypophosphatasia, which effects bones

Table 4.10: Accuracy, Precision Recall, F1 and AUC for the group with the number of drinks greater than 3 of the BUPA data for different methods with logistic loss.

	SDRN	GLM	FNN	GBM	RF	GAM
Accuracy	**0.620**	0.595	0.615	0.615	0.610	0.605
Precision	**0.650**	0.634	0.667	0.672	0.653	0.627
Recall	**0.520**	0.450	0.460	0.450	0.470	0.520
F1	**0.578**	0.526	0.544	0.539	0.547	0.568
AUC	**0.637**	0.618	0.629	0.613	0.608	0.624

and teeth. High levels of alkphos can be an indicator of liver disease or bone disorder.

4.8 Discussion

In this chapter, we propose a sparse deep ReLU network estimator (SDRN) obtained from empirical risk minimization with a Lipschitz loss function satisfying mild conditions. Our framework can be applied to a variety of regression and classification problems in machine learning. In general, deep neural networks are effective tools for lessening the curse of dimensionality under the condition that the target functions have certain special properties. We assume that the unknown target function belongs to Korobov spaces, which are subsets of the Sobolev spaces commonly used in the nonparametric regression literature. Functions in the Korobov spaces need to have partial mixed derivatives rather than a compositional structure, and thus can be more flexible for investigating nonlinear patterns between the response and the predictors.

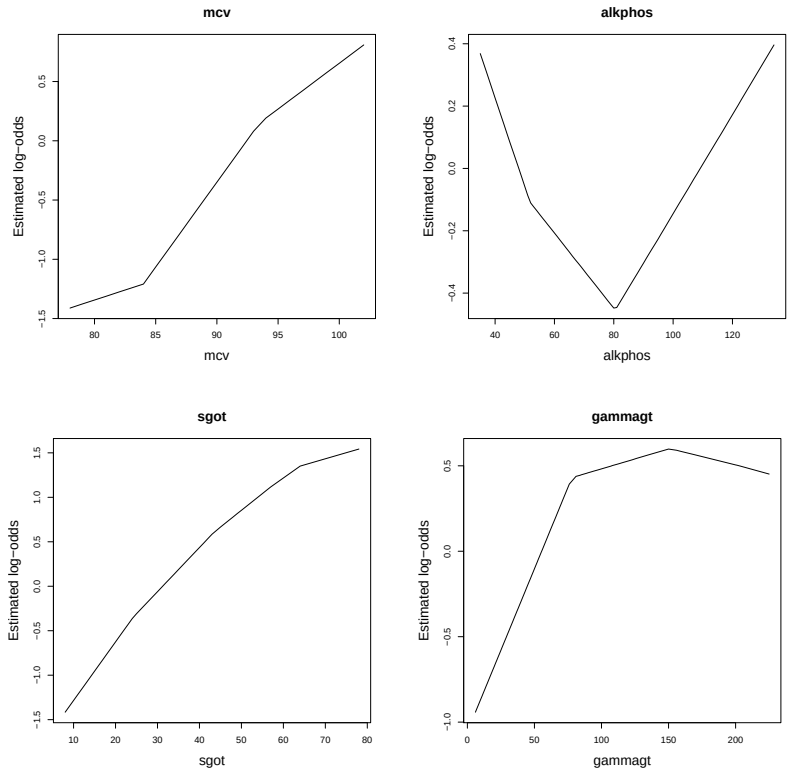


Figure 4.9: The estimated log-odds functions versus mcv, alkphos, sgot and gammagt , respectively, while other covariates are fixed at their mean values, for the BUPA data.

We derive non-asymptotic excess risk bounds for SDRN estimator. Our framework allows the dimension of the feature space to increase with the sample size with a rate slightly slower than $\log(n)$. We further show that our SDRN estimator can achieve the same optimal minimax rate (up to logarithmic factors) as one-dimensional nonparametric regression when the dimension is fixed, and the dimensionality effect is passed on to a logarithmic factor, so the curse of dimensionality is alleviated. The SDRN estimator has a suboptimal rate when the dimension increases with the sample size. Moreover, the depth and the total number of nodes and weights of the network need to increase with the sample size with certain rates established in the paper. These statistical properties provide an important theoretical basis and guidance for the analytic procedures in data analysis. Practically, we illustrate the proposed method through simulation studies and several real data applications. The numerical studies support our theoretical results.

Our proposed method provides a reliable solution for mitigating the curse of dimensionality for modern data analysis. Meanwhile it has opened up several interesting new avenues for further work. One extension is to derive a similar estimator for smoother regression functions with mixed derivatives of order greater than two; Jacobi-weighted Korobov spaces [79] may be considered for this scenario. Our method can be extended to other settings such as semiparametric models, longitudinal data and L_1 penalized regression. Moreover, it can be a promising tool for estimation of the propensity score function or the outcome regression function used in treatment effect studies. These interesting topics deserve thorough investigations for future research.

Chapter 5

Conclusions

In this dissertation, we mainly illustrate the nonparametric machine learning techniques used in subgroup analysis (Chapter 3) and deep neural network regression (Chapter 4), in which they focus on the one-dimensional and high-dimensional problems, respectively.

In Chapter 3, to cluster the heterogeneous longitudinal trajectories of AD data, we propose a subject-specific nonparametric regression model, in which the heterogeneity can be driven by unobserved latent factors. We first use B-splines to estimate the nonparametric functions, and then apply the concave penalty to pairwise the B-spline coefficients so that we can merge the individuals with similar progression curves into the same subgroup. Our proposed method can automatically identify the latent memberships and estimate the parameters in the model simultaneously without knowing the grouping information. For implementation, we develop an ADMM algorithm. Through simulation studies, we demonstrate the promising performance of our proposed method. Meanwhile, the real data (AD data) application also indicates the meaningful subgroups in clinical trials. We hope this

chapter could provide researchers a new idea for subgroup analysis when involving nonparametric components.

In Chapter 4, we study the nonparametric functions in high-dimensional data setting. We propose a sparse deep ReLU network estimator (SDRN) obtained from empirical risk minimization with a Lipschitz loss function satisfying mild conditions. The estimator of the target function is built upon a network architecture of sparsely-connected deep neural networks with the rectified linear unit (ReLU) activation function. We assume that the unknown target function belongs to Korobov spaces, which are subsets of the Sobolev spaces commonly used in the nonparametric regression literature. Rather than having a compositional structure, functions belongs to this space only need to satisfy a smoothness condition. Thus, it is more flexible for capturing the nonlinear patterns between the response and predictors. Our framework is applicable to both regression and classification problems. We also develop statistical properties of the proposed methodology, which provide an important guidance for the data analytic procedures. Practically, we illustrate the proposed method through simulation studies and several real data applications. The numerical studies support our theoretical results. In general, our proposed method provides a reliable solution for mitigating the curse of dimensionality for modern large-scale data analysis.

Bibliography

[1] P. Alquier, V. Cottet, and G. Lecué. Estimation bounds and sharp oracle inequalities of regularized procedures with lipschitz loss functions. *The Annals of Statistics*, 47:2117–2144, 2019.

[2] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 2009.

[3] A. Antoniadis, G. Gregoire, and I. W. McKeague. Bayesian estimation in single-index models. *Statistica Sinica*, 14:1147–1164, 2004.

[4] F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18:1–53, 2017.

[5] B. Bauer and M. Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, 47:2261–2285, 2019.

[6] R. Bellman. *Curse of dimensionality. Adaptive control processes: a guided tour.* Princeton, NJ, 1961.

[7] Y. Bengio and Y. LeCun. *Scaling learning algorithms towards AI. In Bottou, L., Chapelle, O., and DeCoste,D. and Weston, J., eds.,* Large-Scale Kernel Machines. MIT Press, 2007.

[8] Y. Bengio, N. L. Roux, P. Vincent, O. Delalleau, and P. Marcotte. Convex neural networks. *Advances in Neural Information Processing Systems*, pages 123–130, 2005.

[9] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[10] H. J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 2004.

[11] Zongwu Cai, Jianqing Fan, and Qiwei Yao. Functional-coefficient regression models for nonlinear time series. *Journal of the American Statistical Association*, 95(451):941–956, 2000.

[12] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-Theory and Methods*, 3(1):1–27, 1974.

[13] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[14] M. Chen, H. Jiang, W. Liao, and T. Zhao. Nonparametric regression on low-dimensional manifolds using deep relu networks. *preprint, https://arxiv.org/abs/1908.01842*, 2019.

[15] Y. Chen and R. Samworth. Generalized additive and index models with shape constraints. *Journal of the Royal Statistical Society: Series B*, 78:729–754, 2016.

[16] M. Y. Cheng, J. Fan, and J. S. Marron. Minimax efficiency of local polynomial fit estimators at boundaries. *Mimeo Series 2098, University of North Carolina-Chapel Hill*, 1994.

[17] M. Y. Cheng and H. T. Wu. Local linear regression on manifolds and its geometric interpretation. *Journal of the American Statistical Association*, 108:1421–1434, 2013.

[18] F. Cucker and D. Zhou. *Learning theory*. Cambridge Monographs on Applied and Computational Mathematics, 2007.

[19] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[20] Carl De Boor. *A practical guide to splines*. Revised Edition. Springer, New York., 2001.

[21] Paul Doukhan. *Mixing: properties and examples*, volume 85. Springer Science & Business Media, 2012.

[22] D. Dua and C. Graff. Uci machine learning repository [http://archive.ics.uci.edu/ml]. *Irvine, CA: University of California, School of Information and Computer Science*, 2019.

[23] R. Eldan and O. Shamir. The power of depth for feedfoward neural networks. *JMLR: Workshop and Conference Proceedings*, pages 1–34, 2016.

[24] J. Fan and I. Gijbels. *Local polynomials modelling and its applications*. Chapman and Hall, London, 1996.

[25] J. Fan and T. Huang. Profile likelihood inferences on semiparametric varying-coefficient partially linear models. *Bernoulli*, 11:1031–1057, 2005.

[26] J. Fan, C. Ma, and Y. Zhong. A selective overview of deep learning. *Statistical Science*, 36:264–290, 2021.

[27] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[28] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[29] J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76:817–823, 1981.

[30] Christophe Genolini and Bruno Falissard. Kml: A package to cluster longitudinal data. *Computer Methods and Programs in Biomedicine*, 104(3):e112–e121, 2011.

[31] VV Gorodetskii. On the strong mixing property for linear sequences. *Theory of Probability & Its Applications*, 22(2):411–413, 1978.

[32] M. Griebel. Sparse grids and related approximation schemes for higher dimensional problems. *Foundations of Computational Mathematics, Cambridge University Press*, pages 106–161, 2006.

[33] D. Harrison Jr and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5:81–102, 1978.

[34] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[35] Arthur E Hoerl and Robert W Kennard. Ridge regression: applications to nonorthogonal problems. *Technometrics*, 12(1):69–82, 1970.

[36] J. Horowitz and E. Mammen. Rate-optimal estimation for a general class of nonparametric regression models with unknown link functions. *The Annals of Statistics*, 35:2589–2619, 2007.

[37] J. Huang. Projection estimation in multiple regression with application to functional anova models. *The Annals of Statistics*, 26:242–272, 2003.

[38] Jianhua Z Huang. Local asymptotics for polynomial spline regression. *The Annals of Statistics*, 31(5):1600–1635, 2003.

[39] Jianhua Z Huang and Haipeng Shen. Functional coefficient regression models for nonlinear time series: a polynomial spline approach. *Scandinavian Journal of Statistics*, 31(4):515–534, 2004.

[40] Jianhua Z Huang, Colin O Wu, and Lan Zhou. Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statistica Sinica*, pages 763–788, 2004.

[41] Jianhua Z Huang, Liangyue Zhang, and Lan Zhou. Efficient estimation in marginal partially linear models for longitudinal/clustered data using splines. *Scandinavian Journal of Statistics*, 34(3):451–477, 2007.

[42] T. Jung and K. A. S. Wickrama. An introduction to latent class growth analysis and growth mixture modeling. *Social and Personality Psychology Compass*, 2(1):302–317, 2008.

[43] Robert Katzman. Education and the prevalence of dementia and alzheimer's disease. *Neurology*, 43(1):13–20, 1993.

[44] D. P. Kingma and J. L. Ba. Adam: a method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[45] Yuichi Kitamura. Empirical likelihood methods with weakly dependent processes. *The Annals of Statistics*, 25(5):2084–2102, 1997.

[46] J. M. Landwehr, D. Pregibon, and A. C. Shoemaker. Graphical methods for assessing logistic regression models. *Journal of the American Statistical Association*, 79:61–71, 1984.

[47] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[48] H. Liang, X. Liu, R. Li, and C. L. Tsai. Estimation and testing for partially linear single-index models. *The Annals of Statistics*, 38:3811–3836, 2010.

[49] K. Y. Liang and S. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1):13–22, 1986.

[50] S. Liang and R. Srikant. Why deep neural networks for function approximation? *preprint, https://arxiv.org/abs/1610.04161*, 2016.

[51] Rong Liu and L. Yang. Spline-backfitted kernel smoothing of additive coefficient model. *Econometric Theory*, 26(1):29–59, 2010.

[52] GG Lorentz and RA DeVore. *Constructive Approximation, Polynomials and Splines Approximation*. Springer-Verlag, New York, Berlin, Heidelberg, 1993.

[53] S. Ma. Two-step spline estimating equations for generalized additive partially linear models with large cluster sizes. *The Annals of Statistics*, 20:2943–2972, 2012.

[54] S. Ma. A plug-in number of knots selector for polynomial spline regression. *Journal of Nonparametric Statistics*, 26:489–507, 2014.

[55] S. Ma and X. He. Inference for single-index quantile regression models with profile optimization. *The Annals of Statistics*, 44:1234–1268, 2015.

[56] S. Ma, J. Racine, and L. Yang. Spline regression in the presence of categorical predictors. *Journal of Applied Econometrics*, 30:705–717, 2015.

[57] Shujie Ma. Two-step spline estimating equations for generalized additive partially linear models with large cluster sizes. *The Annals of Statistics*, 40(6):2943–2972, 2012.

[58] Shujie Ma. A plug-in the number of knots selector for polynomial spline regression. *Journal of Nonparametric Statistics*, 26(3):489–507, 2014.

[59] Shujie Ma and Xuming He. Inference for single-index quantile regression models with profile optimization. *The Annals of Statistics*, 44(3):1234–1268, 2016.

[60] Shujie Ma and Jian Huang. A concave pairwise fusion approach to subgroup analysis. *Journal of the American Statistical Association*, 112(517):410–423, 2017.

[61] Shujie Ma, Jian Huang, Zhiwei Zhang, and Mingming Liu. Exploration of heterogeneous treatment effects via concave fusion. *The international journal of biostatistics*, 16(1), 2019.

[62] Shujie Ma, Qiongxia Song, and Li Wang. Simultaneous variable selection and estimation in semiparametric modeling of longitudinal/clustered data. *Bernoulli*, 19(1):252–274, 2013.

[63] James G MacKinnon and Halbert White. Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3):305–325, 1985.

[64] J. McDermott and R. S. Forsyth. Diagnosing a disorder in a classification benchmark. *Pattern Recognition Letters*, 73:41–43, 2016.

[65] Paul D McNicholas. Model-based classification using latent gaussian mixture models. *Journal of Statistical Planning and Inference*, 140(5):1175–1181, 2010.

[66] H. Mhaskar, Q. Liao, and T. Poggio. When and why are deep networks better than shallow ones? *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2343–2349, 2017.

[67] H. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14:829–848, 2016.

[68] H. Montanelli and Q. Du. New error bounds for deep relu networks using sparse grids. *SIAM Journal on Mathematics of Data Science*, accepted, 2019.

[69] R. Nakada and M. Imaizumi. Adaptive approximation and generalization of deep neural network with intrinsic dimensionality. *Journal of Machine Learning Research*, 21:1–38, 2020.

[70] Hoh Suk Noh and Byeong U Park. Sparse varying coefficient models for longitudinal data. *Statistica Sinica*, pages 1183–1202, 2010.

[71] T. Poggio, H. N. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14:503–519, 2017.

[72] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[73] George G Roussas and D Ioannides. Moment inequalities for mixing sequences of random variables. *Stochastic Analysis and Applications*, 5(1):60–120, 1987.

[74] D. Ruppert. Empirical-bias bandwidths for local polynomial nonparametric regression and density estimation. *Journal of the American Statistical Association*, 92:1049–1062, 1997.

[75] Mirna Safieh, Amos D Korczyn, and Daniel M Michaelson. Apoe4: an emerging therapeutic target for alzheimer's disease. *BMC medicine*, 17(1):1–17, 2019.

[76] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[77] J. Schmidt-Hieber. Deep relu network approximation of functions on a manifold. *preprint, https://arxiv.org/abs/1908.00695*, 2019.

[78] J. Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48:1875–1897, 2020.

[79] J. Shen and L. L. Wang. Sparse spectral approximations of high-dimensional problems based on hyperbolic cross. *SIAM Journal on Numerical Analysis*, 48:1087–1109, 2010.

[80] Juan Shen and Xuming He. Inference for subgroup analysis with a structured logistic-normal mixture model. *Journal of the American Statistical Association*, 110(509):303–312, 2015.

[81] Z. Shen, H. Yang, and S. Zhang. Deep network approximation characterized by number of neurons. *Communications in Computational Physics*, 28:1768–1811, 2020.

[82] J. J. Song, H. J. Lee, J. S. Morris, and S. Kang. Clustering of time-course gene expression data using functional data analysis. *Computational biology and chemistry*, 31(4):265–274, 2007.

[83] C. J. Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10:1040–1053, 1982.

[84] C. J. Stone. Additive regression and other nonparametric models. *The Annals of Statistics*, 13:689–705, 1985.

[85] C. J. Stone. The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics*, 22:118–184, 1994.

[86] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[87] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.

[88] Hansheng Wang, Runze Li, and Chih-Ling Tsai. Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika*, 94(3):553–568, 2007.

[89] Li Wang, Xiang Liu, Hua Liang, and Raymond J Carroll. Estimation and variable selection for generalized additive partial linear models. *Annals of statistics*, 39(4):1827, 2011.

[90] N. Wang, R. J. Carroll, and X. Lin. Efficient semiparametric marginal estimation for longitudinal/clustered data. *Journal of the American Statistical Association*, 100(469):147–157, 2005.

[91] L. Wasserman. *All of Nonparametric Statistics*. Springer Texts in Statistics, 2006.

[92] Y. Xia. Asymptotic distributions for two estimators of the single-index model. *Econometric Theory*, 22:1112–1137, 2006.

[93] Lan Xue and hua Liang. Polynomial spline estimation for a generalized additive coefficient model. *Scandinavian Journal of Statistics*, 19(1):252–274, 2013.

[94] D. Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 70:103–114, 2017.

[95] G. Ye, Y. Chen, and X. Xie. Efficient variable selection in support vector machines via the alternating direction method of multipliers. *Proceedings of Machine Learning Research*, 15:832–840, 2011.

[96] Konstantina G Yiannopoulou, Aikaterini I Anastasiou, Venetia Zachariou, and Sygkliti H Pelidou. Reasons for failed trials of disease-modifying treatments for alzheimer disease and their contribution in recent research. *Biomedicines*, 7(4):97, 2019.

[97] M. Yuan and D. Zhou. Minimax optimal rates of estimation in high dimensional additive models. *The Annals of Statistics*, 44:2564–2593, 2016.

[98] Cun H Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

[99] X. Zhang, B. U. Park, and J. L. Wang. Time-varying additive models for longitudinal data. *Journal of the American Statistical Association*, 108:983–998, 2013.

[100] Zhongyi Zhu, Wing K Fung, and Xuming He. On the asymptotics of marginal regression splines with longitudinal data. *Biometrika*, 95(4):907–917, 2008.

Appendix A

Supplementary Materials for Chapter 3

A.1 Computational Complexity of ADMM Algorithm

The computational complexity can be expressed in terms of floating point operations per second (flops) required to find the solution [9]. Our ADMM algorithm involves updating the estimates of $\boldsymbol{\delta}$, $\boldsymbol{\upsilon}$ and $\boldsymbol{\gamma}$ given in (3.10), (3.11) and (3.12) through iterations. It costs $O((n-1)nS) = O(n^2S)$ flops for computing the updates of $\boldsymbol{\delta}$ and $\boldsymbol{\upsilon}$, given that $\boldsymbol{\delta}$ and $\boldsymbol{\upsilon}$ are $0.5n(n-1)S \times 1$ vectors, where S is the number of B-spline basis functions. Following [9] (see pages 27-29), the computational cost for updating $\boldsymbol{\gamma}$ given in (3.12) is dominated by $\boldsymbol{V}^{-1}$ and $(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X} + \vartheta\boldsymbol{A}^T\boldsymbol{A})^{-1}$, which require $O(\sum_{i=1}^{n} m_i^3)$ and $O(n^3S^3)$ flops, respectively. Therefore, it takes $O(\sum_{i=1}^{n} m_i^3 + n^3S^3)$ flops to update $\boldsymbol{\gamma}$ in the first iteration. Since the values $(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}+\vartheta\boldsymbol{A}^T\boldsymbol{A})^{-1}$ and $\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{Y}$ remain the same through

iterations, and we only need to compute them once, in the subsequent iterations, the computational cost for updating $\boldsymbol{\gamma}$ is reduced to $O(nS \times 0.5n(n-1)S) = O(n^3S^2)$, which is cost of computing $\boldsymbol{A}^T\left(\boldsymbol{\delta} - \vartheta^{-1}\boldsymbol{\upsilon}\right)$ given in (3.12). In sum, the overall cost of updating $(\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\upsilon})$ is $O(n^2S + \sum_{i=1}^n m_i^3 + n^3S^3) = O(\sum_{i=1}^n m_i^3 + n^3S^3)$ in the first iteration, and it is $O(n^2S + n^3S^2) = O(n^3S^2)$ in the subsequent iterations.

A.2 Consistency and Convergence

Let C denotes a generic constant that might assume different values at different places. Without loss of generality, we consider the following B-spline basis functions that span G, that is, $B_l = S^{1/2}B_l^*$, $l = 1, \ldots, S$, where $\{B_l^*\}_{l=1}^S$ are the B-splines defined in Chapter 5 of [52]. It follows from Theorem 4.2 of [52] that

$$M_1\|\boldsymbol{\gamma}\|_2^2 \leq \int \left\{\sum_{l=1}^S B_l(t)\gamma_l\right\}^2 dt \leq M_2\|\boldsymbol{\gamma}\|_2^2 \tag{A.1}$$

for some constants $0 < M_1 < M_2 < \infty$, where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_S)^T$.

Lemma A.1 *For each i, there exist some constants $0 < M_1 < M_2 < \infty$ such that, except on an event whose probability tends to zero, all the eigenvalues of $\boldsymbol{X}_i^T\boldsymbol{X}_i/m_i$ fall between M_1 and M_2.*

Lemma A.2 *Assume the random variables ξ and η be F_1^k-measurable and F_{k+s}^∞-measurable, respectively. If $E(|\xi|^p) < \infty$, $E(|\eta|^q) < \infty$ for some $p, q > 1$ and $1/p + 1/q < 1$. Then, under α-mixing,*

$$|E(\xi\eta) - E(\xi)E(\eta)| \leq 10\alpha(s)^{1-\frac{1}{p}-\frac{1}{q}}\|\xi\|_p\|\eta\|_q,$$

where $\|\xi\|_p = E^{1/p}(|\xi|^p)$ denotes the L_p-norm of ξ.

The proof of Lemma A.1 and Lemma A.2 can be respectively referred to Lemma 2 of [39] and Theorem 7.3 of [73].

Define $\beta_i^*(t) = \boldsymbol{B}(t)^T\boldsymbol{\gamma}^* \in G$ such that $\|\beta_i^* - \beta_i\|_2 = \inf_{g\in G}\|g - \beta_i\|_2 \triangleq \varpi_i$, it follows from the result on page 149 of [20] that $\varpi_i = J^{-r}$ if $\beta_i(\cdot) \in C^{(r)}$.

A.2.1 Consistency of Initial Estimator

Proposition A.3 *Under conditions (C1)-(C4), the initial estimators $\hat{\beta}_i^{(0)}(t)$, $i = 1, \ldots, n$, satisfy $\|\hat{\beta}_i^{(0)} - \beta_i\|_2^2 = O_p(J/m_i + J^{-2r})$.*

Proof. Recall that

$$\hat{\boldsymbol{\gamma}}_i^{(0)} = \arg\min_{\boldsymbol{\gamma}_i}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i)^T(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i) = (\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{Y}_i,$$

and $\hat{\beta}_i^{(0)}(t) = \boldsymbol{B}(t)^T\hat{\boldsymbol{\gamma}}_i^{(0)}$. Now, define $\tilde{\boldsymbol{\gamma}}_i^{(0)} = (\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\tilde{\boldsymbol{Y}}_i$, $\tilde{\beta}_i^{(0)}(t) = \boldsymbol{B}(t)^T\tilde{\boldsymbol{\gamma}}_i^{(0)}$, where $\tilde{\boldsymbol{Y}}_i = (\beta_i(t_{i1}), \ldots, \beta_i(t_{im_i}))^T$. Obviously,

$$\hat{\boldsymbol{\gamma}}_i^{(0)} - \tilde{\boldsymbol{\gamma}}_i^{(0)} = (\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T(\boldsymbol{Y}_i - \tilde{\boldsymbol{Y}}_i) = (\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{\varepsilon}_i,$$

where $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \ldots, \varepsilon_{im_i})^T$. It follows from Lemma A.1 that there exists a constant $C > 0$, such that

$$E\left(\boldsymbol{\varepsilon}_i^T\boldsymbol{X}_i(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{\varepsilon}_i\right) \le C\frac{1}{m_i^2}E\left(\boldsymbol{\varepsilon}_i^T\boldsymbol{X}_i\boldsymbol{X}_i^T\boldsymbol{\varepsilon}_i\right).$$

Taking $p = q = 4$ in Lemma A.2 and by the properties of B-splines, we can obtain

$$\begin{aligned} E\left(\boldsymbol{\varepsilon}_i^T\boldsymbol{X}_i\boldsymbol{X}_i^T\boldsymbol{\varepsilon}_i\right) &= E\left\{\sum_{l=1}^{S}\left(\sum_{j=1}^{m_i}B_l(t_{ij})\varepsilon_{ij}\right)^2\right\} = E\left\{\sum_{j,j'=1}^{m_i}\sum_{l=1}^{S}\varepsilon_{ij}\varepsilon_{ij'}B_l(t_{ij})B_l(t_{ij'})\right\} \\ &\le S\sum_{j,j'=1}^{m_i}|E(\varepsilon_{ij}\varepsilon_{ij'})| \le 10S\sum_{1\le j,j'\le m_i}\alpha(|j-j'|)^{1/2}[E(|\varepsilon_{ij}|^4)]^{1/4}[E(|\varepsilon_{ij'}|^4)]^{1/4} \\ &= O_p(Jm_i), \end{aligned}$$

where the last equality holds because $\sum_{s=1}^{\infty}\alpha(s)^{1/2}[E(|\varepsilon_{ij}|^4)]^{1/4}[E(|\varepsilon_{ij'}|^4)]^{1/4}$ is bounded from condition (C2). Thus, $\|\hat{\gamma}_i^{(0)}-\tilde{\gamma}_i^{(0)}\|_2^2 = O_p(J/m_i)$. This together with expression (A.1) leads to

$$\|\hat{\beta}_i^{(0)}-\tilde{\beta}_i^{(0)}\|_2^2 = O\left(\|\hat{\gamma}_i^{(0)}-\tilde{\gamma}_i^{(0)}\|_2^2\right) = O_p(J/m_i). \tag{A.2}$$

On the other hand, by Lemma A.1, we have

$$\|\tilde{\gamma}_i^{(0)}-\gamma_i^*\|_2^2 = O_p\left(\frac{1}{m_i}(\tilde{\gamma}_i^{(0)}-\gamma_i^*)^T\boldsymbol{X}_i^T\boldsymbol{X}_i(\tilde{\gamma}_i^{(0)}-\gamma_i^*)\right).$$

Noting that $\boldsymbol{X}_i\tilde{\gamma}_i^{(0)} = \boldsymbol{X}_i(\boldsymbol{X}_i^T\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\tilde{\boldsymbol{Y}}_i$ is an orthogonal projection of $\tilde{\boldsymbol{Y}}_i$. Hence,

$$\begin{aligned}
\frac{1}{m_i}(\tilde{\gamma}_i^{(0)}-\gamma_i^*)^T\boldsymbol{X}_i^T\boldsymbol{X}_i(\tilde{\gamma}_i^{(0)}-\gamma_i^*) &\leq \frac{1}{m_i}(\tilde{\boldsymbol{Y}}_i-\boldsymbol{X}_i\gamma_i^*)^T(\tilde{\boldsymbol{Y}}_i-\boldsymbol{X}_i\gamma_i^*)\\
&= \frac{1}{m_i}\sum_{j=1}^{m_i}(\beta_i(t_{ij})-\beta_i^*(t_{ij}))^2\\
&= O(\varpi_i^2),
\end{aligned}$$

It follows from expression (A.1) that

$$\|\tilde{\beta}_i^{(0)}-\beta_i^*\|_2^2 = O\left(\|\tilde{\gamma}_i^{(0)}-\gamma_i^*\|_2^2\right) = O_p(\varpi_i^2). \tag{A.3}$$

Therefore, by the definition of ϖ_i, equations (A.2)-(A.3) and the triangle inequality, we have

$$\begin{aligned}
&\|\hat{\beta}_i^{(0)}-\beta_i\|_2^2\\
\leq\ &\|\hat{\beta}_i^{(0)}-\tilde{\beta}_i^{(0)}\|_2^2+\|\tilde{\beta}_i^{(0)}-\beta_i^*\|_2^2+\|\beta_i^*-\beta_i\|_2^2\\
=\ &O_p(J/m_i)+O_p(\varpi_i^2)+O_p(\varpi_i^2) = O_p(J/m_i+J^{-2r}).
\end{aligned}$$

This completes the proof.

A.2.2 Convergence of ADMM

Proposition A.4 *Let* $\boldsymbol{r}^{m+1} = \boldsymbol{A}\boldsymbol{\gamma}^{m+1} - \boldsymbol{\delta}^{m+1}$ *and* $\boldsymbol{s}^{m+1} = \vartheta \boldsymbol{A}^T(\boldsymbol{\delta}^{m+1} - \boldsymbol{\delta}^m)$ *respectively be the primal residual and dual residual in the ADMM. Then,* $\lim_{m\to\infty} \|\boldsymbol{r}^{m+1}\|_2^2 = 0$ *and* $\lim_{m\to\infty} \|\boldsymbol{s}^{m+1}\|_2^2 = 0$ *hold for MCP penalty.*

Proof. Taking a careful examination of our constructed objective function $L(\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\nu})$ with that of [61], the conclusion $\lim_{m\to\infty} \|\boldsymbol{r}^{m+1}\|_2^2 = 0$ can be directly derived by a similar proof of proposition 1 in [61]. Recall that $\boldsymbol{\gamma}^{m+1}$ minimize $L(\boldsymbol{\gamma}, \boldsymbol{\delta}^m, \boldsymbol{\nu}^m)$ by definition, thus

$$\begin{aligned} 0 &= \frac{\partial L(\boldsymbol{\gamma}, \boldsymbol{\delta}^m, \boldsymbol{\nu}^m)}{\partial \boldsymbol{\gamma}}\Big|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^{m+1}} = \boldsymbol{X}^T\boldsymbol{V}^{-1}(\boldsymbol{X}\boldsymbol{\gamma}^{m+1} - \boldsymbol{Y}) + \boldsymbol{A}^T\{\boldsymbol{\nu}^m + \vartheta(\boldsymbol{A}\boldsymbol{\gamma}^{m+1} - \boldsymbol{\delta}^m)\} \\ &= \boldsymbol{X}^T\boldsymbol{V}^{-1}(\boldsymbol{X}\boldsymbol{\gamma}^{m+1} - \boldsymbol{Y}) + \boldsymbol{A}^T\{[\boldsymbol{\nu}^{m+1} - \vartheta(\boldsymbol{A}\boldsymbol{\gamma}^{m+1} - \boldsymbol{\delta}^{m+1})] + \vartheta(\boldsymbol{A}\boldsymbol{\gamma}^{m+1} - \boldsymbol{\delta}^m)\} \\ &= \boldsymbol{X}^T\boldsymbol{V}^{-1}(\boldsymbol{X}\boldsymbol{\gamma}^{m+1} - \boldsymbol{Y}) + \boldsymbol{A}^T\boldsymbol{\nu}^{m+1} + \vartheta\boldsymbol{A}^T(\boldsymbol{\delta}^{m+1} - \boldsymbol{\delta}^m), \end{aligned}$$

which implies

$$\boldsymbol{s}^{m+1} = \vartheta\boldsymbol{A}^T(\boldsymbol{\delta}^{m+1} - \boldsymbol{\delta}^m) = -\{\boldsymbol{X}^T\boldsymbol{V}^{-1}(\boldsymbol{X}\boldsymbol{\gamma}^{m+1} - \boldsymbol{Y}) + \boldsymbol{A}^T\boldsymbol{\nu}^{m+1}\}.$$

In view of $\lim_{m\to\infty} \|\boldsymbol{r}^m\|_2^2 = \lim_{m\to\infty} \|\boldsymbol{A}\boldsymbol{\gamma}^m - \boldsymbol{\delta}^m\|_2^2 = 0$, we have

$$0 = \lim_{m\to\infty} \frac{\partial L(\boldsymbol{\gamma}, \boldsymbol{\delta}^m, \boldsymbol{\nu}^m)}{\partial \boldsymbol{\gamma}}\Big|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^{m+1}} = \lim_{m\to\infty} \{\boldsymbol{X}^T\boldsymbol{V}^{-1}(\boldsymbol{X}\boldsymbol{\gamma}^{m+1} - \boldsymbol{Y}) + \boldsymbol{A}^T\boldsymbol{\nu}^{m+1}\} = \lim_{m\to\infty} -\boldsymbol{s}^{m+1}.$$

Therefore, we obtain $\lim_{m\to\infty} \|\boldsymbol{s}^{m+1}\|_2^2 = 0$, this completes the proof.

A.3 Proof of Theorems

To prove the main theoretical results in this article, we first present the following lemma which will be frequently used in the sequel. Let $\bar{\beta}_i(t) = \boldsymbol{B}^T(t)\bar{\boldsymbol{\gamma}}_i$, where

$$\bar{\boldsymbol{\gamma}}_i = \arg\min_{\boldsymbol{\gamma}_i} (\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i)^T\boldsymbol{V}_i^{-1}(\boldsymbol{Y}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i) = (\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{Y}_i. \tag{A.4}$$

Lemma A.5 *Under conditions (C1)-(C5), we have* $\|\bar{\beta}_i - \beta_i\|_2^2 = O_p(J/m_i + J^{-2r})$, $i = 1, \ldots, n$.

Proof. For $i = 1, \ldots, n$, let $\tilde{\beta}_i(t) = \boldsymbol{B}(t)^T\tilde{\boldsymbol{\gamma}}_i$ and $\tilde{\boldsymbol{\gamma}}_i = (\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\tilde{\boldsymbol{Y}}_i$, where $\tilde{\boldsymbol{Y}}_i$ is defined as above. Obviously, $\bar{\beta}_i(t) - \tilde{\beta}_i(t) = \boldsymbol{B}(t)^T(\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{\varepsilon}_i$. By Lemma A.1 and the bounded assumption on the eigenvalues of V, it is easy to verify that there exist two constants $0 < C_1 \leq C_2 < \infty$, such that

$$C_1\frac{1}{m_i^2}E(\boldsymbol{\varepsilon}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{\varepsilon}_i) \leq E\left(\|\bar{\beta}_i(t) - \tilde{\beta}_i(t)\|_2^2\right) \leq C_2\frac{1}{m_i^2}E(\boldsymbol{\varepsilon}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{\varepsilon}_i).$$

According to the operation properties of the trace and expectation, we have

$$\begin{aligned}
E(\boldsymbol{\varepsilon}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{X}_i\boldsymbol{X}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\varepsilon}_i) &= \text{trace}\left\{E(\boldsymbol{\varepsilon}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{X}_i\boldsymbol{X}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\varepsilon}_i)\right\} = E\left(\text{trace}\{\boldsymbol{X}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T\boldsymbol{\Sigma}_i^{-1}\boldsymbol{X}_i\}\right) \\
&= E\left(\text{trace}\{\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{\Sigma}_i\boldsymbol{V}_i^{-1}\boldsymbol{X}_i\}\right) = \text{trace}\left\{E(\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{\Sigma}_i\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)\right\} \\
&= O_p(m_iJ),
\end{aligned}$$

where the last equality holds due to condition (C5) and Lemma A.1. Hence, we obtain

$$\|\bar{\beta}_i - \tilde{\beta}_i\|_2^2 = O_p(J/m_i). \tag{A.5}$$

Furthermore, as $\boldsymbol{V}_i^{-1/2}\boldsymbol{X}_i(\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\tilde{\boldsymbol{Y}}_i$ is an orthogonal projection of $\boldsymbol{V}_i^{-1/2}\tilde{\boldsymbol{Y}}_i$, we have

$$\begin{aligned}
&\frac{1}{m_i}(\tilde{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i^*)^T\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i(\tilde{\boldsymbol{\gamma}}_i - \tilde{\boldsymbol{\gamma}}_i^*) \\
= &\frac{1}{m_i}\left\{(\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\tilde{\boldsymbol{Y}}_i - \tilde{\boldsymbol{\gamma}}_i^*\right\}^T\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1/2}\boldsymbol{V}_i^{-1/2}\boldsymbol{X}_i\left\{(\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\tilde{\boldsymbol{Y}}_i - \tilde{\boldsymbol{\gamma}}_i^*\right\} \\
= &\frac{1}{m_i}\|\boldsymbol{V}_i^{-1/2}\boldsymbol{X}_i(\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^T\boldsymbol{V}_i^{-1}\tilde{\boldsymbol{Y}}_i - \boldsymbol{V}_i^{-1/2}\boldsymbol{X}_i\boldsymbol{\gamma}_i^*\|_2^2 \\
\leq &\frac{1}{m_i}\|\boldsymbol{V}_i^{-1/2}\tilde{\boldsymbol{Y}}_i - \boldsymbol{V}_i^{-1/2}\boldsymbol{X}_i\boldsymbol{\gamma}_i^*\|_2^2 = \frac{1}{m_i}(\tilde{\boldsymbol{Y}}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i^*)^T\boldsymbol{V}_i^{-1}(\tilde{\boldsymbol{Y}}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i^*) \\
= &O_p\left(\frac{1}{m_i}\|\tilde{\boldsymbol{Y}}_i - \boldsymbol{X}_i\boldsymbol{\gamma}_i^*\|_2^2\right) = O_p\left(\|\beta_i - \beta_i^*\|_2^2\right) = O_p(\varpi_i^2),
\end{aligned}$$

where the antepenult equality holds by the bounded assumption on the eigenvalues of V. Combining the expression (A.1), condition (C5) as well as Lemma A.1 leads to

$$\|\tilde{\beta}_i - \beta_i^*\|_2^2 = O\left(\|\tilde{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i^*\|_2^2\right) = O_p\left(\frac{1}{m_i}(\tilde{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i^*)^T \boldsymbol{X}_i^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{X}_i (\tilde{\boldsymbol{\gamma}}_i - \tilde{\boldsymbol{\gamma}}_i^*)\right) = O_p(\varpi_i^2). \quad \text{(A.6)}$$

Consequently, it follows from equations (A.5), (A.6) and the definition of ϖ_i that

$$\|\bar{\beta}_i - \beta_i\|_2^2 \leq \|\bar{\beta}_i - \tilde{\beta}_i\|_2^2 + \|\tilde{\beta}_i - \beta_i^*\|_2^2 + \|\beta_i^* - \beta_i\|_2^2 = O_p(J/m_i + \varpi_i^2) = O_p(J/m_i + J^{-2r}).$$

This finishes the proof.

Proof of Theorem 3.1. Notice that, it is equivalent to individually obtaining $\hat{\boldsymbol{\theta}}_k = \arg\min_{\boldsymbol{\theta}_k} (\boldsymbol{Y}_{(k)} - \boldsymbol{X}_{(k)}\boldsymbol{\theta}_k)^T \boldsymbol{V}_{(k)}^{-1} (\boldsymbol{Y}_{(k)} - \boldsymbol{X}_{(k)}\boldsymbol{\theta}_k) = \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{Y}_{(k)}$ for $k = 1, \ldots, K$, where $\boldsymbol{Y}_{(k)} = \left\{\boldsymbol{Y}_i^T : i \in \mathcal{G}_k\right\}^T, \boldsymbol{X}_{(k)} = \left\{\boldsymbol{X}_i^T : i \in \mathcal{G}_k\right\}^T$ and $\boldsymbol{V}_{(k)} = \operatorname{diag}\{\boldsymbol{V}_i : i \in \mathcal{G}_k\}$. Then $\hat{\alpha}_k(t) = \boldsymbol{B}(t)^T \hat{\boldsymbol{\theta}}_k$ for any $t \in \mathbb{T}$. According to Lemma A.5, we have

$$\|\hat{\alpha}_k - \alpha_k\|_2^2 = O_p(J/N_k + J^{-2r}) \leq O_p(J/N_0 + J^{-2r}),$$

where $\alpha_k(t)$ is the true function in the kth group. As a result,

$$\|\hat{\boldsymbol{\alpha}}^{or} - \boldsymbol{\alpha}\|_2^2 = \sum_{k=1}^{K} \|\hat{\alpha}_k^{or} - \alpha_k\|_2^2 = O_p(J/N_0 + J^{-2r})$$

for any fixed K. This completes the proof.

Proof of Theorem 3.2. Let $\hat{\beta}_i^{or}(t)$ and $\hat{\boldsymbol{\gamma}}_i^{or}$ be the estimated function and estimated B-spline coefficient for subject i given the true membership, respectively. We first prove $\|\hat{\beta}_i - \hat{\beta}_i^{or}\|_2^2 = O_p(J/m_{(n)} + J^{-2r})$ for each i, where . Let $\delta_n = J/m_{(n)} + J^{-2r}$, if one can

show that for any $\omega > 0$, there exists a large enough constant $M > 0$ satisfying

$$P\left\{\inf_{\|\boldsymbol{X}_i(\boldsymbol{\gamma}_i-\hat{\boldsymbol{\gamma}}_i^{or})\|_2^2=M\delta_n} L_n(\boldsymbol{\gamma}) > L_n(\hat{\boldsymbol{\gamma}}^{or})\right\} \geq 1-\omega, \tag{A.7}$$

which means a local minimizer of $L_n(\boldsymbol{\gamma})$ existed in the region $\mathbb{B}_0 = \{\boldsymbol{\gamma} : \|\boldsymbol{X}_i(\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i^{or})\|_2^2 \leq M\delta_n\}$. Then, $\|\hat{\beta}_i - \hat{\beta}_i^{or}\|_2^2 = O_p(J/m_{(n)} + J^{-2r})$ can be proved.

Let $L_{n1} = \frac{1}{2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\gamma})^T\boldsymbol{V}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\gamma})$, thus $\bar{\boldsymbol{\gamma}} = (\bar{\boldsymbol{\gamma}}_1^T, \ldots, \bar{\boldsymbol{\gamma}}_n^T)^T$ minimize L_{n1}, where $\bar{\boldsymbol{\gamma}}_i$, $i = 1, \ldots, n$ are defined in (A.4). It follows from Lemma A.5 that $\|\|\bar{\beta}_i - \beta_i\|_2^2 = O_p(J/m_{(n)} + J^{-2r})$ for each i. Combining this result with Theorem 3.1 leads to

$$\|\bar{\beta}_i - \hat{\beta}_i^{or}\|_2^2 \leq \|\bar{\beta}_i - \beta_i\|_2^2 + \|\hat{\beta}_i^{or} - \beta_i\|_2^2 = O_p(J/m_{(n)} + J^{-2r}),$$

which is equivalent to

$$\|\boldsymbol{X}_i(\bar{\boldsymbol{\gamma}}_i - \hat{\boldsymbol{\gamma}}_i^{or})\|_2^2 \leq C_0\delta_n \tag{A.8}$$

for some constant C_0 from expression (A.1). Moreover, for any $k \neq k'$,

$$\begin{aligned}
\|\hat{\alpha}_k - \hat{\alpha}_{k'}\|_2 &= \|\hat{\alpha}_k - \alpha_k + \alpha_k - \hat{\alpha}_{k'} + \alpha_{k'} - \alpha_{k'}\|_2 \\
&\geq \|\alpha_k - \alpha_{k'}\|_2 - \|\hat{\alpha}_k - \alpha_k\|_2 - \|\hat{\alpha}_{k'} - \alpha_{k'}\|_2 \\
&\geq b - \|\hat{\alpha}_k - \alpha_k\|_2 - \|\hat{\alpha}_{k'} - \alpha_{k'}\|_2.
\end{aligned}$$

Thus, we have $\|\hat{\alpha}_k - \hat{\alpha}_{k'}\|_2 \geq b$ for sufficiently large N_0 from Theorem 3.1. Accordingly, $\|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k'}\|_2 \geq Cb$ for some constant $C > 0$. Similarly, for any $i \in \mathcal{G}_k$, $j \in \mathcal{G}_{k'}$, $k \neq k'$, we can derive that $\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|_2 \geq Cb$ for any $\boldsymbol{\gamma}$ lies in the constraint $\mathbb{B}_0$ and sufficiently large m_n.

In addition, as $P_\tau(\cdot,\lambda) \geq 0$ and $P_\tau(0,\lambda) = 0$, then

$$\begin{aligned} & L_n(\boldsymbol{\gamma}) - L_n(\hat{\boldsymbol{\gamma}}^{or}) \\ \geq \; & L_{n1}(\boldsymbol{\gamma}) - L_{n1}(\hat{\boldsymbol{\gamma}}^{or}) + \sum_{\substack{i \in \mathcal{G}_k, j \in \mathcal{G}_{k'} \\ k \neq k'}} \{P_\tau(\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|_2, \lambda)\} - \sum_{k \neq k'} P_\tau(\|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k'}\|_2, \lambda). \end{aligned}$$

As $\|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k'}\|_2 \geq Cb$ and $\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|_2 \geq Cb$ for any $i \in \mathcal{G}_k$, $j \in \mathcal{G}_{k'}$, $k \neq k'$ from previous arguments, it follows from the condition $Cb \geq \tau\lambda$ that

$$\sum_{\substack{i \in \mathcal{G}_k, j \in \mathcal{G}_{k'} \\ k \neq k'}} \{P_\tau(\|\boldsymbol{\gamma}_i - \boldsymbol{\gamma}_j\|_2, \lambda)\} = 0$$

and

$$\sum_{k \neq k'} P_\tau(\|\hat{\boldsymbol{\theta}}_k - \hat{\boldsymbol{\theta}}_{k'}\|_2, \lambda) = 0$$

Thus,

$$L_n(\boldsymbol{\gamma}) - L_n(\hat{\boldsymbol{\gamma}}^{or}) \geq L_{n1}(\boldsymbol{\gamma}) - L_{n1}(\hat{\boldsymbol{\gamma}}^{or}).$$

Further by the definition of $\bar{\boldsymbol{\gamma}}$ and (A.8), we have $L_{n1}(\boldsymbol{\gamma}) \geq L_{n1}(\hat{\boldsymbol{\gamma}}^{or})$ for any $\boldsymbol{\gamma}$ satisfying $\|\boldsymbol{X}_i(\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i^{or})\|_2^2 = M\delta_n$ with sufficiently large M. Therefore, (A.7) is proved, which means

$$\|\hat{\beta}_i - \hat{\beta}_i^{or}\|_2^2 = O_p(J/m_{(n)} + J^{-2r}).$$

Combing this result with Theorem 3.1 yields

$$\|\hat{\beta}_i - \beta_i\|_2^2 \leq \|\hat{\beta}_i - \hat{\beta}_i^{or}\|_2^2 + \|\hat{\beta}_i^{or} - \beta_i\|_2^2 = O_p(J/m_{(n)} + J^{-2r}).$$

This completes the proof of Theorem 3.2.

Proof of Theorem 3.3. (i) For $i, j \in \mathcal{G}_k$, we have $\beta_i = \beta_j$. Then

$$\begin{aligned} \|\hat{\beta}_i - \hat{\beta}_j\|_2^2 &\leq \|\hat{\beta}_i - \beta_i\|_2^2 + \|\beta_i - \beta_j\|_2^2 + \|\hat{\beta}_j - \beta_j\|_2^2 \\ &\leq 2 \max_i \|\hat{\beta}_i - \beta_i\|_2^2 + 0 = O_p(J/m_{(n)} + J^{-2r}) \to 0 \end{aligned}$$

as $m_{(n)} \to \infty$, where the last equality holds from Theorem 3.2. This means $\hat{\beta}_i$ and $\hat{\beta}_j$ will fall into the same group with probability approaching to 1.

(ii) For any $i \in \mathcal{G}_k$, $j \in \mathcal{G}_{k'}$, $k \neq k'$, it follows from Theorem 3.2 that

$$\begin{aligned} \|\hat{\beta}_i - \hat{\beta}_j\|_2^2 &= \|\hat{\beta}_i - \beta_i + \beta_i - \beta_j + \beta_j - \hat{\beta}_j\|_2^2 \\ &\geq \min_{\substack{i \in \mathcal{G}_k, j \in \mathcal{G}_{k'} \\ k \neq k'}} \|\beta_i - \beta_j\|_2^2 - 2 \max_{1 \leq i \leq n} \|\hat{\beta}_i - \beta_i\|_2^2 \\ &= b^2 - O_p(J/m_{(n)} + J^{-2r}) \to b^2 > 0, \end{aligned}$$

which implies that $\hat{\beta}_i$ and $\hat{\beta}_j$ will fall into the different groups with probability approaching to 1. Therefore, the proof is completed by the combinations of conclusions (i) and (ii).

In what follows, let $a_n \asymp b_n$ mean that a_n/b_n and b_n/a_n are bounded for given sequences of positive numbers a_n and b_n. For a square integrable function g on $\mathbb{T}$, we define the norms $\|g\|^2 = E(g(t)^2)$ and $\|g\|_\infty = \sup_{t \in \mathbb{T}} |g(t)|$.

Proof of Theorem 3.4. We can conclude from the results of Theorems 3.1-3.3 that the proposed penalized estimators performs asymptotically equivalent to the oracle ones as m_n approaching to infinite. Thus, we only need to prove the asymptotic normalities of the oracle estimators $\hat{\boldsymbol{\alpha}}^{or}(t) = (\hat{\alpha}_1(t), \ldots, \hat{\alpha}_K(t))^T = \mathbb{B}(t)\hat{\boldsymbol{\theta}}$, where $\mathbb{B}(t) = \boldsymbol{I}_K \otimes \boldsymbol{B}(t)^T$ (Kronecker product) and $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\theta}}_1^T, \ldots, \hat{\boldsymbol{\theta}}_K^T)^T$. To this end, we first show that

$$\mathrm{Var}\left(\hat{\alpha}_k(t)\right)^{-1/2} \left\{\hat{\alpha}_k(t) - \mathrm{E}(\hat{\alpha}_k(t))\right\} \xrightarrow{d} N(0,1), \qquad k = 1, \ldots, K. \tag{A.9}$$

Recall that for $k = 1, \ldots, K$, we have

$$\hat{\boldsymbol{\theta}}_k = \arg\min_{\boldsymbol{\theta}_k} (\boldsymbol{Y}_{(k)} - \boldsymbol{X}_{(k)}\boldsymbol{\theta}_k)^T \boldsymbol{V}_{(k)}^{-1} (\boldsymbol{Y}_{(k)} - \boldsymbol{X}_{(k)}\boldsymbol{\theta}_k) = \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{Y}_{(k)},$$

where $\boldsymbol{Y}_{(k)} = \left\{\boldsymbol{Y}_i^T : i \in \mathcal{G}_k\right\}^T, \boldsymbol{X}_{(k)} = \left\{\boldsymbol{X}_i^T : i \in \mathcal{G}_k\right\}^T$ and $\boldsymbol{V}_{(k)} = \mathrm{diag}\left\{\boldsymbol{V}_i : i \in \mathcal{G}_k\right\}$. We only consider an fixed k here since other cases can be proved in the same way. For any $t \in \mathbb{T}$, $\hat{\alpha}_k(t) = \boldsymbol{B}(t)^T \hat{\boldsymbol{\theta}}_k = \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{Y}_{(k)}$, and $\mathrm{E}\left(\hat{\alpha}_k(t)\right) = \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \alpha_k(t)$. Thus,

$$\begin{aligned}
\hat{\alpha}_k(t) - \mathrm{E}(\hat{\alpha}_k(t)) &= \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{\varepsilon}_{(k)} \\
&= \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{\Sigma}_{(k)}^{1/2} \boldsymbol{\Sigma}_{(k)}^{-1/2} \boldsymbol{\varepsilon}_{(k)} \\
&\triangleq \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{Z}_{(k)}^T \boldsymbol{e}_{(k)},
\end{aligned}$$

where $\boldsymbol{Z}_{(k)} = \boldsymbol{\Sigma}_{(k)}^{1/2} \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}$ and $\boldsymbol{e}_{(k)} = \boldsymbol{\Sigma}_{(k)}^{-1/2} \boldsymbol{\varepsilon}_{(k)}$. Obviously, we have $\mathrm{E}(\boldsymbol{e}_{(k)}) = 0$ and $\mathrm{Var}(\boldsymbol{e}_{(k)}) = I$, which means that the elements $\{e_{(k)\iota}\}_{\iota=1}^{N_k}$ can be seen as independent random variables with zero mean and unit variance.

Denote $\boldsymbol{Z}_{(k)}^{\iota}$ as a S-dimensional column vector comprised by the ιth row of $\boldsymbol{Z}_{(k)}$, it follows that

$$\hat{\alpha}_k(t) - \mathrm{E}\left(\hat{\alpha}_k(t)\right) = \sum_{\iota=1}^{N_k} \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{Z}_{(k)}^{\iota} e_{(k)\iota} = \sum_{\iota=1}^{N_k} \phi_{(k)\iota} e_{(k)\iota}$$

and

$$\mathrm{Var}(\hat{\alpha}_k(t)) = \sum_{\iota=1}^{N_k} \phi_{(k)\iota}^2,$$

where $\phi_{(k)\iota} = \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)} \right)^{-1} \boldsymbol{Z}_{(k)}^{\iota}$. Therefore, if the Lindeberg condition holds, that is, $\max_\iota \phi_{(k)\iota}^2 / \sum_{\iota=1}^{N_k} \phi_{(k)\iota}^2 \to 0$, we can obtain that

$$\frac{\sum_{\iota=1}^{N_k} \phi_{(k)\iota} e_{(k)\iota}}{\sqrt{\sum_{\iota=1}^{N_k} \phi_{(k)\iota}^2}} \xrightarrow{d} N(0,1), \tag{A.10}$$

which indicates the result (A.9).

In fact, by the definition of $\boldsymbol{Z}_{(k)}$, condition (C5) and Lemma A.1, we have

$$\begin{aligned} \phi_{(k)\iota}^2 &= \left(\boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)} \right)^{-1} \boldsymbol{Z}_{(k)}^{\iota} \right)^2 \asymp \frac{1}{N_k^2} \left(\boldsymbol{B}(t)^T \boldsymbol{Z}_{(k)}^{\iota} \right)^2 \\ &\leq \frac{C}{N_k^2} \sum_{l=1}^{S} B_l^2(t) \sum_{l=1}^{S} B_l^2(t_{(k)\iota}), \end{aligned} \tag{A.11}$$

where the last step holds by the Cauchy-Schwarz inequality and condition (C5). Moreover, based on the same rationale as above, it follows that

$$\begin{aligned} \sum_{\iota=1}^{N_k} \phi_{(k)\iota}^2 &= \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)} \right)^{-1} \sum_{\iota=1}^{N_k} \boldsymbol{Z}_{(k)}^{\iota} \boldsymbol{Z}_{(k)}^{\iota^T} \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)} \right)^{-1} \boldsymbol{B}(t) \\ &= \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)} \right)^{-1} \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{\Sigma}_{(k)} \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)} \right) \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)} \right)^{-1} \boldsymbol{B}(t) \\ &\asymp \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)} \right)^{-1} \boldsymbol{B}(t) \asymp \frac{1}{N_k} \sum_{l=1}^{S} B_l^2(t). \end{aligned} \tag{A.12}$$

Combining the expressions (A.11) and (A.12) leads to

$$\frac{\phi_{(k)\iota}^2}{\sum_{\iota=1}^{N_k} \phi_{(k)\iota}^2} \leq \frac{C}{N_k} \sum_{l=1}^{S} B_l^2(t_{(k)\iota}) \leq \frac{C}{N_k} \sup_{t \in \mathbb{T}} \sum_{l=1}^{S} B_l^2(t).$$

Observing that

$$\sup_{t \in \mathbb{T}} \sqrt{\sum_{l=1}^{S} B_l^2(t)} = \sup_{t \in \mathbb{T}} \sup_{b_l} \frac{\sum_{l=1}^{S} |B_l(t) b_l|}{\sqrt{\sum_{l=1}^{S} b_l^2}} \leq \sup_{b_l} \frac{\sup_{t \in \mathbb{T}} \sum_{l=1}^{S} |B_l(t) b_l|}{\sqrt{\sum_{l=1}^{S} b_l^2}} = \sup_{g \in G} \frac{\|g\|_\infty}{\|g\|_2},$$

where the last step due to expression (A.1). Based on the definitions of norms and condition (C1) that the density function $f(t)$ is uniformly bounded away from 0 and infinity on $\mathbb{T}$, it is easy to verify $\|g\|_2 \asymp \|g\|$. Let $A_n = \sup_{g\in G} \|g\|_\infty/\|g\|$, hence

$$\max_\iota \frac{\phi_{(k)\iota}^2}{\sum_{\iota=1}^{N_k} \phi_{(k)\iota}^2} \leq \frac{C}{N_k} \sup_{g\in G} \frac{\|g\|_\infty^2}{\|g\|_2^2} \leq \frac{C}{N_k} \sup_{g\in G} \frac{\|g\|_\infty^2}{\|g\|^2}(1+o_p(1)) = \frac{CA_n^2}{N_k}(1+o_p(1)).$$

Based on conditions (C1) and (C4), we have $A_n^2 \asymp S$. Therefore,

$$\max_\iota \frac{\phi_{(k)\iota}^2}{\sum_{\iota=1}^{N_k} \phi_{(k)\iota}^2} \leq \frac{CA_n^2}{N_k}(1+o_p(1)) \asymp \frac{S}{N_k}(1+o_p(1)) \to 0,$$

which implies the validation of Lindeberg condition. Consequently, (A.10) is proved, and then (A.9) holds.

On the other hand, according to conditions (C3) and (C4), it is easy to verity that all the conditions assumed in Theorem 5.1 of [38] hold. This implies, by virtue of condition (C5), that

$$|\mathrm{E}\left(\hat{\alpha}_k(t)\right) - \alpha_k(t)| = O_p(J^{-r}).$$

Moreover, it follows from (A.12), condition (C5), Lemma A.1 and the properties of B-spline that

$$\begin{aligned}\mathrm{Var}(\hat{\alpha}_k(t)) &= \boldsymbol{B}(t)^T \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{\Sigma}_{(k)} \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right) \left(\boldsymbol{X}_{(k)}^T \boldsymbol{V}_{(k)}^{-1} \boldsymbol{X}_{(k)}\right)^{-1} \boldsymbol{B}(t) \\ &\asymp S/N_k,\end{aligned}$$

where $\boldsymbol{\Sigma}_{(k)} = \mathrm{diag}\left\{\boldsymbol{\Sigma}_i : i \in \mathcal{G}_k\right\}$. Taking into account of $J/m_{(n)}^{1/(2r+1)} \to \infty$, we have

$$\sup_{t\in\mathbb{T}} \left| \frac{\mathrm{E}\left(\hat{\alpha}_k(t)\right) - \hat{\alpha}_k(t)}{\sqrt{\mathrm{Var}(\hat{\alpha}_k(t))}} \right| = o_p(1). \tag{A.13}$$

Combining the results of (A.9) and (A.13) leads to

$$\mathrm{Var}\left(\hat{\alpha}_k(t)\right)^{-1/2} \left(\hat{\alpha}_k(t) - \alpha_k(t)\right) \xrightarrow{d} N(0,1). \tag{A.14}$$

We further let $\boldsymbol{X}_0 = \text{diag}\left(\boldsymbol{X}_{(1)}, ..., \boldsymbol{X}_{(K)}\right), \boldsymbol{V}_0 = \text{diag}\left(\boldsymbol{V}_{(1)}, ..., \boldsymbol{V}_{(K)}\right)$ and $\boldsymbol{\Sigma}_0 = \text{diag}\left(\boldsymbol{\Sigma}_{(1)}, ..., \boldsymbol{\Sigma}_{(K)}\right)$, where $\boldsymbol{Y}_{(k)} = \left\{\boldsymbol{Y}_i^T : i \in \mathcal{G}_k\right\}^T$, $\boldsymbol{V}_{(k)} = \text{diag}\left\{\boldsymbol{V}_i : i \in \mathcal{G}_k\right\}$, $\boldsymbol{X}_{(k)} = \left\{\boldsymbol{X}_i^T : i \in \mathcal{G}_k\right\}^T$ and $\boldsymbol{\Sigma}_{(k)} = \text{diag}\left\{\boldsymbol{\Sigma}_i : i \in \mathcal{G}_k\right\}$. Finally, by the expression of $\hat{\boldsymbol{\alpha}}^{or}(t)$ and the independence assumption of different subgroup, we can obtain

$$\text{Var}\left(\hat{\boldsymbol{\alpha}}^{or}(t)\right)^{-1/2}\left(\hat{\boldsymbol{\alpha}}^{or}(t) - \boldsymbol{\alpha}(t)\right) \xrightarrow{d} N(\boldsymbol{0}, \boldsymbol{I}_K),$$

where

$$\text{Var}\left(\hat{\boldsymbol{\alpha}}^{or}(t)\right) = \mathbb{B}(t)\left(\boldsymbol{X}_0^T\boldsymbol{V}_0^{-1}\boldsymbol{X}_0\right)^{-1}\left(\boldsymbol{X}_0^T\boldsymbol{V}_0^{-1}\boldsymbol{\Sigma}_0\boldsymbol{V}_0^{-1}\boldsymbol{X}_0\right)\left(\boldsymbol{X}_0^T\boldsymbol{V}_0^{-1}\boldsymbol{X}_0\right)^{-1}\mathbb{B}(t)^T \quad \text{(A.15)}$$

with $\mathbb{B}(t) = \boldsymbol{I}_K \otimes \boldsymbol{B}(t)^T$ (Kronecker product). Therefore, we complete the proof of Theorem 3.4 based on (A.14) and the conclusions of Theorems 3.1-3.3.

Appendix B

Supplementary Materials for

Chapter 4

B.1 Proof of Proposition 4.1

In this section, we provide the proof of Proposition 4.1. The dimension of $W_{\boldsymbol{\ell}}$ satisfies

$$|W_{\boldsymbol{\ell}}| \leq \prod_{j=1}^{d} 2^{\ell_j \vee 2 - 1} = 2^{\sum_{j=1}^{d} \ell_j \vee 2 - d}.$$

Thus,

$$\begin{aligned} |V_m^{(1)}| &\leq \sum_{|\boldsymbol{\ell}|_1 \leq m} 2^{\sum_{j=1}^{d} \ell_j \vee 2 - d} \leq \sum_{|\boldsymbol{\ell}|_1 \leq m} 2^{|\boldsymbol{\ell}|_1 + d} = \sum_{k=0}^{m} 2^{k+d} \binom{d-1+k}{d-1} \\ &= 2^d \sum_{k=0}^{m} 2^k \binom{d-1+k}{d-1} = 2^d \left\{ (-1)^d + 2^{m+1} \sum_{k=0}^{d-1} \binom{m+d}{k} (-2)^{d-1-k} \right\}, \end{aligned}$$

where the last equality follows from (3.62) of [10]. We assume that d is even. The result for odd d can be proved similarly. Then

$$\sum_{k=0}^{d-1}\binom{m+d}{k}(-2)^{d-1-k}=\sum_{v=0}^{d/2-1}2^{2v}\left\{\binom{m+d}{d-(1+2v)}-2\binom{m+d}{d-(2+2v)}\right\}.$$

Moreover,

$$\begin{aligned}
&\binom{m+d}{d-(1+2v)}-2\binom{m+d}{d-(2+2v)}\\
=\;&\frac{(m+d)!}{(d-(1+2v))!(m+1+2v)!}-2\frac{(m+d)!}{(d-(2+2v))!(m+2v+2)!}\\
=\;&\frac{(m+d)!}{(d-(2+2v))!(m+1+2v)!}\left\{\frac{1}{d-(1+2v)}-\frac{2}{m+2v+2}\right\}\\
=\;&\frac{(m+d)!(m+6v-2d+4)}{(d-(1+2v))!(m+2v+2)!}\\
\\
=\;&\frac{(m+d)(m+d-1)\times\cdots\times(m+2v+3)(m+6v-2d+4)}{(d-(1+2v))!}\\
\leq\;&\frac{(m+d)^{(d-(2+2v))}}{(d-(1+2v))!}.
\end{aligned}$$

Thus,

$$\sum_{k=0}^{d-1}\binom{m+d}{k}(-2)^{d-1-k}\leq\sum_{v=0}^{d/2-1}2^{2v}\frac{(m+d)^{(d-(2+2v))}}{(d-(1+2v))!}\leq 2^{d-2}\frac{(m+d)^{(d-2)}}{(d-1)!}.$$

By stirling's formula,

$$(d-1)!\geq\sqrt{2\pi}(d-1)^{d-1/2}e^{-(d-1)}.$$

Therefore,

$$\sum_{k=0}^{d-1}\binom{m+d}{k}(-2)^{d-1-k}\leq 2^{d-2}\frac{(m+d)^{(d-2)}}{\sqrt{2\pi}(d-1)^{d-1/2}e^{-(d-1)}},$$

and hence

$$|V_m^{(1)}|\leq 2^{d+1}2^{m+1}2^{d-2}\frac{(m+d)^{(d-2)}e^{(d-1)}}{\sqrt{2\pi}(d-1)^{d-1/2}}=2\sqrt{\frac{2}{\pi}}\frac{\sqrt{d-1}}{(m+d)}2^m\left(4e\frac{m+d}{d-1}\right)^{d-1}.$$

Moreover, let $\boldsymbol{\ell}_{-1} = (\ell_2, \ldots, \ell_d)^\top$. Then, $|V_0^{(1)}| = \sum_{|\boldsymbol{\ell}|_1=0} \prod_{j=1}^d 2 = 2^d$, and for $m \geq 1$,

$$\begin{aligned} |V_m^{(1)}| &\geq \sum_{|\boldsymbol{\ell}|_1=0} \prod_{j=1}^d 2 + \sum_{1\leq \ell_1 \leq m, |\boldsymbol{\ell}_{-1}|_1=0} (\prod_{j=2}^d 2) 2^{\ell_1 - 1} = 2^d + 2^{d-1} \sum_{1 \leq \ell_1 \leq m} 2^{\ell_1 - 1} \\ &= 2^d + 2^{d-1}(2^m - 1) = 2^{d-1}(2^m - 1 + 2) \geq 2^{d-1}(2^m + 1). \end{aligned}$$

Therefore, $|V_m^{(1)}| \geq 2^{d-1}(2^m + 1)$ for any $m \geq 0$.

B.2 Proof of Proposition 4.2

This section provides the proof of Proposition 4.2. Based on (2.1) and (2.1.2), one has

$$||f_m - f||_2 = || \sum_{\mathbf{1}_d \leq \boldsymbol{\ell} \leq \boldsymbol{\infty}} g_{\boldsymbol{\ell}}(\boldsymbol{x}) - \sum_{|\boldsymbol{\ell}|_1 \leq m} g_{\boldsymbol{\ell}}(\boldsymbol{x})||_2 = || \sum_{|\boldsymbol{\ell}|_1 > m} g_{\boldsymbol{\ell}}(\boldsymbol{x})||_2.$$

By (4.7) and Assumption 3, one has

$$\begin{aligned} || \sum_{|\boldsymbol{\ell}|_1 > m} g_{\boldsymbol{\ell}}(\boldsymbol{x})||_2 &\leq \sum_{|\boldsymbol{\ell}|_1 > m} ||g_{\boldsymbol{\ell}}||_2 \leq \sum_{|\boldsymbol{\ell}|_1 > m} c_\mu 3^{-d} 2^{-2|\boldsymbol{\ell}|_1} ||D^{\mathbf{2}} f||_{L^2} \\ &= c_\mu 3^{-d} ||D^{\mathbf{2}} f||_{L^2} \sum_{|\boldsymbol{\ell}|_1 > m} 2^{-2|\boldsymbol{\ell}|_1}. \end{aligned}$$

Then, one has that for arbitrary $s \in \mathbb{N}$,

$$\begin{aligned} \sum_{|\boldsymbol{\ell}|_1 > m} 2^{-s|\boldsymbol{\ell}|_1} &= \sum_{k'=m+1}^{\infty} 2^{-sk'} \binom{k'+d-1}{d-1} = \sum_{k=0}^{\infty} 2^{-s(k+m+1)} \binom{k+m+1+d-1}{d-1} \\ &= 2^{-s(m+1)} \sum_{k=0}^{\infty} 2^{-sk} \binom{k+m+1+d-1}{d-1} \\ &\leq 2^{-s(m+1)} 2A(d,m), \end{aligned}$$

where $A(d,m) = \sum_{k=0}^{d-1} \binom{m+d}{k}$, where the last inequality follows from Lemma 3.7 of [10].

$$\begin{aligned} ||f_m - f||_2 &\leq \sum_{|\boldsymbol{\ell}|_1 > m} ||g_{\boldsymbol{\ell}}||_2 \leq c_\mu 3^{-d} ||D^{\mathbf{2}} f||_{L^2} 2^{-2(m+1)} 2A(d,m) \\ &= 2^{-1} c_\mu 2^{-2m} 3^{-d} A(d,m) ||D^{\mathbf{2}} f||_{L^2}. \end{aligned}$$

Moreover, for $d \geq 3$, $A(d,m) \leq (d-1)\frac{(m+d)^{d-1}}{(d-1)!} = \frac{(m+d)^{d-1}}{(d-2)!}$, and by stirling's formula,

$$(d-2)! \geq \sqrt{2\pi}(d-2)^{d-3/2}e^{-(d-2)}.$$

Then

$$A(d,m) \leq \frac{(m+d)^{d-1}}{\sqrt{2\pi}(d-2)^{d-3/2}e^{-(d-2)}} = \frac{\sqrt{d-2}}{\sqrt{2\pi}}\left(\frac{m+d}{d-2}\right)^{d-1} e^{(d-2)}.$$

Therefore,

$$\begin{aligned} ||f_m - f||_2 &\leq 2^{-1}c_\mu 2^{-2m}3^{-d}\frac{\sqrt{d-2}}{\sqrt{2\pi}}\left(\frac{m+d}{d-2}\right)^{d-1} e^{(d-2)}||D^{\mathbf{2}}f||_{L^2} \\ &= \widetilde{c}2^{-2m}\sqrt{d-2}\left(\frac{e}{3}\frac{m+d}{d-2}\right)^{d-1}||D^{\mathbf{2}}f||_{L^2}, \end{aligned}$$

where $\widetilde{c} = 2^{-1}c_\mu(3\sqrt{2\pi}e)^{-1}$. For $d=2$, $A(d,m) = m+3$. Thus, $||f_m - f||_2 \leq 2^{-1}3^{-2}c_\mu 2^{-2m}(m+3)||D^{\mathbf{2}}f||_{L^2}$.

B.3 Proof of Proposition 4.3

In this section, we provide the proof of Proposition 4.3. It is clear that $||\widetilde{f}_R - f||_2 = ||\widetilde{f}_R - f_m + f_m - f||_2 \leq ||\widetilde{f}_R - f_m||_2 + ||f_m - f||_2$. The rate of $||f_m - f||_2$ is provided in Proposition 4.2. Next we derive the rate of $||\widetilde{f}_R - f_m||_2$ as follows. By (2.1.2) and (4.11), we have

$$||\widetilde{f}_R - f_m||_2 \leq \sup_{\boldsymbol{x}\in\mathcal{X}} \sum_{|\boldsymbol{\ell}|_1 \leq m} \sum_{s\in I_{\boldsymbol{\ell}}} |\gamma_{\boldsymbol{\ell},\boldsymbol{s}}||\widetilde{\phi}_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x}) - \phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})|.$$

Since a given $\boldsymbol{x}$ belongs to at most one of the disjoint supports for $\phi_{\boldsymbol{\ell},\boldsymbol{s}}(\boldsymbol{x})$, this result together with (4.10) lead to

$$||\widetilde{f}_R - f_m||_2 \leq 3\cdot 2^{-2R-2}(d-1)\sum_{|\boldsymbol{\ell}|_1\leq m} |\gamma_{\boldsymbol{\ell},\boldsymbol{s}_{\boldsymbol{\ell}}}|,$$

for some $\boldsymbol{s}_{\boldsymbol{\ell}}$. Moreover, by (4.5), we have

$$\begin{aligned}\sum_{|\boldsymbol{\ell}|_1 \leq m} |\gamma_{\boldsymbol{\ell},\boldsymbol{s}_{\boldsymbol{\ell}}}| &\leq \sum_{|\boldsymbol{\ell}|_1 \leq m} 6^{-d/2} 2^{-(3/2)|\boldsymbol{\ell}|_1} ||D^{\mathbf{2}} f||_{L^2} \\ &= 6^{-d/2} ||D^{\mathbf{2}} f||_{L^2} \sum_{|\boldsymbol{\ell}|_1 \leq m} 2^{-(3/2)|\boldsymbol{\ell}|_1} = 6^{-d/2} ||D^{\mathbf{2}} f||_{L^2} \sum_{k=0}^{m} 2^{-(3/2)k} \binom{k+d-1}{d-1}.\end{aligned}$$

Since $\sum_{k=0}^{\infty} \left(\frac{1}{\sqrt{8}}\right)^k \left(1-\frac{1}{\sqrt{8}}\right)^d \binom{k+d-1}{d-1} = 1$, it implies that

$$\sum_{k=0}^{m} 2^{-(3/2)k} \binom{k+d-1}{d-1} \leq \sum_{k=0}^{\infty} 2^{-(3/2)k} \binom{k+d-1}{d-1} = \left(1-\frac{1}{\sqrt{8}}\right)^{-d},$$

and thus

$$\sum_{|\boldsymbol{\ell}|_1 \leq m} |\gamma_{\boldsymbol{\ell},\boldsymbol{s}_{\boldsymbol{\ell}}}| \leq \left\{\sqrt{6}(1-\frac{1}{\sqrt{8}})\right\}^{-d} ||D^{\mathbf{2}} f||_{L^2} \leq (\sqrt{3/2})^{-d} ||D^{\mathbf{2}} f||_{L^2}. \tag{B.1}$$

Therefore,

$$\begin{aligned}||\widetilde{f}_R - f_m||_2 &\leq 3 \cdot 2^{-2R-2}(d-1)(\sqrt{3/2})^{-d} ||D^{\mathbf{2}} f||_{L^2} \\ &= (3/4) 2^{-2R}(d-1)(\sqrt{3/2})^{-d} ||D^{\mathbf{2}} f||_{L^2}.\end{aligned}$$

The above result and (4.8) lead to

$$\begin{aligned}||\widetilde{f}_R - f||_2 &\leq ||\widetilde{f}_R - f_m||_2 + ||f_m - f||_2 \\ &\leq \left\{(3/4)2^{-2R}(d-1)(\sqrt{3/2})^{-d} + \widetilde{c} 2^{-2m}\sqrt{d-2}\left(\frac{e}{3}\frac{m+d}{d-2}\right)^{d-1}\right\} ||D^{\mathbf{2}} f||_{L^2} \\ &\leq \left\{\sqrt{\frac{3}{8}} 2^{-2R}(d-1)(\sqrt{\frac{2}{3}})^{d-1} + \widetilde{c} 2^{-2m}\sqrt{d-2}\left(\frac{e}{3}\frac{m+d}{d-2}\right)^{d-1}\right\} ||D^{\mathbf{2}} f||_{L^2},\end{aligned}$$

for $d \geq 3$. The result for $d = 2$ follows from the same procedure. Moreover, the ReLU network used to construct the approximator $\widetilde{f}_R$ has depth $\mathcal{O}(R \log_2 d)$, the computational units $\mathcal{O}(Rd) \times |V_m^{(1)}|$, and the number of weights $\mathcal{O}(Rd) \times |V_m^{(1)}|$. By the upper bound for $|V_m^{(1)}|$ established in (4.1), we have that the number of the computational units is

$\mathcal{O}(Rd)\times\mathcal{O}\left(\frac{\sqrt{d}}{(m+d)}2^m\left(4e\frac{m+d}{d-1}\right)^{d-1}\right)=\mathcal{O}\left(2^m d^{3/2}R\,(m+d)^{-1}\left(4e\frac{m+d}{d-1}\right)^{d-1}\right)$, and the number of weights is $\mathcal{O}\left(2^m d^{3/2}R\,(m+d)^{-1}\left(4e\frac{m+d}{d-1}\right)^{d-1}\right)$.

B.4 Proofs of Proposition 4.4

Under Condition (4.16) given in Assumption 4.5, by the definition of f^0_{RN} given in (4.14) and Proposition 4.3, the approximation error

$$\mathcal{E}(f^0_{RN})-\mathcal{E}(f_0)\le\mathcal{E}(\widetilde{f}_R)-\mathcal{E}(f_0)\le b_\rho||\widetilde{f}_R-f_0||_2^2$$
$$\le b_\rho\left\{\sqrt{\frac{3}{8}}2^{-2R}(d-1)(\sqrt{\frac{2}{3}})^{d-1}+\widetilde{c}2^{-2m}\sqrt{d-1}\left(\frac{e}{3}\frac{m+d}{d-2}\right)^{d-1}\right\}^2||D^{\mathbf{2}}f_0||^2_{L^2},$$

for $d\ge 3$, and

$$\mathcal{E}(f^0_{RN})-\mathcal{E}(f_0)\le b_\rho\left\{\sqrt{\frac{3}{8}}2^{-2R}(d-1)(\sqrt{\frac{2}{3}})^{d-1}+18^{-1}c_\mu 2^{-2m}(m+3)\right\}^2||D^{\mathbf{2}}f_0||^2_{L^2},$$

for $d=2$. Assuming that $m^{-1}=o(1)$ and $m\lesssim R$ as $n\to\infty$, since $\frac{e}{3}\frac{m+d}{d-2}>\sqrt{\frac{2}{3}}$, we have that for sufficiently large n, $\sqrt{\frac{3}{8}}2^{-2R}(d-1)(\sqrt{\frac{2}{3}})^{d-1}<\widetilde{c}2^{-2m}\sqrt{d-1}\left(\frac{e}{3}\frac{m+d}{d-2}\right)^{d-1}$ for $d\ge 3$, and $\sqrt{\frac{3}{8}}2^{-2R}(d-1)(\sqrt{\frac{2}{3}})^{d-1}<18^{-1}c_\mu 2^{-2m}(m+3)$ for $d=2$. Thus,

$$\mathcal{E}(f^0_{RN})-\mathcal{E}(f_0)\le\zeta_{R,m,d},$$

where

$$\begin{aligned}\zeta_{R,m,d} &= 4b_\rho C_f^2\widetilde{c}^2 2^{-4m}d\left(\frac{e}{3}\frac{m+d}{d-2}\right)^{2(d-1)},\ \text{for } d\ge 3\\ \zeta_{R,m,d} &= 81^{-1}b_\rho C_f^2 c_\mu^2 2^{-4m}(m+3)^2,\ \text{for } d=2.\end{aligned}$$

B.5 Proofs of Theorems 4.1 and 4.2

We first introduce a Bernstein inequality which will be used to establish the bounds in Theorems 4.1 and 4.2.

Lemma B.1 *Let $\mathcal{G}$ be a set of scalar-valued functions on $\mathcal{X}\times\mathcal{Y}$ such that for each $\xi(\boldsymbol{X},Y)\in\mathcal{G}$, $\mathbb{E}\{\xi(\boldsymbol{X},Y)\}\geq 0$, $\mathbb{E}\{\xi(\boldsymbol{X},Y)^2\}\leq c_1\mathbb{E}\{\xi(\boldsymbol{X},Y)\}$ and $|\xi(\boldsymbol{X},Y)-\mathbb{E}\{\xi(\boldsymbol{X},Y)\}|\leq c_2$ almost everywhere for some constants $c_1,c_2\in(0,\infty)$. Then for every $\epsilon>0$ and $0<\alpha\leq 1$, we have*

$$\begin{aligned} & P\left\{\sup_{\xi\in\mathcal{G}}\frac{\mathbb{E}\{\xi(\boldsymbol{X},Y)\}-n^{-1}\sum_{i=1}^n\xi(\boldsymbol{X}_i,Y_i)}{\sqrt{\mathbb{E}\{\xi(\boldsymbol{X},Y)\}+\epsilon}}>4\alpha\sqrt{\epsilon}\right\} \\ \leq\ & \mathcal{N}(\alpha\epsilon,\mathcal{G},||\cdot||_\infty)\exp\left(-\frac{\alpha^2 n\epsilon}{2c_1+2c_2/3}\right). \end{aligned}$$

Proof. Let $\{\xi_j\}_{j=1}^J\in\mathcal{G}$ with $J=\mathcal{N}(\alpha\epsilon,\mathcal{G},||\cdot||_\infty)$ being such that $\mathcal{G}$ is covered by $||\cdot||_\infty$- balls centered on ξ_j with radius $\alpha\epsilon$. Denote $\mu(\xi)=\mathbb{E}\{\xi(\boldsymbol{X},Y)\}$ and $\sigma^2(\xi)=$var$\{\xi(\boldsymbol{X},Y)\}$. For each j, the one-side Bernstein inequality in Corollary 3.6 of [18] implies that

$$\begin{aligned} & P\left\{\frac{\mu(\xi_j)-n^{-1}\sum_{i=1}^n\xi_j(\boldsymbol{X}_i,Y_i)}{\sqrt{\mu(\xi_j)+\epsilon}}>\alpha\sqrt{\epsilon}\right\} \\ \leq\ & \exp\left(-\frac{\alpha^2 n(\mu(\xi_j)+\epsilon)\epsilon}{2\{\sigma^2(\xi_j)+c_2\alpha\sqrt{\mu(\xi_j)+\epsilon}\sqrt{\epsilon}/3\}}\right). \end{aligned} \tag{B.2}$$

Since $\sigma^2(\xi_j)\leq\mathbb{E}\{\xi_j(\boldsymbol{X},Y)^2\}\leq c_1\mu(\xi_j)$, then

$$\begin{aligned} & \sigma^2(\xi_j)+c_2\alpha\sqrt{\mu(\xi_j)+\epsilon}\sqrt{\epsilon}/3 \\ \leq\ & c_1\mu(\xi_j)+c_2(\mu(\xi_j)+\epsilon)/3 \\ \leq\ & c_1\left(\mu(\xi_j)+\epsilon\right)+c_2(\mu(\xi_j)+\epsilon)/3 \\ =\ & (c_1+c_2/3)(\mu(\xi_j)+\epsilon). \end{aligned}$$

The above result together with (B.2) implies that

$$\begin{aligned} & P\left\{\frac{\mu(\xi_j)-n^{-1}\sum_{i=1}^n \xi_j\left(\boldsymbol{X}_i,Y_i\right)}{\sqrt{\mu(\xi_j)+\epsilon}}>\alpha\sqrt{\epsilon}\right\} \\ \leq & \exp\left(-\frac{\alpha^2 n(\mu(\xi_j)+\epsilon)\epsilon}{2(c_1+c_2/3)(\mu(\xi_j)+\epsilon)}\right)=\exp\left(-\frac{\alpha^2 n\epsilon}{2(c_1+c_2/3)}\right). \end{aligned} \tag{B.3}$$

For each $\xi \in \mathcal{G}$, there exists some j such that $||\xi-\xi_j||_\infty \leq \alpha\epsilon$. Then $|\mu(\xi)-\mu(\xi_j)|$ and $|n^{-1}\sum_{i=1}^n \xi\left(\boldsymbol{X}_i,Y_i\right)-n^{-1}\sum_{i=1}^n \xi_j\left(\boldsymbol{X}_i,Y_i\right)|$ are both bounded by $\alpha\epsilon$. Hence,

$$\frac{|\mu(\xi)-\mu(\xi_j)|}{\sqrt{\mu(\xi)+\epsilon}}\leq\alpha\sqrt{\epsilon},\ \frac{|n^{-1}\sum_{i=1}^n \xi\left(\boldsymbol{X}_i,Y_i\right)-n^{-1}\sum_{i=1}^n \xi_j\left(\boldsymbol{X}_i,Y_i\right)|}{\sqrt{\mu(\xi)+\epsilon}}\leq\alpha\sqrt{\epsilon}.$$

This implies that

$$\begin{aligned} \mu(\xi_j)+\epsilon &= \mu(\xi_j)-\mu(\xi)+\mu(\xi)+\epsilon \\ &\leq \alpha\sqrt{\epsilon}\sqrt{\mu(\xi)+\epsilon}+\{\mu(\xi)+\epsilon\} \\ &\leq \sqrt{\epsilon}\sqrt{\mu(\xi)+\epsilon}+\{\mu(\xi)+\epsilon\} \\ &\leq 2\{\mu(\xi)+\epsilon\}, \end{aligned}$$

so that $\sqrt{\mu(\xi_j)+\epsilon}\leq 2\sqrt{\{\mu(\xi)+\epsilon\}}$. Therefore, $\{\mu(\xi)-n^{-1}\sum_{i=1}^n \xi\left(\boldsymbol{X}_i,Y_i\right)\}/\sqrt{\mu(\xi)+\epsilon}\geq 4\alpha\sqrt{\epsilon}$ implies that $\{\mu(\xi_j)-n^{-1}\sum_{i=1}^n \xi_j\left(\boldsymbol{X}_i,Y_i\right)\}/\sqrt{\mu(\xi)+\epsilon}\geq 2\alpha\sqrt{\epsilon}$ and thus $\{\mu(\xi_j)-n^{-1}\sum_{i=1}^n \xi_j\left(\boldsymbol{X}_i,Y_i\right)\}/\sqrt{\mu(\xi_j)+\epsilon}\geq\alpha\sqrt{\epsilon}$. This result together with (B.3) implies

$$\begin{aligned} & P\left\{\sup_{\xi\in\mathcal{G}}\frac{\mu(\xi)-n^{-1}\sum_{i=1}^n \xi\left(\boldsymbol{X}_i,Y_i\right)}{\sqrt{\mu(\xi)+\epsilon}}>4\alpha\sqrt{\epsilon}\right\} \\ \leq & \sum_{j=1}^J P\left\{\frac{\mu(\xi_j)-n^{-1}\sum_{i=1}^n \xi_j\left(\boldsymbol{X}_i,Y_i\right)}{\sqrt{\mu(\xi_j)+\epsilon}}>\alpha\sqrt{\epsilon}\right\}\leq J\exp\left(-\frac{\alpha^2 n\epsilon}{2c_1+2c_2/3}\right). \end{aligned}$$

■

Based on the Bernstein inequality given in Lemma B.1, we next provide a probability bound that will be used for establishing an upper bound for the sampling error $\mathcal{E}(\widehat{f}_{RL})-\mathcal{E}(f_{RL}^0)$.

Lemma B.2 *Under Assumptions 4.1-4.4, we have that for any $\epsilon > 0$ and $0 < \alpha \leq 1$,*

$$P\left\{\sup_{f\in\mathcal{F}(\widetilde{\phi},m,B)} \frac{\mathcal{E}(f)-\mathcal{E}(f_{RL}^0)-(\mathcal{E}_n(f)-\mathcal{E}_n(f_{RL}^0))}{\sqrt{\mathcal{E}(f)-\mathcal{E}(f_{RL}^0)+\epsilon}} > 4\alpha\sqrt{\epsilon}\right\}$$
$$\leq \mathcal{N}(\alpha C_\rho^{-1}\epsilon, \mathcal{F}(\widetilde{\phi},m,B), ||\cdot||_\infty)\exp\left(-\frac{\alpha^2 n\epsilon}{2C_\rho^2 a_\rho^{-1}+8M_\rho/3}\right),$$

where C_ρ, a_ρ and M_ρ are constants given in Assumptions 4.2 and 4.4 and Remark 4.7.

Proof. Let $\mathcal{G} = \{\xi(\boldsymbol{x},y) = \rho(f(\boldsymbol{x}),y) - \rho\left(f_{RL}^0(\boldsymbol{x}),y\right); f \in \mathcal{F}(\widetilde{\phi},m,B), (\boldsymbol{x},y) \in \mathcal{X}\times\mathcal{Y}\}$. For any $f \in \mathcal{F}(\widetilde{\phi},m,B)$,

$$\mathbb{E}\{\xi(\boldsymbol{X},Y)\} = \mathbb{E}\{\rho(f(\boldsymbol{X}),Y)\} - \mathbb{E}\{\rho\left(f_{RL}^0(\boldsymbol{X}),Y\right)\} \geq 0,$$

based on the definition of f_{RL}^0 given in (4.14). By Remark 4.7, we have $|\xi(\boldsymbol{x},y)| \leq 2M_\rho$, for almost every $(\boldsymbol{x},y) \in \mathcal{X}\times\mathcal{Y}$, so that

$$|\xi(\boldsymbol{X},Y) - \mathbb{E}\{\xi(\boldsymbol{X},Y)\}| \leq 4M_\rho,$$

almost surely. Moreover, Assumption 4.2 further implies that $|\xi(\boldsymbol{x},y)| \leq C_\rho|f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x})|$ for almost every $(\boldsymbol{x},y) \in \mathcal{X}\times\mathcal{Y}$. Then

$$\mathbb{E}\{\xi(\boldsymbol{X},Y)^2\} \leq C_\rho^2 \int_{\mathcal{X}} |f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x})|^2 d\mu_X(\boldsymbol{x}) = C_\rho^2||f - f_{RL}^0||_2^2. \tag{B.4}$$

Moreover, under Condition (4.15) in Assumption 4.4,

$$||f - f_{RL}^0||_2^2 \leq a_\rho^{-1}\{\mathcal{E}(f) - \mathcal{E}(f_{RL}^0)\}.$$

Thus

$$\mathbb{E}\{\xi(\boldsymbol{X},Y)^2\} \leq a_\rho^{-1}C_\rho^2\{\mathcal{E}(f) - \mathcal{E}(f_{RL}^0)\} = a_\rho^{-1}C_\rho^2\mathbb{E}\{\xi(\boldsymbol{X},Y)\}. \tag{B.5}$$

By the Bernstein inequality given in Lemma B.1, for every $\epsilon > 0$ and $0 < \alpha \leq 1$, we have

$$\begin{aligned} & P\left\{ \sup_{f \in \mathcal{F}(\widetilde{\phi}, m, B)} \frac{\mathcal{E}(f) - \mathcal{E}(f_{RL}^0) - (\mathcal{E}_n(f) - \mathcal{E}_n(f_{RL}^0))}{\sqrt{\mathcal{E}(f) - \mathcal{E}(f_{RL}^0) + \epsilon}} > 4\alpha\sqrt{\epsilon} \right\} \\ \leq \;& \mathcal{N}(\alpha\epsilon, \mathcal{G}, ||\cdot||_\infty) \exp\left(-\frac{\alpha^2 n\epsilon}{2C_\rho^2 a_\rho^{-1} + 8M_\rho/3} \right). \end{aligned}$$

Since $|\rho\left(f(\boldsymbol{x}), y\right) - \rho\left(f_{RL}^0(\boldsymbol{x}), y\right)| \leq C_\rho |f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x})|$ for almost every $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$, it follows that

$$\mathcal{N}(\alpha\epsilon, \mathcal{G}, ||\cdot||_\infty) \leq \mathcal{N}(\alpha C_\rho^{-1}\epsilon, \mathcal{F}(\widetilde{\phi}, m, B), ||\cdot||_\infty).$$

■

Proof of Theorem 4.1. Let $f = \widehat{f}_{RL}$, $\Delta = \mathcal{E}(\widehat{f}_{RL}) - \mathcal{E}(f_{RL}^0)$ and $\alpha = \sqrt{2}/8$. From the result in Lemma B.2, we have

$$P\left\{ \frac{\Delta - (\mathcal{E}_n(\widehat{f}_{RL}) - \mathcal{E}_n(f_{RL}^0))}{\sqrt{\Delta + \epsilon}} > \sqrt{\epsilon/2} \right\} \leq Q, \tag{B.6}$$

where

$$Q = \mathcal{N}(\sqrt{2}C_\rho^{-1}\epsilon/8, \mathcal{F}(\widetilde{\phi}, m, B), ||\cdot||_\infty) \exp\left(-n\epsilon/C^*\right),$$

in which $C^* = 64(C_\rho^2 a_\rho^{-1} + 4M_\rho/3)$.

a) When $\widehat{f}_{RL} = \widehat{f}_{RL}^U$, we have $\mathcal{E}_n(\widehat{f}_{RL}^U) - \mathcal{E}_n(f_{RL}^0) \leq 0$. Then,

$$P(\Delta > \sqrt{\epsilon/2}\sqrt{\Delta + \epsilon}) \leq Q.$$

Moreover,

$$\begin{aligned} & \Delta > \sqrt{\epsilon/2}\sqrt{\Delta + \epsilon} \\ & \iff \Delta^2 > (\epsilon/2)(\Delta + \epsilon) \\ & \iff (\Delta - \epsilon/4)^2 > (9/16)\epsilon^2 \\ & \iff \Delta > \epsilon \text{ or } \Delta < -(1/2)\epsilon. \end{aligned}$$

Since $\Delta \geq 0$, then $P(\Delta > \sqrt{\epsilon/2}\sqrt{\Delta+\epsilon}) \leq Q$ is equivalent to $P(\Delta > \epsilon) \leq Q$.

b) When $\widehat{f}_{RL} = \widehat{f}_{RL}^{P}$, let $\widehat{f}_{RL}^{P}(\boldsymbol{x}) = \widetilde{\phi}(\boldsymbol{x})^{\top}\widehat{\boldsymbol{\gamma}}_{RL}$ and $f_{RL}^{0}(\boldsymbol{x}) = \widetilde{\phi}(\boldsymbol{x})^{\top}\boldsymbol{\gamma}_{RL}^{0}$. Moreover, let $\Delta_n = (\mathcal{E}_n(\widehat{f}_{RL}^{P}) + 2^{-1}\lambda\widehat{\boldsymbol{\gamma}}_{RL}^{\top}\widehat{\boldsymbol{\gamma}}_{RL} - \mathcal{E}_n(\widetilde{f}_R) - 2^{-1}\lambda\boldsymbol{\gamma}_{RL}^{0\top}\boldsymbol{\gamma}_{RL}^{0})$. We have $\Delta_n \leq 0$. Then

$$\begin{aligned}\frac{\Delta - (\mathcal{E}_n(\widehat{f}_{RL}^{P}) - \mathcal{E}_n(f_{RL}^{0}))}{\sqrt{\Delta+\epsilon}} &= \frac{\Delta + (2^{-1}\lambda\widehat{\boldsymbol{\gamma}}_{RL}^{\top}\widehat{\boldsymbol{\gamma}}_{RL} - 2^{-1}\lambda\boldsymbol{\gamma}_{RL}^{0\top}\boldsymbol{\gamma}_{RL}^{0}) - \Delta_n}{\sqrt{\Delta+\epsilon}} \\ &\geq \frac{\Delta}{\sqrt{\Delta+\epsilon}} + \frac{(2^{-1}\lambda\widehat{\boldsymbol{\gamma}}_{RL}^{\top}\widehat{\boldsymbol{\gamma}}_{RL} - 2^{-1}\lambda\boldsymbol{\gamma}_{RL}^{0\top}\boldsymbol{\gamma}_{RL}^{0})}{\sqrt{\Delta+\epsilon}}. \quad \text{(B.7)}\end{aligned}$$

Since $|\widehat{\boldsymbol{\gamma}}_{RL} - \boldsymbol{\gamma}_{RL}^{0}|_2^2 \leq \lambda_{\min,\widetilde{\phi}}^{-1}||\widehat{f}_{RL}^{P} - f_{RL}^{0}||_2^2$, where $\lambda_{\min,\widetilde{\phi}} = \lambda_{\min}\left\{\int \widetilde{\phi}(\boldsymbol{x})\widetilde{\phi}(\boldsymbol{x})^{\top}d\mu_X(\boldsymbol{x})\right\}$, then under Condition (4.15) in Assumption 4.4,, we have

$$|\widehat{\boldsymbol{\gamma}}_{RL} - \boldsymbol{\gamma}_{RL}^{0}|_2^2 \leq \lambda_{\min,\widetilde{\phi}}^{-1}||\widehat{f}_{RL}^{P} - f_{RL}^{0}||_2^2 \leq \lambda_{\min,\widetilde{\phi}}^{-1}a_{\rho}^{-1}\{\mathcal{E}(\widehat{f}_{RL}^{P}) - \mathcal{E}(f_{RL}^{0})\} = \lambda_{\min,\widetilde{\phi}}^{-1}a_{\rho}^{-1}\Delta. \quad \text{(B.8)}$$

Since $|\boldsymbol{\gamma}_{RL}^{0}|_2^2 \leq \lambda_{\min,\widetilde{\phi}}^{-1}||f_{RL}^{0}||_2^2 \leq \lambda_{\min,\widetilde{\phi}}^{-1}B^2$, then by (B.8),

$$\begin{aligned}&2^{-1}\lambda|\widehat{\boldsymbol{\gamma}}_{RL}^{\top}\widehat{\boldsymbol{\gamma}}_{RL} - \boldsymbol{\gamma}_{RL}^{0\top}\boldsymbol{\gamma}_{RL}^{0}| \\ &\leq 2^{-1}\lambda|\widehat{\boldsymbol{\gamma}}_{RL} - \boldsymbol{\gamma}_{RL}^{0}|_2^2 + \lambda|\boldsymbol{\gamma}_{RL}^{0}|_2|\widehat{\boldsymbol{\gamma}}_{RL} - \boldsymbol{\gamma}_{RL}^{0}|_2 \\ &\leq 2^{-1}\lambda\lambda_{\min,\widetilde{\phi}}^{-1}a_{\rho}^{-1}\Delta + \lambda\lambda_{\min,\widetilde{\phi}}^{-1/2}B(\lambda_{\min,\widetilde{\phi}}^{-1}a_{\rho}^{-1}\Delta)^{1/2} \\ &= 2^{-1}\lambda\lambda_{\min,\widetilde{\phi}}^{-1}a_{\rho}^{-1}\Delta + \lambda\lambda_{\min,\widetilde{\phi}}^{-1}Ba_{\rho}^{-1/2}\sqrt{\Delta}.\end{aligned}$$

Assume that $\lambda\lambda_{\min,\widetilde{\phi}}^{-1}a_{\rho}^{-1} \leq 5^{-1}$ and $\lambda\lambda_{\min,\widetilde{\phi}}^{-1}Ba_{\rho}^{-1/2} \leq 5^{-1}\sqrt{\epsilon/2}$.

Then $\lambda\lambda_{\min,\widetilde{\phi}}^{-1} \leq 5^{-1}a_{\rho}^{1/2}\min(a_{\rho}^{1/2}, B\sqrt{\epsilon/2})$, and

$$2^{-1}\lambda|\widehat{\boldsymbol{\gamma}}_{RL}^{\top}\widehat{\boldsymbol{\gamma}}_{RL} - \boldsymbol{\gamma}_{RL}^{0\top}\boldsymbol{\gamma}_{RL}^{0}| \leq \frac{1}{10}\Delta + \frac{1}{5}\sqrt{\epsilon/2}\sqrt{\Delta}.$$

This result together with (B.7) imply that

$$\frac{\Delta - (\mathcal{E}_n(\widehat{f}_{RL}^{P}) - \mathcal{E}_n(f_{RL}^{0}))}{\sqrt{\Delta+\epsilon}} \geq \frac{0.9\Delta}{\sqrt{\Delta+\epsilon}} - \frac{0.2\sqrt{\epsilon/2}\sqrt{\Delta}}{\sqrt{\Delta+\epsilon}} \geq \frac{0.9\Delta}{\sqrt{\Delta+\epsilon}} - 0.2\sqrt{\epsilon/2}.$$

The above result and (B.6) lead to

$$P\left\{\frac{0.9\Delta}{\sqrt{\Delta+\epsilon}} > \sqrt{\epsilon/2} + 0.2\sqrt{\epsilon/2}\right\} \leq Q.$$

Moreover,

$$\frac{0.9\Delta}{\sqrt{\Delta+\epsilon}} > 1.2\sqrt{\epsilon/2} \Longleftrightarrow \frac{\Delta}{\sqrt{\Delta+\epsilon}} > \frac{4}{3}\sqrt{\epsilon/2} \Longleftrightarrow \Delta > \frac{4}{3}\sqrt{\epsilon/2}\sqrt{\Delta+\epsilon}$$
$$\Longleftrightarrow \Delta^2 > \frac{8}{9}\epsilon(\Delta+\epsilon) \Longleftrightarrow (\Delta - \frac{4}{9}\epsilon)^2 > (136/81)\epsilon^2 \Longleftrightarrow \Delta > \frac{4+2\sqrt{34}}{9}\epsilon,$$

where the last step follows from $\Delta > 0$. Therefore, we have $P\left\{\Delta > \frac{4+2\sqrt{34}}{9}\epsilon\right\} \leq Q$. Since $\frac{4+2\sqrt{34}}{9} < 2$, then $P\{\Delta > 2\epsilon\} \leq Q$. ■

Proof of Theorem 4.2. The dimension of the space $\mathcal{F}(\widetilde{\phi}, m, B)$ is $|V_m^{(1)}|$. By Theorem 5.3 of [18], we have

$$\mathcal{N}(\sqrt{2}C_\rho^{-1}\epsilon/8, \mathcal{F}(\widetilde{\phi}, m, B), ||\cdot||_\infty) \leq (\frac{16C_\rho B}{\sqrt{2}\epsilon} + 1)^{|V_m^{(1)}|} \leq (\frac{12C_\rho B}{\epsilon})^{|V_m^{(1)}|},$$

when $\epsilon < C_\rho B/2$. Let

$$\varsigma = (\frac{12C_\rho B}{\epsilon})^{|V_m^{(1)}|} \exp\left(-n\epsilon/C^*\right). \tag{B.9}$$

By the above results and Lemma 4.1, we have

$$P\left(\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_{RN}^0) > c'\epsilon\right) \leq \varsigma, \tag{B.10}$$

where $c' = 1$ when $\widehat{f}_{RN} = \widehat{f}_{RN}^U$, and $c' = 2$ when $\widehat{f}_{RN} = \widehat{f}_{RN}^P$. Moreover, (B.9) leads to $\exp\left(n\epsilon/C^*\right)\epsilon^{|V_m^{(1)}|} = (12C_\rho B\varsigma^{-1/|V_m^{(1)}|})^{|V_m^{(1)}|}$ which is equivalent to

$$\exp(\varkappa\epsilon)\epsilon = \nu \Longleftrightarrow \exp(\varkappa\epsilon)(\varkappa\epsilon) = \varkappa\nu,$$

where $\varkappa = n/(C^*|V_m^{(1)}|)$ and $\nu = 12C_\rho B\varsigma^{-1/|V_m^{(1)}|}$. Applying the monotone increasing Lambert W-function: $W : [0,\infty) \to [0,\infty)$ defined by $W(t\exp(t)) = t$ on both sides of the

above equation, we have

$$W(\varkappa\nu) = \varkappa\epsilon,$$

which is equivalent to $\epsilon = W(\varkappa\nu)/\varkappa \leq \max(1, \log(\varkappa\nu))/\varkappa$, since $W(s) \leq \log(s)$ for all $s \geq e$. Then,

$$\begin{aligned}\epsilon &\leq \max(1, \log(\varkappa\nu))/\varkappa = \frac{C^*|V_m^{(1)}|}{n} \max(1, \log \frac{12C_\rho Bn}{C^*|V_m^{(1)}|\varsigma^{1/|V_m^{(1)}|}}) \\ &\leq \frac{C^*|V_m^{(1)}|}{n} \max(1, \log \frac{12C_\rho Bn}{C^*|V_m^{(1)}|\varsigma}).\end{aligned}$$

Therefore, we have

$$P\left(\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_{RN}^0) > c' \frac{C^*|V_m^{(1)}|}{n} \max(1, \log \frac{C^{**}n}{|V_m^{(1)}|\varsigma})\right) \leq \varsigma,$$

for $C^{**} = 12C_\rho BC^{*-1}$ based on (B.9). ■

B.6 Proofs of Theorem 4.3

Proof of Theorem 4.3. Let $2^m \asymp n^{1/5}$ and $m \lesssim R$. When i) $d \asymp (\log_2 n)^\kappa$ for some constant $\kappa \in (0, 1)$, then for any constant $c \in (0, \infty)$, $\left(c\frac{m+d}{d-2}\right)^{2d} \ll n^\varpi$ for an arbitrary small $\varpi > 0$. Therefore, the bias term satisfies

$$\begin{aligned}\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) &\leq \zeta_{R,m,d} \lesssim n^{-4/5} d \left(\frac{e}{3}\frac{m+d}{d-2}\right)^{-2} \left(\frac{e}{3}\frac{m+d}{d-2}\right)^{2d} \\ &\ll n^{-4/5}(\log_2 n)^\kappa (\log_2 n)^{2(\kappa-1)} n^\varpi = n^{-4/5+\varpi}(\log_2 n)^{3\kappa-2},\end{aligned}$$

where $\zeta_{R,m,d}$ is given in (4.17). Then the bias term satisfies

$$\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) = o(n^{-4/5+\varpi}(\log_2 n)^{3\kappa-2}). \tag{B.11}$$

Moreover, Proposition 4.1 leads to

$$\begin{aligned} |V_m^{(1)}| &\lesssim n^{1/5} d^{1/2} (m+d)^{-1} \left(4e\frac{m+d}{d-1}\right)^{-1} \left(4e\frac{m+d}{d-1}\right)^{d} \\ &\ll n^{1/5}(\log_2 n)^{\kappa/2-1}(\log_2 n)^{\kappa-1} n^{\varpi/2} = n^{1/5+\varpi/2}(\log_2 n)^{3\kappa/2-2}, \end{aligned}$$

and $n^{1/5} \lesssim |V_m^{(1)}|$. Let $\epsilon = n^{-4/5+\varpi/2}(\log_2 n)^{3\kappa/2}$. Then ς given in (B.9) satisfies

$$\begin{aligned} \varsigma &\ll \{n^{4/5-\varpi/2}(\log_2 n)^{-3\kappa/2}\}^{n^{1/5+\varpi/2}(\log_2 n)^{3\kappa/2-2}} \exp\{-n^{1/5+\varpi/2}(\log_2 n)^{3\kappa/2}\} \\ &\leq \exp\{n^{1/5+\varpi/2}(\log_2 n)^{3\kappa/2-2}\log_2 n\} \exp\{-n^{1/5+\varpi/2}(\log_2 n)^{3\kappa/2}\} \\ &= \exp\{n^{1/5+\varpi/2}(\log_2 n)^{3\kappa/2-1}(1-\log_2 n)\} \leq \exp\{-\frac{1}{2}n^{1/5+\varpi/2}(\log_2 n)^{3\kappa/2}\}, \end{aligned}$$

when $n > 4$. Thus, $\varsigma \to 0$ as $n \to \infty$. Therefore, the above results and (B.10) lead to

$$\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_{RN}^0) = \mathcal{O}_p(n^{-4/5+\varpi/2}(\log_2 n)^{3\kappa/2}). \tag{B.12}$$

The rate given in (B.12) is satisfied by both $\widehat{f}_{RN}^U$ and $\widehat{f}_{RN}^P$, For $\widehat{f}_{RN}^P$, the tuning parameter needs to satisfy $\lambda\lambda_{\min,\tilde{\phi}}^{-1} = \mathcal{O}(\sqrt{\epsilon}) = \mathcal{O}(n^{-2/5+\varpi/4}(\log_2 n)^{3\kappa/4})$.

From the results in (B.11) and (B.12), we have

$$\begin{aligned} &\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_0) = \mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_{RN}^0) + \mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) \\ &= \mathcal{O}_p(n^{-4/5+\varpi/2}(\log_2 n)^{3\kappa/2}) + o(n^{-4/5+\varpi}(\log_2 n)^{3\kappa-2}) = o_p(n^{-4/5+\varpi}(\log_2 n)^{3\kappa-2}). \end{aligned}$$

If $R \asymp \log_2 n$, the ReLU network that is used to construct the estimator $\widehat{f}_{RN}$ has depth $\mathcal{O}(R\log_2 d) = \mathcal{O}[\log_2 n\{\log_2(\log_2 n)\}]$, the number of computational units $\mathcal{O}(Rd) \times |V_m^{(1)}| = \mathcal{O}\{(\log_2 n)^{1+\kappa} n^{1/5+\varpi/2}(\log_2 n)^{3\kappa/2-2}\} = \mathcal{O}\{(\log_2 n)^{5\kappa/2-1} n^{1/5+\varpi/2}\}$, and the number of weights $\mathcal{O}(Rd) \times |V_m^{(1)}| = \mathcal{O}\{(\log_2 n)^{5\kappa/2-1} n^{1/5+\varpi/2}\}$.

When ii) $d \asymp 1$, the bias term satisfies

$$\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) \leq \zeta_{R,m,d} \lesssim n^{-4/5} d \left(\frac{e}{3}\frac{m+d}{d-2}\right)^{2d-2} \lesssim n^{-4/5}(d^{-1}\log_2 n)^{2d-2},$$

for $d \geq 3$, and

$$\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) \leq \zeta_{R,m,d} \lesssim n^{-4/5}(\log_2 n)^2 = 2n^{-4/5}(d^{-1}\log_2 n)^{2d-2},$$

for $d = 2$, where $\zeta_{R,m,d}$ is given in (4.17). Then,

$$\mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) = \mathcal{O}(n^{-4/5}(d^{-1}\log_2 n)^{2d-2}). \tag{B.13}$$

Moreover, Proposition 4.1 leads to

$$|V_m^{(1)}| \lesssim n^{1/5} d^{1/2} (m+d)^{-1} \left(4e\frac{m+d}{d-1}\right)^{d-1} = \mathcal{O}(n^{1/5}(\log_2 n)^{d-2}),$$

and $n^{1/5} \lesssim |V_m^{(1)}|$. Let $\epsilon = n^{-4/5}(d^{-1}\log_2 n)^d$. Then ς given in (B.9) satisfies

$$\begin{aligned}
\varsigma &\lesssim \{n^{4/5}(d^{-1}\log_2 n)^{-d}\}^{n^{1/5}(\log_2 n)^{d-2}} \exp\{-n^{1/5}(d^{-1}\log_2 n)^d\} \\
&\lesssim \exp\{n^{1/5}(\log_2 n)^{d-2}(\log_2 n)\} \exp\{-n^{1/5}(\log_2 n)^d\} \\
&= \exp\{n^{1/5}(\log_2 n)^{d-1}(1-\log_2 n)\} \lesssim \exp\{-\frac{1}{2}n^{1/5}(\log_2 n)^d\}.
\end{aligned}$$

Thus, $\varsigma \to 0$ as $n \to \infty$. Therefore, the above results and (B.10) lead to

$$\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_{RN}^0) = \mathcal{O}_p(n^{-4/5}(d^{-1}\log_2 n)^d). \tag{B.14}$$

The rate given in (B.14) is satisfied by both $\widehat{f}_{RN}^U$ and $\widehat{f}_{RN}^P$, For $\widehat{f}_{RN}^P$, the tuning parameter needs to satisfy $\lambda\lambda_{\min,\widetilde{\phi}}^{-1} = \mathcal{O}(\sqrt{\epsilon}) = \mathcal{O}(n^{-2/5}(d^{-1}\log_2 n)^{d/2})$.

From the results in (B.13) and (B.14), we have

$$\mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_0) = \mathcal{E}(\widehat{f}_{RN}) - \mathcal{E}(f_{RN}^0) + \mathcal{E}(f_{RN}^0) - \mathcal{E}(f_0) = \mathcal{O}_p(n^{-4/5}(d^{-1}\log_2 n)^{2d-2}).$$

If $R \asymp \log_2 n$, the ReLU network that is used to construct the estimator $\widehat{f}_{RN}$ has depth $\mathcal{O}(R\log_2 d) = \mathcal{O}(\log_2 n)$, the number of computational units

$$\mathcal{O}(Rd) \times |V_m^{(1)}| = \mathcal{O}\{(\log_2 n)\, n^{1/5}(\log_2 n)^{d-2}\} = \mathcal{O}\{(\log_2 n)^{d-1}\, n^{1/5}\},$$

and the number of weights $\mathcal{O}(Rd) \times |V_m^{(1)}| = \mathcal{O}\{(\log_2 n)^{d-1}\, n^{1/5}\}$. ■

B.7 Proofs of Lemmas 4.1-4.3

A lemma is presented below and it is used to prove the lemmas given in Section 4.5.

Lemma B.3 *For any $f \in \mathcal{F}(\widetilde{\phi}, m, B)$, one has*

$$\lim_{\delta \to 0^+} \frac{\mathcal{E}(f_{RL}^0 + \delta(f - f_{RL}^0)) - \mathcal{E}(f_{RL}^0)}{\delta} \geq 0.$$

Proof. Let $\delta \in (0, 1)$. Based on the definition of $\mathcal{F}(\widetilde{\phi}, m, B)$ given in (4.12), we can see that $f_0 + \delta(f - f_0) \in \mathcal{F}(\widetilde{\phi}, m, B)$. Moreover $\mathcal{E}(f_{RL}^0 + \delta(f - f_{RL}^0)) - \mathcal{E}(f_{RL}^0) \geq 0$ by the definition of f_{RL}^0 given in (4.14). ■

Proof of Lemma 4.1. Denote $t_0 = f_{RL}^0(\boldsymbol{x})$ and $t = f(\boldsymbol{x})$. By Taylor's expansion and Assumption 4.6, we have

$$\rho(t, y) - \rho(t_0, y) = \rho'(t_0, y)(t - t_0) + \int_0^1 2^{-1} \rho''(t_0 + (t - t_0)\omega, y)(t - t_0)^2 d\omega.$$

Moreover, by the dominated convergence theorem and Lemma B.3,

$$\begin{aligned}
&\int_{\mathcal{X}\times\mathcal{Y}} \rho'\left(f_{RL}^0(\boldsymbol{x}), y\right)(f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x}))d\mu(\boldsymbol{x}, y) \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \lim_{\delta \to 0^+} \frac{\rho\left(f_{RL}^0 + \delta(f - f_{RL}^0), y\right) - \rho\left(f_{RL}^0, y\right)}{\delta} d\mu(\boldsymbol{x}, y) \\
&= \lim_{\delta \to 0^+} \int_{\mathcal{X}\times\mathcal{Y}} \frac{\rho\left(f_{RL}^0 + \delta(f - f_{RL}^0), y\right) - \rho\left(f_{RL}^0, y\right)}{\delta} d\mu(\boldsymbol{x}, y) \\
&= \lim_{\delta \to 0^+} \frac{\mathcal{E}(f_{RL}^0 + \delta(f - f_{RL}^0)) - \mathcal{E}(f_{RL}^0)}{\delta} \geq 0.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathcal{E}(f) - \mathcal{E}(f_{RL}^0) &= \int_{\mathcal{X}\times\mathcal{Y}} \{\rho(f(\boldsymbol{x}), y) - \rho\left(f_{RL}^0(\boldsymbol{x}), y\right)\} d\mu(\boldsymbol{x}, y) \\
&\geq \int_{\mathcal{X}\times\mathcal{Y}} a_\rho (f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x}))^2 d\mu(\boldsymbol{x}, y) = a_\rho ||f - f_{RL}^0||_2^2.
\end{aligned}$$

Since $\int_{\mathcal{Y}} \rho'(f_0(\boldsymbol{x}), y) d\mu(y|\boldsymbol{x}) = 0$, then $\int_{\mathcal{X}\times\mathcal{Y}} \rho'\left(f_0(\boldsymbol{x}), y\right)\left(f(\boldsymbol{x}) - f_0(\boldsymbol{x})\right) d\mu(\boldsymbol{x}, y) = 0$. Thus,

$$\begin{aligned}\mathcal{E}(f) - \mathcal{E}(f_0) &= \int_{\mathcal{X}\times\mathcal{Y}} \left(\rho\left(f(\boldsymbol{x}), y\right) - \rho\left(f_0(\boldsymbol{x}), y\right)\right) d\mu(\boldsymbol{x}, y) \\ &\leq \int_{\mathcal{X}\times\mathcal{Y}} b_\rho (f(\boldsymbol{x}) - f_0(\boldsymbol{x}))^2 d\mu(\boldsymbol{x}, y) = b_\rho ||f - f_0||_2^2.\end{aligned}$$

■

Proof of Lemma 4.2. In the following, we will show the results in Lemma 4.2 when the loss function $\rho\left(f(\boldsymbol{x}), y\right)$ is the quantile loss given in (4.3). We follow a proof procedure from [1]. We have

$$\begin{aligned}\mathcal{E}(f) - \mathcal{E}(f_{RL}^0) &= \int_{\mathcal{X}\times\mathcal{Y}} \left(\rho\left(f(\boldsymbol{x}), y\right) - \rho\left(f_{RL}^0(\boldsymbol{x}), y\right)\right) d\mu(\boldsymbol{x}, y) \\ &= \int_{\mathcal{X}} \int_{\mathcal{Y}} \left(\rho\left(f(\boldsymbol{x}), y\right) - \rho\left(f_{RL}^0(\boldsymbol{x}), y\right)\right) d\mu(y|\boldsymbol{x}) d\mu_X(\boldsymbol{x})\end{aligned}$$

Then for all $\boldsymbol{x} \in \mathcal{X}$,

$$\begin{aligned}&\int_{\mathcal{Y}} \rho\left(f(\boldsymbol{x}), y\right) d\mu(y|\boldsymbol{x}) \\ =\ &\int_{\mathcal{Y}} I\{y > f(\boldsymbol{x})\}(y - f(\boldsymbol{x})) d\mu(y|\boldsymbol{x}) + (\tau - 1)\int_{\mathcal{Y}} (y - f(\boldsymbol{x})) d\mu(y|\boldsymbol{x}) \\ =\ &g(\boldsymbol{x}, f(\boldsymbol{x})) + (\tau - 1)\int_{\mathcal{Y}} y d\mu(y|\boldsymbol{x}),\end{aligned}$$

where $g(\boldsymbol{x}, u) = \int_{\mathcal{Y}} I\{y > u\}(1 - \mu(y|\boldsymbol{x}))dy + (1-\tau)u$, and $\mathcal{E}(f) - \mathcal{E}(f_{RL}^0) = \int_{\mathcal{X}} g(\boldsymbol{x}, f(\boldsymbol{x})) d\mu_X(\boldsymbol{x}) - \int_{\mathcal{X}} g(\boldsymbol{x}, f_{RL}^0(\boldsymbol{x})) d\mu_X(\boldsymbol{x})$. Denote $t_0 = f_{RL}^0(\boldsymbol{x})$ and $t = f(\boldsymbol{x})$. By Taylor's expansion, we have

$$g(\boldsymbol{x}, t) - g(\boldsymbol{x}, t_0) = g'(\boldsymbol{x}, t_0)(t - t_0) + \int_0^1 2^{-1} g''\left(\boldsymbol{x}, t_0 + (t - t_0)\omega\right)(t - t_0)^2 d\omega.$$

Since $\frac{g\left(\boldsymbol{x}, f_{RL}^0 + \delta(f - f_{RL}^0)\right) - g(\boldsymbol{x}, f_{RL}^0(\boldsymbol{x}))}{\delta} \leq (2 - \tau)|f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x})|$, by the dominated

convergence theorem and Lemma B.3,

$$\begin{aligned}
&\int_{\mathcal{X}} g'(\boldsymbol{x}, f_{RL}^0(\boldsymbol{x}))(f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x}))d\mu_X(\boldsymbol{x}) \\
&= \int_{\mathcal{X}} \lim_{\delta \to 0^+} \frac{g\left(\boldsymbol{x}, f_{RL}^0 + \delta(f - f_{RL}^0)\right) - g(\boldsymbol{x}, f_{RL}^0(\boldsymbol{x}))}{\delta} d\mu_X(\boldsymbol{x}) \\
&= \lim_{\delta \to 0^+} \int_{\mathcal{X}\times\mathcal{Y}} \frac{g\left(\boldsymbol{x}, f_{RL}^0 + \delta(f - f_{RL}^0)\right) - g(\boldsymbol{x}, f_{RL}^0(\boldsymbol{x}))}{\delta} d\mu_X(\boldsymbol{x}) \\
&= \lim_{\delta \to 0^+} \frac{\mathcal{E}(f_{RL}^0 + \delta(f - f_{RL}^0)) - \mathcal{E}(f_{RL}^0)}{\delta} \geq 0.
\end{aligned}$$

The above results together with $\partial^2 g(\boldsymbol{x}, u)/\partial u^2 = \mu'(u|\boldsymbol{x})$ imply that

$$\begin{aligned}
\mathcal{E}(f) - \mathcal{E}(f_{RL}^0) &= \int_{\mathcal{X}} \{g(\boldsymbol{x}, f(\boldsymbol{x})) - g(\boldsymbol{x}, f_{RL}^0(\boldsymbol{x}))\} d\mu_X(\boldsymbol{x}) \\
&\geq 2^{-1} \int_{\mathcal{X}} (f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x}))^2 \int_0^1 g''\left(\boldsymbol{x}, f_{RL}^0(\boldsymbol{x}) + (f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x}))\omega\right) d\omega d\mu_X(\boldsymbol{x}) \\
&= 2^{-1} \int_{\mathcal{X}} (f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x}))^2 \int_0^1 \mu'(f_{RL}^0(\boldsymbol{x}) + (f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x}))\omega|\boldsymbol{x}) d\omega d\mu_X(\boldsymbol{x}) \\
&\geq \frac{1}{2C_1} \int_{\mathcal{X}} (f(\boldsymbol{x}) - f_{RL}^0(\boldsymbol{x}))^2 d\mu_X(\boldsymbol{x}) = \frac{1}{2C_1} ||f - f_{RL}^0||_2^2
\end{aligned}$$

Note that $\partial g(\boldsymbol{x}, u)/\partial u\,|_{u=f_0} = 0$ and $\partial^2 g(\boldsymbol{x}, u)/\partial u^2 = \mu'(u|\boldsymbol{x})$. Thus by Taylor's expansion,

$$\begin{aligned}
\mathcal{E}(f) - \mathcal{E}(f_0) &= \int_{\mathcal{X}} \{g(\boldsymbol{x}, f(\boldsymbol{x})) - g(\boldsymbol{x}, f_0(\boldsymbol{x}))\} d\mu_X(\boldsymbol{x}) \\
&= 2^{-1} \int_{\mathcal{X}} (f(\boldsymbol{x}) - f_0(\boldsymbol{x}))^2 \int_0^1 g''\left(\boldsymbol{x}, f_0(\boldsymbol{x}) + (f(\boldsymbol{x}) - f_0(\boldsymbol{x}))\omega\right) d\omega d\mu_X(\boldsymbol{x}) \\
&= 2^{-1} \int_{\mathcal{X}} (f(\boldsymbol{x}) - f_0(\boldsymbol{x}))^2 \int_0^1 \mu'(f_0(\boldsymbol{x}) + (f(\boldsymbol{x}) - f_0(\boldsymbol{x}))\omega|\boldsymbol{x}) d\omega d\mu_X(\boldsymbol{x}) \\
&\leq \frac{1}{2C_2} \int_{\mathcal{X}} (f(\boldsymbol{x}) - f_0(\boldsymbol{x}))^2 d\mu_X(\boldsymbol{x}) = \frac{1}{2C_2} ||f - f_0||_2^2.
\end{aligned}$$

■

Proof of Lemma 4.3. The proof of Lemma 4.3 follows the same procedure as the proof of Lemma 4.2, and thus it is omitted. ■