STATISTICAL ANALYSIS FOR NETWORK-BASED MODELS WITH APPLICATIONS TO GENETIC ASSOCIATION AND PREDICTION

By

Xiaoxi Shen

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Statistics — Doctor of Philosophy

2019

ABSTRACT

STATISTICAL ANALYSIS FOR NETWORK-BASED MODELS WITH APPLICATIONS TO GENETIC ASSOCIATION AND PREDICTION

By

Xiaoxi Shen

Network-based models are popular in statistical applications. The main advantage of using a network-based model is that one can understand and evaluate its semantics and properties rather straightforwardly. In this dissertation, we study the statistical properties for some network-based model and apply these models to genetic association studies and genetic risk predictions.

In Chapter 2, we propose a conditional autoregressive (CAR) model to account for possible heterogeneous genetic effects among individuals. In the proposed method, the genetic effect is considered as a random effect and a score test is developed to test the variance component of genetic random effect. Through simulations, we compare the type I error and power performance of the proposed method with those of the generalized genetic random field (GGRF) and the sequence kernel association test (SKAT) methods under different disease scenarios. We find that our method outperforms the other two methods when (i) the rare variants have the major contribution to the disease, or (ii) the genetic effects vary in different individuals or subgroups of individuals. Finally, we illustrate the new method by applying it to the whole genome sequencing data from the Alzheimer's Disease Neuroimaging Initiative.

A kernel-based neural network (KNN) method is proposed in Chapter 3 for genetic risk prediction. KNN inherits the high-dimensional feature from classical kernel methods and the non-linear and non-additive features from neural networks. KNN summarizes a large

number of genetic variants into kernel matrices and uses the kernel matrices as input matrices. Based on these kernel matrices, KNN builds a feedforward neural network to model the complex relationship between genetic variants and a disease outcome. Minimum norm quadratic unbiased estimation (MINQUE) is implemented in KNN to make parameter estimation feasible. Through theoretical proof and simulations, we demonstrate that KNN can attain lower average prediction error than LMM. Finally, we illustrate KNN by an application to the sequencing data from the Alzheimer?s Disease Neuroimaging Initiate.

Nowadays, neural networks have been widely applied in machine learning and artificial intelligence. However, as a statistical model, few researches focuses on statistical properties for neural networks and these will be studied in Chapter 4. A neural network can be classified into a nonlinear regression regression. However, if we consider it parametrically, due to the unidentifiability of the parameters, it is difficult to derive its asymptotic properties. Instead, we consider the estimation problem as a nonparametric regression problem and use the results from sieve estimation to establish the consistency, rates of convergence and asymptotic normality of the neural network estimators. We also illustrate the validity of the theories via simulations.

I dedicate this dissertation to my parents, Xinmei Dai, Xiong Shen and my friends, Chang, Carlos, Fengkan, Julia, Shunjie and many others.

ACKNOWLEDGMENTS

Here, I would like to express my deepest gratitude to my advisors Dr. Qing Lu and Dr. Yuehua Cui for their support and guidance towards my studies and researches. Dr. Lu and Dr. Cui are extremely kind and knowledgeable. They always provide valuable insights and suggestions on the improvements of my work.

I would also like to extend my sincere thanks to my dissertation committee members, Dr. Lyudmila Sakhanenko and Dr. Yimin Xiao. Their comments and suggestions are beneficial for my studies.

I am also grateful to the help I obtained from all the professors in the Department of Statistics and Probability. During my Ph.D. studies, I took most of the courses offered by our department. From each course, I learned various techniques to solve different research problems. For example, I learned how to consider a problem in a Bayesian way and how to interpret the Bayesian results from the course I took from Dr. Tapabrata Maiti.

During my seven years at Michigan State University, I made lots of friends and because of them, I never feel lonely in these years. Many thanks to my senpais: Dr. Honglang Wang, Dr. Tao He, Dr. Bin Gao, Dr. Shunjie Guan and Dr. Jingyi Zhang. They have become successful faculty members and statisticians in big companies. My thanks also go to my friends: Chang Jiang, Shan Zhang, Xiaoran Tong, Kaixu Yang, Yuning Hao, Jinghang Lin, Yuan Zhou and Di Wu. Without your help, I could not be who I am now and I sincerely wish all of you have a wonderful future.

Last but not least, I would like to express my sincere thanks to my parents for their support and confidence in me. I would like to treat this dissertation as my unique gift to both of you.

TABLE OF CONTENTS

LIST OF TABLES . . . . . **viii**

LIST OF FIGURES . . . . . **x**

Chapter 1 Introduction **1**
1.1 Overview 1
1.2 Linear Mixed Models for Genetic Association Studies 3
1.3 Statistical Learning 5
1.3.1 Gaussian Graphical Models 5
1.3.2 Kernel Methods 6
1.3.3 Neural Networks 10
1.4 Objective and Organization 13

Chapter 2 A Conditional Autoregressive Model for Genetic Association Analysis Accounting for Genetic Heterogeneity **15**
2.1 Introduction 15
2.2 Methods 17
2.2.1 Motivation 17
2.2.2 Model Setup 22
2.2.3 Score Test for Variance Component 25
2.2.3.1 γ As a Fixed Constant 26
2.2.3.2 γ As an Unknown Nuisance Parameter 28
2.3 Simulation Results 30
2.3.1 Simulation I: Heterogeneous Genetic Effects Among Individuals or Subgroups 33
2.3.2 Simulation II: Various Causal SNV Rates 37
2.3.3 Simulation III: Misspecification of Weights 38
2.4 Real Data Applications 44
2.5 Discussion 48

Chapter 3 A Kernel-Based Neural Network for High-dimensional Genetic Risk Prediction Analysis **50**
3.1 Introduction 50
3.2 Methodologies 52
3.2.1 Quadratic Estimators for Variance Components 58
3.2.2 MINQUE in KNN 61
3.3 Predictions 66
3.4 Including Fixed Effects 72
3.5 Simulations 76
3.5.1 Nonlinear Random Effect 76
3.5.2 Nonadditive Effects 78

3.5.3 Non-normal Error Distributions . . . 80
3.6 Real Data Application . . . 82
3.7 Discussion . . . 85

Chapter 4 Asymptotic Properties of Neural Network Sieve Estimators . 87
4.1 Introduction . . . 87
4.2 Existence . . . 92
4.3 Consistency . . . 96
4.4 Rate of Convergence . . . 108
4.5 Asymptotic Normality . . . 115
4.6 Simulation Studies . . . 131
4.6.1 Parameter Inconsistency . . . 132
4.6.2 Consistency for Neural Network Sieve Estimators . . . 134
4.6.3 Asymptotic Normality for Neural Network Sieve Estimators . . . 136
4.7 Discussion . . . 142

Chapter 5 Epilogue . . . 144

APPENDICES . . . 147
Appendix A Technical Details and Supplementary Materials for Chapter 2 . . . 148
Appendix B Technical Details and Supplementary Materials for Chapter 3 . . . 155
Appendix C Technical Details and Supplementary Materials for Chapter 4 . . . 167

BIBLIOGRAPHY . . . 171

LIST OF TABLES

Table 2.1: Empirical type I error rates under different weights at level $\alpha = 0.05$ and $\alpha = 0.01$ based on 1000 replicates. Each cell in the table contains the empirical type I error rate. GGRF, SKAT, CAR.FIX and CAR.SUP are the generalized genetic random field model of Li et al. (2014) [41], the sequence kernel association test of Wu et al. (2011) [78], the conditional autoregressive model with fixed nuisance parameter γ, the conditional autoregressive model with maximum score test statistic, respectively 33

Table 2.2: Empirical power comparison of GGRF [41], SKAT [78] and CAR based on 1,000 Monte Carlo replicates. In the simulation, we simulated 8 different subgroups with $\sigma_Z = 0.0001, 0.0005, 0.002, 0.001, 0.003, 0.0001, 0.0005, 0.002$ for BETA and WSS; with $\sigma_Z = 0.01, 0.05, 0.2, 0.1, 0.3, 0.01, 0.05, 0.2$ for UW and LOG. The number in the parenthesis is the standard deviation. . . . 36

Table 2.3: Empirical power comparison of GGRF [41], SKAT [78] and CAR based on 1,000 Monte Carlo replicates. In the simulation, we simulated 8 different subgroups with $\sigma_Z = 0.0001, 0.0001, 0.0002, 0.005, 0.0003, 0.0005, 0.0001, 0.005$ for BETA and WSS; with $\sigma_Z = 0.01, 0.01, 0.02, 0.5, 0.03, 0.05, 0.01, 0.5$ for UW and LOG. The number in the parenthesis is the standard deviation. . 37

Table 2.4: Top 10 hippocampus-associated genes detected by GGRF, SKAT and CAR. 47

Table 3.1: Average mean squared prediction error of KNN with product output kernel matrix, KNN with output kernel matrix as polynomial of order 2 and the BLUP based on LMM. 84

Table 4.1: Comparison of the true parameters and the estimated parameters in a single-layer neural network with 2 hidden units. 132

Table 4.2: Comparison of errors $\|\hat{f}_n - f_0\|_n^2$ and the least square errors $\mathbb{Q}_n(\hat{f}_n)$ after 20,000 iterations under different sample sizes. 136

Table 4.3: p-values for Shapiro-Wilks test and Kolmogorov-Smirnov test for nomality test. We use "NN" to denote the true function as a neural network described in (4.13); "TRI" to denote the true function as a trigonometric function described in (4.14) and "ND" to denote the true function as a continuous function having a non-differential point described in (4.15) 138

Table A.1: Top 10 Genes Detected by GGRF, SKAT and CAR Associated with Entorhinal. 152

Table A.2: Top 10 Genes Detected by GGRF, SKAT and CAR Associated with Ventricle. 153

Table A.3: Top 10 Genes Detected by GGRF, SKAT and CAR Associated with Whole Brain. 154

LIST OF FIGURES

Figure 1.1: Diagram of a neuron with main components (Figure from Wiki). 10

Figure 1.2: A perceptron that mimics the structure of a neuron in a human brain. $x_1, \ldots, x_p$ are input features and x_0 is a bias unit or intercept. Σ is known as a computation unit, where a linear combination of the features, including the bias unit and the parameters $\boldsymbol{\theta}$ to be learned is calculated and then an activation function is applied. In this example, a standard sigmoid activation function $\sigma(x) = (1 + e^{-x})^{-1}$ is applied. 11

Figure 1.3: A neural network with one hidden layer. $x_1, \ldots, x_p$ are input features and x_0 is a bias unit or intercept. Units in the dashed rectangle are known as hidden computation units, where a linear combination of features from previous layer and associated parameters is calculated and an activation function is applied to obtain outputs for these hidden units. Σ at the end is an output computation unit, in which a linear combination of the output features from the last hidden layers and associated weights is calcuated and an acitivation function is applied. 12

Figure 2.1: An illustration of a Gaussian graphical model. For the CAR model, the graph represents a network of the genetic effects among individuals. In the graph, each node stands for the gentic effect of an indiviudal and the existence of an edge between two individuals indicates dependence between genetic effects between these two individuals. This is characterized by the precision matrix in the joint normal distribution of these genetic effects. . 20

Figure 2.2: The Distribution of minor allele frequency of in sequencing variants on chromosome 17 from the 1,000 Genome project. 32

Figure 2.3: Empirical Power Comparison of CAR, SKAT and GGRF by varying the levels of genetic heterogeneity among individuals under four different weights. The x-axis is the effect size σ_Z, used in the simulation. The effect sizes are chosen as 0.01, 0.05, 0.075, 0.1, 0.3, 0.5 for the UW and LOG weights, and the effect sizes are set as 0.0005, 0.00075, 0.001, 0.0015, 0.002 for the BETA and WSS weights. 35

Figure 2.4: Comparison of empirical power of CAR, SKAT and GGRF with different causal SNV rates under the BETA weight and the WSS weight. The standard deviation σ_Z used in the simulation is gradually increased. $\sigma_Z = 0.0005, 0.001, 0.002, 0.005$, respectively in each column from left to right. 39

Figure 2.5: Comparison of empirical power of CAR, SKAT and GGRF with different causal SNV rates under the UW weight and the LOG weight. The standard deviation σ_Z used in the simulation is gradually increased. $\sigma_Z = 0.05, 0.1, 0.2, 0.5$ respectively in each column from left to right. . . . 40

Figure 2.6: Comparison of empirical power of CAR, SKAT and GGRF with misspecified weights when the true weight is BETA and WSS. In each column from left to right, we used BETA, LOG, UW and WSS weight, respectively. 42

Figure 2.7: Comparison of empirical power of CAR, SKAT and GGRF with misspecified weights when the true weight is UW and LOG. In each column from left to right, we used BETA, LOG, UW and WSS weight, respectively. . 43

Figure 2.8: Histograms of the Phenotypes used for Real Data Analyses 45

Figure 3.1: An illustration of the hierarchical structure of the kernel neural network model with L input kernel matrices $\boldsymbol{K}_1(\boldsymbol{G}), \ldots, \boldsymbol{K}_L(\boldsymbol{G})$ and J hidden kernel matrices $\boldsymbol{K}_1(\boldsymbol{U}), \ldots, \boldsymbol{K}_J(\boldsymbol{U})$. The output layer is the prediction for the random effect $\boldsymbol{f}$. 59

Figure 3.2: The intuition under the assumption $\tilde{\sigma}_R^2 \leq \tilde{\tau}\xi$ in Proposition 3.3.2. 70

Figure 3.3: The boxplots for linear mixed models (LMM) and kernel neural networks (KNN) in terms of prediction errors. The left panel shows the results when a linear function is used and the right panel shows the results when a sine function is used. In the horizontal axis, “1" corresponds to the LMM; “2" corresponds to the KNN with product input kernel and product output kernel; “3" corresponds to the KNN with product input and polynomial output; “4" corresponds to the KNN with polynomial input and product output and “5" corresponds to the polynomial input and polynomial output. 77

Figure 3.4: The boxplots for linear mixed models (LMM) and kernel neural networks (KNN) in terms of prediction errors based on the simulation model focusing on the interaction effect. The vertical axis is scaled to 0-5 by removing some outliers to make the comparison visually clear. In the horizontal axis, “1" corresponds to the LMM; “2" corresponds to the KNN with product input kernel and product output kernel; “3" corresponds to the KNN with product input and polynomial output; “4" corresponds to the KNN with polynomial input and product output and “5" corresponds to the polynomial input and polynomial output. 79

Figure 3.5: The boxplots for linear mixed models (LMM) and kernel neural networks (KNN) in terms of prediction errors based on the simulation model using dominant coding (left figure) and recessive coding (right figure) for SNPs. In the horizontal axis, “1" corresponds to the LMM; “2" corresponds to the KNN with product input kernel and product output kernel; “3" corresponds to the KNN with product input and polynomial output; “4" corresponds to the KNN with polynomial input and product output and “5" corresponds to the polynomial input and polynomial output. . . 81

Figure 3.6: The boxplots for linear mixed models (LMM) and kernel neural networks (KNN) in terms of prediction errors based on the simulation model using t-distribution for error (left figure) and centered χ^2_1-distribution for error (right figure). In the horizontal axis, “1" corresponds to the LMM; “2" corresponds to the KNN with product input kernel and product output kernel; “3" corresponds to the KNN with product input and polynomial output; “4" corresponds to the KNN with polynomial input and product output and “5" corresponds to the polynomial input and polynomial output. 82

Figure 4.1: Comparison of the true function and fitted function under the simulation model (4.12). The black curve is the true function defined in (4.13) and the blue dashed curve is the fitted curve obtained after fitting the neural network model. 133

Figure 4.2: Figures on comparison of the true function and the fitted function used in simulations. The top panel shows the scenario when the true function is a single layer neural network; the middle panel shows the scenario when the true function is a sine function and the bottom panel show the scenario when the true function is a continuous function having a non-differentiable point. As we can see from all the cases, the fitted curve becomes closer to the truth as the sample size increases. 137

Figure 4.3: Normal Q-Q plot for $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ under 200 iterations and various sample sizes. The true function f_0 is a single-layer neural network with 2 hidden units as defined in (4.13). 139

Figure 4.4: Normal Q-Q plot for $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ under 200 iterations and various sample sizes. The true function f_0 is a trigonometric function as defined in (4.14). 140

Figure 4.5: Normal Q-Q plot for $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ under 200 iterations and various sample sizes. The true function f_0 is a continuous function having a non-differential point as defined in (4.15). 141

Figure B.1: The boxplots for linear mixed models (LMM) and kernel neural network (KNN) in terms of prediction errors. The left panel shows the results when an inverse logistic function is used and the right panel shows the results when a polynomial function of order 2 is used. In the horizontal axis, “1" corresponds to the LMM; “2" corresponds to the KNN with product input kernel and product output kernel; “3" corresponds to the KNN with product input and polynomial output; “4" corresponds to the KNN with polynomial input and product output and “5" corresponds to the polynomial input and polynomial output. 166

Chapter 1

Introduction

1.1 Overview

During the past decades, genome-wide association studies (GWAS) become a powerful tool for investigating the associations between human genome and diseases. There are many successful examples. For example, Scott et al. (2007)[62] identified type 2 diabetes (T2D)-associated variants in an intergenic region of chromosome 11p12, near the genes *IGF2BP2* and *CDKAL1* and the region of *CDKN2A* and *CDKN2B*. They also confirmed that variants near *TCF7L2*, *SLC30A8*, *HHEX*, *FTO*, *PPARG* and *KCNJ11* are associated with T2D risk. Through case-control comparisons, the Wellcome Trust Case Control Consortium [14] identified 24 independent association signals at $p < 5 \times 10^{-7}$ for seven common diseases including bipolar disorder, type 1 and type 2 diabetes, etc. With the development of next-generation sequencing (NGS) technology, it is now possible to sequence the whole human genome with little cost. As pointed out in Goldstein et al. (2013)[24], GWAS primarily make use of markers that are intended to represent causal variation indirectly, whereas, NGS can directly identify the cause variants. Many novel genetic association analyses nowadays are based on sequencing data (see for example, Wu et al. (2011)[78] and Li et al. (2014)[41]) and have achieved various levels of success.

As precision medicine will be one of the focus on healthcare in the future, another im-

portant topic in statistical genetics is risk prediction. Accurate risk prediction can enable targeted preventative treatments, including fitness regimens for patients at risk of cardiovascular disease, or increased mammogram frequency for patients with high breast cancer risk [35]. Abraham and Inouye (2015) [1] mentioned that the main aim of risk prediction is maximization of predictive power, including the validity and robustness of model predictions, while a causal interpretation is not strictly necessary for a good predictive model. Therefore, a model that is capable of jointly estimating the predictive effects of all single-nucleotide variant (SNV) sets and achieves high and robust prediction accuracy is critically important for risk prediction research [72].

With the breakthrough in computational technologies, we are in the era of heading to artificial intelligence. Many learning methods based on network structure have been proposed for classification and regression. Among which, neural networks have been receiving more and more popularity due to its ability to capture nonlinear structures in the data. Due to the complex relationship between genetic variants and human diseases, statistical models that are capable for taking nonlinear relationship into account may perform better (e.g., reaching higher prediction accuracy) than current methods. As an example, Wang et al. (2016)[71] used a Deep Convolutional Neural Fields (DeepCNF) to predict protein secondary structure and achieves higher prediction accuracy than that of the best predictors' accuracy used before.

In this chapter, we will first review linear mixed models along with their popular applications in genetic association studies in section 1.2. Then in section 1.3, we will briefly introduce some methods in statistical learning, including Gaussian graphical models, kernel methods and neural networks.

1.2 Linear Mixed Models for Genetic Association Studies

Linear mixed models are powerful tools in statistics to model correlated continuous outcomes. The general formulation of a linear mixed model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{\epsilon}, \tag{1.1}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is a vector of continuous response; $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ is the design matrix for fixed effects and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the fixed effect coefficients; $\boldsymbol{Z} \in \mathbb{R}^{n\times q}$ is the design matrix for random effects and $\boldsymbol{b} \in \mathbb{R}^q$ is the random effect and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a vector of random error. Typical assumptions on the distribution of random vectors $\boldsymbol{b}$ and $\boldsymbol{\epsilon}$ are normality assumptions:

$$\boldsymbol{b} \sim \mathcal{N}_q(\boldsymbol{0}, \sigma_b^2 \boldsymbol{D}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{R}),$$

where $\boldsymbol{D} \in \mathbb{R}^{q\times q}$ and $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ are the covariance matrices for $\boldsymbol{b}$ and $\boldsymbol{\epsilon}$ respectively. In most applications, $\boldsymbol{R}$ is chosen to be the identity matrix $\boldsymbol{I}_n$.

In terms of applications in genetic association studies, the fixed effect design matrix $\boldsymbol{X}$ is usually composed of clinical covariates such as age, gender, ethnicity groups etc. The random effect design matrix $\boldsymbol{Z}$ consists of genetic variants such as single-neucleotide polymorphisms (SNPs). Sequence Kernel Association Test (SKAT)[78], which is a commonly used method in genetic association analysis, is based on such model setup. Under this situation, to test whether there is an association between genetic variants and the response, it suffices to test whether the variance component with respect to the genetic variants, which is σ_b^2 in model

(1.1), is zero or not. That is, we need to test

$$H_0 : \sigma_b^2 = 0 \text{ vs } H_1 : \sigma_b^2 \neq 0. \tag{1.2}$$

In SKAT, this is accomplished based on a score-type test. One of the significant advantage of modeling genetic variants for testing using random effects rather than fixed effects is to reduce the dimensionality of the parameters to be tested. For example, to test whether a specific gene is associated with the response, if we model the genetic variants as fixed effect, multiple testing corrections need to be conducted and since a gene may contain thousands of SNPs, it may require an extremely small p-value to conclude statistical significance, which will lead to power loss. On the other hand, by regarding genetic effect as a random effect, we only need to test a few variance components.

Note that we may write model (1.1) as follows:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{a} + \boldsymbol{\epsilon}, \tag{1.3}$$

where $\boldsymbol{a} \in \mathbb{R}^n$ is a random vector with $\boldsymbol{a} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma_b^2 \boldsymbol{Z}\boldsymbol{D}\boldsymbol{Z}^T)$. As pointed out in Yang et al. (2011)[79], we may consider $\boldsymbol{a}$ as the total genetic effects of the individuals and assume $\boldsymbol{a} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma_a^2 \boldsymbol{A})$ for some positive definite matrix $\boldsymbol{A}$ and $\boldsymbol{A}$ is interpreted as the genetic relationship matrix between individuals. Again, testing the association between genetic variants and response is equivalent to test $H_0 : \sigma_a^2 = 0$. We will also see later in this chapter that such a linear mixed model is equivalent to a semi-parametric model when the function in the non-parametric part belongs to a certain function space.

1.3 Statistical Learning

Statistical learning has received more and more attention nowadays. Due to a wide range of topics in statistical learning, it is impossible to list all of them in several pages. We refer interested readers to Friedman et al. (2001)[20] for more details. Here, we will focus on Gaussian graphical models, kernel methods and neural networks.

1.3.1 Gaussian Graphical Models

Graphical models are powerful tools in statistical learning. In short, graphical models use a graph-based representation as the basis for compactly encoding a complex distribution over a high-dimensional space. In short, a graphical model is a combination of multivariate statistics and graphical structure and it is indeed a probability distribution. The data is the nodal information in the graph with each node being treated as a random variable, while the edges represent a set of independencies that hold in the distribution. One of the benefits we can get from graphical models is that the number of parameters in the model could be reduced dramatically.

Directed graphs and undirected graphs are two types of commonly used graphs. A directed graphical model based on a directed acyclic graph (DAG) is known as a Bayesian network and directed edges give causal relationships as well as the characterization of the conditional distributions among the random variables. An undirected graphical model is known as a Markov random field and the undirected edges give correlations between variables.

A Gaussian graphical model is a particular case of a graphical model where the joint distribution of the random variables are Gaussian. Here we will focus on undirected Gaussian graphical models. Suppose $\boldsymbol{X} = [X_1, \ldots, X_n]^T \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ be a labeled

graph with $\mathcal{X} = \{1, \ldots, n\}$ and $\mathcal{E}$ be the edge set such that there is no edge between X_i and X_j if and only if $X_i \perp X_j | \boldsymbol{X}_{-ij}$. Such property is known as the pairwise Markov property [58]. Due to the Gaussianity, such conditional independencies can be characterized by the zero elements in the precision matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$.

Theorem 1.3.1 (Theorem 2.2 in [58]). *Let $\boldsymbol{X}$ be normally distributed with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{Q}$ being symmetric positive definite. Then for $i \neq j$,*

$$X_i \perp X_j | \boldsymbol{X}_{-ij} \Leftrightarrow Q_{ij} = 0.$$

Theorem 1.3.1 implies that the nonzero pattern of $\boldsymbol{Q}$ determines $\mathcal{G}$, so we can read off from $\boldsymbol{Q}$ whether X_i and X_j are conditionally independent given others. If $\boldsymbol{Q}$ is a completely dense matrix, then $\mathcal{G}$ is fully connected.

1.3.2 Kernel Methods

A kernel arises as a similarity measure that can be though of as an inner product in a so-called feature space [60]. More specifically, let $\mathcal{X}$ be the input domain and $\mathcal{F}$ be the feature space and it connects the input domain $\mathcal{X}$ via a map Φ, i.e.,

$$\begin{aligned} \Phi : \quad & \mathcal{X} \to \mathcal{F} \\ & x \mapsto \boldsymbol{x} := \Phi(x). \end{aligned}$$

Then a kernel K is defined to be the inner product in the feature space, that is,

$$K: \quad \mathcal{X} \times \mathcal{X} \to \mathbb{C}$$
$$(x, x') \mapsto K(x, x') := \left\langle \Phi(x), \Phi(x') \right\rangle$$

Therefore, a kernel function can be thought of as a generalized inner product. Since Φ can be a nonlinear function, a kernel function is able to characterized nonlinear features among the input variables. A particular kernel function that plays an important role in statistical learning is known as the reproducing kernel.

Definition 1.3.1 (Reproducing Kernel [6]). *Let E be a nonempty abstract set. A function*

$$K: \quad E \times E \to \mathbb{C}$$
$$(s, t) \mapsto K(s, t)$$

is a reproducing kernel of a Hilbert space $\mathcal{H}$ if and only if

(i) $\forall t \in E$, $K(\cdot, t) \in \mathcal{H}$;

(ii) $\forall t \in E$, $\forall \phi \in \mathcal{H}$, $\langle \phi, K(\cdot, t) \rangle = \phi(t)$

A Hilbert space $\mathcal{H}$ possessing a reproducing kernel K is called a Reproducing Kernel Hilbert Space (RKHS) and is denoted by $\mathcal{H}_K$.

The second property in the previous definition is known as the reproducing property: the value of the function ϕ at the point t is reproduced by the inner product of ϕ with $K(\cdot, t)$.

From the reproducing property, we can easily see that

$$\langle K(\cdot, t), K(\cdot, t')\rangle = K(t, t').$$

This identity forms the basis for the kernel trick, which is frequently used in statistical learning.

"Given an algorithm which is formulated in terms of a positive definite kernel K, one can construct an alternative algorithm by replacing K by another positive definite kernel $\tilde{K}$."[60]

It has been shown in Liu et al. (2007)[42] that there is a close connection between kernel methods and linear mixed model. In fact, such connection is due to the following Nonparametric Representer Theorem [59]:

Theorem 1.3.2 (Nonparametric Representer Theorem [59]). *Suppose we are given a nonempty set $\mathcal{X}$, a positive definite real-valued kernel K on $\mathcal{X}\times\mathcal{X}$, a training sample $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$, a strictly monotonically increasing real-valued function g on $[0, \infty)$, an arbitrary cost function $c : (\mathcal{X} \times \mathbb{R}^2)^n \to \bar{\mathbb{R}}$, and a class of functions*

$$\mathcal{F} = \left\{ f \in \mathbb{R}^{\mathcal{X}} : f(\cdot) = \sum_{i=1}^{\infty} \beta_i K(\cdot, z_i), \beta_i \in \mathbb{R}, z_i \in \mathcal{X}, \|f\|_{\mathcal{H}_K} < \infty \right\},$$

where $\|\cdot\|_{\mathcal{H}_K}$ is the norm in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ associated with K, i.e., for any $z_i \in \mathcal{X}$, $\beta_i \in \mathbb{R}$ $(i \in \mathbb{N})$,

$$\left\| \sum_{i=1}^{\infty} \beta_i K(\cdot, z_i) \right\|_{\mathcal{H}_K}^2 = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \beta_i \beta_j K(z_i, z_j).$$

Then for any $f \in \mathcal{F}$ minimizing the regularized risk functional

$$c((x_1, y_1, f(x_1)), \ldots, (x_n, y_n, f(x_n))) + g(\|f\|_{\mathcal{H}_K})$$

admits a representation of the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, x_i).$$

Here are some details in Liu et al. (2007)[42]. Consider a partial linear model

$$y_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + h(\boldsymbol{z}_i) + \epsilon_i, \quad \epsilon_i \sim \text{i.i.d.} \mathcal{N}(0, \sigma^2)$$

where $\boldsymbol{x}_i \in \mathbb{R}^q$ is the vector of clinical covariates for subject i; $\boldsymbol{z}_i \in \mathbb{R}^p$ is a vector of gene expression pathway and h lies in some RKHS $\mathcal{H}_K$, the estimator for $\boldsymbol{\beta}$ and h obtained by minimizing the penalized quadratic loss function

$$J(h, \boldsymbol{\beta}) = \frac{1}{2} \sum_{i=1}^{n} \Big(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta} - h(\boldsymbol{z}_i) \Big)^2 + \frac{\lambda}{2} \|h\|_{\mathcal{H}_K}^2$$

is the same as the best linear unbiased predictor (BLUP) of the linear mixed effect model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{h} + \boldsymbol{\epsilon}$$

$$\boldsymbol{h} \sim \mathcal{N}_n(\boldsymbol{0}, \tau \boldsymbol{K})$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n),$$

where the (i, j)th element of $\boldsymbol{K}$ is $K(\boldsymbol{z}_i, \boldsymbol{z}_j)$ and $\tau = \lambda^{-1}\sigma^2$. Hence to conduct hypothesis

testing on $H_0 : h = 0$, it is equivalent to test $H_0 : \tau = 0$. In fact, the normality assumptions here is not necessary since as we shall see below, the equivalence comes from the Henderson's mixed model equation [28], which does not depend on the normality assumptions.

1.3.3 Neural Networks

A neural network is a learning algorithm used to solve nonlinear classification and regression problems. The origins of neural network are algorithms that try to mimic the function of a human brain. Figure 1.1 provides the structure of a neuron in a human brain.

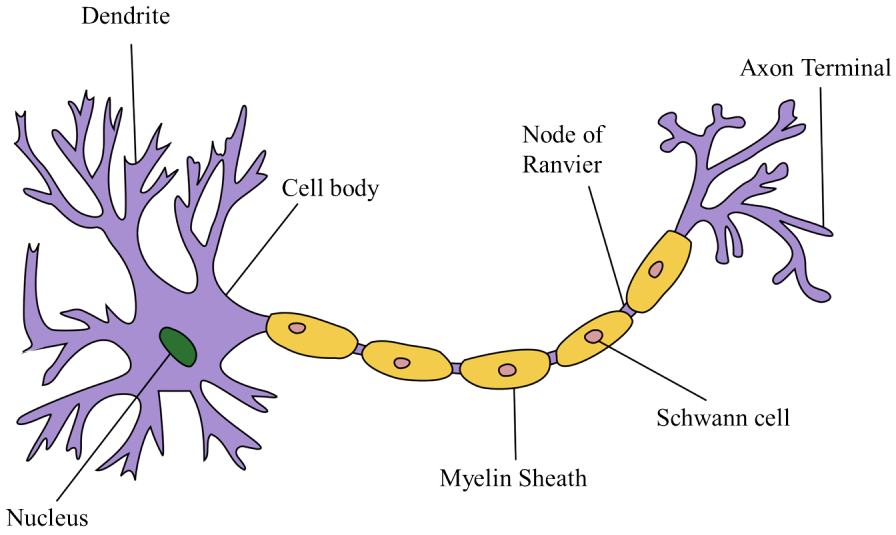

Figure 1.1: Diagram of a neuron with main components (Figure from Wiki).

In a neuron, dendrites are "input wires" receiving signals from other locations and axon is an "output wire" which sends signals to other neurons. So basically, we have some inputs and within the neuron, some calculations are processed and then send the results through axons. Hence a neuron model (logistic unit) can be constructed as shown in Figure 1.2. Such model is known as a perceptron. In this simplified structure, $x_1, \ldots, x_p$ are input units. The node represented by Σ is called a computation unit, which plays a role analogous to the body

of the neuron. The arrows from the input units to the computation unit are “input wires", mimic the dendrites of a neuron, while the arrow originated from the computation unit is an “output wire", analogous to the axon of a neuron.

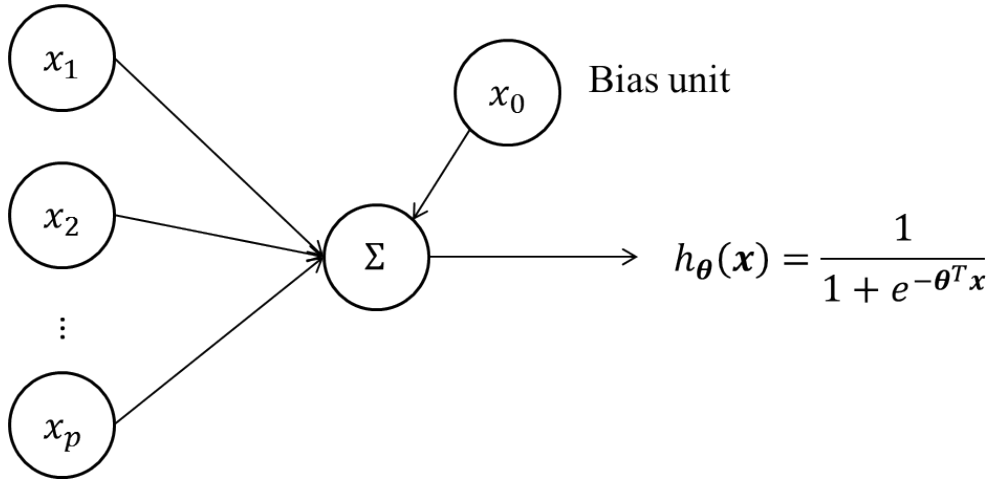

Figure 1.2: A perceptron that mimics the structure of a neuron in a human brain. $x_1, \ldots, x_p$ are input features and x_0 is a bias unit or intercept. Σ is known as a computation unit, where a linear combination of the features, including the bias unit and the parameters $\boldsymbol{\theta}$ to be learned is calculated and then an activation function is applied. In this example, a standard sigmoid activation function $\sigma(x) = (1 + e^{-x})^{-1}$ is applied.

In this perceptron model, the computation unit first calculates the linear combination of the inputs (including the bias unit) $\boldsymbol{x}$ and the parameters (also known as weights) $\boldsymbol{\theta}$ and then apply a sigmoid activation function $\sigma(x) = (1 + e^{-x})^{-1}$. Other activation functions typically used in neural networks are

- **Hyperbolic Tangent (Tanh)**: $a(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$;
- **Rectified Linear Unit (ReLU)**: $a(x) = x_+ = \max\{x, 0\}$;
- **Leaky ReLU**: $a(x) = \max\{\epsilon x, x\}$ for some small positive number ϵ.

A neural network is simply a group of perceptrons combined together as shown in Figure 1.3. This is an example of a neural network with one hidden layer in that the computation

units in the dashed rectangle are known as hidden units. Modern deep learning methods are based on deep neural networks, which are simply neural networks with multiple hidden layers. But in this dissertation, we will only focus on neural networks with one hidden layer.

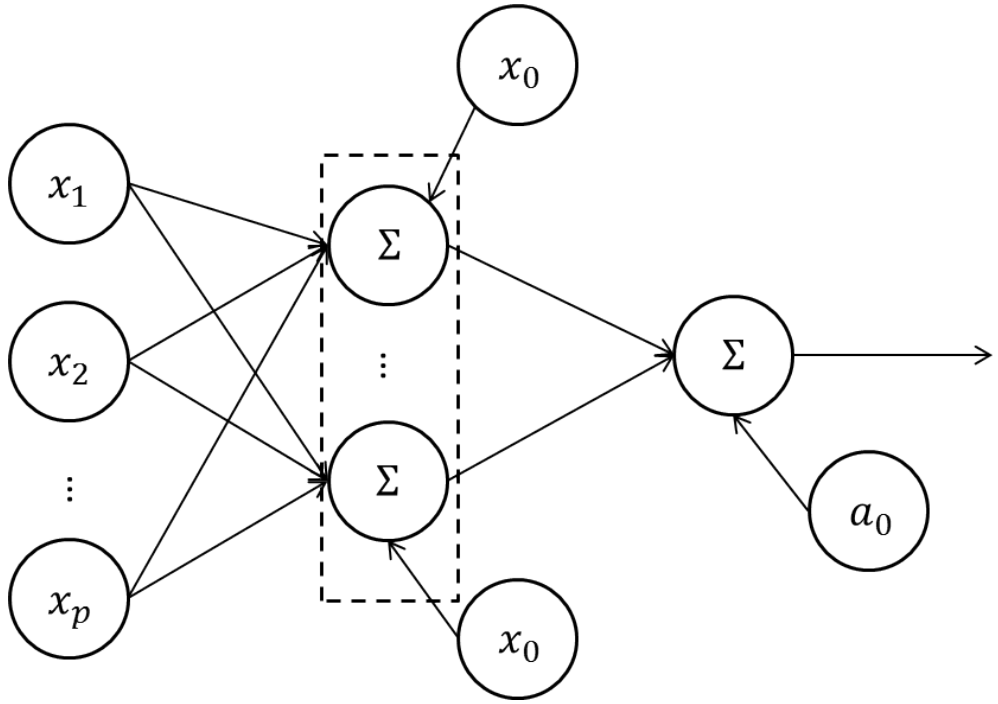

Figure 1.3: A neural network with one hidden layer. $x_1, \ldots, x_p$ are input features and x_0 is a bias unit or intercept. Units in the dashed rectangle are known as hidden computation units, where a linear combination of features from previous layer and associated parameters is calculated and an activation function is applied to obtain outputs for these hidden units. Σ at the end is an output computation unit, in which a linear combination of the output features from the last hidden layers and associated weights is calcuated and an acitivation function is applied.

One promising property for neural networks is the universal approximation property, which simply says that a neural network with one hidden layer can approximate a continuous function on a compact support with arbitrary degree of accuracy. Formally, this is given in the following theorem.

Theorem 1.3.3 (Universal Approximation Theorem (Theorem 30.4 in [18]))**.** *For every continuous function* $f : [a, b]^d \to \mathbb{R}$ *and for every* $\epsilon > 0$*, there exists a neural network with*

one hidden layer $\psi(\boldsymbol{x})$ *such that*

$$\sup_{\boldsymbol{x}\in[a,b]^d} |f(\boldsymbol{x}) - \psi(\boldsymbol{x})| < \epsilon.$$

1.4 Objective and Organization

In current literature of genetic association analysis based on linear mixed models, the covariance matrix of the genetic random effects are usually specified directly based on some kernel matrices. Our proposed approach used an indirect method. Instead Due to the Gaussian assumption, the zero element in the precision matrix indicates a missing edge in the graph representing the connections of the genetic effect among the subjects. Therefore, the CAR model is based on the assumption that similar genetic variants leads to similar genetic effects. Therefore, our model is able to capture more genetic heterogeneity among individuals.

Since a linear mixed model with a kernel matrix as the covariance matrix of the random effect is equivalent to a semi-parametric model, the prediction performance for linear mixed model is usually satisfactory. By adding another hierarchy to a linear mixed model, we can show that it is equivalent to a nonlinear mixed effect model so that it may have even better prediction performance in genetic risk prediction than classical linear mixed model. This is the basic idea and objective of our proposed kernel neural network.

Most researches on neural networks focus on improving the prediction accuracy of neural networks on testing data set. From a statistical point of view, a neural network is simply a nonlinear regression problem. Therefore, it is worthwhile to establish the asymptotic properties, including the consistency and asymptotic normality of neural network estimators since once we have establish these properties, we may conduct statistical inference and may

apply to genetic association analysis later on. However, as we will see, classical results on nonlinear regression cannot be applied directly to neural networks. Instead, we used some techniques in empirical processes and nonparametric regression to establish consistency and asymptotic normality of neural network sieve estimators.

The remaining dissertation is organized as follows. In Chapter 2, we focus on a conditional autoregressive model with its application to genetic association analysis on sequencing data. In Chapter 3, we focus on a kernel-based neural network and discuss its performance in genetic risk prediction. In Chapter 4, we derive the asymptotic properties for neural network sieve estimators and in Chapter 5, we discuss possible extensions of these methods and future work.

Chapter 2

A Conditional Autoregressive Model for Genetic Association Analysis Accounting for Genetic Heterogeneity

2.1 Introduction

Substantial evidence from a wide range of diseases (e.g., breast cancer and hearing loss) indicates that complex diseases are characterized by remarkable genetic heterogeneity [46]. Evolutionary studies also suggest that individually rare mutations generated from each generation create vast genetic heterogeneity in human diseases and could collectively play a substantial role in causing diseases. The recently developed whole-genome sequencing technology generates a deep catalog of genetic variants, especially those rare variants, and allows researchers to comprehensively investigate their role in human diseases. Although new technology holds promise for uncovering novel disease-associated variants, the massive amount of sequencing data and low frequent of rare variants bring tremendous analytical challenges to sequencing data analysis. Further challenge comes from sequencing variants, especially those rare variants, which could be highly heterogeneous: (1) the same gene may harbor many (hundreds of even thousands) different rare mutations; and (2) the same variant may

have heterogeneous effects in different individuals or subgroups of individuals [46]. These degree of genetic heterogeneity often have been neglected in existing statistical frameworks, adding another layer of difficulty to the discovery process.

Many new statistical methods have been proposed to deal with the joint association analysis of single nucleotide variants (SNVs), including rare variants. The burden test developed by Morgenthaler and Thilly (2007), Li and Leal (2008), Madsen and Browning (2009)[40, 44, 49], as a pioneer in testing the genetic association on sequencing data, collapses all the genetic information through a weighted sum. The burden test performs well if the effects of SNVs are in the same direction and same magnitude. Nevertheless, it is subject to power loss if the assumption fails. Similarity-based methods have been proposed to address this issue. One of the most popular methods is the sequence kernel association test (SKAT) [78], which is a semi-parametric method. SKAT is closely related to the sum of square score (SSU) test [52], and can detect both uni-directional and bi-directional genetic effects. More recently, a genetic random field model (GenRF) [27] was proposed in analyzing sequencing data for continuous phenotypes, and a generalized genetic random field model (GGRF) [41] was proposed to generalize the random field model for other types of phenotypes. Most of these methods were based on the idea that individuals with similar genotypes tend to have similar phenotypes. Compared with other similarity-based methods (e.g., SKAT), GenRF and GGRF have nice asymptotic properties, and can by applied to small-scale sequencing studies without small-sample adjustment.

Most of the existing methods assume the disease under investigation as one unified phenotype with homogeneous genetic causal. When genetic heterogeneity is present, the existing methods will likely yield attenuated estimates for genetic variants with heterogeneous effects, leading to low testing power. To consider the genetic heterogeneity in sequencing studies,

we proposed a conditional autoregressive (CAR) model. Different from the previous GGRF model, which applies the conditional autoregressive model on the phenotypes directly, we use a linear mixed model with the genetic effect being considered as a random effect to account for heterogeneous genetic effects. By using a score test for variance components, it has advantage of computational efficiency since we only need to obtain estimators under the null hypothesis. On the other hand, it shares a nice asymptotic feature with GenRF and GGRF, which makes it appealing for small sample size studies. Simulation studies also showed that our proposed method can have high power when rare variants play an important role or when variants have different genetic effects among different individuals or subgroups of individuals. Therefore, CAR provides a powerful alternative approach to search for disease-associated variants, especially those rare or having heterogeneous effects.

The remaining chapter is arranged as follow: In section 2.2, we propose a conditional autoregressive model for genetic association analysis of sequencing data and a score test for statistical inference. In section 2.3, we conduct simulation studies to compare performance of our method with two existing methods (i.e., SKAT and GGRF) under different scenarios. Finally, in section 2.4, we apply our model to the whole-genome sequencing data from the Alzheimer's Disease Neuroimaging Initiative.

2.2 Methods

2.2.1 Motivation

The linear mixed model has been commonly used to assess the association of a set of SNVs with a continuous phenotype. It models the effect of each SNV as a random effect and assumes the genetic effects of all SNVs in the set (e.g., a gene) follow an arbitrary distribution.

By testing the variance component of the random effect, it evaluates the joint effects of SNVs in the set on the phenotype of interest. One of the most popularly used linear mixed model for sequencing studies is the sequence kernel association test (SKAT) developed by Wu et al. (2011) [78]. It is a semi-parametric method that uses a kernel function to deal with high-dimensional genetic data and uses a score-based variance component test to assess the association. While SKAT has many advantages, such as being robust for the direction and magnitude of genetic effects, it does not consider the heterogeneous effects of genetic variants among individuals or subgroups of individuals (e.g., gender and race groups). If the disease of interest undergoes heterogeneous genetic etiological processes (i.e., genetic causes differs among individuals), the traditional linear mixed model (e.g., SKAT), which typically assumes the genetic effects are similar across all the samples, can suffer from power loss. To consider the genetic heterogeneity in association analysis, we propose a conditional autoregressive model. Similar to SKAT, the genetic effect of an individual is considered as a random effect. In fact, according to the CAR model construction, it is based on the assumption that similar genotype leads to similar genetic effect and therefore accounts for the genetic heterogeneity among individuals.

All SKAT [78], genome-wide complex trait analysis (GCTA) [79] and CAR are based on the following linear mixed model:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{a} + \boldsymbol{\epsilon},$$

where $\boldsymbol{a}$ is the total genetic random effects of the individuals; $\boldsymbol{X}$ is the design matrix containing clinical covariates such as age, gender, etc. and $\boldsymbol{\epsilon}$ is the random error. It is natural to assume the total genetic effect $\boldsymbol{a}$ follows a multivariate normal distribution with

$\boldsymbol{a} \sim \mathcal{N}_n(\mathbf{0}, \sigma_a^2 \boldsymbol{\Sigma})$. A natural question is how to choose the covariance matrix $\boldsymbol{\Sigma}$. In SKAT, $\boldsymbol{\Sigma} = \boldsymbol{G}\boldsymbol{W}\boldsymbol{G}^T$, where $\boldsymbol{G}$ is a matrix of all SNVs and $\boldsymbol{W} = \text{diag}\{w_1, \ldots, w_p\}$ is a diagonal matrix containing the weights of the p genetic variants. In GCTA [79], the (i, j)-th element in $\boldsymbol{\Sigma}$ is defined to be

$$\boldsymbol{\Sigma}_{ij} = \frac{1}{p} \sum_{k=1}^{p} \frac{(g_{ik} - 2p_k)(g_{jk} - 2p_k)}{2p_k(1 - p_k)},$$

where p_k is the frequency of the reference allele for the kth SNV. Both methods use a direct way to define the marginal covariance of the genetic effects between two subjects. For CAR model, we consider an indirect way to model the covariance of the genetic effects. Specifically, if we use a graph to represent the connections of the genetic effects among the individuals as shown in Figure 2.1 where a circle represents a person's genetic effect and an edge represents a link between the genetic effects of two subjects, due to the Gaussian assumption on the random effect $\boldsymbol{a}$, an edge exists if and only if $a_i \perp a_j | \boldsymbol{a}_{-i,j}$, or equivalently $p(a_i|\boldsymbol{a}_{-i}) = p(a_i|\boldsymbol{a}_{-i,j})$. So it is worth investigating the conditional distribution of $a_i|\boldsymbol{a}_{-i}$.

A reasonable conditional model can be assumed as follow

$$a_i | a_j, j \neq i \sim \mathcal{N}\left(\sum_{j \neq i} b_{ij} a_j, \tau_i^2\right),$$

To find the joint distribution of $\boldsymbol{a}$, we first quote the Brook's Lemma.

Lemma 2.2.1 ([10]). *Let $\pi(\boldsymbol{x})$ be the density for $\boldsymbol{x} \in \mathbb{R}^n$ and define $\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : \pi(\boldsymbol{x}) > 0\}$. Let $\boldsymbol{x}, \boldsymbol{x'} \in \Omega$, then*

$$\begin{aligned}
\frac{\pi(\boldsymbol{x})}{\pi(\boldsymbol{x'})} &= \prod_{i=1}^{n} \frac{\pi(x_i | x_1, \ldots, x_{i-1}, x'_{i+1}, \ldots, x'_n)}{\pi(x'_i | x_1, \ldots, x_{i-1}, x'_{i+1}, \ldots, x'_n)} \\
&= \prod_{i=1}^{n} \frac{\pi(x_i | x'_1, \ldots, x'_{i-1}, x_{i+1}, \ldots, x_n)}{\pi(x'_i | x'_1, \ldots, x'_{i-1}, x_{i+1}, \ldots, x_n)}
\end{aligned}$$

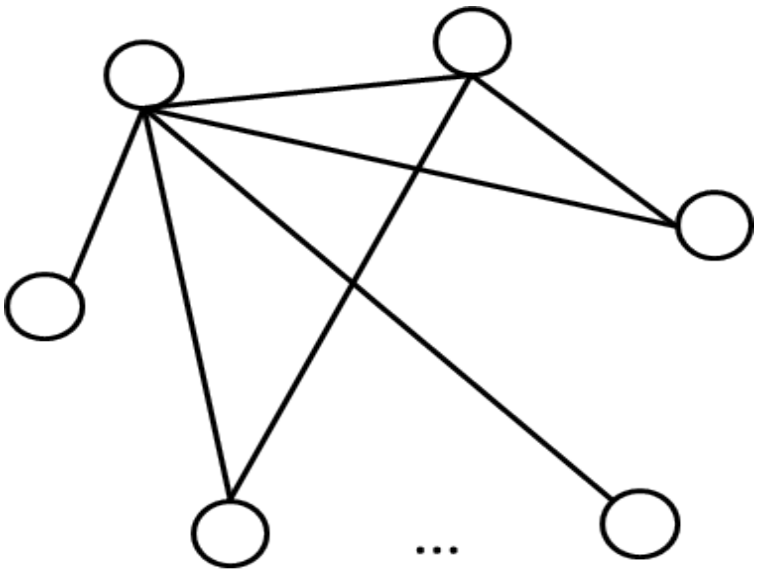

Figure 2.1: An illustration of a Gaussian graphical model. For the CAR model, the graph represents a network of the genetic effects among individuals. In the graph, each node stands for the gentic effect of an indiviudal and the existence of an edge between two individuals indicates dependence between genetic effects between these two individuals. This is characterized by the precision matrix in the joint normal distribution of these genetic effects.

Now we are able to transfer a conditional distribution to a joint distribution, which is established in 2.2.1 and its proof can be found in Appendix A.

Proposition 2.2.1. *Let* $\boldsymbol{a} \in \mathbb{R}^n$ *be a random vector with*

$$a_i | a_j, j \neq i \sim \mathcal{N}\left(\sum_{j \neq i} b_{ij} a_j, \tau_i^2\right),$$

then the joint density function of $\boldsymbol{a}$ *is given by*

$$\pi(\boldsymbol{a}) \propto \exp\left\{-\frac{1}{2}\boldsymbol{a}^T \boldsymbol{\Delta}^{-1}(\boldsymbol{I} - \boldsymbol{B})\boldsymbol{a}\right\},$$

where $\boldsymbol{B} = [b_{ij}]$ *is a symmetric matrix with* $b_{ii} = 0$ *and* $\boldsymbol{\Delta} = diag\{\tau_1^2, \ldots, \tau_n^2\}$. *This shows that* $\boldsymbol{a} \sim \mathcal{N}_n(\boldsymbol{0}, (\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\Delta})$.

Proposition 2.2.1 shows that $\boldsymbol{a} \sim \mathcal{N}_n(\mathbf{0}, (\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{\Delta})$. The first thing we need to ensure is that $\boldsymbol{\Delta}^{-1}(\boldsymbol{I} - \boldsymbol{B})$ is symmetric and a simple condition is $\boldsymbol{\Delta}^{-1}\boldsymbol{B}$ is symmetric, which becomes

$$\frac{b_{ij}}{\tau_i^2} = \frac{b_{ji}}{\tau_j^2} \text{ for all } i, j.$$

Given a similarity matrix $\boldsymbol{S}$, this can be accomplished by setting $b_{ij} = s_{ij}/\sum_{j \neq i} s_{ij}$ and $\tau_i^2 = \sigma_a^2/\sum_{j \neq i} s_{ij}$ and the CAR model becomes

$$a_i | a_j, j \neq i \sim \mathcal{N}\left(\frac{1}{\sum_{j \neq i} s_{ij}} \sum_{j \neq i} s_{ij} a_j, \frac{\sigma_a^2}{\sum_{j \neq i} s_{ij}}\right). \tag{2.1}$$

In such case, the joint distribution of $\boldsymbol{a}$ is

$$\pi(\boldsymbol{a}) \propto \exp\left\{-\frac{1}{2\sigma_a^2}\boldsymbol{a}^T(\boldsymbol{D} - \boldsymbol{S})\boldsymbol{a}\right\},$$

where $\boldsymbol{D} = \text{diag}\left\{\sum_{j \neq 1} s_{1j}, \ldots, \sum_{j \neq n} s_{nj}\right\}$. However, one issue here is that $\boldsymbol{D} - \boldsymbol{S}$ is singular since $(\boldsymbol{D} - \boldsymbol{S})\mathbf{1}_n = \mathbf{0}$. Such impropriety can be remedied by redefining the precision matrix of $\boldsymbol{a}$ as $\boldsymbol{D} - \gamma\boldsymbol{S}$, where γ is chosen to make $\boldsymbol{D} - \gamma\boldsymbol{S}$ nonsingular. In fact, as will be shown in Proposition 2.2.2, the nonsingularity of $\boldsymbol{D} - \gamma\boldsymbol{S}$ is guaranteed by choosing $|\gamma| < 1$. The proof of Proposition 2.2.2 can be found in Appendix A.

Proposition 2.2.2. *Let $\boldsymbol{D} - \gamma\boldsymbol{S}$ be as defined in the main text. Then it is nonsingular if $|\gamma| < 1$.*

In such case, (2.1) becomes

$$a_i | a_j, j \neq i \sim \mathcal{N}\left(\frac{\gamma}{\sum_{j \neq i} s_{ij}} \sum_{j \neq i} s_{ij} a_j, \frac{\sigma_a^2}{\sum_{j \neq i} s_{ij}}\right), \tag{2.2}$$

which will be used in the model defined in section 2.2.2.

The main difference between CAR model and the other methods is that the precision matrix of the random effect $\boldsymbol{a}$, which is the inverse of the covariance matrix, is specified first. It is well-known that the (i, j)-th element in the precision matrix is in fact the partial covariance between a_i and a_j given all the others. The generalized genetic random fields (GGRF) model proposed by Li et al.(2014)[41] has a similar structure. A GGRF model is based on the idea of similar genotypes leads to similar phenotypes, while a CAR model can be interpreted as similar genotypes leads to similar genetic effects and the variations among phenotypes is related to the variations among genetic effects of individuals.

In section 2.2.2, a CAR model will be introduced to account for the genetic heterogeneity. In section 2.2.3, a linear score test for testing the variance component will be derived.

2.2.2 Model Setup

Suppose that there are n subjects. Let y_i be a quantitative trait for the ith subject. To relate the phenotype with genetic variants, we consider the following linear mixed model, also known as the conditional autoregressive (CAR) model:

$$\begin{aligned} y_i &= \boldsymbol{x}_i^T \boldsymbol{\beta} + a_i + \epsilon_i, \quad i = 1, \ldots, n \\ a_i | a_j, j \neq i &\sim \mathcal{N}\left(\frac{\gamma}{\sum_{j \neq i} s_{ij}} \sum_{j \neq i} s_{ij} a_j, \frac{\sigma_a^2}{\sum_{j \neq i} s_{ij}} \right) \\ \epsilon_1, \ldots \epsilon_n &\sim \text{i.i.d. } \mathcal{N}(0, \sigma^2), \end{aligned} \tag{2.3}$$

where $\boldsymbol{x}_i = [x_{i1}, \ldots, x_{ip}]^T$ is the covariates of the ith subject; $\boldsymbol{\beta}$ is the fixed effect of the covariates; a_i is the genetic random effect and ϵ_i is the random error of the ith subject;

γ measures the overall genetic correlation among all subjects. A larger value of γ implies strong overall genetic correlation among all samples; s_{ij} is the genetic similarity between the ith and the jth subjects, which is measured on a set of SNVs. s_{ij} does not need to be estimated and it is calculated based on some given similarity metrics (e.g. IBS kernel); σ_a^2 measures the variation of the genetic effects.

As we have seen in section 2.2.1, the joint distribution of $\boldsymbol{a} = [a_1, \ldots, a_n]^T$ can be written as

$$\boldsymbol{a} \sim \mathcal{N}_n(\boldsymbol{0}, \sigma_a^2(\boldsymbol{D} - \gamma \boldsymbol{S})^{-1}),$$

where $\mathcal{N}_n$ denotes an n-dimensional multivariate normal distribution; $\boldsymbol{S}$ is the genetic similarity matrix and $\boldsymbol{D}$ is a diagonal matrix with diagonal elements being the row sums of $\boldsymbol{S}$.

$$\boldsymbol{S} = \begin{bmatrix} 0 & s_{12} & s_{13} & \cdots & s_{1n} \\ s_{21} & 0 & s_{23} & \cdots & s_{2n} \\ s_{31} & s_{32} & 0 & \cdots & s_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & s_{n3} & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{D} = \begin{bmatrix} \sum_{j \neq 1} s_{1j} & & & & \\ & \sum_{j \neq 2} s_{2j} & & & \\ & & & \ddots & \\ & & & & \sum_{j \neq n} s_{nj} \end{bmatrix}.$$

Therefore, model (2.3) can be written into the following matrix form:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{a} + \boldsymbol{\epsilon} \\ \boldsymbol{a} &\sim \mathcal{N}_n(\boldsymbol{0}, \sigma_a^2(\boldsymbol{D} - \gamma \boldsymbol{S})^{-1}) \\ \boldsymbol{\epsilon} &\sim \mathcal{N}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n), \end{aligned} \tag{2.4}$$

where $\boldsymbol{y} = [y_1, \ldots, y_n]^T$ is the phenotype and $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T$ is the covaraite matrix. Since all the genetic information are contained in the total genetic random effects $\boldsymbol{a}$, in order to test the association of a SNV set with the response, it is sufficient to test $H_0 : \sigma_a^2 = 0$.

Let $\boldsymbol{G}_i = [g_{i1}, \ldots, g_{iK}]^T$ be the the genotypes of K SNVs in a genetic region (e.g. a gene) for the ith subject, where $g_{ij}, j = 1, \ldots, K$ is coded as additive, i.e. $\{0, 1, 2\}$. Given the set of SNVs for subjects i and j, a commonly-used kernel function measuring genetic similarity is the weighted identity-by-state (IBS) function [78], which is defined by

$$s(\boldsymbol{G}_i, \boldsymbol{G}_j) = \sum_{k=1}^{K} \omega_k \{2 - |g_{ik} - g_{jk}|\},$$

where ω_k is the prespecified weight for the kth variant, which is usually a function of minor allele frequency (MAF). In our model, we consider the scaled version of the weighted IBS similarity defined by

$$s_{ij} = \sum_{j=1}^{K} \frac{\omega_k \{2 - |g_{ik} - g_{jk}|\}}{2 \sum_{k=1}^{K} \omega_k}.$$

For the weights ω_k, we considered four different weight functions based on the minor allele frequencies (MAF) as considered in Li et al. (2014) [41]. Among these four weight functions, unweighted (UW) assigns same weights to both common and rare variants; weighted sum statistics type of weight (WSS), on the other hand, put almost all the weights on rare variants and nearly no weights on the common variants. The Beta distribution type of weights (BETA) and the logarithm of MAFs (LOG) lie between UW and WSS with LOG putting more weights on common variants than BETA.

1. Unweighted (UW)

$$\omega_k = 1, \quad 1 \leq k \leq p$$

2. Beta distribution type of weights (BETA)

$$\omega_k = \text{dbeta}(\text{MAF}_k, 1, 25)^2, \quad 1 \leq k \leq p,$$

i.e. the weight is the square of the probability density of the Beta distribution with parameters 1 and 25.

3. Weighted sum statistics type of weights (WSS)

$$\omega_k = \frac{1}{\text{MAF}_k(1 - \text{MAF}_k)}, \quad 1 \leq k \leq p$$

4. Logarithm of MAFs as weights (LOG)

$$\omega_k = -\log_{10}(\text{MAF}_k), \quad 1 \leq k \leq p$$

2.2.3 Score Test for Variance Component

In order to perform the hypothesis testing, we also need to consider the nuisance parameter γ. In the following discussion, we considered two cases. In the first case, we treat γ as a fixed constant and use the linear score test proposed by Qu et al. (2013) [54]. In the second case, γ is treated as an unknown nuisance parameter. Based on the idea of Davies (1987) [17], a maximum statistic is proposed and the corresponding p-value is obtained via a simulation based method.

To evaluate whether there is a genetic association between a SNV set and the trait of interest, we test $H_0 : \sigma_a^2 = 0$. By introducing the ratio between two variance components $\lambda = \sigma_a^2/\sigma^2$, it is equivalent to test $H_0 : \lambda = 0$ vs $H_1 : \lambda > 0$.

2.2.3.1 γ As a Fixed Constant

When γ is fixed as a constant (e.g. overall mean of SNV correlations), a linear score test procedure [54] can be used to test $H_0 : \lambda = 0$ vs $H_1 : \lambda > 0$. Based on model (2.4), we have the marginal distribution of $\boldsymbol{y}$:

$$\boldsymbol{y} \sim \mathcal{N}_n(\boldsymbol{X}\boldsymbol{\beta}, \tilde{\boldsymbol{V}}),$$

where $\tilde{\boldsymbol{V}} = \sigma_a^2(\boldsymbol{D} - \gamma\boldsymbol{S})^{-1} + \sigma^2\boldsymbol{I}_n = \sigma^2\left[\boldsymbol{I}_n + \lambda(\boldsymbol{D} - \gamma\boldsymbol{S})^{-1}\right] := \sigma^2\boldsymbol{V}(\lambda)$. Let $\boldsymbol{\theta} = [\lambda, \sigma^2, \boldsymbol{\beta}^T]^T$ be the vector of unknown parameters, where σ^2 and $\boldsymbol{\beta}$ are nuisance parameters. The nuisance parameter $\boldsymbol{\beta}$ does not need to be estimated when we use the restricted log-likelihood method. We can find a $q \times n$ matrix $\boldsymbol{K}$ such that $\boldsymbol{K}\boldsymbol{X} = 0$ and $\boldsymbol{K}\boldsymbol{K}^T = \boldsymbol{I}_q$, where $q = n - \text{rank}(\boldsymbol{X})$. Such a matrix can be found by using the QR decomposition of $\boldsymbol{X}$ [47]. Since $\boldsymbol{y} \sim \mathcal{N}_n(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2\boldsymbol{V}(\lambda))$, we obtain

$$\boldsymbol{y}^* := \boldsymbol{K}\boldsymbol{y} \sim \mathcal{N}_q(\boldsymbol{0}, \sigma^2\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T), \tag{2.5}$$

where $\boldsymbol{y}^*$ is known as the error contrast since $\mathbb{E}[\boldsymbol{y}^*] = \boldsymbol{0}$, which does not depend on $\boldsymbol{X}$. The restricted log-likelihood function, which provides an unbiased estimator of σ^2, can be formed as

$$\ell(\lambda, \sigma^2|\boldsymbol{y}^*) \propto -\frac{q}{2}\log\sigma^2 - \frac{1}{2}\log|\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T| - \frac{1}{2\sigma^2}\boldsymbol{y}^{*T}(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{y}^*. \tag{2.6}$$

Moreover, by considering the profiled version of the restricted log-likelihood function in (2.6), we can get the profiled REML of σ^2 for a given λ:

$$\tilde{\sigma}^2(\lambda) = \arg\max_{\sigma^2} \ell(\sigma^2|\boldsymbol{y}^*, \lambda) = \frac{1}{q}\boldsymbol{y}^{*T}(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{y}^*.$$

Back substituting $\tilde{\sigma}^2(\lambda)$ into the restricted log-likelihood function (2.6), we get the profiled restricted log-likelihood function as follow:

$$\ell_p(\lambda|\boldsymbol{y}^*) \propto -\frac{q}{2}\log[\boldsymbol{y}^{*T}(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{y}^*] - \frac{1}{2}\log|\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T|. \tag{2.7}$$

As pointed out in Stern and Welsh (2000) [66], the profiled restricted log-likelihood function is score and information unbiased and hence it can be used for inference. The score statistic $S(\lambda)$ can be calculated as follow:

$$\begin{aligned}
S(\lambda) &= \frac{\partial \ell_p}{\partial \lambda} \\
&= \frac{q}{2}\frac{\boldsymbol{y}^{*T}(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{K}\frac{\partial \boldsymbol{V}(\lambda)}{\partial\lambda}\boldsymbol{K}^T(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{y}^*}{\boldsymbol{y}^{*T}(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{y}^*} - \\
&\qquad\qquad \frac{1}{2}\mathrm{tr}\left[(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{K}\frac{\partial \boldsymbol{V}(\lambda)}{\partial\lambda}\boldsymbol{K}^T\right] \\
&= \frac{q}{2}\frac{\boldsymbol{y}^{*T}(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{K}(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\boldsymbol{K}^T(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{y}^*}{\boldsymbol{y}^{*T}(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{y}^*} - \\
&\qquad\qquad \frac{1}{2}\mathrm{tr}\left[(\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)^{-1}\boldsymbol{K}(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\boldsymbol{K}^T\right].
\end{aligned}$$

Under the null hypothesis $H_0 : \lambda = 0$, $\boldsymbol{V}(\lambda) = \boldsymbol{V}(0) = \boldsymbol{I}_n$ so that the score statistic under H_0 can be expressed as:

$$S(0) = \frac{q}{2}\frac{\boldsymbol{y}^{*T}\boldsymbol{K}(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\boldsymbol{K}^T\boldsymbol{y}^*}{\boldsymbol{y}^{*T}\boldsymbol{y}^*} - \frac{1}{2}\mathrm{tr}\left[(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\right].$$

To test $H_0 : \lambda = 0$ vs $H_1 : \lambda > 0$, it is equivalent to test $H_0 : S(0) = 0$ vs $H_1 : S(0) \neq 0$. The exact way to calculate the p-value of the test is given in Qu et al. (2013) [54]

$$\mathbb{P}(S(0) > s) = \mathbb{P}\left(\sum_{j=1}^{q} \lambda_j Z_j^2 > 0\right),$$

where s is the observed value of $S(0)$, $\lambda_1, \ldots, \lambda_q$ are the eigenvalues of the following matrix

$$\boldsymbol{B} = \boldsymbol{K}(\boldsymbol{D} - \gamma \boldsymbol{S})^{-1}\boldsymbol{K}^T - \left[\frac{2s}{q} + \frac{1}{q}\mathrm{tr}\left((\boldsymbol{D} - \gamma \boldsymbol{S})^{-1}\right)\right]\boldsymbol{I}_q.$$

$Z_1^2, \ldots, Z_q^2$ are independent chi-square-distributed random variables and the p-value can be calculated by using the Davis method [16].

2.2.3.2 γ As an Unknown Nuisance Parameter

In practice, γ is usually unknown so that it is necessary to consider how to test $H_0 : \lambda = 0$ vs $H_1 : \lambda > 0$ under such case. One natural idea is to estimate γ and plug in the estimator (e.g. MLE). However, since γ is embedded in the variance-covariance matrix of $\boldsymbol{y}$, it is unlikely that the maximum likelihood estimator of γ has an analytic form. Moreover, γ only appears in the alternative model, therefore it is not even possible to evaluate the MLE of γ under H_0. Instead, we follow the idea of Davies (1987) [17] and construct a maximum score statistic for the association test. More specifically, following the notations we used in section 2.2.3.1, we have

$$S_\gamma(0) = \frac{q}{2}\frac{\boldsymbol{y}^{*T}\boldsymbol{K}(\boldsymbol{D} - \gamma \boldsymbol{S})^{-1}\boldsymbol{K}^T\boldsymbol{y}^*}{\boldsymbol{y}^{*T}\boldsymbol{y}^*} - \frac{1}{2}\mathrm{tr}\left[(\boldsymbol{D} - \gamma \boldsymbol{S})^{-1}\right],$$

where $\boldsymbol{y}^* \sim \mathcal{N}_q(\boldsymbol{0}, \sigma^2\boldsymbol{K}\boldsymbol{V}(\lambda)\boldsymbol{K}^T)$ with $\boldsymbol{V}(\lambda) = \boldsymbol{I}_n + \lambda(\boldsymbol{D} - \gamma \boldsymbol{S})^{-1}$. Now, note that under the null hypothesis $H_0 : \lambda = 0$, we have $\boldsymbol{y}^* \sim \mathcal{N}_q(\boldsymbol{0}, \sigma^2\boldsymbol{I}_q)$ so that $\frac{1}{\sigma}\boldsymbol{y}^* \sim \mathcal{N}_q(\boldsymbol{0}, \boldsymbol{I}_q)$. We

form the maximum score test statistic as follow:

$$\begin{aligned}
T &= \sup_{\gamma\in\Gamma} S_\gamma(0) \\
&= \sup_{\gamma\in\Gamma} \frac{q{\boldsymbol{y}^*}^T \boldsymbol{K}(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\boldsymbol{K}^T\boldsymbol{y}^* - {\boldsymbol{y}^*}^T \operatorname{tr}\left[(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\right]\boldsymbol{I}_q\boldsymbol{y}^*}{2{\boldsymbol{y}^*}^T\boldsymbol{y}^*} \\
&= \frac{1}{2\left(\frac{1}{\sigma}\boldsymbol{y}^*\right)^T\left(\frac{1}{\sigma}\boldsymbol{y}^*\right)} \sup_{\gamma\in\Gamma} \left(\frac{1}{\sigma}\boldsymbol{y}^*\right)^T \boldsymbol{\Sigma}(\gamma)\left(\frac{1}{\sigma}\boldsymbol{y}^*\right) \\
&:= (2\|\boldsymbol{Z}\|_2^2)^{-1} \sup_{\gamma\in\Gamma} \boldsymbol{Z}^T\boldsymbol{\Sigma}(\gamma)\boldsymbol{Z},
\end{aligned}$$

where $\boldsymbol{Z} = \frac{1}{\sigma}\boldsymbol{y}^* \sim \mathcal{N}_q(\boldsymbol{0}, \boldsymbol{I}_q)$, $\boldsymbol{\Sigma}(\gamma) = q\boldsymbol{K}(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\boldsymbol{K}^T - \operatorname{tr}\left[(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\right]\boldsymbol{I}_q$. $\gamma \in \left(1/\mu_{(1)}, 1/\mu_{(n)}\right)$, where $\mu_{(1)} < \mu_{(2)} < \cdots < \mu_{(n)}$ are the ordered eigenvalues of $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{S}\boldsymbol{D}^{-\frac{1}{2}}$ ensuring that the matrix $\boldsymbol{D}-\gamma\boldsymbol{S}$ is nonsingular (see [5]). Since the parameter γ can be interpreted as a measurement of correlation coefficient, we set $\Gamma = \left(1/\mu_{(1)}, 1/\mu_{(n)}\right) \cap (-1, 1)$. To calculate the p-value, let t be the observed value of T, then

$$\begin{aligned}
\mathbb{P}(T > t) &= \mathbb{P}\left((2\|\boldsymbol{Z}\|_2^2)^{-1} \sup_{\gamma\in\Gamma} \boldsymbol{Z}^T\boldsymbol{\Sigma}(\gamma)\boldsymbol{Z} > t\right) \\
&= \mathbb{P}\left(\sup_{\gamma\in\Gamma} \boldsymbol{Z}^T\left(\boldsymbol{\Sigma}(\gamma) - 2t\boldsymbol{I}_q\right)\boldsymbol{Z} > 0\right) \\
&= \mathbb{P}\left(\sup_{\gamma\in\Gamma} \sum_{i=1}^{q} \lambda_j(\gamma) Z_j^2 > 0\right),
\end{aligned}$$

where $\lambda_1(\gamma), \ldots, \lambda_q(\gamma)$ are the eigenvalues of

$$\frac{1}{q}(\boldsymbol{\Sigma}(\gamma) - 2t\boldsymbol{I}_q) = \boldsymbol{K}(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\boldsymbol{K}^T - \left(\frac{2t}{q} + \frac{1}{q}\operatorname{tr}\left[(\boldsymbol{D}-\gamma\boldsymbol{S})^{-1}\right]\right)\boldsymbol{I}_q$$

and $Z_1^2, \ldots, Z_q^2$ are independent chi-square-distributed random variables with degrees of freedom 1. We use the following procedures to approximate this probability:

1. Partition the index set Γ as $\gamma_1 < \gamma_2 < \cdots < \gamma_M$. Γ is an open interval, denoted by (L, U). When we implement this step, we choose a small $\epsilon > 0$ and let $\gamma_1 = L + \epsilon$ and $\gamma_M = U - \epsilon$;

2. Generate an $N \times q$ matrix $\boldsymbol{\Upsilon}$ with each element being a χ_1^2 random variable;

3. For each γ_i, $i = 1, \ldots, M$,

   Calculate $\lambda_1(\gamma_i), \ldots, \lambda_q(\gamma_i)$, which are the eigenvalues of $\boldsymbol{\Sigma}(\gamma_i) - 2t\boldsymbol{I}_q$;

   Calculate $\sum_{j=1}^q \lambda_j(\gamma_i)\boldsymbol{\Upsilon}_{kj}$, $k = 1, \ldots, N$, where $\boldsymbol{\Upsilon}_{kj}$ is the (k, j)-th element of $\boldsymbol{\Upsilon}$;

4. Approximate the p-value as

$$\mathbb{P}\left(\sup_{\gamma\in\Gamma}\sum_{j=1}^{q}\lambda_j(\gamma)Z_j^2 > 0\right) \approx \frac{\#\left\{k : \max_{\gamma_i, i=1,\ldots,M}\sum_{j=1}^q \lambda_j(\gamma_i)\boldsymbol{\Upsilon}_{kj} > 0\right\}}{N}.$$

2.3 Simulation Results

Simulation studies were conducted to compare the CAR model with the original GGRF model and the commonly used SKAT method. For the CAR model, we evaluated both scenarios where γ is fixed (denoted by CAR.FIX) and γ is treated as an unknown nuisance parameter (denoted by CAR.SUP). In the simulation, we compared the empirical type I error rates and empirical power of the three methods under various disease models with heterogeneous genetic effects. In addition, we compared the empirical power of the three

methods under different percentage of causal SNVs and under the situation when the weights were misspecified. In order to mimic the real structure of sequencing data, the genetic data used for simulation was simulated based on the real sequencing data of Chromosome 17: 7344328 - 8344327 from the 1,000 Genome project [13]. The minor allele frequencies (MAF) of SNVs in this region range from 0.046% to 49.954% with a distribution highly skewed to the right. Figure 2.2 summarizes the distribution of the MAF with MAF<0.05. For each setting, we simulated 1,000 Monte Carlo replicates to calculate the empirical type I error rates and the empirical power of the three methods. In each replicate, we randomly selected a 30Kb segment from the region and used all the genetic variants in that segment for the association analysis.

We first examined the type I error performances of the three methods. In the simulation, we considered two significance levels, $\alpha = 0.05$ and $\alpha = 0.01$, under the following null model,

$$y_i = \varepsilon_i \sim \mathcal{N}(0, 1).$$

Table 2.1 summarizes the empirical type I error rates of the three methods under four different weights (i.e., UW, BETA, WSS and LOG). As we observe from the results, the empirical type I error rates of GGRF, CAR.FIX and CAR.SUP are well controlled at the level of 0.05 or 0.01, while the type I error rate for SKAT is conservative under WSS (for $\alpha = 0.05$) and LOG (for $\alpha = 0.01$). Because the score test of CAR and the test procedure of GGRF are exact test procedures without any asymptotic approximation, the type I error rates of the two methods are well controlled.

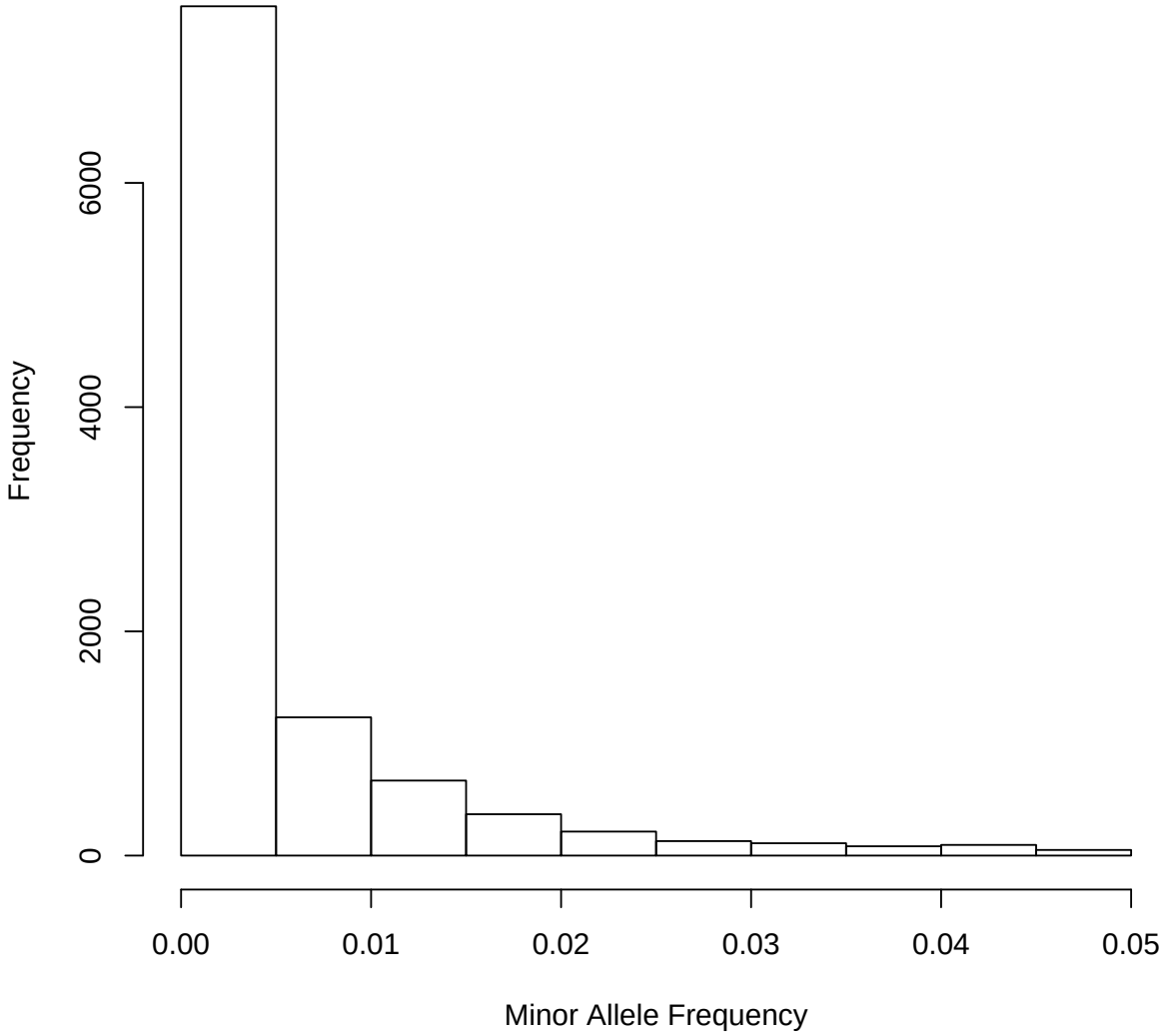

Figure 2.2: The Distribution of minor allele frequency of in sequencing variants on chromosome 17 from the 1,000 Genome project.

Table 2.1: Empirical type I error rates under different weights at level $\alpha = 0.05$ and $\alpha = 0.01$ based on 1000 replicates. Each cell in the table contains the empirical type I error rate. GGRF, SKAT, CAR.FIX and CAR.SUP are the generalized genetic random field model of Li et al. (2014) [41], the sequence kernel association test of Wu et al. (2011) [78], the conditional autoregressive model with fixed nuisance parameter γ, the conditional autoregressive model with maximum score test statistic, respectively

	$\alpha = 0.05$				$\alpha = 0.01$			
Methods	UW	BETA	WSS	LOG	UW	BETA	WSS	LOG
GGRF	0.047	0.041	0.054	0.036	0.006	0.012	0.015	0.009
SKAT	0.041	0.040	0.038	0.042	0.006	0.009	0.007	0.004
CAR.FIX	0.051	0.052	0.057	0.052	0.010	0.010	0.010	0.012
CAR.SUP	0.041	0.053	0.055	0.048	0.010	0.011	0.009	0.012

2.3.1 Simulation I: Heterogeneous Genetic Effects Among Individuals or Subgroups

We first considered a complex disease scenario in which each individual has a different genetic effect, which we refer as the heterogeneous genetic effect. The following model was used to simulate the phenotype:

$$y_i = \sum_{k=1}^{K} w_k^* g_{i,k} Z_{i,k} + \varepsilon_i, \quad 1 \le i \le n, \tag{2.8}$$

where K is the number of genetic variants in a 30 Kb segment; $g_{i,k}$ is the genotype of the kth SNV for individual i, coded as additive (i.e., $g_{i,k} = 0$ for genotype AA, $g_{i,k} = 1$ for genotype Aa and $g_{i,k} = 2$ for genotype aa). We set the percentage of causal SNVs as 50%. $w_k^* = 0$ if the kth genetic variant is not a causal variant and $w_k^* = \omega_k$ if the kth genetic variant is a causal variant, where ω_k is the weight defined in section 2. $Z_{i,1}, \ldots, Z_{i,K}, 1 \le i \le n$ are genetic effects for the ith individual, which follow a normal distribution with mean 0 and standard deviation σ_Z. $\varepsilon_1, \ldots, \varepsilon_n$ are iid random variables distributed as $\mathcal{N}(0, 1)$. Note that

in (2.8), the coefficients of the genetic variants are unique for each individual, and therefore different individuals could have different genetic effects.

Figure 2.3 summarizes the empirical power of three methods under different degrees of genetic heterogeneity. As we observe from the simulation results, the CAR model outperforms the other two methods with the increasing level of genetic heterogeneity (i.e., increasing σ_Z). For instance, when common variants (i.e., UW and LOG) have more contributions to the phenotype, our model has substantial power increase with the increased level of genetic heterogeneity (i.e., σ_Z), while the power of SKAT and GGRF remain low even with the increased genetic effect. When rare variants play an important role (i.e., BETA and WSS), GGRF and SKAT have some power increase with the increased genetic effect, but still have lower power than CAR. As also shown in Figure 2.3, CAR.FIX and CAR.SUP have very similar performance. Therefore, for simplicity, the further analysis is performed based on CAR.FIX, in which we fix the nuisance parameter γ as the overall mean of the correlation matrix of SNVs.

Next, we considered a scenario where genetic heterogeneity exist among subgroups (e.g., ethnic groups). Under such case, individuals within a group had homogeneous genetic effects, but the genetic effects among groups were different. In this simulation, we partitioned the total number of individuals of 500 into 8 groups, with the number of individuals in each groups being 200, 100, 50, 25, 40, 35, 45 and 5, respectively. Within each group, we used the following model to simulate the phenotypes, so that the effect sizes were different for different groups.

$$y_i = \sum_{k=1}^{p} w_k^* g_{i,k} Z_k + \varepsilon_i, \quad 1 \leq i \leq n. \tag{2.9}$$

The only difference in model (2.9) from model (2.8) in the main text is that in model (2.9),

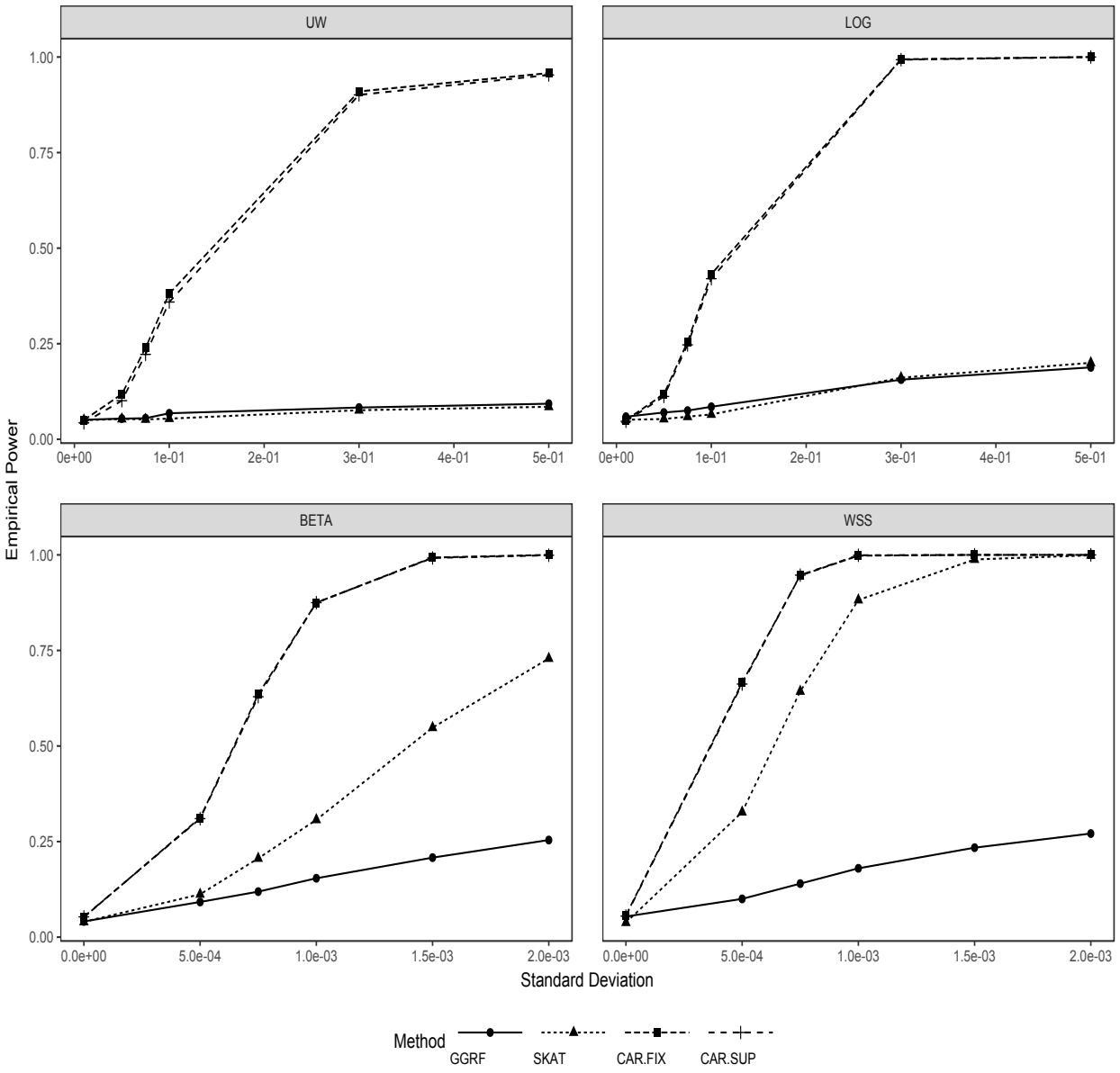

Figure 2.3: Empirical Power Comparison of CAR, SKAT and GGRF by varying the levels of genetic heterogeneity among individuals under four different weights. The x-axis is the effect size σ_Z, used in the simulation. The effect sizes are chosen as 0.01, 0.05, 0.075, 0.1, 0.3, 0.5 for the UW and LOG weights, and the effect sizes are set as 0.0005, 0.00075, 0.001, 0.0015, 0.002 for the BETA and WSS weights.

Z_k does not depend on the individual i and we also assume that $Z_1, \ldots, Z_n \sim \mathcal{N}(0, \sigma_Z^2)$ so that all the individual have the same genetic effect. We set the effect sizes σ_Z as 0.0001, 0.0005, 0.002, 0.001, 0.003, 0.0001, 0.0005 and 0.002 for the eight groups for the BETA and WSS weights, and 0.01, 0.05, 0.2, 0.1, 0.3, 0.01, 0.05 and 0.2 for the UW and LOG weights.

Table 2.2 summarizes the results of the simulation. Consistent with previous findings, CAR has higher power than the other two methods for all four scenarios. SKAT has good power under the WSS weight, but has reduced power under the other weights. While all three methods have low power when common variants play an important role (i.e., UW and LOG), CAR still performs better than SKAT and GGRF.

Table 2.2: Empirical power comparison of GGRF [41], SKAT [78] and CAR based on 1,000 Monte Carlo replicates. In the simulation, we simulated 8 different subgroups with $\sigma_Z = 0.0001, 0.0005, 0.002, 0.001, 0.003, 0.0001, 0.0005, 0.002$ for BETA and WSS; with $\sigma_Z = 0.01, 0.05, 0.2, 0.1, 0.3, 0.01, 0.05, 0.2$ for UW and LOG. The number in the parenthesis is the standard deviation.

Methods	UW	BETA	WSS	LOG
GGRF	0.298(1.505E-02)	0.180(1.243E-02)	0.189(1.226E-02)	0.148(1.140E-02)
SKAT	0.328(1.497E-02)	0.454(1.622E-02)	0.805(1.265E-02)	0.210(1.287E-02)
CAR	0.378(1.548E-02)	0.817(1.248E-02)	0.952(0.670E-02)	0.445(1.611E-02)

Since rare mutation occurs in a small number of individuals and could have a larger effect than common variants, we also modified the effect sizes of the eight groups so that groups with a small number of individuals tended to have higher effect sizes. Specifically, we set σ_Z as 0.0001, 0.0001, 0.0002, 0.005, 0.0003, 0.0005, 0.0001 and 0.005 for the eight groups for the BETA and WSS weights, and 0.01, 0.01, 0.02, 0.5, 0.03, 0.05, 0.01 and 0.5 for the UW and LOG weights. Under such setting, the groups with the sizes of 25 and 5 had the highest effect sizes.

From Table 2.3, we find that all three methods have some power losses than the previous case, but the conclusion is similar to the previous case. CAR still outperforms the other two

under all four scenarios. SKAT performs well when WSS is used, while GGRF has a good performance under the UW weight.

Table 2.3: Empirical power comparison of GGRF [41], SKAT [78] and CAR based on 1,000 Monte Carlo replicates. In the simulation, we simulated 8 different subgroups with $\sigma_Z = 0.0001, 0.0001, 0.0002, 0.005, 0.0003, 0.0005, 0.0001, 0.005$ for BETA and WSS; with $\sigma_Z = 0.01, 0.01, 0.02, 0.5, 0.03, 0.05, 0.01, 0.5$ for UW and LOG. The number in the parenthesis is the standard deviation.

Methods	UW	BETA	WSS	LOG
GGRF	0.201(1.263E-02)	0.149(1.093E-02)	0.154(1.161E-02)	0.115(1.024E-02)
SKAT	0.217(1.288E-02)	0.374(1.534E-02)	0.618(1.545E-02)	0.149(1.096E-02)
CAR	0.382(1.576E-02)	0.672(1.472E-02)	0.769(1.342E-02)	0.428(1.544E-02)

2.3.2 Simulation II: Various Causal SNV Rates

Since the percentage of causal SNVs in a genetic region (e.g., a gene) also have impact on the power of the association test, in simulation II, we examined the performances of GGRF, SKAT and CAR by varying the percentage of causal SNVs. Similar as simulation I, the phenotypes were simulated by using the simulation model (2.8). In this simulation, we varied the percentages of causal SNPs and investigated the effect of 50%, 40%, 30%, 20%, 10%, 5% and 1% causal SNV rates on the power performance of the three methods.

Figures 2.4 and 2.5 illustrate the empirical power of GGRF, SKAT and CAR under different causal SNV rates and four weights. Figure 2.4 summarizes the results under the WSS and BETA wights, i.e the weights focusing more on rare variants. Overall, the CAR model outperforms the other two methods. As the causal SNV rates increase, the empirical power of CAR increases significantly. The similar trend can also be found for SKAT, especially when the WSS weights are used. For the BETA weight, SKAT also attain high empirical power but not as high as that of CAR. GGRF has lower empirical power as compared to CAR and SKAT and its empirical power increases slowly with the increase of causal SNV

rates. We also find that under the WSS weight, both SKAT and CAR attain decent performance when the genetic causes are heterogeneous (i.e., increasing σ_Z) and the causal SNV rates are moderately high.

Figure 2.5 summarizes the results under the LOG and UW weights, i.e the weights focusing more on the effects of common variants. Same as the case of rare variants, the CAR model performs the best for all the scenarios. Even the empirical power of CAR under the LOG and UW weights is not high as those under the BETA and WSS weights, it can still reach high power when the causal SNV rate is high. The CAR model could also gain more substantial power than the other two methods with the increased levels of genetic heterogeneity (i.e., increased σ_Z). Neither SKAT nor GGRF performs as well as they do under the WSS or BETA weights, even with high causal SNV rate and high genetic effects. As we can see from Figure 2.5, if the common variants play an important role in disease phenotypes and have heterogeneous effects, both SKAT and GGRF suffer from extreme power loss, while the CAR model could still obtain moderate or high power.

2.3.3 Simulation III: Misspecification of Weights

In practice, we do not know whether common variants or rare variants play a more important role in the disease process. Since the performance of GGRF, SKAT and CAR depend on the prespecified weights, it is important to investigate whether these models could still obtain reasonable power when the weights are misspecified. We used similar simulation settings as before except that we randomly selected 50 SNV as the causal variants in this particular simulation.

We first focused on rare variants, and simulated phenotypes using either the WSS weight or the BETA weight. When we applied GGRF, SKAT and CAR to the simulation data, all

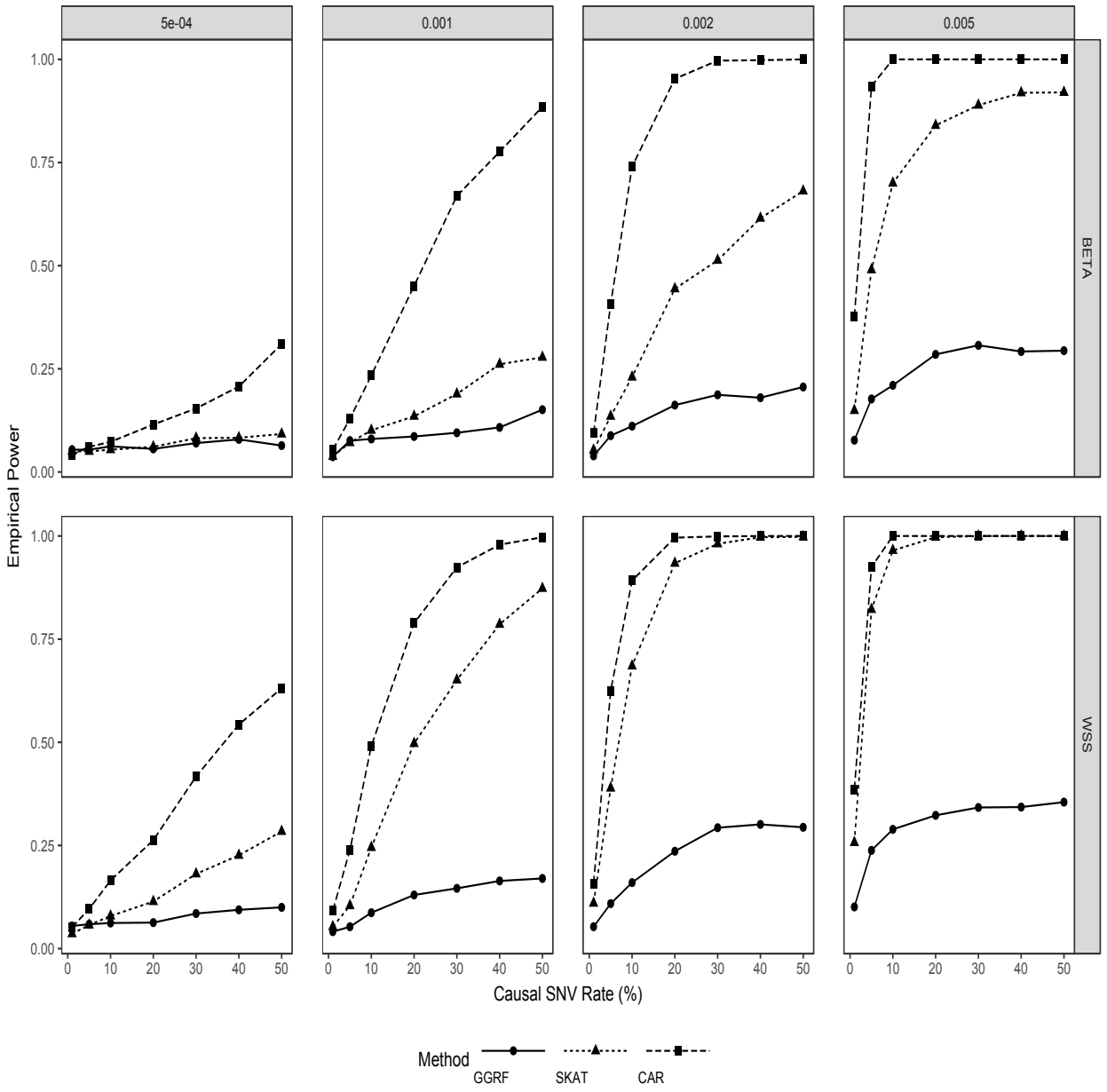

Figure 2.4: Comparison of empirical power of CAR, SKAT and GGRF with different causal SNV rates under the BETA weight and the WSS weight. The standard deviation σ_Z used in the simulation is gradually increased. $\sigma_Z = 0.0005, 0.001, 0.002, 0.005$, respectively in each column from left to right.

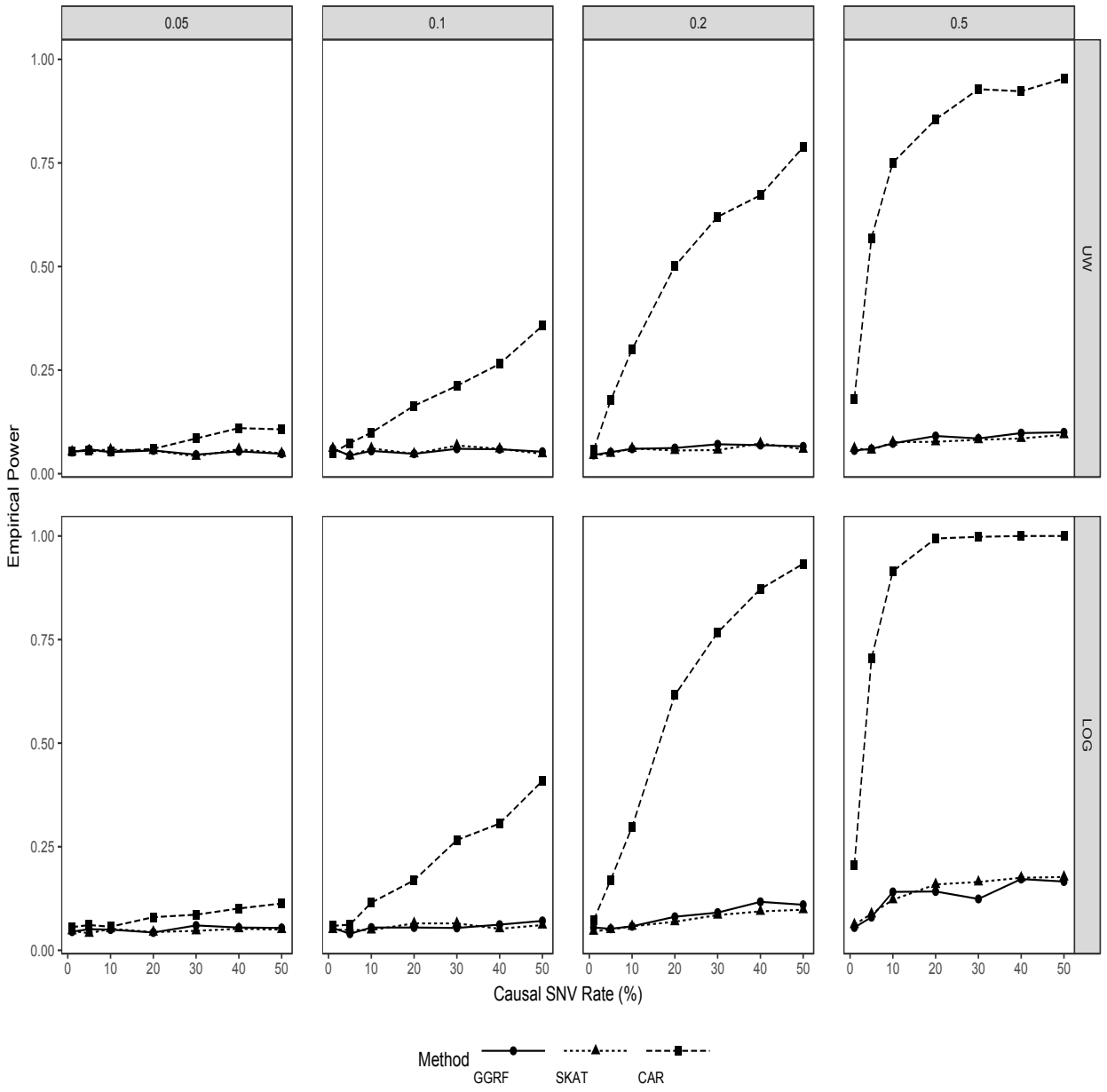

Figure 2.5: Comparison of empirical power of CAR, SKAT and GGRF with different causal SNV rates under the UW weight and the LOG weight. The standard deviation σ_Z used in the simulation is gradually increased. $\sigma_Z = 0.05, 0.1, 0.2, 0.5$ respectively in each column from left to right.

the weights (UW, BETA, WSS and LOG) were used so that we were able to evaluate their performances under both misspecified and correctly specified weights.

Figure 2.6 summarizes the results when the underlying weights in the simulation are WSS and BETA. From Figure 2.6, since WSS put extremely high weights to the rare variants, all the three methods attain high empirical power when the weights are specified as WSS or BETA. As expected, all the methods have the highest empirical power when the weight function is correctly specified (i.e., WSS). We also find that neither SKAT nor GGRF performs well when the weight is misspecified as UW or LOG. On the other hand, as we can see from the figures, as long as there is genetic heterogeneity, CAR has a good power performance even though the weight is misspecified.

Same conclusions hold for BETA. Both GGRF and CAR will obtain highest power when the BETA weight is applied, while SKAT reaches highest power when WSS is used. CAR outperforms SKAT and GGRF in all cases. With the increased genetic heterogeneity, CAR remains high power even with misspecified weights. To conclude, in the case when the disease is mainly caused by rare variants, both SKAT and CAR can have high power, but for SKAT, we need to use the right weights (BETA or WSS).

Next, we focused on common variants and simulated phenotypes using UW and LOG. Figure 2.7 summarizes the results based on UW and LOG. In the first row, the true weight used in the simulation is UW. As expected, CAR attains high power when the specified weight functions focus on common variants (i.e., UW and LOG). With the increased heterogeneity, CAR can obtain high power with the LOG or UW weights. However, it could suffer from power loss when the weight is misspecified (i.e., WSS an BETA). Overall, CAR outperforms SKAT and GGRF, even when the weight is misspecified.

The same conclusion holds for the LOG weight. Since LOG also puts some weight on

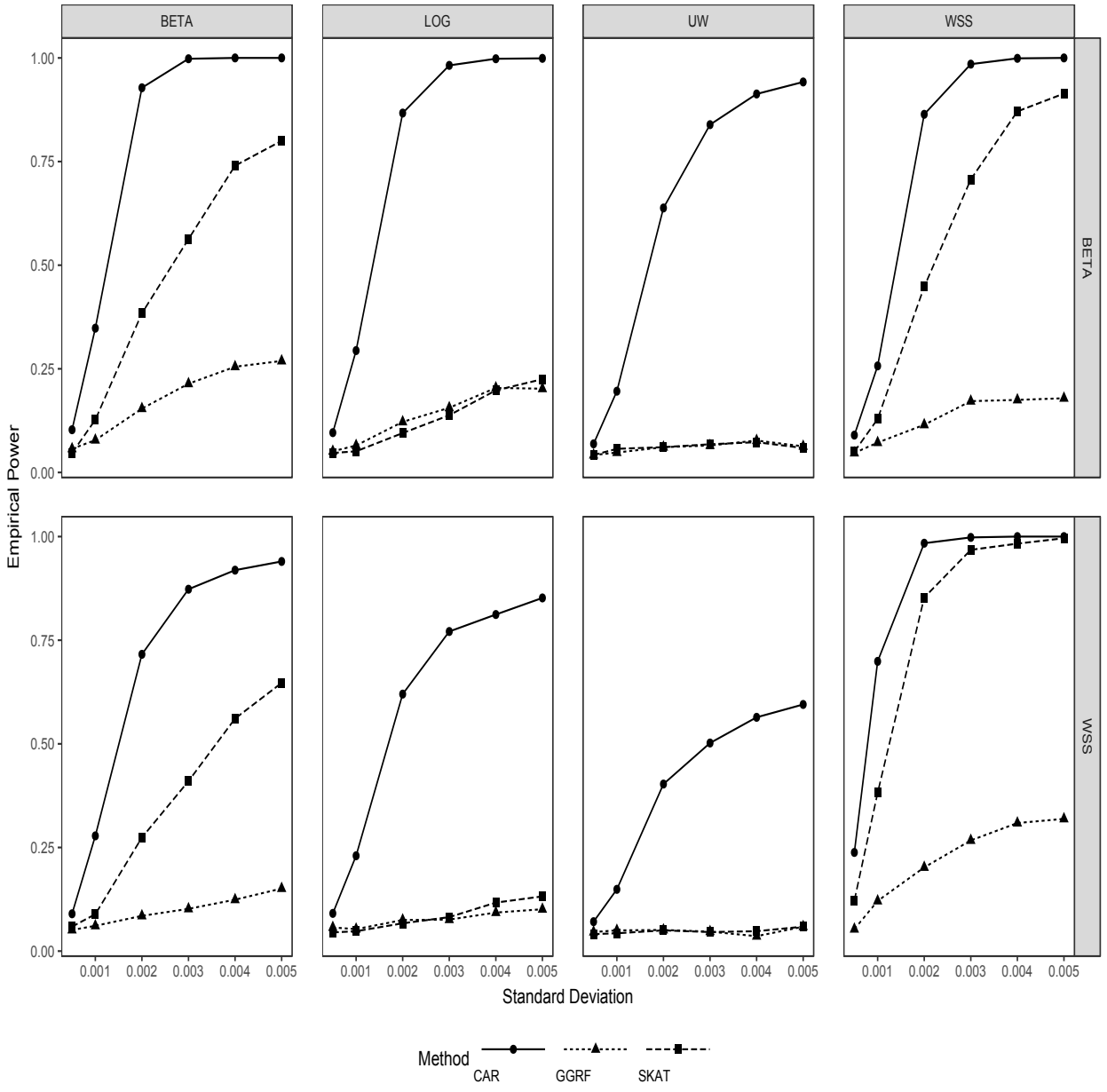

Figure 2.6: Comparison of empirical power of CAR, SKAT and GGRF with misspecified weights when the true weight is BETA and WSS. In each column from left to right, we used BETA, LOG, UW and WSS weight, respectively.

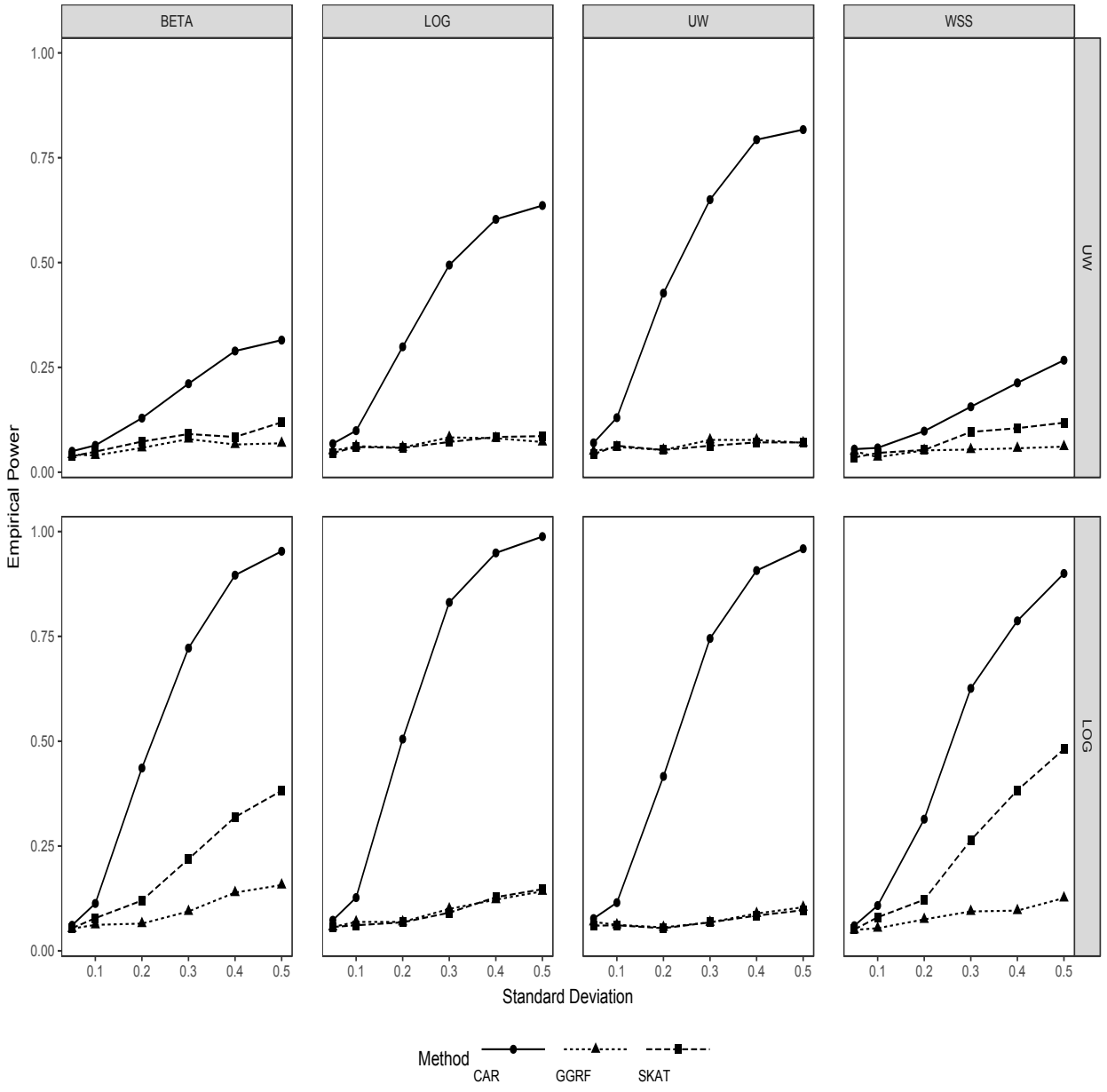

Figure 2.7: Comparison of empirical power of CAR, SKAT and GGRF with misspecified weights when the true weight is UW and LOG. In each column from left to right, we used BETA, LOG, UW and WSS weight, respectively.

rare variants, we find that SKAT has a good performance as compared with its performance under the UW weight. However, SKAT attains highest power by using the WSS weight. This can be explained by the fact that WSS can help to catch the heterogeneous genetic effect. As expected, CAR attains the highest power when the true weight (i.e., LOG) is specified. When LOG is the underlying weight, we can see from Figure 2.7 that there is no significant difference of the four weights used in the CAR model. Overall, GGRF has lower power than CAR and SKAT.

2.4 Real Data Applications

We applied our method to the whole genome sequencing data from Alzheimer's Disease Neuroimaging Initiative (ADNI) and performed a genome-wide gene based association analysis. A total of 808 samples at the screening and baseline of the ADNI1 and ADNI2 studies have the whole genome sequencing data, from which we extracted 21069 genes based on the GRCh 37 assembly. ADNI also provides pre-calculated volumes of cortical regions. We chose four of them as the phenotypes of interests, which are hippocampus, entorhinal, whole brain and ventricles. The motivation of choosing these four volumes as the phenotypes is from previous biological findings. More specifically, the hippocampus, a brain area playing an important role in learning and memory, is especially vulnerable to damage at early stages of Alzheimer's disease (AD) [51]. The volume of hippocampus changes over time and could have a large impact on Alzheimer's disease [61]. The entorhinal cortex is also crucial in declarative memories. In mild AD patients, the loss in the entorhinal volume is evident. The entorhinal cortex is also highly correlated with the severity of the disease [36]. Similarly, the whole brain volume decreases significantly in patients with AD [67]. The brain ventricles also play

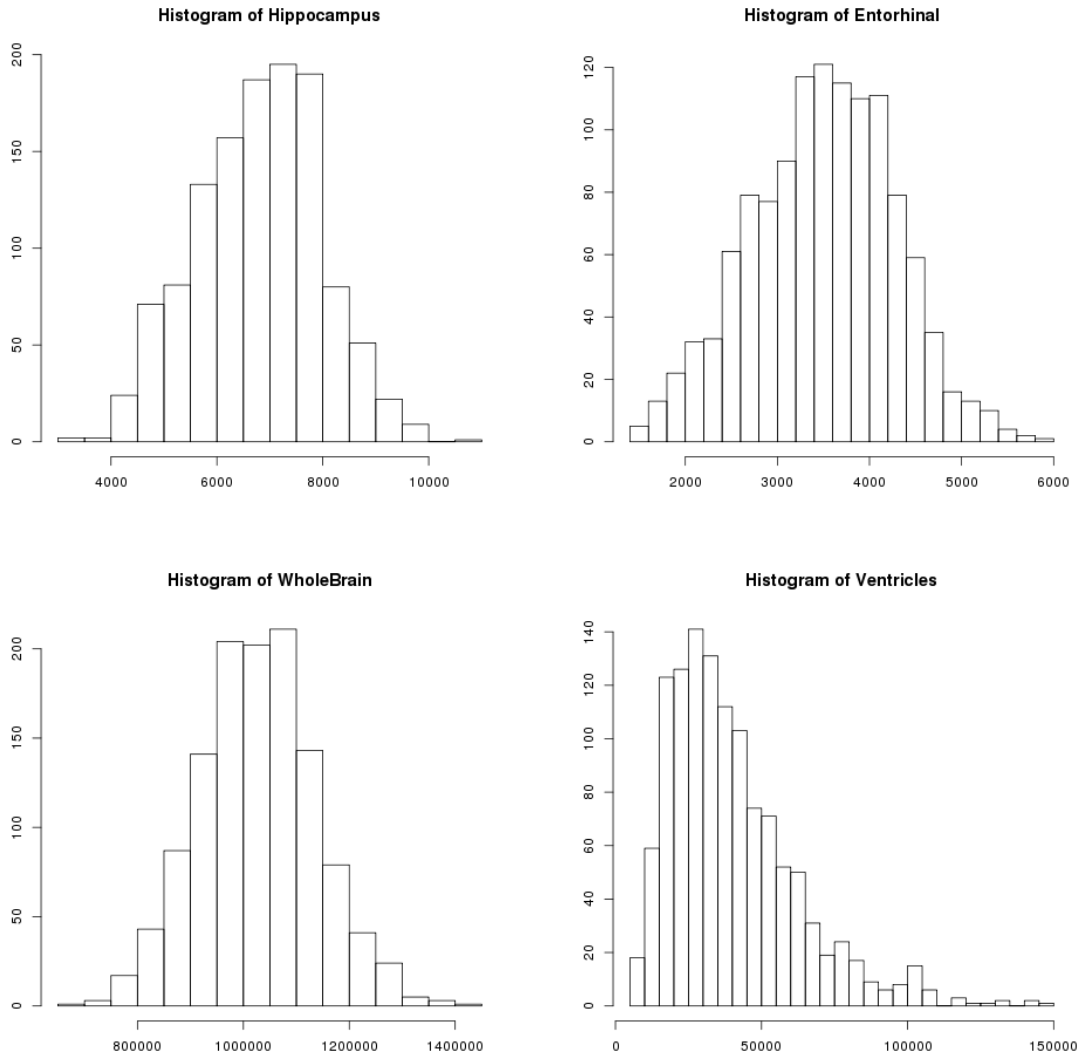

Figure 2.8: Histograms of the Phenotypes used for Real Data Analyses

an important role in AD and it is well known that ventricular volume is significantly higher in AD patients [19].

Figure 2.8 plots the histograms of the four phenotypes used in the analysis. As we can see from the histograms, hippocampus, entorhinal and whole brain are nearly normally distributed. For Ventricles, we first applied a normal quantile transformation to the phenotype and then applied our method.

We restricted our analyses to individuals with caucasian ancestry due to the small sample

size of non-caucasion samples and the issue of population stratification. In the analysis, age, gender, years of education, marriage status, and $APoE\varepsilon 4$ were used as covariates. After removing individuals with missing phenotypes or covariates, a total of 588 subjects remained for the analysis. In this analysis, we also compared our method with SKAT and GGRF. The BETA weight was chosen for all the three methods, and the linear kernel was used in SKAT.

Table 2.4 summarizes the top 10 genes (based on the p-values) associated with hippocampus found by the three methods (CAR, GGRF and SKAT). As we can see from the table, the results of GGRF and SKAT are similar and are different from that of CAR. This could be due to the fact that both methods are similar in terms of assuming genetic homogeneity, while CAR is developed based on a different model, which accounts for genetic heterogeneity. Nonetheless, all the three methods found gene *RPL27* on chromosome 17 associated with hippocampus. In addition to hippocampus, Appendix A also provides results of the top 10 genes related to entorhinal, ventricle and whole brain. Given the limited sample size of the study, none of the association reached statistical significance after multiple-testing adjustment. Nevertheless, few genes (e.g., *RPL27*) have been identified by all three methods, which might worth further investigation. In addition, some of the genes (e.g. *LINC01449*) was detected by the CAR model only, which could also be further evaluated for potential genetic heterogeneity.

Table 2.4: Top 10 hippocampus-associated genes detected by GGRF, SKAT and CAR.

CAR			GGRF			SKAT		
CHR	Gene Name	*P*-value	CHR	Gene Name	*P*-value	CHR	Gene Name	*P*-value
7	*LINC01449*	4.98E-05	11	*CDKN1C*	6.37E-05	17	*RPL27*	4.04E-05
17	*RGS9*	8.09E-05	17	*RPL27*	7.87E-05	17	*IFI35*	6.15E-05
10	*MAPK8*	8.79E-05	2	*LOC400997*	1.35E-04	2	*LOC400997*	1.57E-04
17	*RPL27*	8.81E-05	17	*IFI35*	1.44E-04	4	*MMRN1*	2.03E-04
2	*WTH3DI*	1.21E-04	1	*AJAP1*	1.98E-04	17	*RUNDC1*	2.53E-04
18	*PQLC1*	1.60E-04	12	*GRIN2B*	2.83E-04	12	*CCDC59*	3.56E-04
1	*NTNG1*	1.71E-04	19	*NLRP11*	3.03E-04	1	*AK4*	3.61E-04
14	*PSMC1*	1.81E-04	8	*EFCAB1*	4.10E-04	19	*ISYNA1*	3.99E-04
15	*SECISBP2L*	2.88E-04	1	*AK4*	4.69E-04	10	*C10orf107*	4.48E-04
1	*F5*	2.93E-04	1	*MROH9*	5.43E-04	2	*LOC101929260*	5.29E-04

2.5 Discussion

We have proposed a conditional autoregressive model for genetic association analysis of sequencing data. Our simulations show that the CAR model can obtain high power under the scenario (i) when rare variants are related to the phenotypes and (ii) when genetic variants have different genetic effects among individuals or subgroups of individuals. Moreover, we derive the exact form of the test statistic, which makes the method computationally efficient for large-scale sequencing data analysis. Unlike SKAT, which uses the asymptotic distribution for its test statistic, the exact distribution of the CAR test statistic under the null hypothesis is not conservative. Therefore, no additional small sample size adjustment is needed for the CAR model.

In our simulations, γ is chosen as the average correlation of the genetic correlation coefficients among all individuals. Alternatively, we could also use the average of pairwise linkage disequilibrium (LD) correlation coefficient. Simulation results find that there are no substantial differences between two values (results not shown). However, calculating LD correlation coefficient is more time consuming than estimating the traditional correlation coefficient. It is also interesting to consider the model when $\gamma = 1$, i.e. assuming the genetic random effects are highly correlated. As mentioned in Rue and Held (2005) [58], when $\gamma = 1$, the genetic random effect is then modeled by an intrinsic random field. In this case, the joint distribution of the genetic random effects is not a proper distribution so that traditional frequentist approach may not work well. One potential solution is to use Bayesian method as proposed in Rue and Held (2005) [58].

The work introduced in this paper focuses on the continuous phenotype with normal distribution. Nevertheless, the model can be extended to the case when the phenotypes

follow distributions in the exponential family by using the generalized linear mixed model (GLMM). One potential challenge of extending the CAR model to GLMM is that the test statistic and the likelihood function may not have closed form, which requires alternative approach (e.g. Monte Carlo method) to numerically estimate both of them. This is a future work worth further study.

Chapter 3

A Kernel-Based Neural Network for High-dimensional Genetic Risk Prediction Analysis

3.1 Introduction

Neural-network-based (NN) methods, such as deep learning, has made impressive advances in areas, such as imaging recognition and nature language precessing [38]. There is an increasing interest in using NN methods in genomic studies, especially in the field of regulatory genomic, where NN methods and software have been developed to improve the capacity of modeling DNA and RNA targets of regulatory proteins [53].

Even though NN methods also hold great promise in revealing the complex relationship between human diseases and genetic variants, few research has been done in this field. One of the key challenges is the massive genomic data, which often includes millions of genetic variants. Directly applying the classic NN methods to such a large number of genetic variants requires the estimation of millions of parameters, which is computationally difficult and likely causes serious over-fitting issue. Instead of modeling individual effects of millions of genetic variants, we can model the genetic effect of these variants as a random effect by adopting a

linear mixed model (LMM) framework. LMM has been widely used in genetic data analysis, starting from complex segregation analysis [50] to recent genome-wide complex trait analysis [79] and set-based association analysis[78].

Based on LMM, we propose a method, referred to as the kernel neural network (KNN), for high-dimensional genetic data analysis. As we show in the later section of the paper, under certain conditions, KNN is equivalent to a nonlinear mixed effect model so that a linear mixed model is a special case of KNN. In addition to its ability to handle high-dimensional data, KNN inherits properties from the neural network, which allows the method to consider nonlinear and nonadditive effects. Due to the complexity of such method, it is difficult to estimate the parameters from KNN using the conventional approach. Moreover, the parameters in KNN are unidentifiable. To address such issue, instead of using the restricted maximum likelihood estimator (REML) [15], we use the minimum quadratic unbiased estimator (MINQUE) proposed by Rao (1970, 1971, 1972)[55, 56, 57] to estimate the "variance components". We show both theoretically and empirically that MINQUE has nice properties.

The remaining chapter is arranged as follows: Section 2 provides the description of the KNN model and the MINQUE estimation procedure; Section 3 provides theoretical comparison of prediction performance between KNN and LMM and Section 4 discusses the inclusion of fixed effects; results from simulation study and a real data application are presented in Sections 5 and 6, respectively. Before we proceeding to the main text, we first summarize notations that will be frequently used in this paper.

Notations: Throughout the chapter, capital bold italic letters $\boldsymbol{A}, \boldsymbol{B}, \ldots, \boldsymbol{\Gamma}, \boldsymbol{\Theta}, \ldots$ will be used to denote matrices; small bold italic letters $\boldsymbol{a}, \boldsymbol{b}, \ldots, \boldsymbol{\alpha}, \boldsymbol{\beta}, \ldots$ will be used to denote vectors and other small letters will be used to denote scalars. $\boldsymbol{I}_n$ will be used to denote an $n \times n$ identity matrix and the symbol "$\precsim$" will be used to denote asymptotically less than.

3.2 Methodologies

Kernel methods are widely used in recent machine learning area due to its capability of capturing nonlinear features from the data so that the prediction error can be diminished. As has been mentioned in Shawe-Taylor et al. (2004)[63], given a kernel and a training set, the kernel matrix acts as an information bottleneck, as all the information available to a kernel algorithm must be extracted from this matrix. On the other hand, linear mixed effect models are also widely used in the area of genetic risk prediction. Therefore, it seems natural to combine these two methods together. A naive way is to change the covariance matrix of the random effect in the linear mixed model to a kernel matrix as was did in [78] and [79]. As we have seen in section 1.3.2 that for a linear mixed model with a kernel matrix as the covariance matrix of the random effect is equivalent to a semiparametric regression model.

Now, we move a little bit further. By extending the Representer Theorem, we will see that a nonlinear mixed effect model is equivalent to a hierarchical linear mixed model. Specifically, consider m i.i.d. random features $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ and the following nonlinear mixed effect model:

$$y_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + f(u_{i1}, \ldots, u_{im}) + \epsilon_i, \quad i = 1, \ldots, n, \tag{3.1}$$

where $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. $\mathcal{N}(0, \phi)$ random variables and $f \in \mathcal{H}_K$ for some RKHS $\mathcal{H}_K$. We consider the following loss function:

$$L(\boldsymbol{\beta}, f) = \frac{1}{2\phi}\mathbb{E}_{\boldsymbol{U}}\left[(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - f(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m))^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - f(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m))\right] + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}_K}. \tag{3.2}$$

The first thing we are going to do is to extend Theorem 1.3.2 to meet with our need.

Theorem 3.2.1 (Extended Representer Theorem)**.** *For a fixed kernel K, let $\mathcal{H}_K$ be the*

corresponding RKHS and let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ *be random variables. Then for a convex function* $L : \mathbb{R}^n \to \mathbb{R}$ *and non-decreasing* $\Omega : \mathbb{R} \to \mathbb{R}$*, if the optimization problem can be expressed as*

$$J(f^*) = \min_{f \in \mathcal{H}_K} J(f) = \min_{f \in \mathcal{H}_K} \left\{ \mathbb{E}\left[L(f(\boldsymbol{X}_1), \ldots, f(\boldsymbol{X}_n))\right] + \Omega\left(\|f\|^2_{\mathcal{H}_K}\right) \right\},$$

then the solution can be expressed as

$$f^* = \sum_{i=1}^{n} \alpha_i \mathbb{E}\left[K(\cdot, \boldsymbol{X}_i)\right]$$

Proof. Consider the subspace $\mathcal{S} \subset \mathcal{H}_K$ given by

$$\mathcal{S} = \operatorname{span}\left\{\mathbb{E}\left[K(\cdot, \boldsymbol{X}_i)\right] : i = 1, \ldots, n\right\}.$$

Since $\mathcal{S}$ is a finite dimensional space, it is therefore closed. The projection theorem then implies $\mathcal{H}_K = \mathcal{S} \oplus \mathcal{S}^{\perp}$, i.e., for every $f \in \mathcal{H}_K$, we can uniquely write $f = f_{\mathcal{S}} + f_{\perp}$, where $f_{\mathcal{S}} \in \mathcal{S}$ and $f_{\perp} \in \mathcal{S}^{\perp}$. Noting that $\langle f_{\perp}, \mathbb{E}\left[K(\cdot, \boldsymbol{X}_i)\right]\rangle = 0$ for each i, the reproducing property implies

$$\begin{aligned}
\mathbb{E}\left[f(\boldsymbol{X}_i)\right] &= \mathbb{E}\left[\langle f, K(\cdot, \boldsymbol{X}_i)\rangle\right] = \langle f, \mathbb{E}[K(\cdot, \boldsymbol{X}_i)]\rangle \\
&= \langle f_{\mathcal{S}}, \mathbb{E}\left[K(\cdot, \boldsymbol{X}_i)\right]\rangle = \mathbb{E}\left[f_{\mathcal{S}}(\boldsymbol{X}_i)\right] \\
&= f_{\mathcal{S}}(\boldsymbol{X}_i)
\end{aligned}$$

On the other hand, Pythagoras Theorem implies that

$$\|f\|^2_{\mathcal{H}_K} = \|f_{\mathcal{S}}\|^2_{\mathcal{H}_K} + \|f_{\perp}\|_{\mathcal{H}_K} \geq \|f_{\mathcal{S}}\|^2_{\mathcal{H}_K}.$$

Since Ω is non-decreasing, we can know that $\Omega\left(\|f\|^2_{\mathcal{H}_K}\right) \geq \Omega\left(\|f_{\mathcal{S}}\|^2_{\mathcal{H}_K}\right)$. Then it follows from Jensen's inequality that

$$\begin{aligned} J(f) &= \mathbb{E}\left[L(f(\boldsymbol{X}_1), \ldots, f(\boldsymbol{X}_n))\right] + \lambda\Omega\left(\|f\|^2_{\mathcal{H}_K}\right) \\ &\geq L\left(\mathbb{E}[f(\boldsymbol{X}_1)], \ldots, \mathbb{E}\left[f(\boldsymbol{X}_n)\right]\right) + \lambda\Omega\left(\|f_{\mathcal{S}}\|^2_{\mathcal{H}_K}\right) \\ &= L\left(f_{\mathcal{S}}(\boldsymbol{X}_1), \ldots, f_{\mathcal{S}}(\boldsymbol{X}_n)\right) + \lambda\Omega\left(\|f_{\mathcal{S}}\|^2_{\mathcal{H}_K}\right) \\ &= \mathbb{E}\left[L\left(f_{\mathcal{S}}(\boldsymbol{X}_1), \ldots, f_{\mathcal{S}}(\boldsymbol{X}_n)\right)\right] + \lambda\Omega\left(\|f_{\mathcal{S}}\|^2_{\mathcal{H}_K}\right). \end{aligned}$$

Hence, $J(f)$ is minimized if $f \in \mathcal{S}$ and we can express the minimizer as

$$f^*(\cdot) = \sum_{i=1}^{n} \alpha_i \mathbb{E}[K(\cdot, \boldsymbol{X}_i)].$$

□

Let $\boldsymbol{W}_i$ be the ith row of the random matrix $\boldsymbol{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_m]$, based on Theorem 3.2.1, the general solution for $f(\cdot)$ in (3.2) can be expressed as

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i \mathbb{E}\left[K(\cdot, \boldsymbol{W}_i)\right],$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^T$ are unknown parameters. Substituting this back into (3.2), we

have

$$
\begin{aligned}
J(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{1}{2\phi}\mathbb{E}_{\boldsymbol{U}}\left[(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha})\right] \\
&\qquad + \frac{\lambda}{2}\left\langle \sum_{i=1}^{n}\alpha_i\mathbb{E}[K(\cdot,\boldsymbol{W}_i)], \sum_{j=1}^{n}\alpha_j\mathbb{E}\left[K(\cdot,\boldsymbol{W}_j)\right]\right\rangle_{\mathcal{H}_K} \\
&= \frac{1}{2\phi}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha}) \\
&\qquad + \frac{\lambda}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\mathbb{E}\left[\left\langle K(\cdot,\boldsymbol{W}_i), K(\cdot,\boldsymbol{W}_j)\right\rangle_{\mathcal{H}_K}\right] \\
&= \frac{1}{2\phi}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha})^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha}) + \frac{\lambda}{2}\boldsymbol{\alpha}^T\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha},
\end{aligned}
$$

where $\boldsymbol{K}(\boldsymbol{U})$ is an $n\times n$ matrix whose (i,j)th element if $K(\boldsymbol{W}_i,\boldsymbol{W}_j)$. Differentiating $J(\boldsymbol{\alpha},\boldsymbol{\beta})$ with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we get

$$
\begin{aligned}
\frac{\partial J}{\partial \boldsymbol{\beta}} &= -\phi^{-1}\boldsymbol{X}^T\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\mathbb{E}_{\boldsymbol{U}}\left[\boldsymbol{K}(\boldsymbol{U})\right]\boldsymbol{\alpha}\right) \\
\frac{\partial J}{\partial \boldsymbol{\alpha}} &= -\phi^{-1}\mathbb{E}_{\boldsymbol{U}}\left[\boldsymbol{K}(\boldsymbol{U})\right]^T\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}-\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha}\right) + \lambda\mathbb{E}_{\boldsymbol{U}}\left[\boldsymbol{K}(\boldsymbol{U})\right]\boldsymbol{\alpha}
\end{aligned}
$$

By setting the second equation to 0, we get by the symmetry and positive definiteness of $\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]$,

$$
\begin{aligned}
&\lambda\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha} + \phi^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]^T\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha} = \phi^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]^T(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) \\
\Rightarrow &\boldsymbol{\alpha} = (\lambda\phi)^{-1}\left(\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\right)^{-1}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}).
\end{aligned}
$$

Similarly, for the first equation, we have

$$\begin{aligned}
\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} &= \boldsymbol{X}^T(\boldsymbol{y} - \mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\boldsymbol{\alpha}) \\
&= \boldsymbol{X}^T\left(\boldsymbol{y} - (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})](\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})])^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\right) \\
&= \boldsymbol{X}^T(\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})])^{-1}\boldsymbol{y} \\
&\quad + (\lambda\phi)^{-1}\boldsymbol{X}^T\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})](\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})])^{-1}\boldsymbol{X}\boldsymbol{\beta},
\end{aligned}$$

which implies that

$$\boldsymbol{X}^T\left(\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\right)^{-1}\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{X}^T\left(\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\right)^{-1}\boldsymbol{y}.$$

Hence, we get

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= \left[\boldsymbol{X}^T\left(\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\right)^{-1}\boldsymbol{X}\right]^{-1}\boldsymbol{X}^T\left(\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\right)^{-1}\boldsymbol{y} \\
\hat{\boldsymbol{\alpha}} &= (\lambda\phi)^{-1}\left(\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\right)^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}),
\end{aligned}$$

so that

$$\hat{\boldsymbol{f}} = \mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\hat{\boldsymbol{\alpha}} = (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\left(\boldsymbol{I}_n + (\lambda\phi)^{-1}\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\right)^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

Through some simple calculations, it can be seen that $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\alpha}}$ is the solution to the following equation:

$$\begin{bmatrix}
\boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{X} & \boldsymbol{X}^T\boldsymbol{R}^{-1} \\
\boldsymbol{R}^{-1}\boldsymbol{X} & \boldsymbol{R}^{-1} + \tau^{-1}\left(\mathbb{E}_{\boldsymbol{U}}[\boldsymbol{K}(\boldsymbol{U})]\right)^{-1}
\end{bmatrix}
\begin{bmatrix}
\boldsymbol{\beta} \\
\boldsymbol{f}
\end{bmatrix} =
\begin{bmatrix}
\boldsymbol{X}^T\boldsymbol{R}^{-1}\boldsymbol{y} \\
\boldsymbol{R}^{-1}\boldsymbol{y}
\end{bmatrix},$$

where $\tau = \lambda\phi$, $\boldsymbol{R} = \phi \boldsymbol{I}_n$ and this equation corresponds exactly to the Henderson's mixed model equation of the linear mixed model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f} + \boldsymbol{\epsilon}, \tag{3.3}$$

where $\mathbb{E}[\boldsymbol{f}] = \boldsymbol{0}$, $\mathrm{Var}[\boldsymbol{f}] = \tau \mathbb{E}_{\boldsymbol{U}}\left[\boldsymbol{K}(\boldsymbol{U})\right]$ and $\mathbb{E}[\boldsymbol{\epsilon}] = \boldsymbol{0}$, $\mathrm{Var}[\boldsymbol{\epsilon}] = \phi \boldsymbol{I}_n$. This is equivalent to the following hierarchical model. In the model, we consider a more general setup that the matrix $\boldsymbol{K}(\boldsymbol{U})$ is a positive linear combination of different kernel matrices.

$$\begin{aligned} \boldsymbol{y}|\boldsymbol{f} &\sim \mathcal{N}_n\left(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{f}, \phi \boldsymbol{I}_n\right) \\ \boldsymbol{f}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m &\sim \mathcal{N}_n\left(\boldsymbol{0}, \sum_{j=1}^{J} \tau_j \boldsymbol{K}_j(\boldsymbol{U})\right) \\ \boldsymbol{u}_1, \ldots, \boldsymbol{u}_m &\sim \text{ i.i.d. } \mathcal{N}_n\left(\boldsymbol{0}, \sum_{l=1}^{L} \xi_l \boldsymbol{K}_l(\boldsymbol{G})\right), \end{aligned} \tag{3.4}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m] \in \mathbb{R}^{n \times m}$; $\boldsymbol{K}_j(\boldsymbol{U})$, $j = 1, \ldots, J$ are $n \times n$ kernel matrices constructed based on the latent variables $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ and $\boldsymbol{K}_l(\boldsymbol{G})$, $l = 1, \ldots, L$ are kernel matrices constructed based on the genetic variables. For instance, if we have p genetic variants (p can be greater than n), we can define $\boldsymbol{K}(\boldsymbol{G}) = p^{-1}\boldsymbol{G}\boldsymbol{G}^T$, which is known as the product kernel or linear kernel. To see the connection between model (3.4) and model (3.3), we note that

$$\begin{aligned} \mathbb{E}[\boldsymbol{f}] &= \mathbb{E}_{\boldsymbol{U}}\left(\mathbb{E}[\boldsymbol{f}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]\right) = \boldsymbol{0} \\ \mathrm{Var}[\boldsymbol{f}] &= \mathbb{E}_{\boldsymbol{U}}\left[\mathrm{Var}\left(\boldsymbol{f}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\right)\right] + \mathrm{Var}\left[\mathbb{E}\left(\boldsymbol{f}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\right)\right] \\ &= \mathbb{E}_{\boldsymbol{U}}\left[\sum_{j=1}^{J} \tau_j \boldsymbol{K}_j(\boldsymbol{U})\right] = \sum_{j=1}^{J} \tau_j \mathbb{E}_{\boldsymbol{U}}\left[\boldsymbol{K}_j(\boldsymbol{U})\right]. \end{aligned}$$

This is the same as we obtained from model (3.3).

As an illustration, the basic hierarchical structure of the model can be seen from Figure 3.1. Due to the similarity in the network structure as in the case of popular neural networks, we thus call our model a kernel neural network (KNN). KNN has several nice features. First, by considering genetic effects as random effects, the method can simultaneously deal with millions of variants. This addresses the limitation of the fixed-effect conventional neural network, which is computationally prohibitive on such a large number of variables. Second, by using random genetic effects, the model complexity is also greatly reduced, as we no longer require to estimate a large number of fixed genetic effects. Third, KNN allows for a large number of hidden units without increasing model complexity. Finally, using hidden units, KNN can capture non-linear and non-additive effects, and therefore is able to model complex functions beyond linear models.

In the remaining part of this section and section 3.3, we will focus on the scenario where there is no fixed effects, that is $\boldsymbol{\beta} = \mathbf{0}$. In section 3.4, we will consider the general estimation procedure when $\boldsymbol{\beta} \neq \mathbf{0}$ and we will also calculate the prediction error.

3.2.1 Quadratic Estimators for Variance Components

Popular estimation strategies for variance components in linear models are the maximum likelihood estimator (MLE) and the restricted maximum likelihood estimator (REML) [15]. However, both methods depend on the marginal distribution of $\boldsymbol{y}$. In our KNN model, it is generally difficult to obtain the marginal distribution of $\boldsymbol{y}$, which involves high dimensional integration with respect to $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$. Moreover, the $\boldsymbol{u}_i$'s are embedded in the kernel matrices $\boldsymbol{K}_j(\boldsymbol{U})$, $j = 1, \ldots, J$, which makes the integration even more complicated.

On the other hand, given the KNN model describes above, we can easily obtain the

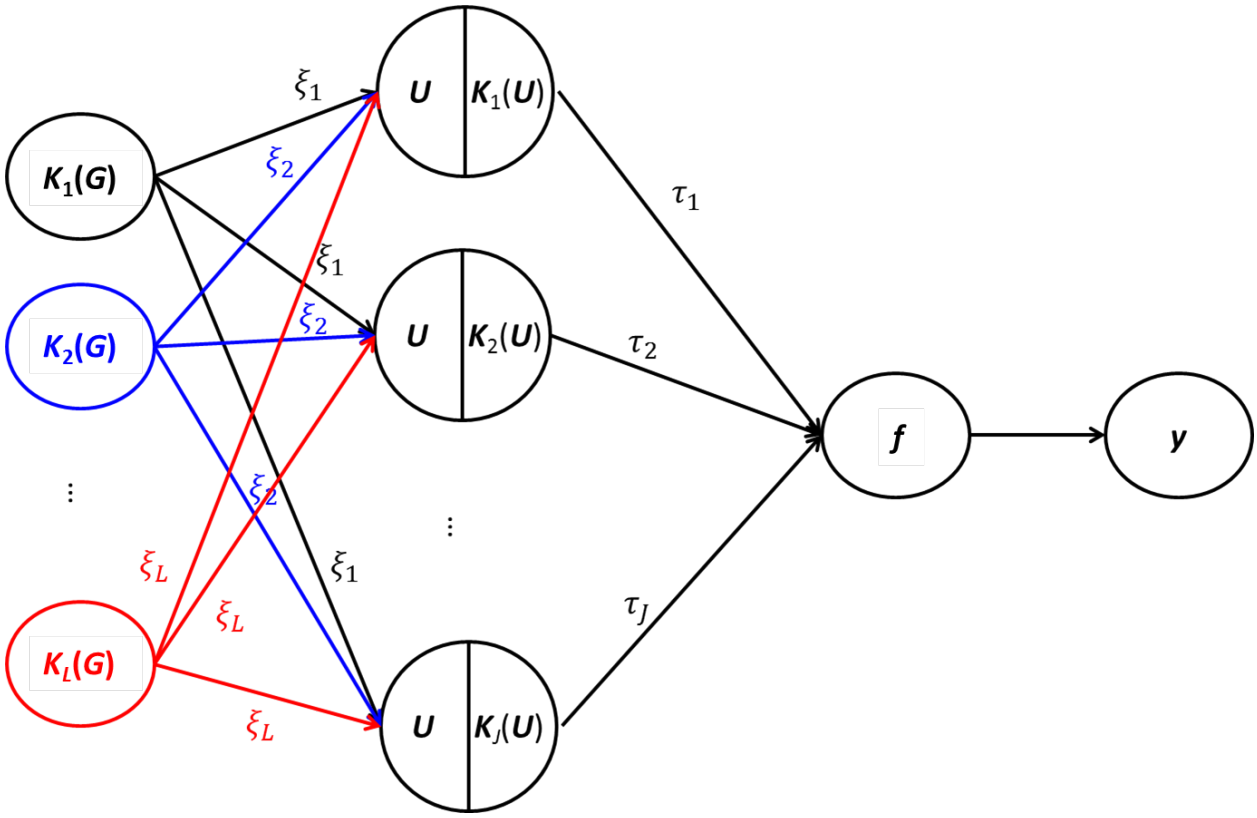

Figure 3.1: An illustration of the hierarchical structure of the kernel neural network model with L input kernel matrices $\boldsymbol{K}_1(\boldsymbol{G}), \ldots, \boldsymbol{K}_L(\boldsymbol{G})$ and J hidden kernel matrices $\boldsymbol{K}_1(\boldsymbol{U}), \ldots, \boldsymbol{K}_J(\boldsymbol{U})$. The output layer is the prediction for the random effect $\boldsymbol{f}$.

conditional distribution of $\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ is given by

$$\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m \sim \mathcal{N}_n \left(\mathbf{0}, \sum_{j=1}^{J} \tau_j \boldsymbol{K}_j(\boldsymbol{U}) + \phi \boldsymbol{I}_n \right).$$

Then the marginal mean and variance of $\boldsymbol{y}$ can be obtained via conditioning arguments:

$$\mathbb{E}[\boldsymbol{y}] = \mathbb{E}\left(\mathbb{E}[\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]\right) = \mathbf{0}. \tag{3.5}$$

$$\begin{aligned}
\mathrm{Var}[\boldsymbol{y}] &= \mathbb{E}\left[\mathrm{Var}(\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)\right] + \mathrm{Var}\left[\mathbb{E}(\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{J} \tau_j \boldsymbol{K}_j(\boldsymbol{U}) + \phi \boldsymbol{I}_n\right] \\
&= \sum_{j=1}^{J} \tau_j \mathbb{E}[\boldsymbol{K}_j(\boldsymbol{U})] + \phi \boldsymbol{I}_n \\
&:= \sum_{j=0}^{J} \tau_j \mathbb{E}[\boldsymbol{K}_j(\boldsymbol{U})],
\end{aligned} \tag{3.6}$$

where $\tau_0 = \phi$ and $\boldsymbol{K}_0(\boldsymbol{U}) = \boldsymbol{I}_n$. Given the marginal mean and covariance matrix, the MInimum Quadratic Unbiased Estimator (MINQUE) proposed by Rao (1970, 1971, 1972)[55, 56, 57] can be used to estimate the variance components. The basic idea of MINQUE is to use a quadratic form $\boldsymbol{y}^T \boldsymbol{\Theta} \boldsymbol{y}$ to estimate a linear combination of variance components. The MINQUE matrix $\boldsymbol{\Theta}$ is obtained by minimizing a suitable matrix norm, which is typically chosen to be the Frobenius norm, of the difference between $\boldsymbol{\Theta}$ and the matrix in the quadratic estimator by assuming that we know the random components in the linear mixed models. The constraint in the optimization is to guarantee the unbiasedness of the estimators. One advantage of MINQUE is that it has a closed form solution provided by Lemma 3.4 in Rao (1971)[56] so that it can be computed efficiently. However, MINQUE can also provide a negative estimator for a single variance component. When this occurs, we simply set the

negative estimators to be zero, as we usually do for MLE or REML of variance components, except for the error variance component. If the MINQUE estimator for the error variance component becomes negative, we project the MINQUE matrix $\boldsymbol{\Theta}$ onto the positive semi-definite cone $\mathcal{S}_n^+$. Specifically, the modified estimator for error variance component becomes

$$\hat{\tau}_0 = \boldsymbol{y}^T \boldsymbol{O} \begin{bmatrix} \max\{\lambda_1, 0\} & & \\ & \ddots & \\ & & \max\{\lambda_n, 0\} \end{bmatrix} \boldsymbol{O}^T \boldsymbol{y},$$

where $\boldsymbol{O}\text{diag}\{\lambda_1, \ldots, \lambda_n\}\boldsymbol{O}^T$ is the eigen-decomposition of $\boldsymbol{\Theta}$.

3.2.2 MINQUE in KNN

For the ease of theoretical justifications, throughout the remaining of the chapter, we illustrate the method with one hidden kernel matrix (i.e. $J = 1$) with the form of $\boldsymbol{K}_{ij}(\boldsymbol{U}) = g\left[\frac{1}{n}\boldsymbol{w}_i^T\boldsymbol{w}_j\right]$, where $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$ are the rows of the matrix $\boldsymbol{U}$ and $g[\boldsymbol{A}]$ means that the function g is applied to each element in the matrix $\boldsymbol{A}$.

We start by considering the simplest case $g(x) = x$, in which case, the hidden kernel matrix becomes $\boldsymbol{K}(\boldsymbol{U}) = \frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T$. In this case, we have an explicit form of the marginal variance for $\boldsymbol{y}$. Since $\boldsymbol{U}\boldsymbol{U}^T \sim \mathcal{W}_n(m, \sum_{l=1}^L \xi_l \boldsymbol{K}_l(\boldsymbol{G}))$, where $\mathcal{W}_n(m, \boldsymbol{A})$ is a Wishart distribution with m degrees of freedom and covariance matrix $\boldsymbol{A}$, we have

$$\boldsymbol{V} := \text{Var}[\boldsymbol{y}] = \tau\mathbb{E}_{\boldsymbol{U}}\left[\boldsymbol{K}(\boldsymbol{U})\right] + \phi\boldsymbol{I} = \sum_{l=1}^{L} \tau\xi_l \boldsymbol{K}_l(\boldsymbol{G}) + \phi\boldsymbol{I}_n. \tag{3.7}$$

From equation (3.7), we can see that there is an identifiability issue if we directly estimate τ and ξ_l's. To solve this issue, we reparameterize the covariance matrix by letting $\theta_l = \tau\xi_l$,

$\theta_0 = \phi$, $\boldsymbol{K}_0(\boldsymbol{G}) = \boldsymbol{I}_n$ and rewrite $\mathrm{Var}[\boldsymbol{y}]$ as

$$\boldsymbol{V} = \sum_{l=0}^{L} \theta_l \boldsymbol{K}_l(\boldsymbol{G}) = \sum_{l=0}^{L} \theta_l \boldsymbol{S}_l(\boldsymbol{G}) \boldsymbol{S}_l^T(\boldsymbol{G}), \tag{3.8}$$

where $\boldsymbol{S}_0(\boldsymbol{G}), \ldots, \boldsymbol{S}_L(\boldsymbol{G})$ are the Cholesky lower triangles for the kernel matrices $\boldsymbol{K}_0(\boldsymbol{G}), \ldots, \boldsymbol{K}_L(\boldsymbol{G})$, respectively. Then the parameters $\theta_0, \ldots, \theta_L$ can be estimated via MINQUE. Specifically,

$$\begin{aligned} \hat{\theta}_0 &= \boldsymbol{y}^T \hat{\boldsymbol{A}}_0 \boldsymbol{y} \\ \hat{\theta}_i &= \boldsymbol{y}^T \boldsymbol{A}_i \boldsymbol{y} \vee 0, \quad i = 1, \ldots, L, \end{aligned}$$

where

$$\boldsymbol{A}_i = \sum_{l=0}^{L} \eta_l \left[\sum_{l=0}^{L} \boldsymbol{K}_l(\boldsymbol{G}) \right]^{-1} \boldsymbol{K}_l(\boldsymbol{G}) \left[\sum_{l=0}^{L} \boldsymbol{K}_l(\boldsymbol{G}) \right]^{-1}, \quad i = 0, \ldots, L,$$

and $\eta_0, \ldots, \eta_L$ are the solutions to $\boldsymbol{\Gamma}\boldsymbol{\eta} = \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is a vector of zero except that the $(i+1)$th elements is 1 and

$$\boldsymbol{\Gamma}_{ij} = \mathrm{tr} \left(\left[\sum_{l=0}^{L} \boldsymbol{K}_l(\boldsymbol{G}) \right]^{-1} \boldsymbol{K}_i(\boldsymbol{G}) \left[\sum_{l=0}^{L} \boldsymbol{K}_l(\boldsymbol{G}) \right]^{-1} \boldsymbol{K}_j(\boldsymbol{G}) \right).$$

Moreover, $\hat{\boldsymbol{A}}_0 = P_{\mathcal{S}_n^+} \boldsymbol{A}_0$ as mentioned above. For a general kernel matrix of the form $\boldsymbol{K}(\boldsymbol{U}) = g\left[\frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T\right]$, we focus on the case where g satisfies the following property, which we called the Generalized Linear Separable Condition:

$$g\left(\sum_{\alpha=1}^{k} c_\alpha x_\alpha \right) = \sum_{\alpha=1}^{k'} \pi_\alpha(c_1, \ldots, c_k) \gamma_\alpha(x_1, \ldots, x_k), \tag{3.9}$$

where $c_1, \ldots, c_k \in \mathbb{R}$ are coefficients and $\pi_1, \ldots, \pi_{k'}, \gamma_1, \ldots, \gamma_{k'}$ are some functions. Exam-

ples of kernel functions satisfying the condition are polynomial kernels. We know that

$$\boldsymbol{K}_{st}(\boldsymbol{U}) = f\left(\frac{\boldsymbol{w}_s^T \boldsymbol{w}_t}{m}\right), \quad s,t = 1,\ldots,n \tag{3.10}$$

and

$$\begin{bmatrix} \boldsymbol{w}_{i1} \\ \boldsymbol{w}_{j1} \end{bmatrix}, \ldots, \begin{bmatrix} \boldsymbol{w}_{im} \\ \boldsymbol{w}_{jm} \end{bmatrix} \sim \text{i.i.d. } \mathcal{N}_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{ii} & \sigma_{ij} \\ \sigma_{ij} & \sigma_{jj} \end{bmatrix}\right), \tag{3.11}$$

where $\sigma_{ii}, \sigma_{jj}, \sigma_{ij}$ are the corresponding elements in $\boldsymbol{\Sigma} = \sum_{l=1}^{L} \xi_l \boldsymbol{K}_l(\boldsymbol{G})$. We can then use Taylor expansion to obtain an approximation of $\mathbb{E}[\boldsymbol{K}_{ij}(\boldsymbol{U})]$, which is given in Lemma 3.2.1. The proof of Lemma 3.2.1 can be found in Appendix B.

Lemma 3.2.1. *Let the random vectors $\boldsymbol{w}_i, \boldsymbol{w}_j \in \mathbb{R}^m$ be as in (3.11) and the $\boldsymbol{K}_{ij}(\boldsymbol{U})$ as defined in (3.10). Then if*

$$\left| g''\left(\lambda\sigma_{ij} + (1-\lambda)\frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{m}\right)\right| \leq M, \quad a.s., \tag{3.12}$$

for some $M > 0$ and all $\lambda \in [0,1]$, we have

$$\mathbb{P}\left(\left|\boldsymbol{K}_{ij}(\boldsymbol{U}) - \hat{\boldsymbol{K}}_{ij}(\boldsymbol{U})\right| > \delta\right) \leq 4\exp\left\{-m\left(1 \wedge \frac{\delta}{20Ms_{ij}} \wedge \frac{\delta}{M\sigma_{ij}^2} \wedge \frac{1}{4|\sigma_{ij}|}\sqrt{\frac{2\delta}{M}}\right)\right\},$$

where $\hat{\boldsymbol{K}}_{ij}(\boldsymbol{U}) = g(\sigma_{ij}) + g'(\sigma_{ij})\left(\frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{m} - \sigma_{ij}\right)$.

Lemma 3.2.1 and Remark B.0.1 in Appendix B show that when we use $\hat{K}(\boldsymbol{U}) = [\hat{\boldsymbol{K}}_{ij}(\boldsymbol{U})]$ to approximate $\boldsymbol{K}(\boldsymbol{U})$ and when the number of hidden features $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ is large enough, the approximation will be sufficiently small. Hence, we can write $\boldsymbol{K}(\boldsymbol{U})$ as follow:

$$\boldsymbol{K}(\boldsymbol{U}) = \hat{\boldsymbol{K}}(\boldsymbol{U}) + o_P(1) = g[\boldsymbol{\Sigma}] + g'[\Sigma] \odot \left(\frac{1}{m} \boldsymbol{U}\boldsymbol{U}^T - \boldsymbol{\Sigma} \right) + o_P(1), \tag{3.13}$$

where $\odot$ means the Hadamard product of two matrices. Moreover, the Strong Law of Large Numbers implies $\frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T \to \boldsymbol{\Sigma}$ a.s. Therefore, equation (3.13) can be further written as

$$\boldsymbol{K}(\boldsymbol{U}) = g[\boldsymbol{\Sigma}] + o_P(1),$$

i.e., $\boldsymbol{K}(\boldsymbol{U}) \xrightarrow{P} f[\boldsymbol{\Sigma}]$ as $m \to \infty$ element-wisely. Following from the version of Dominated Convergence Theorem based on convergence in probability, the following lemma can be easily shown.

Lemma 3.2.2. *Under the assumptions of Lemma 3.2.1, if* $g''\left(\eta_{ij}\right)\left(\frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{m} - \sigma_{ij}\right)^2 \in L^1(\mathbb{P})$*, then*

$$\mathbb{E}\left[\frac{1}{2} g''\left(\eta_{ij}\right)\left(\frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{m} - \sigma_{ij}\right)^2\right] = o(1),$$

where $\eta_{ij} = \lambda\sigma_{ij} + (1-\lambda)\frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{m}$ *for some* $\lambda \in [0,1]$.

Based on Lemma 3.2.2, the marginal variance-covariance matrix $\boldsymbol{V}$ can be written as

$$\begin{aligned}
\boldsymbol{V} &= \tau\mathbb{E}\left[\boldsymbol{K}(\boldsymbol{U})\right] + \phi\boldsymbol{I}_n \\
&\simeq \tau\mathbb{E}\left[\hat{\boldsymbol{K}}(\boldsymbol{U})\right] + \phi\boldsymbol{I}_n \\
&\simeq \tau g[\boldsymbol{\Sigma}] + \phi\boldsymbol{I}_n \\
&= \tau \sum_{l=1}^{L'} \pi_l(\xi_1, \ldots, \xi_L)\gamma_l\left[\boldsymbol{K}_1(\boldsymbol{G}), \ldots, \boldsymbol{K}_L(\boldsymbol{G})\right] + \phi\boldsymbol{I}_n \\
&= \sum_{l=0}^{L'} \theta_l \boldsymbol{S}_l(\boldsymbol{G})\boldsymbol{S}_l^T(\boldsymbol{G}),
\end{aligned}$$

where $\theta_0 = \phi$, $\theta_l = \tau\pi_l(\xi_1, \ldots, \xi_L)$, $l = 1, \ldots, L'$ and $S_0(\boldsymbol{G}) = \boldsymbol{I}_n$, $\boldsymbol{S}_l(\boldsymbol{G})$ is the Cholesky lower triangle for the matrix $\gamma_l\left[\boldsymbol{K}_1(\boldsymbol{G}), \ldots, \boldsymbol{K}_L(\boldsymbol{G})\right]$, $l = 1, \ldots, L$. As an example, we may consider $g(x) = (1+x)^2$, which corresponds to the output polynomial kernel. $g[\boldsymbol{\Sigma}] = (\boldsymbol{J}_n + \xi_1 \frac{1}{p}\boldsymbol{G}\boldsymbol{G}^T)^{\circledwedge 2}$, where the symbol $\circledwedge 2$ means the elementwise square. In this case, $L = 1$ and $\boldsymbol{K}_1(\boldsymbol{G}) = p^{-1}\boldsymbol{G}\boldsymbol{G}^T$. Then note that

$$\begin{aligned} g[\boldsymbol{\Sigma}] &= \boldsymbol{J}_n + \frac{2\xi_1}{p}\boldsymbol{J} \odot \boldsymbol{G}\boldsymbol{G}^T + \frac{\xi_1^2}{p^2}(\boldsymbol{G}\boldsymbol{G}^T)^{\circledwedge 2} \\ &= \boldsymbol{J}_n + 2\xi_1 \frac{1}{p}\boldsymbol{G}\boldsymbol{G}^T + \xi_1^2 \frac{1}{p^2}(\boldsymbol{G}\boldsymbol{G}^T)^{\circledwedge 2}. \end{aligned}$$

This shows that for polynomial output kernel and one input product kernel, $L' = 3$ with $\pi_1(\xi_1) = 1, \pi_2(\xi_1) = 2\xi_1, \pi_3(\xi_1) = \xi_1^2$ and $\gamma_1[\boldsymbol{K}_1(\boldsymbol{G})] = \boldsymbol{J}_n, \gamma_2[\boldsymbol{K}_1(\boldsymbol{G})] = \boldsymbol{K}_1(\boldsymbol{G}) = p^{-1}\boldsymbol{G}\boldsymbol{G}^T, \gamma_3[\boldsymbol{K}_1(\boldsymbol{G})] = p^{-2}(\boldsymbol{G}\boldsymbol{G}^T)^{\circledwedge 2}$. The parameters $\theta_0, \ldots, \theta_L$ can be estimated via MINQUE as well. Based on the above discussion, we can see that the estimation of the variance components in KNN through MINQUE is an approximation. What we basically do here is to use a complex mixed model to approximate the KNN.

3.3 Predictions

In this section, we make a theoretical comparison of prediction performance between KNN and LMM. Based on our model, the best predictor for $\boldsymbol{f}$ is given by

$$\begin{aligned}
\hat{\boldsymbol{y}} &= \mathbb{E}\left[\boldsymbol{f}|\boldsymbol{y}\right] = \mathbb{E}_{\boldsymbol{U}}\left[\mathbb{E}\left(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\right)\right] \\
&= \mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\tau_j \boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tau_j \boldsymbol{K}_j(\boldsymbol{U}) + \phi \boldsymbol{I}_n\right)^{-1}\right]\boldsymbol{y} \\
&= \mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\phi^{-1}\tau_j \boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\phi^{-1}\tau_j \boldsymbol{K}_j(\boldsymbol{U}) + \boldsymbol{I}_n\right)^{-1}\right]\boldsymbol{y} \\
&:= \mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j \boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j \boldsymbol{K}_j(\boldsymbol{U}) + \boldsymbol{I}_n\right)^{-1}\right]\boldsymbol{y},
\end{aligned}$$

where $\tilde{\tau}_j = \tau_j \phi^{-1}, j = 1, \ldots, m$. The prediction error based on $\hat{\boldsymbol{y}}$ is given by

$$\begin{aligned}
R &= \left(\boldsymbol{y} - \hat{\boldsymbol{y}}\right)^T \left(\boldsymbol{y} - \hat{\boldsymbol{y}}\right) \\
&= \boldsymbol{y}^T \left(\boldsymbol{I}_n - \mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j \boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j \boldsymbol{K}_j(\boldsymbol{U}) + \boldsymbol{I}_n\right)^{-1}\right]\right)^T \\
&\qquad \left(\boldsymbol{I}_n - \mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j \boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j \boldsymbol{K}_j(\boldsymbol{U}) + \boldsymbol{I}_n\right)^{-1}\right]\right)\boldsymbol{y}.
\end{aligned}$$

Note that

$$\begin{aligned}
&\boldsymbol{I}_n - \mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]\\
=&\mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n-\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]\\
=&\mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right],
\end{aligned}$$

we have

$$R=\boldsymbol{y}^T\left(\mathbb{E}_{\boldsymbol{U}}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]\right)^2\boldsymbol{y}.$$

Direct evaluation of the prediction error R is complicated. Instead, we approximate it based on asymptotic results. Same as the above, we focus on the case where $J=1$ and $\boldsymbol{K}(\boldsymbol{U})=g\left[\frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T\right]$. The proof of the following Lemma can be found in Appendix B.

Lemma 3.3.1 (Approximation of Prediction Error)**.** *(i) When $g(x)=x$, then as $m\to\infty$,*

$$R\simeq\boldsymbol{y}^T\left(\sum_{l=1}^{L}\tilde{\tau}\xi\boldsymbol{K}_l(\boldsymbol{G})+\boldsymbol{I}_n\right)^{-2}\boldsymbol{y}.$$

(ii) When g is continuous and $g[\boldsymbol{\Sigma}]\in\mathcal{S}_+^n$, then as $m\to\infty$,

$$R\simeq\boldsymbol{y}^T\left(\tilde{\tau}g\left[\sum_{l=1}^{L}\xi\boldsymbol{K}_l(\boldsymbol{G})\right]+\boldsymbol{I}_n\right)^{-2}\boldsymbol{y}.$$

Now we compare the average prediction error between kernel neural network and linear mixed model. For a linear mixed model, the prediction error using best predictor can be

obtained as follow. The proof can be found in Appendix B.

Proposition 3.3.1 (Prediction Error for a Linear Mixed Model)**.** *Consider the linear mixed effect model*

$$\begin{aligned}
\boldsymbol{y} &= \boldsymbol{f} + \boldsymbol{\epsilon};\\
\boldsymbol{f} &\sim \mathcal{N}_n\left(\mathbf{0}, \sigma_R^2 \boldsymbol{\Sigma}\right);\\
\boldsymbol{\epsilon} &\sim \mathcal{N}_n\left(\mathbf{0}, \phi \boldsymbol{I}_n\right).
\end{aligned}$$

The prediction error based on quadratic loss and the best predictor $\hat{\boldsymbol{y}} = \mathbb{E}[\boldsymbol{f}|\boldsymbol{y}] = \tilde{\sigma}_R^2 \boldsymbol{\Sigma}\left(\tilde{\sigma}_R^2 \boldsymbol{\Sigma} + \boldsymbol{I}_n\right)^{-1} \boldsymbol{y}$ $(\tilde{\sigma}_R^2 = \sigma_R^2 \phi^{-1})$ *is given by*

$$PE_{LMM} = \phi \sum_{i=1}^{n} \left(\tilde{\sigma}_R^2 \lambda_i(\boldsymbol{\Sigma}) + 1\right)^{-1},$$

where PE_{LMM} *is the average prediction error for the linear mixed model and* $\lambda_1(\boldsymbol{\Sigma}), \ldots, \lambda_n(\boldsymbol{\Sigma})$ *are the eigenvalues of* $\boldsymbol{\Sigma}$*.*

Proposition 3.3.2. *Assuming that* $\sigma^2 = \phi$ *and* $\tilde{\sigma}_R^2 \leq \tilde{\tau} \min_{1 \leq l \leq L} \xi_l$*, we have*

$$PE_{KNN} \precsim PE_{LMM},$$

where PE_{KNN} *stands for average prediction error for kernel neural network.*

Proof. Let $\boldsymbol{A} = \mathbb{E}_{\boldsymbol{U}}\left[(\tilde{\tau}\boldsymbol{K}(\boldsymbol{U}) + \boldsymbol{I}_n)^{-1}\right]$. Then for the kernel neural network, the average

prediction error is given by

$$
\begin{aligned}
PE_{KNN} &= \mathbb{E}\left[\boldsymbol{y}^T\boldsymbol{A}^2\boldsymbol{y}\right] = \mathbb{E}_{\boldsymbol{U}}\left[\mathbb{E}\left(\boldsymbol{y}^T\boldsymbol{A}^2\boldsymbol{y}|\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m\right)\right] \\
&= \phi\mathbb{E}_{\boldsymbol{U}}\left[\mathrm{tr}\left(\boldsymbol{A}^2\left(\tilde{\tau}\boldsymbol{K}(\boldsymbol{U})+\boldsymbol{I}_n\right)\right)\right] \\
&= \phi\mathrm{tr}\left\{\left(\mathbb{E}_{\boldsymbol{U}}\left[\left(\tilde{\tau}\boldsymbol{K}(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]\right)^2\mathbb{E}_{\boldsymbol{U}}\left[\tilde{\tau}\boldsymbol{K}(\boldsymbol{U})+\boldsymbol{I}_n\right]\right\} \\
&\simeq \phi\mathrm{tr}\left\{\left(\tilde{\tau}\sum_{l=1}^{L}\xi_l\boldsymbol{K}_l(\boldsymbol{G})+\boldsymbol{I}_n\right)^{-2}\left(\tilde{\tau}\sum_{l=1}^{L}\xi_l\boldsymbol{K}_l(\boldsymbol{G})+\boldsymbol{I}_n\right)\right\} \\
&= \phi\mathrm{tr}\left\{\left(\tilde{\tau}\sum_{l=1}^{L}\xi_l\boldsymbol{K}_l(\boldsymbol{G})+\boldsymbol{I}_n\right)^{-1}\right\} \\
&\leq \phi\sum_{i=1}^{n}\left(\tilde{\tau}\min_{1\leq l\leq L}\xi_l\lambda_i\left(\sum_{l=1}^{L}\boldsymbol{K}_l(\boldsymbol{G})\right)+1\right)^{-1}
\end{aligned}
$$

Under the assumptions in this proposition and the linear mixed model with $\boldsymbol{\Sigma} = \sum_{l=1}^{L}\boldsymbol{K}_l(\boldsymbol{G})$, we have

$$
\frac{\phi\left(\tilde{\tau}\min_{1\leq l\leq L}\lambda_i(\boldsymbol{\Sigma})+1\right)^{-1}}{\sigma^2\left(\tilde{\sigma}_R^2\lambda_i(\boldsymbol{\Sigma})+1\right)^{-1}} = \frac{\phi}{\sigma^2}\frac{\tilde{\sigma}_R^2\lambda_i(\boldsymbol{\Sigma})+1}{\tilde{\tau}\min_{1\leq l\leq L}\xi_l\lambda_i(\boldsymbol{\Sigma})+1} \leq 1,
$$

which implies that $PE_{KNN} \precsim PE_{LMM}$. □

Remark 3.3.1. *The result for Proposition 3.3.2 can be illustrated by using Figure 3.2. As shown in the Figure 3.2 for the case of $L = 1$, there are two paths from the kernel matrix based on $\boldsymbol{G}$ to the response $\boldsymbol{y}$. One is the kernel neural network path (solid line) and the other is the linear mixed model path (dash-dotted line). The intuition behind the assumption $\tilde{\sigma}_R^2 \leq \tilde{\tau}\xi$ is that the kernel neural network should explain more variations than the linear mixed model as it has two portions.*

We then extend the result to $\boldsymbol{K}(\boldsymbol{U}) = g\left[\frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T\right]$, where g is as described in Lemma

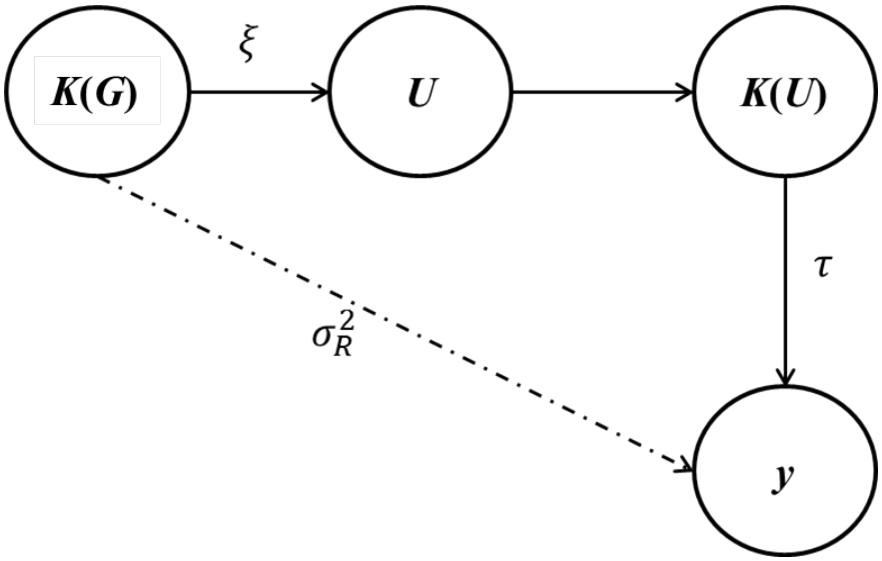

Figure 3.2: The intuition under the assumption $\tilde{\sigma}_R^2 \leq \tilde{\tau}\xi$ in Proposition 3.3.2.

3.3.1(ii).

Proposition 3.3.3. *Under the above notations, assuming that* $\sigma^2 = \phi$, $\tilde{\sigma}_R^2 \leq \tilde{\tau} \min_{1\leq l\leq L} \xi_l$ *and* $\lambda_1\left((g-\iota)\left[\sum_{l=1}^L \xi_l \boldsymbol{K}_l(\boldsymbol{G})\right]\right) \geq 0$ *with* $|g''(x)| \leq M$ *for some* $M > 0$ *and all* x *between* $\min_{i,j} \sigma_{ij}$ *and* $\max_{i,j} \frac{\boldsymbol{v}_i^T \boldsymbol{v}_j}{m}$, *we have*

$$PE_{KNN} \precsim PE_{LMM},$$

where $\lambda_1(\boldsymbol{\Sigma})$ *is the smallest eigenvalue of the matrix* $\boldsymbol{\Sigma}$.

Proof. Note that

$$\begin{aligned}
PE_{KNN} &= \phi \mathrm{tr}\left\{\left(\mathbb{E}\left[(\tilde{\tau}\boldsymbol{K}(\boldsymbol{U})+\boldsymbol{I}_n)^{-1}\right]\right)^2 \mathbb{E}\left[\tilde{\tau}\boldsymbol{K}(\boldsymbol{U})+\boldsymbol{I}_n\right]\right\} \\
&\simeq \phi \mathrm{tr}\left\{\left(\tilde{\tau} g\left[\sum_{l=1}^{L}\xi_l \boldsymbol{K}_l(\boldsymbol{G})\right]+\boldsymbol{I}_n\right)^{-1}\right\} \\
&= \phi \sum_{i=1}^{n} \frac{1}{\tilde{\tau}\lambda_i\left(g\left[\sum_{l=1}^{L}\xi_l \boldsymbol{K}_l(\boldsymbol{G})\right]\right)+1} \\
&\leq \phi \sum_{i=1}^{n} \frac{1}{\tilde{\tau}\lambda_i\left((g-\iota)\left[\sum_{l=1}^{L}\xi_l \boldsymbol{K}_l(\boldsymbol{G})\right]+\min_{1\leq l\leq L}\xi_l \sum_{l=1}^{L}\boldsymbol{K}_l(\boldsymbol{G})\right)+1},
\end{aligned}$$

where $\iota : \boldsymbol{\Sigma} \mapsto \boldsymbol{\Sigma}$ is the identity map. Corollary 4.3.15 in Horn and Johnson (2012)[32] implies that

$$PE_{KNN} \precsim \phi \sum_{i=1}^{n} \frac{1}{\tilde{\tau}\min_{1\leq l\leq L}\xi_l\lambda_i(\boldsymbol{K}_l(\boldsymbol{G}))+\tilde{\tau}\lambda_1\left((g-\iota)\left[\sum_{l=1}^{L}\xi_l \boldsymbol{K}_l(\boldsymbol{G})\right]\right)+1}$$

□

Corollary 3.3.1. *If* $g\left[\sum_{l=1}^{L}\xi_l \boldsymbol{K}_l(\boldsymbol{G})\right]-\sum_{l=1}^{L}\xi_l \boldsymbol{K}_l(\boldsymbol{G})$ *is positive semidefinite, then*

$$PE_{KNN} \precsim PE_{LMM}.$$

Example 3.3.1 (Polynomial Kernels)**.** *For a polynomial kernel of degree d, i.e.,* $\boldsymbol{K}_{ij}(\boldsymbol{U}) = \left(c+\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m}\right)^d$*, we have* $g(x) = (c+x)^d = \sum_{k=0}^{d}\binom{d}{k}c^{d-k}x^k$ *so that*

$$(g-\iota)(x) = c^d + (dc^{d-1}-1)x + \sum_{k=2}^{d}\binom{d}{k}c^{d-k}x^k.$$

Theorem 4.1 in Hiai (2009)[29] states that for a real function on $(-\alpha, \alpha)$, $0 < \alpha \leq \infty$, it is Schur positive[1] *if and only if it is analytic and $g^{(k)}(0) \geq 0$ for all $k \geq 0$. Since $g - \iota$ is a polynomial function, it is clearly analytic. We can then expand $g(x)$ using Taylor expansion around 0 and obtain*

$$\binom{d}{k} c^{d-k} = \frac{g^{(k)}(0)}{k!} \Rightarrow g^{(k)}(0) = \frac{d!}{(d-k)!} c^{d-k}, \quad k = 0, \ldots, d.$$

Hence, we have

$$(g - \iota)^{(k)}(x) = \begin{cases} dc^{d-1} - 1 & \text{if } k = 1 \\ \frac{d!}{(d-k)!} c^{d-k} & \text{if } k \in \{0, 1, \ldots, d\} \setminus \{1\} \\ 0 & k \geq d + 1 \end{cases} .$$

To make $g - \iota$ Schur positive, we only need to require $c \geq \sqrt[d-1]{\frac{1}{d}} \geq 0$ so that the minimum eigenvalue condition of Proposition 3.3.3 holds.

3.4 Including Fixed Effects

In the previous discussions, we focus on the case where the marginal distribution of the response variable $\boldsymbol{y}$ has mean $\mathbf{0}$. In many applications, as the one we see in the ADNI real data application, there are many important covariates which may have large effects to the response variable. In this part, we are going to extend the proposed KNN model to take the

[1] For a real function f on $(-\alpha, \alpha)$ and for $n \in \mathbb{N}$, it is Schur-positive of order n if $f[\boldsymbol{A}]$ is positive semidefinite for all positive semidefinite $\boldsymbol{A} \in \mathcal{M}_n(\mathbb{R})$ with entries in $(-\alpha, \alpha)$.

covariates into account. As we have mentioned in (3.4), the general structure of KNN is

$$\begin{aligned} \boldsymbol{y}|\boldsymbol{f} &\sim \mathcal{N}_n\left(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{f}, \phi \boldsymbol{I}_n\right) \\ \boldsymbol{f}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m &\sim \mathcal{N}_n\left(\mathbf{0}, \sum_{j=1}^{J} \tau_j \boldsymbol{K}_j(\boldsymbol{U})\right) \\ \boldsymbol{u}_1, \ldots, \boldsymbol{u}_m &\sim \text{ i.i.d. } \mathcal{N}_n\left(\mathbf{0}, \sum_{l=1}^{L} \xi_l \boldsymbol{K}_l(\boldsymbol{G})\right). \end{aligned} \tag{3.14}$$

Similar to the situation considered before, given the latent variables $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$, we have

$$\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m \sim \mathcal{N}_n\left(\boldsymbol{X}\boldsymbol{\beta}, \sum_{j=1}^{J} \tau_j \boldsymbol{K}_j(\boldsymbol{U}) + \phi \boldsymbol{I}_n\right)$$

and then the marginal mean and covariance matrix of $\boldsymbol{y}$ can be obtained as follow:

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{y}\right] &= \mathbb{E}_{\boldsymbol{U}}\left[\mathbb{E}\left[\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\right]\right] = \mathbb{E}_{\boldsymbol{U}}\left[\boldsymbol{X}\boldsymbol{\beta}\right] = \boldsymbol{X}\boldsymbol{\beta} \\ \operatorname{Var}\left[\boldsymbol{y}\right] &= \operatorname{Var}_{\boldsymbol{U}}\left[\mathbb{E}\left[\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\right]\right] + \mathbb{E}_{\boldsymbol{U}}\left[\operatorname{Var}\left[\mathbb{E}\left[\boldsymbol{y}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\right]\right]\right] \\ &= \operatorname{Var}_{\boldsymbol{U}}[\boldsymbol{X}\boldsymbol{\beta}] + \mathbb{E}_{\boldsymbol{U}}\left[\sum_{j=1}^{J} \tau_j \boldsymbol{K}_j(\boldsymbol{U}) + \phi \boldsymbol{I}_n\right] \\ &= \sum_{j=0}^{J} \tau_j \mathbb{E}_{\boldsymbol{U}}\left[\boldsymbol{K}_j(\boldsymbol{U})\right], \end{aligned}$$

where $\tau_0 = \phi$ and $\boldsymbol{K}_0(\boldsymbol{U}) = \boldsymbol{I}_n$. Again, we focus on the case where $J = 1$ and $\boldsymbol{K}(\boldsymbol{U})$ is of the form $g\left[\frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T\right]$ with g satisfying the generalized linear separable condition so that after suitable reparameterization, the variance components become estimable. A natural way to obtain the estimators for $\boldsymbol{\beta}$ and the variance components $\boldsymbol{\theta}$ is to first obtain a "good" estimates for the variance components $\hat{\boldsymbol{\theta}}$ and then plug it into the Aitken equation to obtain the estimates for fixed-effect parameters $\hat{\boldsymbol{\beta}}$. Kackar and Harville (1981)[37] showed that as

long as $\hat{\boldsymbol{\theta}}$ is even and translation-invariant, $\boldsymbol{p}^T\hat{\boldsymbol{\beta}} + \boldsymbol{q}^T\hat{\boldsymbol{f}}$ is an unbiased prediction for the quantity $\boldsymbol{p}^T\boldsymbol{\beta} + \boldsymbol{q}^T\boldsymbol{f}$.

Let $\boldsymbol{R}$ be an $r \times n$ matrix with $r = n - \text{rank}(\boldsymbol{X})$ such that $\boldsymbol{RX} = \boldsymbol{O}$ and $\boldsymbol{RR}^T = \boldsymbol{I}_r$. Such a matrix can be found using the QR decomposition of $\boldsymbol{X}$ [47]. The estimators for the variance components can then be estimated based on the transformed model:

$$\begin{aligned}
\tilde{\boldsymbol{y}}|\tilde{\boldsymbol{f}} &\sim \mathcal{N}_r(\tilde{\boldsymbol{f}}, \phi\boldsymbol{I}_r) \\
\tilde{\boldsymbol{f}}|\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m &\sim \mathcal{N}_r\left(\boldsymbol{0}, \sum_{j=1}^{J} \tau_j \boldsymbol{R}\boldsymbol{K}_j(\boldsymbol{U})\boldsymbol{R}^T\right) \\
\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m &\sim \text{ i.i.d. } \mathcal{N}_n\left(\boldsymbol{0}, \sum_{l=1}^{L} \xi_l \boldsymbol{K}_l(\boldsymbol{G})\right),
\end{aligned}$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{Ry}$ and $\tilde{\boldsymbol{f}} = \boldsymbol{Rf}$, which is the same model framework we mainly discussed in section 3.2. As we have seen, the estimators for the variance components after reparameterization is of the form $\tilde{\boldsymbol{y}}^T\boldsymbol{\Theta}\tilde{\boldsymbol{y}} = \boldsymbol{y}^T\boldsymbol{R}^T\boldsymbol{\Theta}\boldsymbol{Ry}$. Since it is a quadratic form, clearly it is an even function in $\boldsymbol{y}$. To see that it is translation invariant, we note that for any $\boldsymbol{c} \in \mathcal{C}(\boldsymbol{X})$, we can know that $\boldsymbol{c} = \boldsymbol{Xb}$ for some vector $\boldsymbol{b} \in \mathbb{R}^r$ and we have

$$\begin{aligned}
(\boldsymbol{y} - \boldsymbol{c})^T\boldsymbol{R}^T\boldsymbol{\Theta}\boldsymbol{R}(\boldsymbol{y} - \boldsymbol{c}) &= (\boldsymbol{y} - \boldsymbol{Xb})^T\boldsymbol{R}^T\boldsymbol{\Theta}\boldsymbol{R}(\boldsymbol{y} - \boldsymbol{Xb}) \\
&= \boldsymbol{y}^T\boldsymbol{R}^T\boldsymbol{\Theta}\boldsymbol{Ry} - 2\boldsymbol{y}^T\boldsymbol{R}^T\boldsymbol{\Theta}\boldsymbol{RXb} + \boldsymbol{b}^T\boldsymbol{X}^T\boldsymbol{R}^T\boldsymbol{\Theta}\boldsymbol{RXb} \\
&= \boldsymbol{y}^T\boldsymbol{R}^T\boldsymbol{\Theta}\boldsymbol{Ry},
\end{aligned}$$

where the last equality follows since $\boldsymbol{RX} = \boldsymbol{O}$ as defined. Therefore, we can know that the obtained estimators for variance components are also translation-invariant and based on the results in Kackar and Harville (1981)[37], the obtained estimator for $\boldsymbol{\beta}$ by plugging in the

variance component estimators is unbiased.

For the prediction error, we note that when the covariates present, the predictor for $\boldsymbol{y}$ is given by $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{f}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \mathbb{E}[\boldsymbol{f}|\boldsymbol{y}]$. Based on the result from linear mixed models, we know that

$$\begin{aligned}\hat{\boldsymbol{f}} &= \mathbb{E}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right](\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}) \\ &= \mathbb{E}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]\left[\boldsymbol{I}_n - \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-}\boldsymbol{X}^T\boldsymbol{V}^{-1}\right]\boldsymbol{y}\end{aligned}$$

where $\boldsymbol{V} = \mathrm{Var}[\boldsymbol{y}] = \sum_{j=1}^{J}\tau_j\mathbb{E}[\boldsymbol{K}_j(\boldsymbol{U})] + \phi\boldsymbol{I}_n$ and then

$$\begin{aligned}\boldsymbol{y}-\hat{\boldsymbol{y}} &= \left[\boldsymbol{I}_n - \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-}\boldsymbol{X}^T\boldsymbol{V}^{-1}\right]\boldsymbol{y} \\ &\quad - \mathbb{E}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]\left[\boldsymbol{I}_n - \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-}\boldsymbol{X}^T\boldsymbol{V}^{-1}\right]\boldsymbol{y} \\ &= \left(\boldsymbol{I}_n - \mathbb{E}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]\right)\left[\boldsymbol{I}_n - \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-}\boldsymbol{X}^T\boldsymbol{V}^{-1}\right]\boldsymbol{y} \\ &= \mathbb{E}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]\left[\boldsymbol{I}_n - \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-}\boldsymbol{X}^T\boldsymbol{V}^{-1}\right]\boldsymbol{y},\end{aligned}$$

where the last equality follows by noting that

$$\boldsymbol{I}_n - \mathbb{E}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})\right)\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right] = \mathbb{E}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]$$

as shown above. Therefore, by letting $\boldsymbol{A} = \mathbb{E}\left[\left(\sum_{j=1}^{J}\tilde{\tau}_j\boldsymbol{K}_j(\boldsymbol{U})+\boldsymbol{I}_n\right)^{-1}\right]$ and $\boldsymbol{P}_{\boldsymbol{V}} =$

$\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-}\boldsymbol{X}^T\boldsymbol{V}^{-1}$, the prediction error is obtained as follow:

$$(\boldsymbol{y}-\hat{\boldsymbol{y}})^T(\boldsymbol{y}-\hat{\boldsymbol{y}}) = \boldsymbol{y}^T\left(\boldsymbol{I}_n - \boldsymbol{P}_{\boldsymbol{V}}^T\right)\hat{\boldsymbol{A}}^2\left(\boldsymbol{I}_n - \boldsymbol{P}_{\boldsymbol{V}}\right)\boldsymbol{y}.$$

When $J=1$, as we have shown in the proof of Lemma 3.3.1, $\mathbb{E}\left[(\tilde{\tau}\boldsymbol{K}(\boldsymbol{U})+\boldsymbol{I}_n)^{-1}\right] \simeq \left(\tilde{\tau}f\left[\sum_{l=1}^{L}\xi_l\boldsymbol{K}_l(\boldsymbol{G})\right]+\boldsymbol{I}_n\right)^{-1}$ so that we can estimate the prediction error by using simple plug-in estimators.

$$\widehat{(\boldsymbol{y}-\hat{\boldsymbol{y}})^T(\boldsymbol{y}-\hat{\boldsymbol{y}})} \simeq \boldsymbol{y}^T\left(\boldsymbol{I}_n - \boldsymbol{P}_{\hat{\boldsymbol{V}}}^T\right)\left(\hat{\tilde{\tau}}f\left[\sum_{l=1}^{L}\hat{\xi}_l\boldsymbol{K}_l(\boldsymbol{G})\right]+\boldsymbol{I}_n\right)^{-2}\left(\boldsymbol{I}_n - \boldsymbol{P}_{\hat{\boldsymbol{V}}}\right)\boldsymbol{y}.$$

3.5 Simulations

In this section, we conducted some simulations to compare the prediction performance of KNN with MINQUE estimation to the prediction performance of the Best Linear Unbiased Estimator (BLUP) in linear mixed models. All the simulations are based on 100 individuals with 500 Monte Carlo iterations.

3.5.1 Nonlinear Random Effect

As we have seen in section 3.2 that a KNN is equivalent to a nonlinear fixed effect model and we have also shown that the prediction error for KNN will be smaller than that of linear mixed models. Here, we use a simulation to validate previous observations through some simulation studies. We used the following model to simulate the response:

$$\boldsymbol{y} = \mathbf{1}_n + 2\boldsymbol{\zeta} + f(\boldsymbol{u}) + \boldsymbol{\epsilon}, \quad \boldsymbol{u} \sim \mathcal{N}_n\left(\mathbf{0}, \frac{1}{p}\boldsymbol{G}\boldsymbol{G}^T\right), \tag{3.15}$$

where $\boldsymbol{G}$ is an $n \times p$ matrix containing the genetic information (SNP) and $\boldsymbol{\zeta}, \boldsymbol{\epsilon} \sim$ i.i.d. $\mathcal{N}_n(\mathbf{0}, \boldsymbol{I}_n)$. Here $\boldsymbol{\zeta}$ is served as the fixed effect in the formulation of a KNN. In this simulation, four types of functions f are considered, which are linear ($f(x) = x$), sine ($f(x) = \sin(2\pi x)$), inverse logistic ($f(x) = 1/(1+e^{-x})$) and polynomial function of order 2 ($f(x) = x^2$). When applying the kernel neural network, we set $L = J = 1$ and choose $\boldsymbol{K}(\boldsymbol{G})$ and $\boldsymbol{K}(\boldsymbol{U})$ as either product kernel or polynomial kernel of order 2. Figure 3.3 shows the results when f is a linear or a sine function. The cases where f is an inverse logistic function or a polynomial function of order 2 are summarized in Appendix B.

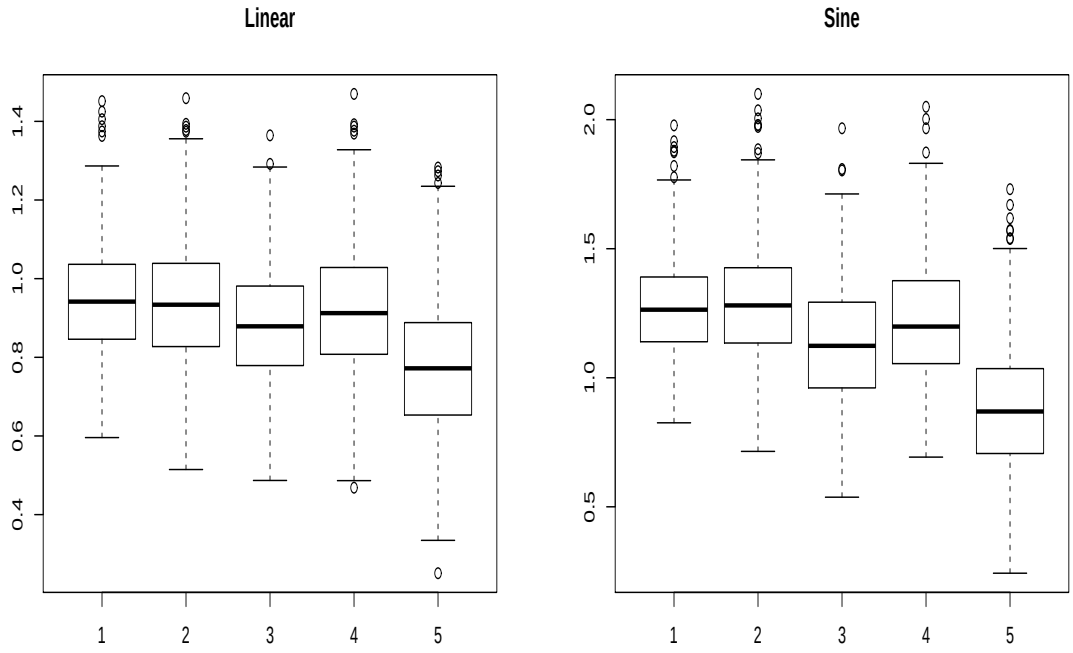

Figure 3.3: The boxplots for linear mixed models (LMM) and kernel neural networks (KNN) in terms of prediction errors. The left panel shows the results when a linear function is used and the right panel shows the results when a sine function is used. In the horizontal axis, "1" corresponds to the LMM; "2" corresponds to the KNN with product input kernel and product output kernel; "3" corresponds to the KNN with product input and polynomial output; "4" corresponds to the KNN with polynomial input and product output and "5" corresponds to the polynomial input and polynomial output.

As we can observe from the boxplots, when the output kernel is chosen to be the product kernel, the performance of KNN is similar to the performance of LMM although when the input kernel is chosen to be polynomial kernel, KNN gets a slightly better prediction error, which we think is not significantly different from that of LMM. However, KNN performs significantly better than LMM when the output kernel is chosen to be polynomial kernel. As one can tell from the box plots, when both the input and output kernels are chosen to be polynomial, the KNN has the best performance in terms of the prediction error, which is consistent for all nonlinear functions simulated in this section.

3.5.2 Nonadditive Effects

In this simulation, we explore the performances of both method under non-additive effects. We conducted two simulations in terms of two different types of non-additive effects. In the first simulation, we mainly focus on the interaction effect and generate the response using the following model:

$$\boldsymbol{y} = f(\boldsymbol{G}) + \boldsymbol{\epsilon},$$

where $\boldsymbol{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_p] \in \mathbb{R}^{n \times p}$ is the SNP data and $\boldsymbol{\epsilon} \sim \mathcal{N}_n(\boldsymbol{0}, \boldsymbol{I}_n)$. When applying both methods, the mean is adjusted so that the response has marginal mean 0. In the simulation, we randomly pick 10 causal SNPs, denoted by $\boldsymbol{g}_{i_1}, \ldots, \boldsymbol{g}_{i_{10}}$ and consider the following function,

$$f(\boldsymbol{G}) = \sum_{1 \le j_1 < j_2 \le 10} \boldsymbol{g}_{i_{j_1}} \odot \boldsymbol{g}_{i_{j_2}},$$

where $\odot$ stands for the Hadamard product. For LMM, the product kernel was used as the covariance matrix to generate the random effect. The result is shown in Figure 3.4. It is interesting to notice from the boxplots that both LMM and KNN have many outliers when

product kernel was used. Overall, LMM has larger variations compared to KNN in this simulation. When the output kernel in KNN is the polynomial kernel, the performance of KNN is much better than that of LMM.

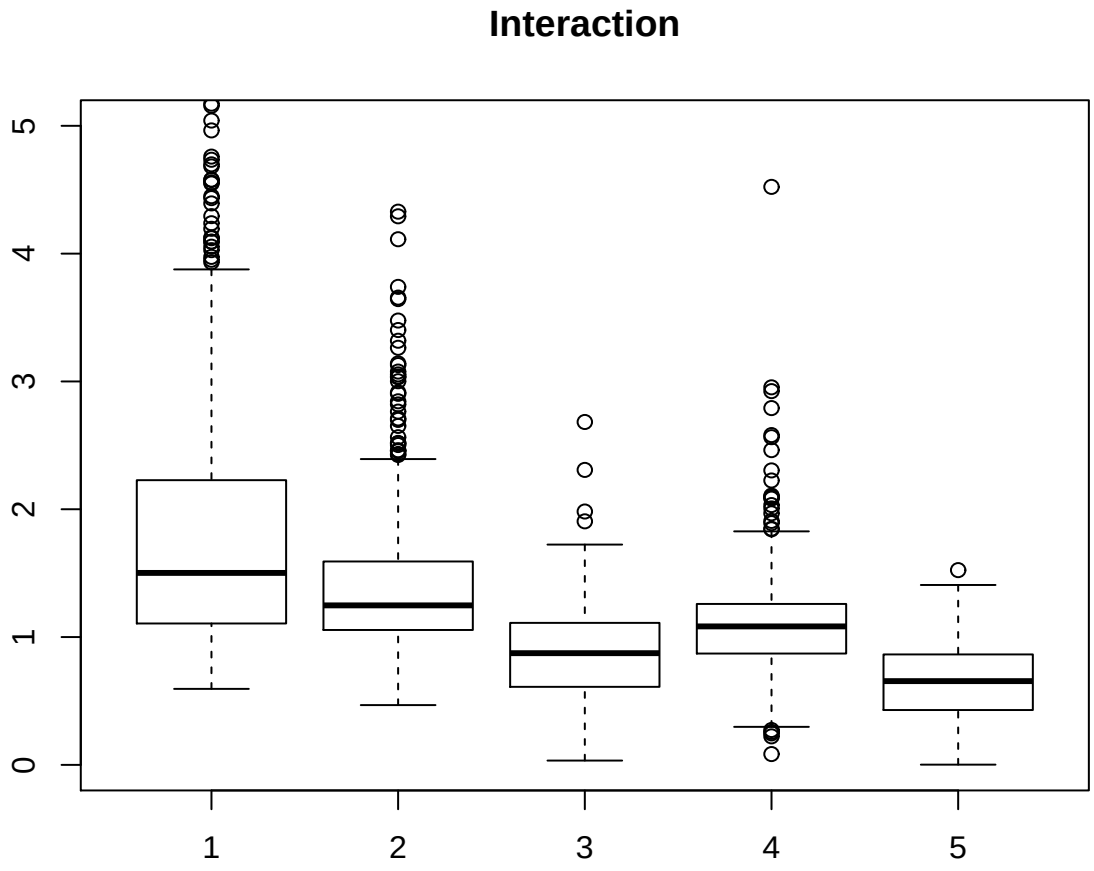

Figure 3.4: The boxplots for linear mixed models (LMM) and kernel neural networks (KNN) in terms of prediction errors based on the simulation model focusing on the interaction effect. The vertical axis is scaled to 0-5 by removing some outliers to make the comparison visually clear. In the horizontal axis, "1" corresponds to the LMM; "2" corresponds to the KNN with product input kernel and product output kernel; "3" corresponds to the KNN with product input and polynomial output; "4" corresponds to the KNN with polynomial input and product output and "5" corresponds to the polynomial input and polynomial output.

There are three main modes of inheritance: additive, dominant and recessive. In many situations, the additive coding (AA=0, Aa=1, aa=2) is used. In the second simulation, we

consider the dominant coding (AA=1, Aa=1, aa=0) and the recessive coding (AA=0, Aa=1, aa=1). The response was simulated based on the model:

$$\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{\epsilon}, \quad \boldsymbol{a} \sim \mathcal{N}_n\left(\mathbf{0}, \frac{1}{p}\boldsymbol{G'}\boldsymbol{G'}^T\right), \quad \boldsymbol{\epsilon} \sim \mathcal{N}_n(\mathbf{0}, \boldsymbol{I}_n),$$

where $\boldsymbol{G'}$ is a SNP data matrix based on dominant coding or recessive coding so that each element in $\boldsymbol{G'}$ takes only two possible values 0 and 1. Figure 3.5 summarizes the simulation results. By comparing the two boxplots in Figure 3.5, the performances look similar in both cases. Similar as before, KNN with input polynomial kernel and output polynomial kernel achieves the lowest prediction error.

3.5.3 Non-normal Error Distributions

In this simulation, we consider different types of error distributions and explore the performance in terms of prediction error between LMM and KNN. Specifically, we focus on two types of error distributions. The first one is a t-distribution with degrees of freedom 2, which is a heavy-tailed distribution and the second one is a centered χ^2_1-distribution, which is a non-symmetric distribution. The response was simulated based on the model as in section 5.1 except that the function applied to the random effect is $f(x) = x$ and the distribution for the error term ϵ is either t_2 or centered χ^2_1 distribution. The results are summarized in Figure 3.6. From the results, we can know that the KNN with polynomial input and output kernel is slightly more robust to non-normal distributions compared with LMM and KNN with other combinations of input and output kernels considered in the simulations. Since t_2-distribution is a heavy-tailed distribution, we can find that there are more outliers than in the normal case. In terms of the number of outliers, KNN with polynomial input and

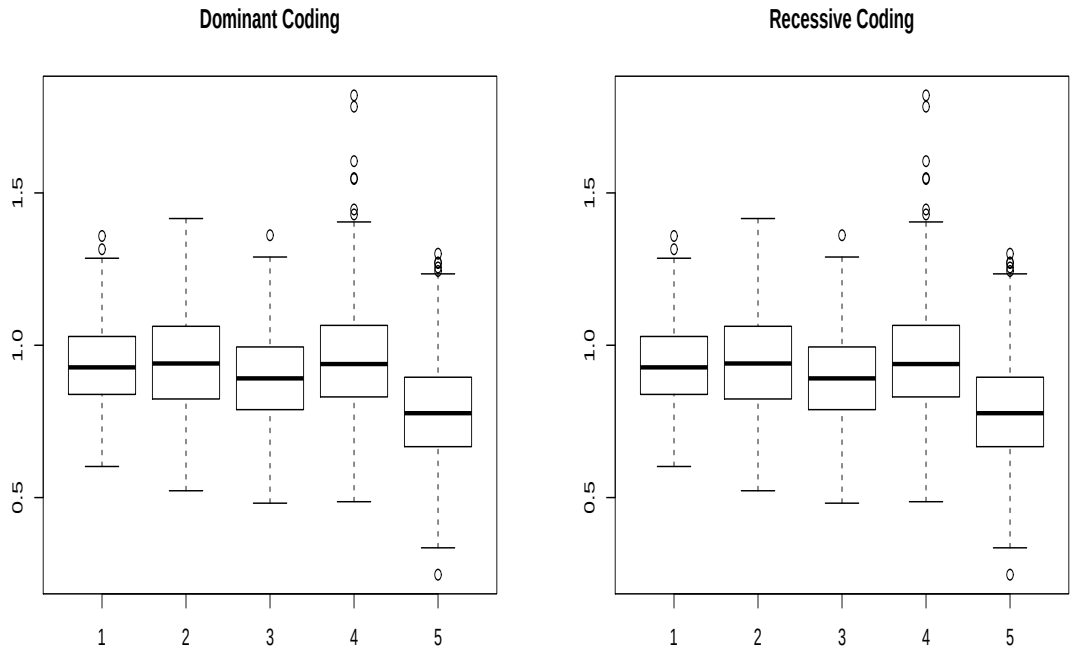

Figure 3.5: The boxplots for linear mixed models (LMM) and kernel neural networks (KNN) in terms of prediction errors based on the simulation model using dominant coding (left figure) and recessive coding (right figure) for SNPs. In the horizontal axis, "1" corresponds to the LMM; "2" corresponds to the KNN with product input kernel and product output kernel; "3" corresponds to the KNN with product input and polynomial output; "4" corresponds to the KNN with polynomial input and product output and "5" corresponds to the polynomial input and polynomial output.

output kernel still performs the best.

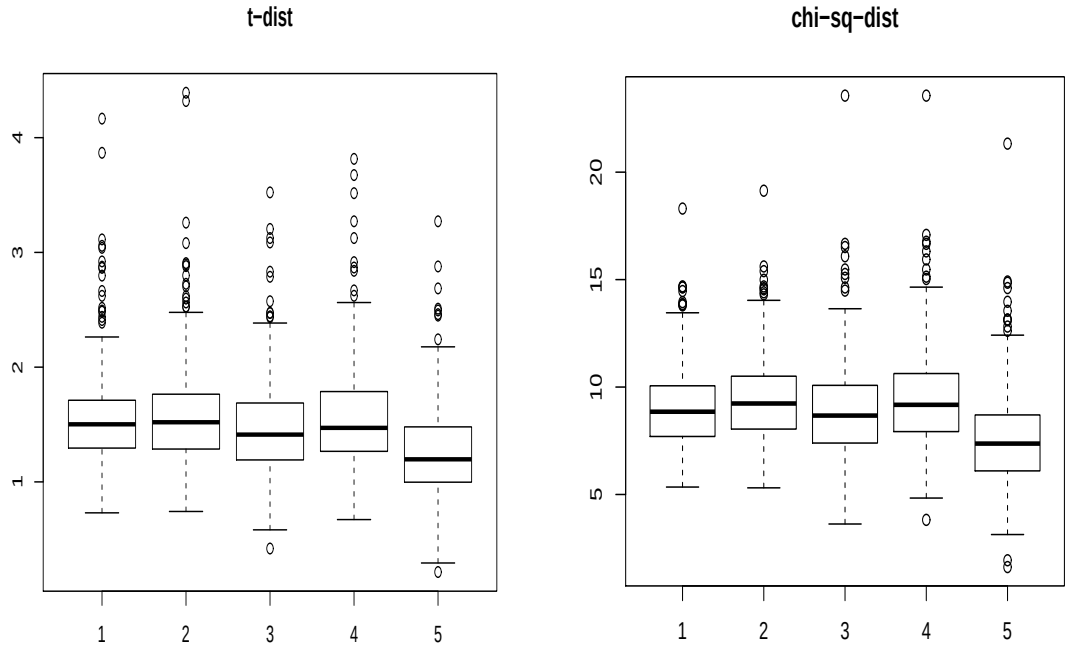

Figure 3.6: The boxplots for linear mixed models (LMM) and kernel neural networks (KNN) in terms of prediction errors based on the simulation model using t-distribution for error (left figure) and centered χ^2_1-distribution for error (right figure). In the horizontal axis, "1" corresponds to the LMM; "2" corresponds to the KNN with product input kernel and product output kernel; "3" corresponds to the KNN with product input and polynomial output; "4" corresponds to the KNN with polynomial input and product output and "5" corresponds to the polynomial input and polynomial output.

3.6 Real Data Application

We applied our method to the whole genome sequencing data from Alzheimer's Disease Neuroimaging Initiative (ADNI) as well and made predictions on the responses. A total of 808 individuals at the baseline of the ADNI1 and ADNI2 studies have the whole genome sequencing data. We dropped the single nucleotide polymorphisms (SNPs) with low calling rate (<0.9), or low minor allele frequencies (MAF) (<0.01), or those failed to pass the

Hardy Weinberg exact test (p-value<1e-6), and non-European American samples were also dropped. The data was then uploaded to the server in the University of Michigan for posterior likelihood imputation (`https://imputationserver.sph.umich.edu/index.html`). From the imputed data, we extracted SNPs with allelic $R^2 > 0.9$ and then the covariance kernel matrix the normalized identity-by-state (IBS) kernel matrix were constructed for analysis. Specifically, the $(i, j)$the element in each of the two kernel matrix is defined as follows:

$$\boldsymbol{K}^{\text{cov}}(\boldsymbol{g}_i, \boldsymbol{g}_j) = \frac{1}{p-1} \sum_{k=1}^{p} \left(g_{ik} - \frac{1}{p} \sum_{k} g_{ik} \right) \left(g_{jk} - \frac{1}{p} \sum_{k} g_{jk} \right)$$

$$\boldsymbol{K}^{\text{ibs}}(\boldsymbol{g}_i, \boldsymbol{g}_j) = \frac{1}{2p} \sum_{k=1}^{p} \left[2 - |g_{ik} - g_{jk}| \right],$$

where p is the number of SNPs in all expressions.

Four volume measures of cortical regions, which are hippocampus, ventricles, entorhinal and whole brain volumes were used as phenotypes of interest. We chose these four cortical regions since they play important roles in the Alzheimer's disease (AD). The loss in the volumes of the whole brain, hippocampus and entorhinal and the increment in the ventricular volume can be detected among AD patients. When we applied both the KNN method and LMM method, we only include the subjects having both genetic information and phenotypic information, which results in 513 individuals for hippocampus; 564 individuals for ventricles; 516 individuals for entorhinal and 570 individuals for whole brain volumes.

The response variable was chosen to be the natural logarithm of the volumes of the four cortical regions and the covariates were chosen as the age, gender, education status and

APOE4. Then the KNN is based on the model

$$\begin{aligned}\boldsymbol{y}|\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m &\sim \mathcal{N}_n\left(\boldsymbol{X}\boldsymbol{\beta}, \tau\boldsymbol{K}(\boldsymbol{U}) + \phi\boldsymbol{I}_n\right)\\ \boldsymbol{u}_1,\ldots,\boldsymbol{u}_m &\sim \mathcal{N}_n\left(\boldsymbol{0}, \xi_1\boldsymbol{K}^{\text{cov}} + \xi_2\boldsymbol{K}^{\text{ibs}}\right),\end{aligned}$$

and the LMM is based on the following model assumption:

$$\boldsymbol{y} \sim \mathcal{N}_n\left(\boldsymbol{X}\boldsymbol{\beta}, \tau_1\boldsymbol{K}^{\text{cov}} + \tau_2\boldsymbol{K}^{\text{ibs}} + \tau_3\boldsymbol{I}_n\right). \tag{3.16}$$

The restricted maximum likelihood estimates for τ_i, $i = 1, 2, 3$ were then calculated based on the Fisher scoring methods and the BLUP for the LMM were computed based on these estimators. Similarly, when we applied the KNN methods, the two kernel matrices $\boldsymbol{K}^{\text{cov}}$ and $\boldsymbol{K}^{\text{ibs}}$ were used as the input kernel matrices and the output kernel matrix is chosen to be either the product kernel or the polynomial kernel of order 2. The average mean square errors on predictors of both methods were summarized in the Table 3.1.

Table 3.1: Average mean squared prediction error of KNN with product output kernel matrix, KNN with output kernel matrix as polynomial of order 2 and the BLUP based on LMM.

	KNN(prod)	KNN(poly)	LMM
Hippocampus	2.01e-06	4.17e-05	2.11e-02
Ventricles	2.44e-03	1.98e-03	2.24e-01
Entorhinal	1.01e-05	1.06e-04	4.05e-02
Whole Brain Volume	1.25e-07	3.04e-08	3.68e-11

As we can see from the table, both KNN and LMM have good prediction errors. However, we would say that the prediction errors from KNN are more realistic. It is interesting to note that the average prediction errors of LMM for entorhinal and whole brain volume are

extremely close to zero. We checked the estimates of the variance components in this case and noticed that the variance component associated with the identity matrix, which is τ_3 as in (3.16). Since the BLUP of the random effect in this case is given by

$$BLUP = \boldsymbol{X}\hat{\boldsymbol{\beta}} + \left(\tau_1 \boldsymbol{K}^{\text{cov}} + \tau_2 \boldsymbol{K}^{\text{ibs}}\right)\left(\tau_1 \boldsymbol{K}^{\text{cov}} + \tau_2 \boldsymbol{K}^{\text{ibs}} + \tau_3 \boldsymbol{I}_n\right)^{-1}(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

so that if τ_3 is very close to 0, we can know that the BLUP is very close to the response $\boldsymbol{y}$, which leads to the extremely low prediction error. On the other hand, for KNN, since when the estimate of error variance becomes negative, we project the MINQUE matrix onto the positive semidefinite cone so that we will always get a positive estimate for the error variance component, which makes the calculation of prediction error more reasonable.

3.7 Discussion

In this chapter, a kernel-based neural network model was proposed for prediction analyses. The kernel-based neural network can be thought of as an extension of linear mixed model since it can reduce to a linear mixed model through choosing product kernel matrix as the output kernel matrix and via reparameterization. A modified MINQUE strategy is used to obtain the estimators of variance components in the KNN model if the output kernel satisfies the generalized linear separable condition. Empirical simulation studies and real data application show that the KNN model can achieve better performance in terms of the mean squared prediction error when the output kernel matrix is chosen as the polynomial kernel matrix. This is analogous to the popular neural network model, where better prediction accuracy can be achieved when nonlinear activation functions are applied.

Many extensions can be made to make the KNN model more flexible and have much broader applications. First, it may be possible to consider conducting base kernel matrix selection. Although in this paper, we do not consider how to choose the number of base kernel matrix L, but too many kernel matrices will certainly increase the amount of redundant information. So it may be beneficial to propose a criterion on choosing the base kernel matrices. An other possible extension of the KNN model is more challenging. The theoretical properties discussed in this paper mainly focus on the case where only one output kernel matrix is used and the kernel function has a specific form. It is natural to consider more general kernel functions, but the estimation procedure of the variance component would be more complex. Moreover, it is also advisable to consider the performance of KNN if deep network structures were applied.

Chapter 4

Asymptotic Properties of Neural Network Sieve Estimators

4.1 Introduction

With the widespread usage of machine learning and artificial intelligence, neural networks have regained their popularity. More and more learning machines are based on deep neural networks and have achieved great classification and prediction accuracy. We refer interested readers to Goodfellow et al. (2016) [25] for more background and details. In classical statistical learning theory, the consistency and the rate of convergence of the empirical risk minimization principle are of the great interest. Many upper bounds have been obtained for the empirical risk and the sample complexity based on the growth function and the Vapnik-Chervonenkis dimension (see for example, [70, 4, 18]). However, not many studies focus on the asymptotic properties for neural networks. As Thomas J. Sargent said, "artificial intelligence is actually statistics, but in a very gorgeous phrase, it is statistics." So it is natural and worthwhile to explore whether a neural network model possesses nice asymptotic properties since if it does, it may be possible to conduct statistical inference based on this model. Throughout this chapter, we will focus on the asymptotic properties of neural networks with one hidden layer.

Using the language in statistics, fitting a neural network with one hidden layer is simply a parametric nonlinear regression problem:

$$y_i = \alpha_0 + \sum_{j=1}^{r} \alpha_j \sigma\left(\boldsymbol{\gamma}_j^T \boldsymbol{x}_i + \gamma_{0,j}\right) + \epsilon_i,$$

where $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. random errors with $\mathbb{E}[\epsilon] = 0$ and $\mathbb{E}[\epsilon^2] = \sigma^2 < \infty$ and $\sigma(\cdot)$ is an activation function, for example $\sigma(z) = 1/(1 + e^{-z})$, which will be the main focus in this paper. White and Racine (2001)[75] obtained the asymptotic distribution of the resulting estimators under the assumption that the true parameters are unique. In fact, the authors implicitly assumed that the number of hidden units r is known. However, even if we assume that we know the number of hidden units, it is difficult to establish the asymptotic properties for the parameter estimators. In section 4.6.1, we conduct a simulation based on a single-layer neural network with 2 hidden units. Even for such a simple model, the simulation result suggests that reaching consistency is highly unlikely. Moreover, since the number of hidden units is usually unknown in practice, such assumption can be easily violated. For example, as pointed out in Fukumizu (1996)[21] and Fukumizu et al. (2003) [22], if the true function is $f_0(x) = \alpha\sigma(\gamma x)$, that is the true number of hidden units is 1, and if we fit the model using a neural network with two hidden units, then any parameter $\boldsymbol{\theta} = [\alpha_0, \alpha_1, \ldots, \alpha_r, \gamma_{0,1}, \ldots, \gamma_{0,r}, \boldsymbol{\gamma}_1^T, \ldots, \boldsymbol{\gamma}_r^T]^T$ in the high-dimensional set

$$\{\boldsymbol{\theta} : \gamma_1 = \gamma, \alpha_1 = \alpha, \gamma_{0,1} = \gamma_{0,2} = \alpha_2 = \alpha_0 = 0\} \cup$$

$$\{\boldsymbol{\theta} : \gamma_1 = \gamma_2 = \gamma, \gamma_{0,1} = \gamma_{0,2} = \alpha_0 = 0, \alpha_1 + \alpha_2 = \alpha\}$$

realizes the true function $f_0(x)$. Therefore, when the number of hidden units is unknown,

the parameters in this parametric nonlinear regression problem are unidentifiable. Theorem 1 in Wu (1981)[77] showed that a necessary condition for the weak consistency of nonlinear least square estimators is that

$$\sum_{i=1}^{n}[f(\boldsymbol{x}_i, \boldsymbol{\theta}) - f(\boldsymbol{x}_i, \boldsymbol{\theta}')]^2 \to \infty, \text{ as } n \to \infty,$$

for all $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$ in the parameter space as long as the distribution of the errors has finite Fisher information. Such condition implies that when the parameters are not identifiable, the resulting nonlinear least squares estimators will be inconsistent, which hinders further explorations on the asymptotic properties for the neural network estimators. Liu and Shao (2003)[43] and Zhu and Zhang (2006)[80] proposed some techniques to conduct hypothesis testing under loss of identifiability. However, their theoretical results are hard to implement in real world applications.

Even though a function can have different neural network parametrizations, the function itself can be considered as unique. Moreover, due to the Universal Approximation Theorem [33], any continuous function on a compact support can be approximated arbitrarily well by a neural network with one hidden layer. So it seems natural to consider a nonparametric regression setting and approximate the function class through a class of neural networks with one hidden layer. Specifically, suppose that the true nonparametric regression model is

$$y_i = f_0(\boldsymbol{x}_i) + \epsilon_i,$$

where $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. random variables defined on a complete probability space $(\Omega, \mathcal{A}, \mathbb{P})$ with $\mathbb{E}[\epsilon] = 0$, $\text{Var}[\epsilon] = \sigma^2 < \infty$; $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X} \subset \mathbb{R}^d$ are vectors of covariates with $\mathcal{X}$ being

a compact set in $\mathbb{R}^d$ and f_0 is an unknown function needed to be estimated. We assume that $f_0 \in \mathcal{F}$, where $\mathcal{F}$ is the class of continuous functions with compact supports. Clearly, f_0 minimizes the population criterion function

$$\begin{aligned} Q_n(f) &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2 + \sigma^2. \end{aligned}$$

A least squares estimator of the regression function can be obtained by minimizing the empirical squared error loss $\mathbb{Q}_n(f)$:

$$\hat{f}_n = \text{argmin}_{f\in\mathcal{F}}\mathbb{Q}_n(f) = \text{argmin}_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2.$$

However, if the class of functions $\mathcal{F}$ is too rich, the resulting least squares estimator may have undesired property such as inconsistency [68, 65, 64]. Instead, we may optimize the squared error loss over some less complex function space $\mathcal{F}_n$, which is an approximation of $\mathcal{F}$ while the approximation error tends to 0 as the sample size increases. In the language of Grenander (1981)[26], such a sequence of function classes is known as a sieve. More precisely, we consider a sequence of function classes

$$\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_n \subseteq \mathcal{F}_{n+1} \subseteq \cdots \subseteq \mathcal{F}$$

approximating $\mathcal{F}$ in the sense that $\bigcup_{n=1}^{\infty}\mathcal{F}_n$ is dense in $\mathcal{F}$, that is for each $f \in \mathcal{F}$, there exists $\pi_n f \in \mathcal{F}_n$ such that $d(f, \pi_n f) \to 0$ as $n \to \infty$, where $d(\cdot,\cdot)$ is some pseudo-metric defined on $\mathcal{F}$. With some abuse of notation, an approximate sieve estimator $\hat{f}_n$ is defined to

be

$$\mathbb{Q}_n(\hat{f}_n) \leq \inf_{f \in \mathcal{F}_n} \mathbb{Q}_n(f) + \mathcal{O}_p(\eta_n), \tag{4.1}$$

where $\eta_n \to 0$ as $n \to \infty$.

Throughout the rest of the chapter, we focus on the sieve of neural networks with one hidden layer and sigmoid activation function. Specifically, we let

$$\mathcal{F}_{r_n} = \left\{ \alpha_0 + \sum_{j=1}^{r_n} \alpha_j \sigma\left(\boldsymbol{\gamma}_j^T \boldsymbol{x} + \gamma_{0,j}\right) : \boldsymbol{\gamma}_j \in \mathbb{R}^d, \alpha_j, \gamma_{0,j} \in \mathbb{R}, \right.$$
$$\left. \sum_{j=0}^{r_n} |\alpha_j| \leq V_n \text{ for some } V_n > 4 \text{ and } \max_{1 \leq j \leq r_n} \sum_{i=0}^{d} |\gamma_{i,j}| \leq M_n \text{ for some } M_n > 0 \right\}, \tag{4.2}$$

where $r_n, V_n, M_n \uparrow \infty$ as $n \to \infty$. Such method has been discussed in many works (see for example [73] and [74]). In those papers, consistency of the neural network sieve estimators has been established under a random design. However, there are few results on the asymptotic distribution of the neural network sieve estimators, which will be established in this paper. Moreover, throughout this paper, we focus on the fixed design. [33] showed that $\bigcup_n \mathcal{F}_{r_n}$ is dense in $\mathcal{F}$ under the sup-norm. But when considering the asymptotic properties of the sieve estimators, we use the pseudo-norm $\|f\|_n^2 = n^{-1} \sum_{i=1}^n f^2(\boldsymbol{x}_i)$ (see Proposition C.0.1 in Appendix C) defined on $\mathcal{F}$ and $\mathcal{F}_{r_n}$.

In section 4.2, we discuss the existence of neural network sieve estimators. The weak consistency and rate of convergence of the neural network sieve estimators will be established in section 4.3 and section 4.4, respectively. Section 4.5 focuses on the asymptotic distribution of the neural network sieve estimators. Several simulations results are presented in section 4.6.

Notation: Throughout the rest of the chapter, bold font alphabetic letters and Greek letters are vectors. $C(\mathcal{X})$ is the set of continuous functions defined on $\mathcal{X}$. The symbol $\lesssim$ means "is bounded above up to a universal constant" and $a_n \sim b_n$ means $\frac{a_n}{b_n} \to 1$ as $n \to \infty$. For a pseudo-metric space (T, d), $N(\epsilon, T, d)$ is its covering number, that is the minimum number of ϵ-balls needed to cover T. Its natural logarithm is the entropy number and is denoted by $H(\epsilon, T, d)$.

4.2 Existence

A natural question to ask is whether or not the sieve estimator based on neural networks exists. Before we enter the main discussion, we first look at some simple properties of $\mathcal{F}_{r_n}$. Proposition 4.2.1 shows that the sigmoid function is a Lipschitz function with Lipschitz constant $L = 1/4$.

Proposition 4.2.1. *A sigmoid function $\sigma(z) = e^z/(1 + e^z)$ is a Lipschitz function on $\mathbb{R}$ with Lipschitz constant $1/4$.*

Proof. For all $z_1, z_2 \in \mathbb{R}$, we know that $\sigma(z)$ is continuous on $[z_1, z_2]$ and is differentiable on (z_1, z_2). Note that

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)) \leq \frac{1}{4} \quad \forall z \in \mathbb{R}.$$

By Mean Value Theorem, we know that

$$\sigma(z_1) - \sigma(z_2) = \sigma'(\lambda z_1 + (1 - \lambda) z_2)(z_1 - z_2)$$

for some $\lambda \in [0,1]$. Hence

$$|\sigma(z_1) - \sigma(z_2)| = |\sigma'(\lambda z_1 + (1-\lambda)z_2)||z_1 - z_2| \leq \frac{1}{4}|z_1 - z_2|,$$

which means that $\sigma(z)$ is a Lipschitz function on $\mathbb{R}$ with Lipschitz constant $1/4$. □

The second proposition provides an upper bound for the envelope function $\sup_{f \in \mathcal{F}_{r_n}} |f|$.

Proposition 4.2.2. *For each fixed n*

$$\sup_{f \in \mathcal{F}_{r_n}} \|f\|_\infty \leq V_n.$$

Proof. For any $f \in \mathcal{F}_{r_n}$ with n fixed, note that for all $\boldsymbol{x} \in \mathcal{X}$, we have

$$\begin{aligned} |f(\boldsymbol{x})| &= \left| \alpha_0 + \sum_{j=1}^{r_n} \alpha_j \sigma\left(\boldsymbol{\gamma}_j^T \boldsymbol{x} + \gamma_{0,j}\right) \right| \\ &\leq |\alpha_0| + \sum_{j=1}^{r_n} |\alpha_j| \sigma\left(\boldsymbol{\gamma}_j^T \boldsymbol{x} + \gamma_{0,j}\right) \leq \sum_{j=0}^{r_n} |\alpha_j| \leq V_n. \end{aligned}$$

Since the right hand side does not depend on $\boldsymbol{x}$ and f, we get

$$\sup_{f \in \mathcal{F}_{r_n}} \|f\|_\infty = \sup_{f \in \mathcal{F}_{r_n}} \sup_{\boldsymbol{x} \in \mathcal{X}} |f(\boldsymbol{x})| \leq V_n.$$

□

Now we quote a general result from White and Wooldridge (1991) [76]. The theorem tells us that under some mild conditions, there exists a sieve approximate estimator and that such an estimator is also measurable.

Theorem 4.2.1 (Theorem 2.2 in White and Wooldeidge (1991)[76]). *Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a complete probability space and let (Θ, ρ) be a pseudo-metric space. Let $\{\Theta_n\}$ be a sequence of compact subsets of Θ. Let $\mathbb{Q}_n : \Omega \times \Theta_n \to \bar{\mathbb{R}}$ be $\mathcal{A} \otimes \mathcal{B}(\Theta_n)/\mathcal{B}(\bar{\mathbb{R}})$-measurable, and suppose that for each $\omega \in \Omega$, $\mathbb{Q}_n(\omega, \cdot)$ is lower semicontinuous on Θ_n, $n = 1, 2, \ldots$. Then for each $n = 1, 2, \ldots$, there exists $\hat{\theta}_n : \Omega \to \Theta_n$, $\mathcal{A}/\mathcal{B}(\Theta_n)$-measurable such that for each $\omega \in \Omega$, $\mathbb{Q}_n(\omega, \hat{\theta}_n(\omega)) = \inf_{\theta \in \Theta_n} \mathbb{Q}_n(\omega, \theta)$.*

Note that

$$\begin{aligned}
\mathbb{Q}_n(f) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(f_0(\boldsymbol{x}_i) + \epsilon_i - f(\boldsymbol{x}_i))^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2 - 2\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) + \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2.
\end{aligned}$$

Since the randomness only comes from ϵ_i's, it is clear that $\mathbb{Q}_n$ is a measurable function and for a fixed ω, $\mathbb{Q}_n$ is continuous in f. Therefore, to show the existence of the sieve estimator, it suffices to show that $\mathcal{F}_{r_n}$ is compact in $C(\mathcal{X})$, which is proved in the following lemma.

Lemma 4.2.1. *Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$. Then for each fixed n, $\mathcal{F}_{r_n}$ is a compact set.*

Proof. For each fixed n, let $\boldsymbol{\theta}_n = [\alpha_0, \ldots, \alpha_{r_n}, \boldsymbol{\gamma}_{0,1}, \ldots, \gamma_{0,r_n}, \boldsymbol{\gamma}_1^T, \ldots, \boldsymbol{\gamma}_{r_n}^T]^T$ belong to $[-V_n, V_n]^{r_n+1} \times [-M_n, M_n]^{r_n(d+1)} := \Theta_n$. For n fixed, Θ_n is a bounded closed set and hence it is a compact

set in $\mathbb{R}^{r_n(d+2)+1}$. Consider a map

$$H : (\Theta_n, \|\cdot\|_2) \to (\mathcal{F}_{r_n}, \|\cdot\|_n)$$
$$\boldsymbol{\theta}_n \mapsto H(\boldsymbol{\theta}_n) = \alpha_0 + \sum_{j=1}^{r_n} \alpha_j \sigma\left(\boldsymbol{\gamma}_j^T \boldsymbol{x} + \gamma_{0,j}\right)$$

Note that $\mathcal{F}_{r_n} = H(\Theta_n)$. Therefore, to show that $\mathcal{F}_{r_n}$ is a compact set, it suffices to show that H is a continuous map due to the compactness of Θ_n. Let $\boldsymbol{\theta}_{1,n}, \boldsymbol{\theta}_{2,n} \in \Theta_n$, then

$$\begin{aligned}
&\|H(\boldsymbol{\theta}_{1,n}) - H(\boldsymbol{\theta}_{2,n})\|_n^2 \\
=&\frac{1}{n}\sum_{i=1}^{n}\left[\alpha_0^{(1)} + \sum_{j=1}^{r_n}\alpha_j^{(1)}\sigma\left({\boldsymbol{\gamma}_j^{(1)}}^T\boldsymbol{x}_i + \gamma_{0,j}^{(1)}\right) - \alpha_0^{(2)} - \sum_{j=1}^{r_n}\alpha_j^{(2)}\sigma\left({\boldsymbol{\gamma}_j^{(2)}}^T\boldsymbol{x}_i + \gamma_{0,j}^{(2)}\right)\right]^2 \\
\leq&\frac{1}{n}\sum_{i=1}^{n}\left[\left|\alpha_0^{(1)} - \alpha_0^{(2)}\right| + \sum_{j=1}^{r_n}\left|\alpha_j^{(1)}\sigma\left({\boldsymbol{\gamma}_j^{(1)}}^T\boldsymbol{x}_i + \gamma_{0,j}^{(1)}\right) - \alpha_j^{(2)}\sigma\left({\boldsymbol{\gamma}_j^{(2)}}^T\boldsymbol{x}_i + \gamma_{0,j}^{(2)}\right)\right|\right]^2 \\
=&\frac{1}{n}\sum_{i=1}^{n}\left[\left|\alpha_0^{(1)} - \alpha_0^{(2)}\right| + \sum_{j=1}^{r_n}|\alpha_j^{(1)}|\left|\sigma\left({\boldsymbol{\gamma}_j^{(1)}}^T\boldsymbol{x}_i + \gamma_{0,j}^{(1)}\right) - \sigma\left({\boldsymbol{\gamma}_j^{(2)}}^T\boldsymbol{x}_i + \gamma_{0,j}^{(2)}\right)\right| + \right. \\
&\qquad\qquad \left. |\alpha_j^{(1)} - \alpha_j^{(2)}|\sigma\left({\boldsymbol{\gamma}_j^{(2)}}^T\boldsymbol{x}_i + \gamma_{0,j}^{(2)}\right)\right]^2 \\
\leq&\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=0}^{r_n}|\alpha_j^{(1)} - \alpha_j^{(2)}| + \frac{V_n}{4}\sum_{j=1}^{r_n}\left|\left(\boldsymbol{\gamma}_j^{(1)} - \boldsymbol{\gamma}_j^{(2)}\right)^T\boldsymbol{x}_i\right| + \left|\gamma_{0,j}^{(1)} - \gamma_{0,j}^{(2)}\right|\right]^2 \\
\leq&\left[\sum_{j=0}^{r_n}|\alpha_j^{(1)} - \alpha_j^{(2)}| + \frac{V_n}{4}(1 \vee \|\boldsymbol{x}\|_\infty)\sum_{j=1}^{r_n}\left\|\boldsymbol{\gamma}_j^{(1)} - \boldsymbol{\gamma}_j^{(2)}\right\|_1 + \left|\gamma_{0,j}^{(1)} - \gamma_{0,j}^{(2)}\right|\right]^2 \\
\leq&\left(\frac{V_n}{4}(1 \vee \|\boldsymbol{x}\|_\infty)\right)^2 [r_n(d+1)]\|\boldsymbol{\theta}_{1,n} - \boldsymbol{\theta}_{2,n}\|_2^2.
\end{aligned}$$

Hence, for any $\epsilon > 0$, by choosing $\delta = \epsilon / \left(\frac{V_n}{4}(1 \vee \|\boldsymbol{x}\|_\infty)\sqrt{r_n(d+1)}\right)$, we observe that when

$\|\boldsymbol{\theta}_{1,n} - \boldsymbol{\theta}_{2,n}\|_2 < \delta$, we have

$$\|H(\boldsymbol{\theta}_{1,n}) - H(\boldsymbol{\theta}_{2,n})\|_n < \epsilon,$$

which implies that H is a continuous map and hence $\mathcal{F}_{r_n}$ is a compact set for each fixed n. □

As a simple corollary of Lemma 4.2.1 and Theorem 4.2.1, we can easily obtain the existence of sieve estimator.

Corollary 4.2.1. *Under the notations above, for each $n = 1, 2, \ldots$, there exists $\hat{f}_n : \Omega \to \mathcal{F}_{r_n}$, $\mathcal{A}/\mathcal{B}(\mathcal{F}_{r_n})$-measurable such that $\mathbb{Q}_n(\hat{f}_n(\omega)) = \inf_{f \in \mathcal{F}_{r_n}} \mathbb{Q}_n(f)$.*

4.3 Consistency

In this section, we are going to show the result on the consistency of the neural network sieve estimator. The consistency result leans heavily on the following Uniform Law of Large Numbers. We start by considering a simple case with $V_n \equiv V$ for all n. In such a case, $\bigcup_n \mathcal{F}_{r_n}$ is not dense in $\mathcal{F}$ but instead in a subset of $\mathcal{F}$ containing functions satisfying a certain smoothness condition.

Lemma 4.3.1. *Let $\epsilon_1, \ldots, \epsilon_n$ be i.i.d. sub-Gaussian random variables with sub-Gaussian parameter σ_0. Then if $[r_n(d+2)+1]\log[r_n(d+2)+1] = o(n)$, we have*

$$\sup_{f \in \mathcal{F}_{r_n}} |\mathbb{Q}_n(f) - Q_n(f)| \xrightarrow{p^*} 0.$$

Proof. For any $\delta > 0$, we have

$$\begin{aligned}
&\mathbb{P}^*\left(\sup_{f\in\mathcal{F}_{r_n}} |\mathbb{Q}_n(f) - Q_n(f)| > \delta\right) \\
=&\mathbb{P}^*\left(\sup_{f\in\mathcal{F}_{r_n}} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i^2 - \sigma^2 - 2\frac{1}{n}\sum_{i=1}^n \epsilon_i \left(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right)\right| > \delta\right) \\
\leq&\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n \epsilon_i^2 - \sigma^2\right| > \frac{\delta}{2}\right) + \mathbb{P}^*\left(\sup_{f\in\mathcal{F}_{r_n}} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right| > \frac{\delta}{4}\right) \\
:=&(I) + (II).
\end{aligned}$$

For (I), by Weak Law of Large Numbers, we know that there exists $N_1 > 0$ such that for all $n \geq N_1$ we have

$$(I) = \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n \epsilon_i^2 - \sigma^2\right| > \frac{\delta}{2}\right) < \frac{\delta}{2}.$$

Now, we are going to evaluate (II). From the sub-Gaussianity of $\epsilon_1, \ldots, \epsilon_n$, we know that $\epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))$ is also sub-Gaussian with mean 0 and sub-Gaussian parameter $\sigma_0|f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)|$. Hence, by Hoeffding inequality

$$\begin{aligned}
\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n \epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right| > \frac{\delta}{4}\right) &= \mathbb{P}\left(\left|\sum_{i=1}^n \epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right| > \frac{n\delta}{4}\right) \\
&\leq 2\exp\left\{-\frac{n^2\delta^2}{32\sigma_0^2\sum_{i=1}^n (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2}\right\}.
\end{aligned}$$

From Proposition 4.2.2, we know that $\sup_{f\in\mathcal{F}_{r_n}} \|f\|_n \leq V$. Hence, based on Corollary 8.3 in van de Geer (2000)[68], (II) will have an exponential bound if there exists some constant C and $\delta > 0, \sigma > 0$ satisfying $V > \delta/\sigma$ and

$$\sqrt{n}\delta \geq 2C\left(\int_{\delta/(8\sigma)}^V H^{1/2}(u, \mathcal{F}_{r_n}, \|\cdot\|_n)\mathrm{d}u \vee V\right). \tag{4.3}$$

Now, we are going to show that (4.3) holds in our case. It follows from Theorem 14.5 in Anthony and Barglett (2009)[4] that an upper bound of the covering number for $\mathcal{F}_{r_n}$ is given by

$$N(\epsilon, \mathcal{F}_{r_n}, \|\cdot\|_\infty) \leq \left(\frac{4e[r_n(d+2)+1]\left(\frac{1}{4}V\right)^2}{\epsilon\left(\frac{1}{4}V - 1\right)} \right)^{r_n(d+2)+1} := \tilde{A}_{r_n,d,V}\epsilon^{-[r_n(d+2)+1]},$$

where $\tilde{A}_{r_n,d,V} = \left(e[r_n(d+2)+1]V^2/(V-4)\right)^{r_n(d+2)+1}$. By letting

$$\begin{aligned} A_{r_n,d,V} &= \log \tilde{A}_{r_n,d,V} - [r_n(d+2)+1] \\ &= [r_n(d+2)+1]\left(\log \frac{e[r_n(d+2)+1]V^2}{V-4} - 1\right) \\ &= [r_n(d+2)+1]\log \frac{[r_n(d+2)+1]V^2}{V-4}, \end{aligned}$$

and note that $V^2 - eV + 4e \geq 0$ for all V, we get $\log \frac{[r_n(d+2)+1]V^2}{V-4} \geq \log \frac{V^2}{V-4} \geq \log \frac{e(V-4)}{V-4} = 1$ and then

$$\begin{aligned} H(\epsilon, \mathcal{F}_{r_n}, \|\cdot\|_\infty) &= \log N(\epsilon, \mathcal{F}_{r_n}, \|\cdot\|_\infty) \\ &= \log \tilde{A}_{r_n,d,V} + [r_n(d+2)+1]\log \frac{1}{\epsilon} \\ &\leq A_{r_n,d,V} + [r_n(d+2)+1]\frac{1}{\epsilon} \quad (\text{since } \log x \leq x-1 \text{ for all } x > 0) \\ &\leq A_{r_n,d,V}\left(1 + \frac{1}{\epsilon}\right). \end{aligned}$$

Note that

$$\|f\|_n^2 = \frac{1}{n}\sum_{i=1}^n f^2(\boldsymbol{x}_i) \leq \left(\sup_{\boldsymbol{x}} |f(\boldsymbol{x})|\right)^2 = \|f\|_\infty^2,$$

we have $H(\epsilon, \mathcal{F}_{r_n}, \|\cdot\|_n) \leq H(\epsilon, \mathcal{F}_{r_n}, \|\cdot\|_\infty)$. Then

$$\begin{aligned}
\int_{\delta/(8\sigma)}^{V} H^{1/2}(\epsilon, \mathcal{F}_{r_n}, \|\cdot\|_n)\mathrm{d}\epsilon &\leq A_{r_n,d,V}^{1/2} \int_0^V \left(1+\frac{1}{\epsilon}\right)^{1/2} \mathrm{d}\epsilon \\
&= A_{r_n,d,V}^{1/2} \left[\int_0^1 \left(1+\frac{1}{\epsilon}\right)^{1/2} \mathrm{d}\epsilon + \int_1^V \left(1+\frac{1}{\epsilon}\right)^{1/2} \mathrm{d}\epsilon\right] \\
&\leq A_{r_n,d,V}^{1/2} \left[\sqrt{2}\int_0^1 \epsilon^{-\frac{1}{2}} \mathrm{d}\epsilon + \sqrt{2}(V-1)\right] \\
&\leq A_{r_n,d,V}^{1/2} \left[2\sqrt{2} + 2\sqrt{2}(V-1)\right] \\
&= 2\sqrt{2} A_{r_n,d,V}^{1/2} V.
\end{aligned}$$

Clearly $2\sqrt{2}A_{r_n,d,V}^{1/2} V \geq V$ and under the assumption that $[r_n(d+2)+1]\log[r_n(d+2)+1] = o(n)$, we get for any $\delta > 0$ there exists $N_2 > 0$ such that for all $n \geq N_2$,

$$4\sqrt{2}V\left(\frac{1}{n}A_{r_n,d,V}\right)^{1/2} < \frac{\delta}{4},$$

i.e. (4.3) holds with $C = 1$ and $n \geq N_2$. Hence, based on Corollary 8.3 in van de Geer (2000)[68], we get for $n \geq N_2$,

$$\mathbb{P}^*\left(\sup_{f\in\mathcal{F}_{r_n}} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right| > \frac{\delta}{4} \wedge \frac{1}{n}\sum_{i=1}^n \epsilon_i^2 \leq \sigma^2\right) \leq \exp\left\{-\frac{n\delta^2}{64V^2}\right\}. \tag{4.4}$$

Since $\int_0^V H^{1/2}(\epsilon, \mathcal{F}_{r_n}, \|\cdot\|_n)\mathrm{d}\epsilon < \infty$, we may take $\sigma \to \infty$ in (4.4) to get

$$\mathbb{P}^*\left(\sup_{f\in\mathcal{F}_{r_n}} \left|\frac{1}{n}\sum_{i=1}^n \epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right| > \frac{\delta}{4}\right) \leq \exp\left\{-\frac{n\delta^2}{64V^2}\right\}.$$

Let $N_3 = \frac{64V^2}{\delta^2} \log \frac{2}{\delta}$, then for $n \geq \max\{N_2, N_3\}$, we have

$$(II) = \mathbb{P}^* \left(\sup_{f \in \mathcal{F}_{r_n}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) \right| > \frac{\delta}{4} \right) \leq \frac{\delta}{2}.$$

Thus we conclude that for any $\delta > 0$, by taking $n \geq \max\{N_1, N_2, N_3\}$, we have

$$\mathbb{P}^* \left(\sup_{f \in \mathcal{F}_{r_n}} |\mathbb{Q}_n(f) - Q_n(f)| > \delta \right) < \delta,$$

which proves the desired result. □

Remark 4.3.1. *Lemma 4.3.1 shows that if we have a fixed number of features, the desired Uniform Law of Large Numbers holds when the number of hidden units in the neural network sieve does not grow too fast.*

Now, we are going to extend the result to a more general case. In Lemma 4.3.1, we assumed that the errors $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. sub-Gaussian and $V_n \equiv V$. In the following lemma, we are going to relax both restrictions.

Lemma 4.3.2. *Under the assumption that*

$$[r_n(d+2)+1] V_n^2 \log(V_n[r_n(d+2)+1] = o(n), \ \textit{as } n \to \infty,$$

we have

$$\sup_{f \in \mathcal{F}_{r_n}} |\mathbb{Q}_n(f) - Q_n(f)| \xrightarrow{p^*} 0, \ \textit{as } n \to \infty.$$

Proof. As in the proof of Lemma 4.3.1, it suffices to show that

$$\mathbb{P}^*\left(\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i))\right|>\frac{\delta}{4}\right)\to 0,\ \text{as } n\to\infty. \tag{4.5}$$

By Markov's inequality, (4.5) holds if we can show

$$\mathbb{E}^*\left[\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i))\right|\right]\to 0,\ \text{as } n\to\infty.$$

Now, since $\mathbb{E}[\epsilon]=0$ and note that for each $f\in\mathcal{F}_{r_n}$, it has its corresponding parametrization $\boldsymbol{\theta}_n$. Since $\boldsymbol{\theta}_n$ is in a compact set, we know that there exists a sequence $\boldsymbol{\theta}_{n,k}\to\boldsymbol{\theta}_n$ as $k\to\infty$ with $\boldsymbol{\theta}_{n,k}\in\mathbb{Q}^{r_n(d+2)+1}\cap([-V_n,V_n]^{r_n+1}\times[-M_n,M_n]^{r_n(d+1)})$. Each $\boldsymbol{\theta}_{n,k}$ corresponds to a function $f_k\in\mathcal{F}_{r_n}$. By continuity, we have $f_k(\boldsymbol{x})\to f(\boldsymbol{x})$ for each $\boldsymbol{x}\in\mathcal{X}$. By Example 2.3.4 in van der Vaart and Wellner (1996)[69], we know that $\mathcal{F}_{r_n}$ is P-measurable and by symmetrization inequality we have

$$\begin{aligned}&\mathbb{E}^*\left[\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i))\right|\right]\\ \le&2\mathbb{E}_\epsilon\mathbb{E}_\xi\left[\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i\epsilon_i\left(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)\right)\right|\right],\end{aligned}$$

where $\xi_1,\ldots,\xi_n$ are i.i.d. Rademacher random variables independent of $\epsilon_1,\ldots,\epsilon_n$. Based on the Strong Law of Large Numbers, there exists $N_1>0$, such that for all $n\ge N_1$,

$$\frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2<\sigma^2+1,\ \text{a.s.}$$

For fixed $\epsilon_1,\ldots,\epsilon_n$, $\sum_{i=1}^{n}\xi_i\epsilon_i(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i))$ is a sub-Gaussian process indexed by $f\in\mathcal{F}_{r_n}$. Suppose that $(\Xi,\mathcal{C},\mu)$ is the probability space on which $\xi_1,\ldots,\xi_n$ are defined and let

$Y(f,\omega) = \sum_{i=1}^n \xi_i(\omega)\epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))$ with $f \in \mathcal{F}_{r_n}$ and $\omega \in \Xi$. As we have shown above, we have $f_k \to f$ and by continuity, $Y(f_k,\omega) \to Y(f,\omega)$ for any $\omega \in \Xi$. This shows that $\{Y(f,\omega), f \in \mathcal{F}_{r_n}\}$ is a separable sub-Gaussian process. Hence Corollary 2.2.8 in van der Vaart and Wellner (1996)[69] implies that there exists a universal constant K and for any $f_n^* \in \mathcal{F}_{r_n}$ with $n \geq N_1$,

$$
\begin{aligned}
&\mathbb{E}_\xi\left[\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{n}\sum_{i=1}^n \xi_i\epsilon_i(f(\boldsymbol{x}_i)) - f_0(\boldsymbol{x}_i))\right|\right] \\
&\quad= \mathbb{E}_\xi\left[\frac{1}{\sqrt{n}}\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^n \xi_i\epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right|\right] \\
&\quad\leq \mathbb{E}_\xi\left[\left|\frac{1}{n}\sum_{i=1}^n \xi_i\epsilon_i(f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right|\right] + K\int_0^\infty \sqrt{\frac{\log N\left(\frac{1}{2}\eta, \mathcal{F}_{r_n}, d\right)}{n}}\mathrm{d}\eta \\
&\quad= \mathbb{E}_\xi\left[\left|\frac{1}{n}\sum_{i=1}^n \xi_i\epsilon_i(f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right|\right] + K\int_0^{2V_n} \sqrt{\frac{\log N\left(\frac{1}{2}\eta, \mathcal{F}_{r_n}, d\right)}{n}}\mathrm{d}\eta \\
&\quad\leq \mathbb{E}_\xi\left[\left|\frac{1}{n}\sum_{i=1}^n \xi_i\epsilon_i(f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right|\right] + K\int_0^{2V_n} \sqrt{\frac{\log N\left(\frac{1}{2\sqrt{\sigma^2+1}}\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty\right)}{n}}\mathrm{d}\eta,
\end{aligned}
$$

where the second equality follows by Proposition 4.2.2 and for $f, g \in \mathcal{F}_{r_n}$,

$$
\begin{aligned}
d(f,g) &= \left[\sum_{i=1}^n\left(\frac{1}{\sqrt{n}}\epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) - \frac{1}{\sqrt{n}}\epsilon_i(g(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right)^2\right]^{1/2} \\
&= \left(\frac{1}{n}\sum_{i=1}^n \epsilon_i^2(f(\boldsymbol{x}_i) - g(\boldsymbol{x}_i))^2\right)^{1/2}
\end{aligned}
$$

so that the last inequality follows by noting that

$$
d(f,g) \leq \|f-g\|_\infty\left(\frac{1}{n}\sum_{i=1}^n \epsilon_i^2\right)^{1/2}.
$$

Now we are going to evaluate these two terms. For the first term, for $n \geq N_1$, by Cauchy-Schwarz inequality, we have

$$\begin{aligned}
\mathbb{E}_\xi \left[\left| \frac{1}{n} \sum_{i=1}^{n} \xi_i \epsilon_i (f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) \right| \right] &\leq \frac{1}{n} \sum_{i=1}^{n} |\epsilon_i| |f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)| \\
&\leq \left(\frac{1}{n} \sum_{i=1}^{n} \epsilon_i^2 \right)^{1/2} \left(\frac{1}{n} \sum_{i=1}^{n} (f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2 \right)^{1/2} \\
&\leq \sqrt{\sigma^2 + 1} \sup_{\boldsymbol{x} \in \mathcal{X}} |f_n^*(\boldsymbol{x}) - f_0(\boldsymbol{x})|, \text{ a.s.}
\end{aligned}$$

By choosing $f_n^* = \pi_{r_n} f_0$ and from the universal approximation theorem of neural network by Hornik et al. (1989)[33], we know that $\sup_{x \in \mathcal{X}} |f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)| \to 0$ as $n \to \infty$ so that for any $\zeta > 0$, there exists $N_2 > 0$, such that for all $n \geq N_2$,

$$\sup_{x \in \mathcal{X}} |f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)| < \frac{\zeta}{\sqrt{\sigma^2 + 1}}.$$

Therefore, by choosing $n \geq N_1 \vee N_2$, we get

$$\mathbb{E}_\xi \left[\left| \frac{1}{n} \sum_{i=1}^{n} \xi_i \epsilon_i (f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) \right| \right] < \zeta \text{ a.s.}$$

For the second term, we use the same bound from Theorem 14.5 in Anthony and Bartlett (2009)[4] as we did in the proof of Lemma 4.3.1:

$$\begin{aligned}
N\left(\frac{1}{2\sqrt{\sigma^2 + 1}} \eta, \mathcal{F}_{r_n}, \| \cdot \|_\infty \right) &\leq \left(\frac{8\sqrt{\sigma^2 + 1} e [r_n(d+2) + 1] \left(\frac{1}{4} V_n \right)^2}{\eta \left(\frac{1}{4} V_n - 1 \right)} \right)^{r_n(d+2)+1} \\
&:= \tilde{B}_{r_n, d, V_n} \eta^{-[r_n(d+2)+1]},
\end{aligned}$$

where $\tilde{B}_{r_n,d,V_n} = \left(2\sqrt{\sigma^2+1}e[r_n(d+2)+1]V_n^2/(V_n-4)\right)^{r_n(d+2)+1}$. By letting

$$\begin{aligned}
B_{r_n,d,V_n} &= \log \tilde{B}_{r_n,d,V_n} - [r_n(d+2)+1] \\
&= [r_n(d+2)+1]\left(\log \frac{2\sqrt{\sigma^2+1}e[r_n(d+2)+1]V_n^2}{V_n-4} - 1\right) \\
&= [r_n(d+2)+1]\left(\log \frac{[r_n(d+2)+1]V_n^2}{V_n-4} + \log(2\sqrt{\sigma^2+1})\right) \\
&\leq 2[r_n(d+2)+1]\log \frac{[r_n(d+2)+1]V_n^2}{V_n-4}, \text{ for all } n \geq N_1 \vee N_3,
\end{aligned}$$

where by choosing N_3 so that $r_n(d+2)+1 \geq 2\sqrt{\sigma^2+1}$, the last inequality follows by noting that $V_n^2 - V_n + 4 \geq 0$ for all V_n so that $\log \frac{[r_n(d+1)+1]V_n^2}{V_n-4} \geq \log \frac{2\sqrt{\sigma^2+1}(V_n-4)}{V_n-4} = \log(2\sqrt{\sigma^2+1})$. We also have for

$$\begin{aligned}
H\left(\frac{1}{2\sqrt{\sigma^2+1}}\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty\right) &= \log N\left(\frac{1}{2\sqrt{\sigma^2+1}}\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty\right) \\
&= \log \tilde{B}_{r_n,d,V_n} + [r_n(d+2)+1]\log\frac{1}{\eta} \\
&\leq B_{r_n,d,V_n} + [r_n(d+2)+1]\frac{1}{\eta} \\
&\leq B_{r_n,d,V_n}\left(1+\frac{1}{\eta}\right)
\end{aligned}$$

and hence for all $n \geq N_1 \vee N_3$,

$$\begin{aligned}
&\int_0^{2V_n} H^{1/2}\left(\frac{1}{2\sqrt{\sigma^2+1}}\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty\right) \mathrm{d}\eta \\
&\quad \leq B_{r_n,d,V_n}^{1/2} \int_0^{2V_n} \left(1+\frac{1}{\eta}\right)^{1/2} \mathrm{d}\eta \\
&\quad = B_{r_n,d,V_n}^{1/2} \left[\int_0^1 \left(1+\frac{1}{\eta}\right)^{1/2} \mathrm{d}\eta + \int_1^{2V_n} \left(1+\frac{1}{\eta}\right)^{1/2} \mathrm{d}\eta\right] \\
&\quad \leq B_{r_n,d,V_n}^{1/2} \left[\sqrt{2}\int_0^1 \eta^{-1/2} \mathrm{d}\eta + \sqrt{2}(2V_n - 1)\right] \\
&\quad \leq 4\sqrt{2} B_{r_n,d,V_n}^{1/2} V_n,
\end{aligned}$$

which implies that

$$\begin{aligned}
\int_0^{2V_n} \sqrt{\frac{H\left(\frac{1}{2\sqrt{\sigma^2+1}}\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty\right)}{n}} \mathrm{d}\eta &\leq 4\sqrt{2} n^{-1/2} B_{r_n,d,V_n}^{1/2} V_n \\
&\sim 8\sqrt{\frac{[r_n(d+2)+1]V_n^2 \log(V_n[r_n(d+2)+1])}{n}},
\end{aligned}$$

where the last part follows by noting that $\log \frac{V_n^2}{V_n - 4} \sim \log V_n$. Under the assumption given in the Lemma, there exists $N_4 > 0$, such that for all $n \geq N_4$, we have

$$\sqrt{\frac{[r_n(d+2)+1]V_n^2 \log(V_n[r_n(d+2)+1])}{n}} < \frac{\zeta}{8}.$$

Therefore, by choosing $n \geq N_1 \vee N_2 \vee N_3 \vee N_4$, we get

$$\mathbb{E}_\xi\left[\sup_{f \in \mathcal{F}_{r_n}} \left|\frac{1}{n}\sum_{i=1}^n \xi_i \epsilon_i (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right|\right] < 2\zeta \text{ a.s.},$$

i.e. $\mathbb{E}_\xi\left[\sup_{f \in \mathcal{F}_{r_n}} \left|\frac{1}{n}\sum_{i=1}^n \xi_i \epsilon_i (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))\right|\right] \to 0$ a.s.. Moreover, based on what we

have shown, we know that for n sufficiently large,

$$\mathbb{E}_\xi\left[\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i\epsilon_i(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i))\right|\right] \leq \sqrt{\sigma^2+1}\|\pi_{r_n}f_0-f_0\|_\infty + 4\sqrt{2}KB^{1/2}_{r_n,d,V_n}n^{-1/2}V_n \to 0, \text{ a s.}$$

and since $\mathbb{E}_\epsilon\left[\sqrt{\sigma^2+1}\|\pi_{r_n}f_0-f_0\|_\infty + 4\sqrt{2}KB^{1/2}_{r_n,d,V_n}n^{-1/2}V_n\right] = \sqrt{\sigma^2+1}\|\pi_{r_n}f_0-f_0\|_\infty + 4\sqrt{2}KB^{1/2}_{r_n,d,V_n}n^{-1/2}V_n \to 0 < \infty$, by Generalized Dominated Convergence Theorem, we know that

$$\begin{aligned}&\mathbb{E}^*\left[\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i))\right|\right]\\ &\leq 2\mathbb{E}_\epsilon\mathbb{E}_\xi\left[\sup_{f\in\mathcal{F}_{r_n}}\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i\epsilon_i\left(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)\right)\right|\right]\to 0,\end{aligned}$$

which completes the proof. □

Based on the above lemmas, we are ready to state the theorem on the consistency of neural network sieve estimators.

Theorem 4.3.1. *Under the notation given above, if*

$$[r_n(d+2)+1]V_n^2\log(V_n[r_n(d+2)+1] = o(n), \text{ as } n\to\infty, \tag{4.6}$$

then

$$\|\hat{f}_n - f_0\|_n \xrightarrow{p} 0.$$

Proof. Since Q is continuous at $f_0 \in \mathcal{F}$ and $Q(f_0) = \sigma^2 < \infty$, we note that for any $\epsilon > 0$,

$$\inf_{f:\|f-f_0\|_n \geq \epsilon} Q_n(f) - Q_n(f_0) = \inf_{f:\|f-f_0\|_n \geq \epsilon} \frac{1}{n} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2 \geq \epsilon^2 > 0.$$

Hence, it follows from Lemma 4.2.1, Lemma 4.3.2 and Corollary 2.6 in White and Wooldridge (1991)[76] that

$$\|\hat{f}_n - f_0\|_n \xrightarrow{p} 0.$$

□

Remark 4.3.2. *We discuss the condition (4.6) in Theorem 4.3.1 via some simple examples here. If $\alpha_j = \mathcal{O}(1)$ for $j = 1, \ldots, r_n$, then $V_n = \mathcal{O}(r_n)$ and*

$$[r_n(d+2)+1]V_n^2 \log(V_n[r_n(d+2)+1]) = \mathcal{O}(r_n^3 \log r_n).$$

Therefore, a possible growth rate for the number of hidden units in a neural network is $r_n = o\left((n/\log n)^{1/3}\right)$. On the other hand, if we have a slow growth rate for the number of hidden units in the neural network, such as $r_n = \log V_n$, then we have

$$[r_n(d+2)+1]V_n^2 \log(V_n[r_n(d+2)+1]) = \mathcal{O}((V_n \log V_n)^2).$$

Hence, a possible growth rate for the upper bound of the weights from the hidden layer to the output layer is $V_n = o\left(n^{1/2}/\log n\right)$.

4.4 Rate of Convergence

To obtain the rate of convergence for neural network sieves, we are going to apply Theorem 3.4.1 in van der Vaart and Wellner (1996)[69].

Lemma 4.4.1. *Let $f_n^* = \pi_{r_n} f_0 \in \mathcal{F}_{r_n}$. Then under the notations given above, for every n and $\delta > 8\|f_n^* - f_0\|_n$, we have*

$$\sup_{\frac{\delta}{2} < \|f - f_n^*\|_n \leq \delta, f \in \mathcal{F}_{r_n}} Q_n(f_n^*) - Q_n(f) \lesssim -\delta^2.$$

Proof. First, we note that

$$\begin{aligned} Q_n(f_n^*) - Q_n(f) &= \frac{1}{n}\sum_{i=1}^{n}(f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2 + \sigma^2 - \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2 - \sigma^2 \\ &= \|f_n^* - f_0\|_n^2 - \|f - f_0\|_n^2. \end{aligned}$$

So to show the result, we need to provide an upper bound for $Q_n(f_n^*) - Q_n(f)$ in terms of $\|f - f_n^*\|_n$. Due to the fact that $\|\cdot\|_n$ is a pseudo-norm, the triangle inequality gives

$$\begin{aligned} \|f - f_n^*\|_n &\leq \|f - f_0\|_n + \|f_n^* - f_0\|_n \\ &= \|f - f_0\|_n - \|f_n^* - f_0\|_n + 2\|f_n^* - f_0\|_n. \end{aligned}$$

Therefore, we have

$$\|f - f_0\|_n - \|f_n^* - f_0\|_n \geq \|f - f_n^*\| - 2\|f_n^* - f_0\|_n,$$

so that for every f such that $\|f - f_n^*\|_n^2 \geq 16\|f_n^* - f_0\|_n^2$, i.e., $\|f - f_n^*\|_n \geq 4\|f_n^* - f_0\|_n$,

$$\|f - f_0\|_n - \|f_n^* - f_0\|_n \geq \|f - f_n^*\|_n - \frac{1}{2}\|f - f_n^*\|_n = \frac{1}{2}\|f - f_n^*\|_n \geq 0.$$

By squaring both sides, we obtain

$$\begin{aligned}
\frac{1}{4}\|f - f_n^*\|_n^2 &\leq \|f - f_0\|_n^2 + \|f_n^* - f_0\|_n^2 - 2\|f - f_0\|_n\|f_n^* - f_0\|_n \\
&\leq \|f - f_0\|_n^2 + \|f_n^* - f_0\|_n^2 - 2\|f_n^* - f_0\|_n^2 \\
&= \|f - f_0\|_n^2 - \|f_n^* - f_0\|_n^2
\end{aligned}$$

and hence

$$\begin{aligned}
\sup_{\frac{\delta}{2} < \|f - f_n^*\|_n \leq \delta, f \in \mathcal{F}_{r_n}} Q_n(f_n^*) - Q_n(f) &\leq \sup_{\|f - f_n^*\|_n > \frac{\delta}{2}, f \in \mathcal{F}_{r_n}} \|f_n^* - f_0\|_n^2 - \|f - f_0\|_n^2 \\
&\leq \sup_{\|f - f_n^*\|_n > \frac{\delta}{2}, f \in \mathcal{F}_{r_n}} \left(-\frac{1}{4}\|f - f_n^*\|_n^2 \right) \\
&\lesssim -\delta^2.
\end{aligned}$$

□

Lemma 4.4.2. *For every sufficiently large n and $\delta > 8\|f_n^* - f_0\|_n$, we have*

$$\mathbb{E}^* \left[\sup_{\frac{\delta}{2} < \|f - f_n^*\|_n \leq \delta, f \in \mathcal{F}_{r_n}} \sqrt{n} \left[(\mathbb{Q}_n - Q_n)(f_n^*) - (\mathbb{Q}_n - Q_n)(f) \right]^+ \right] \lesssim \int_0^\delta \sqrt{\log N(\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty)} d\eta$$

Proof. Note that

$$(\mathbb{Q}_n - Q_n)(f_n^*) = \frac{1}{n}\sum_{i=1}^n \epsilon_i^2 - \sigma^2 - \frac{2}{n}\sum_{i=1}^n \epsilon_i (f_n^*(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))$$
$$(\mathbb{Q}_n - Q_n)(f_n^*) = \frac{1}{n}\sum_{i=1}^n \epsilon_i^2 - \sigma^2 - \frac{2}{n}\sum_{i=1}^n \epsilon_i (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)),$$

we have

$$(\mathbb{Q}_n - Q_n)(f_n^*) - (\mathbb{Q}_n - Q_n)(f) = \frac{2}{n}\sum_{i=1}^n \epsilon_i (f(\boldsymbol{x}_i) - f_n^*(\boldsymbol{x}_i)).$$

Hence by following similar arguments as in the proof of Lemma 4.3.2 as well as applying Corollary 2.2.8 in van der Vaart and Wellner (1996)[69], we have

$$\begin{aligned}
&\mathbb{E}^* \left[\sup_{\frac{\delta}{2} < \|f - f_n^*\|_n \leq \delta, f \in \mathcal{F}_{r_n}} \sqrt{n} \left[(\mathbb{Q}_n - Q_n)(f_n^*) - (\mathbb{Q}_n - Q_n)(f) \right]^+ \right] \\
\leq &\mathbb{E}^* \left[\sup_{\frac{\delta}{2} < \|f - f_n^*\|_n \leq \delta, f \in \mathcal{F}_{r_n}} \left| \frac{1}{\sqrt{n}} \sum_{i=1}^n \epsilon_i (f(\boldsymbol{x}_i) - f_n^*(\boldsymbol{x}_i)) \right| \right] \\
\leq &2\mathbb{E}_\epsilon \mathbb{E}_\xi \left[\sup_{\frac{\delta}{2} < \|f - f_n^*\|_n \leq \delta, f \in \mathcal{F}_{r_n}} \left| \frac{1}{\sqrt{n}} \sum_{i=1}^n \xi_i \epsilon_i \left(f(\boldsymbol{x}_i) - f_n^*(\boldsymbol{x}_i) \right) \right| \right] \\
\lesssim &\int_0^\delta \sqrt{\log N(\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty)} \mathrm{d}\eta,
\end{aligned}$$

where the last inequality follows since $f_n^* \in \mathcal{F}_{r_n}$ for n large enough. □

Now we are ready to apply Theorem 3.4.1 in van der Vaart and Wellner (1996)[69] to obtain the rate of convergence for neural network sieve estimators.

Theorem 4.4.1. *Under the above notations, if*

$$\eta_n = \mathcal{O}\left(\min\{ \|\pi_{r_n} f_0 - f_0\|_n^2, r_n(d+2)\log(r_n V_n(d+2))/n, r_n(d+2)\log n/n \} \right),$$

then

$$\|\hat{f}_n - f_0\|_n = \mathcal{O}_p\left(\max\left\{\|\pi_{r_n} f_0 - f_0\|_n, \sqrt{\frac{r_n(d+2)\log[r_n V_n(d+2)]}{n}}, \sqrt{\frac{r_n(d+2)\log n}{n}}\right\}\right).$$

Proof. Use the same bound from Theorem 14.5 in Anthony and Bartlett (2009)[4] as we did before, we have

$$\begin{aligned}
\log N(\eta, \mathcal{F}_{r_n}, \|\cdot\|_n) &\leq \log N(\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty) \\
&\leq \log\left(\frac{4e[r_n(d+2)+1]\left(\frac{1}{4}V_n\right)^2}{\eta\left(\frac{1}{4}V_n - 1\right)}\right)^{r_n(d+2)+1} \\
&= [r_n(d+2)+1]\log\frac{\tilde{C}_{r_n,d,V_n}}{\eta},
\end{aligned}$$

where $\tilde{C}_{r_n,d,V_n} = \frac{e[r_n(d+2)+1]V_n^2}{V_n - 4} > e$. Then it follows from Lemma 3.8 in Mendelson (2003)[48] that for $\delta < 1$,

$$\begin{aligned}
\int_0^\delta \sqrt{\log N(\eta, \mathcal{F}_{r_n}, \|\cdot\|_n)}\mathrm{d}\eta &\leq [r_n(d+2)+1]^{1/2}\int_0^\delta \sqrt{\log\frac{\tilde{C}_{r_n,d,V_n}}{\eta}}\mathrm{d}\eta \\
&\lesssim [r_n(d+2)+1]^{1/2}\delta\sqrt{\log\frac{\tilde{C}_{r_n,d,V_n}}{\delta}} \\
&:= \phi_n(\delta).
\end{aligned}$$

Define $h : \delta \mapsto \phi_n(\delta)/\delta^\alpha = [r_n(d+2)+1]^{1/2}\delta^{1-\alpha}\sqrt{\log \frac{\tilde{C}_{r_n,d,V_n}}{\delta}}$, then since for $0 < \delta < 1$

$$\begin{aligned}
h'(\delta) &= [r_n(d+2)+1]^{1/2}\left((1-\alpha)\delta^{-\alpha}\sqrt{\log\frac{\tilde{C}_{r_n,d,V_n}}{\delta}} - \frac{1}{2}\frac{\delta^2}{\tilde{C}_{r_n,d,V_n}}\frac{\tilde{C}_{r_n,d,V_n}}{\delta^2}\log^{-1/2}\frac{\tilde{C}_{r_n,d,V_n}}{\delta}\right) \\
&= [r_n(d+2)+1]^{1/2}\left((1-\alpha)\delta^{-\alpha}\sqrt{\log\frac{\tilde{C}_{r_n,d,V_n}}{\delta}} - \frac{1}{2}\log^{-1/2}\frac{\tilde{C}_{r_n,d,V_n}}{\delta}\right) \\
&< 0
\end{aligned}$$

for $1 < \alpha < 2$, we can know that $\delta \mapsto \phi_n(\delta)/\delta^\alpha$ is decreasing on $(0,\infty)$. Let $\rho_n \lesssim \|\pi_{r_n} f_0 - f_0\|_n^{-1}$, then note that

$$\begin{aligned}
\rho_n^2\phi_n\left(\frac{1}{\rho_n}\right) &= \rho_n[r_n(d+2)+1]^{1/2}\log^{1/2}\left(\rho_n\tilde{C}_{r_n,d,V_n}\right) \\
&= [r_n(d+2)+1]^{1/2}\rho_n\sqrt{\log\rho_n + \log\tilde{C}_{r_n,d,V_n}}
\end{aligned}$$

and

$$\begin{aligned}
\log\tilde{C}_{r_n,d,V_n} &= 1 + \log\frac{[r_n(d+2)+1]V_n^2}{V_n-4} \lesssim \log\frac{[r_n(d+2)+1]V_n^2}{V_n-4} \\
&\sim \log[r_nV_n(d+2)],
\end{aligned}$$

we have

$$\rho_n^2\phi_n\left(\frac{1}{\rho_n}\right) \lesssim \sqrt{n} \Leftrightarrow r_n(d+2)\rho_n^2\left(\log\rho_n + \log[r_nV_n(d+2)]\right) \lesssim n.$$

This shows that for

$$\rho_n \lesssim \min\left\{\left(\frac{n}{r_n(d+2)\log[r_nV_n(d+2)]}\right)^{1/2}, \left(\frac{n}{r_n(d+2)\log n}\right)^{1/2}\right\},$$

we have $\rho_n^2\phi_n\left(\frac{1}{\rho_n}\right) \lesssim \sqrt{n}$. Based on these observation, Lemma 4.4.1, Lemma 4.4.2 and Theorem 3.4.1 in van der Vaart and Wellner (1996)[69] imply that

$$\|\hat{f}_n - \pi_{r_n}f_0\|_n = \mathcal{O}_p\left(\max\left\{\|\pi_{r_n}f_0 - f_0\|_n, \sqrt{\frac{r_n(d+2)\log[r_nV_n(d+2)]}{n}}, \sqrt{\frac{r_n(d+2)\log n}{n}}\right\}\right).$$

By triangle inequality, we can further get

$$\begin{aligned}\|\hat{f}_n - f_0\|_n &\le \|\hat{f}_n - \pi_{r_n}f_0\|_n + \|\pi_{r_n}f_0 - f_0\|_n \\ &= \mathcal{O}_p\left(\max\left\{\|\pi_{r_n}f_0 - f_0\|_n, \sqrt{\frac{r_n(d+2)\log[r_nV_n(d+2)]}{n}}, \sqrt{\frac{r_n(d+2)\log n}{n}}\right\}\right).\end{aligned}$$

□

Remark 4.4.1. *Recall that a sufficient condition to ensure consistency is* $r_n(d+2)V_n^2\log[r_nV_n(d+2)] = o(n)$. *In this case,* $r_n(d+2)\log[r_nV_n(d+2)] \le n$ *and the rate of convergence can be simplified to*

$$\|\hat{f}_n - f_0\|_n = \mathcal{O}_p\left(\max\left\{\|\pi_{r_n}f_0 - f_0\|_n, \sqrt{\frac{r_n(d+2)\log n}{n}}\right\}\right).$$

If we assume $f_0 \in \mathcal{F}$ where $\mathcal{F}$ is the space of functions, which have finite first absolute

moments of the Fourier magnitude distributions, i.e.,

$$\mathcal{F} = \left\{ f : \mathbb{R}^d \to \mathbb{R} : f(\boldsymbol{x}) = \int \exp\left\{ i\boldsymbol{a}^T\boldsymbol{x} \right\} \mathrm{d}\mu_f(\boldsymbol{a}), \right.$$
$$\left. \|\mu_f\|_1 := \int \max(\|\boldsymbol{a}\|_1, 1)\mathrm{d}|\mu_f|(\boldsymbol{a}) \leq C \right\}, \tag{4.7}$$

where μ_f is a complex measure on $\mathbb{R}^d$; $|\mu_f|$ denotes the total variation of μ_f, i.e., $|\mu|(A) = \sup \sum_{n=1}^{\infty} |\mu(A_n)|$ and the supremum is taken over all measurable partitions $\{A_n\}_{n=1}^{\infty}$ of A; and $\|\boldsymbol{a}\|_1 = \sum_{i=1}^{d} |a_i|$ for $\boldsymbol{a} = [a_1, \ldots, a_d]^T \in \mathbb{R}^d$. Theorem 3 in Markovoz (1996)[45] shows that $\delta_n := \|f_0 - \pi_{r_n} f_0\|_n \lesssim r_n^{-1/2-1/(2d)}$. So if we let d fixed and $\rho_n = \delta_n^{-1}$ and $V_n \equiv V$ in the proof of Theorem 4.4.1, δ_n must also satisfy the following inequality:

$$\begin{aligned}
& \rho_n^2 \phi\left(\frac{1}{\rho_n}\right) \lesssim \rho_n r_n^{1/2} \log^{1/2}\left(\rho_n \tilde{C}_{r_n,d,V_n}\right) \lesssim \sqrt{n} \\
\Rightarrow \quad & \rho_n^2 r_n \log \rho_n + \rho_n^2 r_n \log r_n \lesssim n \\
\Rightarrow \quad & \delta_n^{-2}\left(-r_n \log \delta_n + r_n \log r_n\right) \lesssim n \\
\Rightarrow \quad & r_n^{1+\frac{1}{d}} r_n \log r_n \lesssim n
\end{aligned}$$

One possible choice of r_n to satisfy such condition is $r_n \asymp (n/\log n)^{\frac{d}{2+d}}$. In such a case, we obtain

$$\|\hat{f}_n - f_0\|_n = \mathcal{O}_p\left(\left(\frac{n}{\log n}\right)^{-\frac{1+1/d}{4(1+1/(2d))}}\right),$$

which is the same rate obtained in Chen and Shen (1998)[12]. It is interesting to note that in the case where $d = 1$, we have $\|\hat{f}_n - f_0\|_n = \mathcal{O}_p\left((n/\log n)^{-1/3}\right)$. Such rate is close to the $\mathcal{O}_p(n^{-1/3})$, which is the convergence rate in non-parametric least square problems when the class of functions considered has bounded variation in $\mathbb{R}$ (see Example 9.3.3 in van de Geer

(2000)[68]) and as shown in Proposition C.0.3 in the Appendix C, $\mathcal{F}_{r_n}$ is a class of functions with bounded variation in $\mathbb{R}$ so that the convergence rate we obtained makes sense.

4.5 Asymptotic Normality

Through out this section, we will assume that $\mathbb{E}[|\epsilon|^{2+\lambda}] < \infty$ for some $\lambda > 0$. To establish the asymptotic normality of sieve estimator based on neural network, we follow the idea in Shen (1997)[64]. We start by calculating the Gâteaux derivative of the empirical criterion function $\mathbb{Q}_n(f) = n^{-1}\sum_{i=1}^n (y_i - f(\boldsymbol{x}_i))^2$.

$$\begin{aligned}
\mathbb{Q}'_{n,f_0}[f - f_0] &= \lim_{t\to 0} \frac{1}{t}\left[\frac{1}{n}\sum_{i=1}^n (y_i - f_0(\boldsymbol{x}_i) - t(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)))^2 - \frac{1}{n}\sum_{i=1}^n (y_i - f_0(\boldsymbol{x}_i))^2\right] \\
&= \lim_{t\to 0} \frac{1}{n}\sum_{i=1}^n \frac{1}{t}\Big[(y_i - f_0(\boldsymbol{x}_i))^2 - 2t(y_i - f_0(\boldsymbol{x}_i))(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) \\
&\qquad\qquad + t^2(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2 - (y_i - f_0(\boldsymbol{x}_i))^2\Big] \\
&= -\frac{2}{n}\sum_{i=1}^n \epsilon_i (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)).
\end{aligned}$$

Then the remainder of first-order functional Taylor series expansion is

$$\begin{aligned}
R_n[f - f_0] &= \mathbb{Q}_n(f) - \mathbb{Q}_n(f_0) - \mathbb{Q}'_{n,f_0}[f - f_0] \\
&= \frac{1}{n}\sum_{i=1}^n (y_i - f(\boldsymbol{x}_i))^2 - \frac{1}{n}\sum_{i=1}^n (y_i - f(\boldsymbol{x}_i))^2 + \frac{2}{n}\sum_{i=1}^n \epsilon_i (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) \\
&= \frac{1}{n}\sum_{i=1}^n (\epsilon_i + f_0(\boldsymbol{x}_i) - f(\boldsymbol{x}_i))^2 - \frac{1}{n}\sum_{i=1}^n \epsilon_i^2 + \frac{2}{n}\sum_{i=1}^n \epsilon_i (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) \\
&= \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2 \\
&= \|f - f_0\|_n^2.
\end{aligned}$$

As will be seen in the proof of asymptotic normality, the rate of convergence for the empirical process $\{n^{-1/2}\sum_{i=1}^{n}\epsilon_i(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)) : f \in \mathcal{F}_{r_n}\}$ plays an important role. Here we establish a lemma, which will be used to find the desired rate of convergence.

Lemma 4.5.1. *Let $X_1, \ldots, X_n$ be independent random variables with $X_i \sim P_i$. Define the empirical process $\{\nu_n(f)\}$ as*

$$\nu_n(f) = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}[f(X_i) - P_i f].$$

Let $\mathcal{F}_n = \{f : \|f\|_\infty \le V_n\}$. Let $\epsilon > 0$ and $\alpha \ge \sup_{f\in\mathcal{F}_n} n^{-1}\sum_{i=1}^{n} Var[f(X_i)]$ be arbitrary. Define t_0 by $H(t_0, \mathcal{F}_n, \|\cdot\|_\infty) = \frac{\epsilon}{4}\psi(M,n,\alpha)$, where $\psi(M,n,\alpha) = M^2/\left[2\alpha\left(1+\frac{MV_n}{2\sqrt{n\alpha}}\right)\right]$. If

$$H(u, \mathcal{F}_n, \|\cdot\|_\infty) \le A_n u^{-r}, \tag{4.8}$$

for some $0 < r < 2$ and $u \in (0, a]$, where a is a small positive number, and there exists a positive constant $K_i = K_i(r,\epsilon)$, $i = 1, 2$ such that

$$M \ge K_1 A_n^{\frac{2}{r+2}} V_n^{\frac{2-r}{r+2}} n^{\frac{r-2}{2(r+2)}} \vee K_2 A_n^{1/2}\alpha^{\frac{2-r}{4}}.$$

Then

$$\mathbb{P}^*\left(\sup_{f\in\mathcal{F}_n}|\nu_n(f)| > M\right) \le 5\exp\left\{-(1-\epsilon)\psi(M,n,\alpha)\right\}.$$

Proof. The proof of the lemma is similar to the proof of Corollary 2.2 in Alexander (1984) [2] and the proof of Lemma 1 in Shen and Wong (1994)[65]. Since $H(u, \mathcal{F}_n, \|\cdot\|_\infty) \le A_n u^{-r}$

for some $0 < r < 2$, then we have

$$I(s,t) := \int_s^t H^{1/2}(u, \mathcal{F}_n, \|\cdot\|_\infty)\mathrm{d}u \leq 2(2-r)^{-1} A_n^{\frac{1}{2}} t^{1-\frac{r}{2}}.$$

Based on our assumption,

$$A_n t_0^{-r} \geq H(t_0, \mathcal{F}_n, \|\cdot\|_\infty) = \frac{\epsilon}{4}\psi(M,n,\alpha) \Rightarrow t_0 \leq \left[\frac{4A_n}{\epsilon\psi}\right]^{1/r}.$$

Note that if $M \leq 3\sqrt{n}\alpha/V_n$, then $\psi(M,n,\alpha) \geq M^2/(4\alpha)$ and if $M \geq 3\sqrt{n}\alpha/V_n$, then $2(\sqrt{n}\alpha + MV_n/3) \leq 4MV_n/3$ and hence $\psi(M,n,\alpha) \geq 3\sqrt{n}M/(4V_n)$. In summary,

$$\psi(M,n,\alpha) \geq \begin{cases} M^2/(4\alpha) & \text{if } M < 3\sqrt{n}\alpha/V_n, \\ 3\sqrt{n}M/(4V_n) & \text{if } M \geq 3\sqrt{n}\alpha/V_n \end{cases}.$$

Therefore, if $M \geq 3\sqrt{n}\alpha/V_n$,

$$\begin{aligned} 2^8\epsilon^{-3/2} I\left(\frac{\epsilon M}{64\sqrt{n}}, t_0\right) &\leq 2^9\epsilon^{-3/2}(2-r)^{-1} A_n^{1/2} t_0^{1-\frac{r}{2}} \\ &\leq 2^9\epsilon^{-3/2}(2-r)^{-1}\left(\frac{4}{\epsilon}\right)^{\frac{1}{r}-\frac{1}{2}} A_n^{1/r}\left(\frac{3}{4V_n}\sqrt{n}M\right)^{\frac{1}{2}-\frac{1}{r}} \\ &= \tilde{K}_1 A_n^{1/r} V_n^{\frac{1}{r}-\frac{1}{2}} n^{\frac{1}{4}-\frac{1}{2r}} M^{\frac{1}{2}-\frac{1}{r}}, \end{aligned}$$

where $\tilde{K}_1 = 2^9\epsilon^{-3/2}(2-r)^{-1}\left(\frac{4}{\epsilon}\right)^{\frac{1}{r}-\frac{1}{2}}\left(\frac{3}{4}\right)^{\frac{1}{2}-\frac{1}{r}}$. Hence

$$\begin{aligned}
2^8\epsilon^{-3/2} I\left(\frac{\epsilon M}{64\sqrt{n}}, t_0\right) < M &\Leftrightarrow \tilde{K}_1 A_n^{1/r} V_n^{\frac{1}{r}-\frac{1}{2}} n^{\frac{1}{4}-\frac{1}{2r}} M^{\frac{1}{2}-\frac{1}{r}} < M \\
&\Leftrightarrow \tilde{K}_1 A_n^{1/r} V_n^{\frac{1}{r}-\frac{1}{2}} n^{\frac{r-2}{4r}} < M^{\frac{1}{r}+\frac{1}{2}} \\
&\Leftrightarrow M > K_1 A_n^{\frac{2}{r+2}} V_n^{\frac{2-r}{r+2}} n^{\frac{r-2}{2(r+2)}},
\end{aligned}$$

where $K_1 = \tilde{K}_1^{\frac{2r}{r+2}}$. On the other hand, if $M < 3\sqrt{n}\alpha/V_n$,

$$\begin{aligned}
2^8\epsilon^{-3/2} I\left(\frac{\epsilon M}{64\sqrt{n}}, t_0\right) &\leq 2^9\epsilon^{-3/2}(2-r)^{-1} A_n^{1/2} t_0^{1-\frac{r}{2}} \\
&\leq 2^9\epsilon^{-3/2}(2-r)^{-1}\left(\frac{4}{\epsilon}\right)^{\frac{1}{r}-\frac{1}{2}} A_n^{1/r}\left(\frac{M^2}{4\alpha}\right)^{\frac{1}{2}-\frac{1}{r}} \\
&= \tilde{K}_2 A_n^{1/r} M^{1-\frac{2}{r}} \alpha^{\frac{1}{r}-\frac{1}{2}},
\end{aligned}$$

where $\tilde{K}_2 = 2^9\epsilon^{-3/2}(2-r)^{-1}\left(\frac{4}{\epsilon}\right)^{\frac{1}{r}-\frac{1}{2}}\left(\frac{1}{4}\right)^{\frac{1}{2}-\frac{1}{r}}$. Hence

$$\begin{aligned}
2^8\epsilon^{-3/2} I\left(\frac{\epsilon M}{64\sqrt{n}}, t_0\right) < M &\Leftrightarrow \tilde{K}_2 A_n^{1/r} M^{1-\frac{2}{r}} \alpha^{\frac{1}{r}-\frac{1}{2}} < M \\
&\Leftrightarrow \tilde{K}_2 A_n^{1/r} \alpha^{\frac{2-r}{2r}} < M^{\frac{2}{r}} \\
&\Leftrightarrow M > K_2 A_n^{1/2} \alpha^{\frac{2-r}{4}},
\end{aligned}$$

where $K_2 = \tilde{K}_2^{r/2}$. In conclusion, if $M \geq K_1 A_n^{\frac{2}{r+2}} V_n^{\frac{2-r}{r+2}} n^{\frac{r-2}{2(r+2)}} \vee K_2 A_n^{1/2} \alpha^{\frac{2-r}{4}}$, then $2^8\epsilon^{-3/2} I\left(\frac{\epsilon M}{64\sqrt{n}}, t_0\right) < M$ and by Theorem 2.1 in Alexander (1984) [2], we get the desired result. □

As a Corollary to Lemma 4.5.1, we can get the rate of convergence for the the empirical

process $\{n^{-1/2}\sum_{i=1}^{n}\epsilon_i(f(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)) : f\in\mathcal{F}_{r_n}\}$.

Corollary 4.5.1. *Let ρ_n be such that $\rho_n\|\hat{f}_n-f_0\|_n=\mathcal{O}_p(1)$ and $\mathcal{F}_{r_n}$ be the class of neural network sieves as defined in (4.2). Suppose that $\mathbb{E}[|\epsilon|^{2+\lambda}]<\infty$ for some $\lambda>0$. Then under the conditions*

(C1) $r_n(d+2)V_n\log[r_nV_n(d+2)]=o(n^{1/4})$;

(C2) $n\rho_n^{-2}/V_n^{\lambda}=o(1)$,

we have

$$\sup_{\|f-f_0\|_n\leq\rho_n^{-1},f\in\mathcal{F}_{r_n}}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\epsilon_i(f-f_0)(\boldsymbol{x}_i)\right|=o_p(1).$$

Proof. To establish the desired result, we apply the truncation device.

$$\begin{aligned}
&\mathbb{P}^*\left(\sup_{\|f-f_0\|_n\leq\rho_n^{-1},f\in\mathcal{F}_{r_n}}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\epsilon_i(f-f_0)(\boldsymbol{x}_i)\right|\gtrsim M\right)\\
\leq&\mathbb{P}^*\left(\sup_{\|f-f_0\|_n\leq\rho_n^{-1},f\in\mathcal{F}_{r_n}}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\epsilon_i\mathbb{I}_{\{|\epsilon_i|\leq V_n\}}(f-f_0)(\boldsymbol{x}_i)\right|\gtrsim M\right)\\
&\qquad+\mathbb{P}^*\left(\sup_{\|f-f_0\|_n\leq\rho_n^{-1},f\in\mathcal{F}_{r_n}}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\epsilon_i\mathbb{I}_{\{|\epsilon_i|>V_n\}}(f-f_0)(\boldsymbol{x}_i)\right|\gtrsim M\right)\\
:=&(I)+(II)
\end{aligned}$$

For (I), we can apply Lemma 4.5.1 directly. Note that $|\epsilon\mathbb{I}_{\{|\epsilon|\leq V_n\}}(f-f_0)(\boldsymbol{x})|\leq V_n(V_n+$

$\|f_0\|_\infty) \lesssim V_n^2$ since $\|f_0\|_\infty < \infty$ and for $0 < \eta < 1$,

$$\begin{aligned}
\log N(\eta, \mathcal{F}_{r_n}, \|\cdot\|_\infty) &\leq \log \left(\frac{4e[r_n(d+2)+1]\left(\frac{1}{4}V_n\right)^2}{\eta\left(\frac{1}{4}V_n - 1\right)} \right)^{r_n(d+2)+1} \\
&= [r_n(d+2)+1]\left(\log \tilde{C}_{r_n,d,V_n} + \log\frac{1}{\eta}\right) \\
&\leq [r_n(d+2)+1]\left(\log \tilde{C}_{r_n,d,V_n} + \frac{1}{\eta} - 1\right) \\
&= C_{r_n,d,V_n}\left(1 + \frac{1}{\eta}\right) \\
&\leq 2C_{r_n,d,V_n}\frac{1}{\eta},
\end{aligned}$$

where $\tilde{C}_{r_n,d,V_n} = \frac{e[r_n(d+2)+1]V_n^2}{V_n - 4}$ and

$$\begin{aligned}
C_{r_n,d,V_n} &= [r_n(d+2)+1]\log \tilde{C}_{r_n,d,V_n} - [r_n(d+2)+1] \\
&= [r_n(d+2)+1]\log\frac{[r_n(d+2)+1]V_n^2}{V_n - 4} \\
&\sim r_n(d+2)\log[r_n V_n(d+2)].
\end{aligned}$$

Therefore, equation (4.8) is satisfied with $r = 1$ and $A_n = 2C_{r_n,d,V_n}$ and it follows from Lemma 4.5.1 that for $M \gtrsim C_{r_n,d,V_n}^{2/3} V_n^{2/3} n^{-1/6} \vee C_{r_n,d,V_n}^{1/2}\alpha^{1/4}$, we have $(I) \leq 5\exp\{-(1-\epsilon)\psi(M,n,\alpha)\}$ and hence

$$\sup_{\|f-f_0\|\leq\rho_n^{-1}, f\in\mathcal{F}_{r_n}} \left| \frac{1}{\sqrt{n}}\epsilon_i \mathbb{I}_{\{|\epsilon_i|\leq V_n\}}(f - f_0)(\boldsymbol{x}_i) \right| = \mathcal{O}_p\left(\frac{C_{r_n,d,V_n}^{2/3} V_n^{2/3}}{n^{1/6}} \right).$$

Then since by (C1),

$$\frac{C_{r_n,d,V_n}^{2/3} V_n^{2/3}}{n^{1/6}} \sim \left(\frac{r_n(d+2)V_n \log[r_n V_n(d+2)]}{n^{1/4}} \right)^{2/3} = o_p(1).$$

For (II), by Cauchy-Schwarz inequality, we have

$$\left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \mathbb{I}_{\{|\epsilon_i| > V_n\}} (f - f_0)(\boldsymbol{x}_i) \right| \leq \left(\frac{1}{n} \sum_{i=1}^{n} \epsilon_i^2 \mathbb{I}_{\{|\epsilon_i| > V_n\}} \right)^{1/2} \|f - f_0\|_n.$$

Then it follows from Markov inequality that

$$\begin{aligned}
(II) &= \mathbb{P}^* \left(\sup_{\|f-f_0\|_n \leq \rho_n^{-1}, f \in \mathcal{F}_{r_n}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \mathbb{I}_{\{|\epsilon_i| > V_n\}} (f - f_0)(\boldsymbol{x}_i) \right| \gtrsim M n^{-1/2} \right) \\
&\leq \mathbb{P} \left(\left(\frac{1}{n} \sum_{i=1}^{n} \epsilon_i^2 \mathbb{I}_{\{|\epsilon_i| > V_n\}} \right)^{1/2} \rho_n^{-1} \gtrsim M n^{-1/2} \right) \\
&= \mathbb{P} \left(\frac{1}{n} \sum_{i=1}^{n} \epsilon_i^2 \mathbb{I}_{\{|\epsilon_i| > V_n\}} \gtrsim M^2 n^{-1} \rho_n^2 \right) \\
&\lesssim M^{-2} n \rho_n^{-2} \mathbb{E}[\epsilon^2 \mathbb{I}_{|\epsilon| > V_n}] \\
&\lesssim M^{-2} n \rho_n^{-2} \frac{\mathbb{E}[|\epsilon|^{2+\lambda}]}{V_n^{\lambda}}.
\end{aligned}$$

It then follows from condition (C2) that $(II) \to 0$ and hence

$$\sup_{\|f-f_0\|_n \leq \rho_n^{-1}, f \in \mathcal{F}_{r_n}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \mathbb{I}_{\{|\epsilon_i| > V_n\}} (f - f_0)(\boldsymbol{x}_i) \right| = o_p(1).$$

Combining the results we obtained above, we get

$$\sup_{\|f-f_0\|_n \leq \rho_n^{-1}, f \in \mathcal{F}_{r_n}} \left| \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \epsilon_i (f - f_0)(\boldsymbol{x}_i) \right| = o_p(1)$$

□

Remark 4.5.1. *Condition (C2) can be further simplified using the results we obtained in Theorem 4.4.1. If* $\eta_n = \mathcal{O}\left(\min\{\|\pi_{r_n} f_0 - f_0\|_n^2, r_n(d+2)\log(r_n V_n(d+2))/n, r_n(d+2)\log n/n\}\right)$, *then* $\rho_n^{-1} \asymp \max\left\{\|\pi_{r_n} f_0 - f_0\|_n, \sqrt{r_n(d+2)\log[r_n V_n(d+2)]/n}, \sqrt{r_n(d+2)\log n/n}\right\}$. *It follows from condition (C1), that* $\rho_n^{-1} \asymp \max\left\{\|\pi_{r_n} f_0 - f_0\|_n, \sqrt{r_n(d+2)\log n/n}\right\}$. *For simplicity, here we assume that* $\rho_n^{-1} \asymp \sqrt{r_n(d+2)\log n/n}$. *This holds for functions having finite first absolute moments of the Fourier magnitude distributions as we have discussed at the end of section 4.4. Then in this case,*

$$n\rho_n^{-2}/V_n^{\lambda} \asymp r_n(d+2)\log n/V_n^{\lambda},$$

so that condition (C2) becomes $r_n(d+2)\log n/V_n^{\lambda} \to 0$.

Now we are going to establish the asymptotic normality for neural network estimators. For $f \in \{f \in \mathcal{F}_{r_n} : \|f - f_0\|_n \leq \rho_n^{-1}\}$, we consider a local alternative

$$\tilde{f}_n(f) = (1-\delta_n)f + \delta_n(f_0 + \iota), \tag{4.9}$$

where $0 \leq \delta_n = \eta_n^{1/2} = o(n^{-1/2})$ is chosen such that $\rho_n \delta_n = o(1)$ and $\iota(\boldsymbol{x}) \equiv 1$.

Theorem 4.5.1 (Asymptotic Normality)**.** *Suppose that* $0 \leq \eta_n = o(n^{-1})$ *and conditions (C1) and (C2) in Corollary 4.5.1 hold. Assume further that the following two conditions hold*

(C3) $\sup_{f \in \mathcal{F}_{r_n}: \|f - f_0\|_n \leq \rho_n^{-1}} \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n = \mathcal{O}_p(\rho_n \delta_n^2)$;

(C4) $\sup_{f \in \mathcal{F}_{r_n}: \|f - f_0\|_n \leq \rho_n^{-1}} \frac{1}{n}\sum_{i=1}^n \epsilon_i \left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i) - \tilde{f}_n(f)(\boldsymbol{x}_i)\right) = \mathcal{O}_p(\delta_n^2)$,

then

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)\right]\xrightarrow{d}\mathcal{N}(0,\sigma^2).$$

Before we proceed to the proof of the theorem, let us focus on the conditions given in the theorem. For condition (C1), note that if (C1) holds, then

$$r_n(d+2)V_n^2\log[r_nV_n(d+2)]\leq[r_n(d+2)]^4V_n^4\left(\log[r_nV_n(d+2)]\right)^4=o(n),$$

so it is a sufficient condition to ensure the consistency of the neural network sieve estimator. As in Remark 4.3.2, we consider some simple scenarios here. If $V_n=\mathcal{O}(r_n)$, then $r_n(d+2)V_n\log[r_nV_n(d+2)]=\mathcal{O}\left(r_n^2\log r_n\right)$ so that a possible growth rate for r_n is $r_n=o\left(n^{1/8}/(\log n)^2\right)$. On the other hand, if $r_n=\log V_n$, then $r_n(d+2)V_n\log[r_nV_n(d+2)]=\mathcal{O}\left(V_n(\log V_n)^2\right)$ and a possible growth rate for V_n is $V_n=o(n^{1/4}/(\log n)^2)$. Thus, we can see that in both cases, the growth rate required for the asymptotic normality of neural network sieve estimator is slower than the growth rate required for the consistency as given in Remark 4.3.2. One explanation for this is that due to the Universal Approximation Theorem, a neural network with one hidden layer can approximate a continuous function on compact support arbitrarily well if the number of hidden units is sufficiently large. Therefore, if the number of hidden units is too large, the neural network sieve estimator $\hat{f}_n$ may be very close to the best projector of the true function f_0 in $\mathcal{F}_{r_n}$ so that the error $\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)\right]$ may be close to zero and the variations in this quantity may be small. By allowing slower growth rate of the number of hidden units can increase the variations in the quantity $\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)\right]$, which makes the asymptotic normality more reasonable. On the other hand, condition (C3) and condition (C4) are similar conditions as in Shen (1997)[64], which are known for conditions on approximation error and these conditions dictate that the approximation rate of a

single layer neural network cannot be too slow, otherwise it may require a huge number of samples to reach the desired approximation error. Therefore, the conditions in the theorem can be considered as a trade-off between bias and variance.

Proof of Theorem 4.5.1. The main idea of the proof is to use the functional Taylor series expansion for $\mathbb{Q}_n(f)$ and carefully bound each term in the expansion. For any $f \in \{f \in \mathcal{F}_{r_n} : \|f - f_0\|_n \leq \rho_n^{-1}\}$,

$$\begin{aligned}\mathbb{Q}_n(f) &= \mathbb{Q}_n(f_0) + \mathbb{Q}'_{n,f_0}[f - f_0] + R_n[f - f_0] \\ &= \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 - \frac{2}{n}\sum_{i=1}^{n}\epsilon_i(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) + \frac{1}{n}\sum_{i=1}^{n}(f(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i))^2. \end{aligned} \tag{4.10}$$

Note that

$$\begin{aligned}\|\tilde{f}_n(f) - f_0\|_n &= \|(1-\delta_n)\hat{f}_n + \delta_n(f_0 + \iota) - f_0\|_n \\ &= \|(1-\delta_n)(\hat{f}_n - f_0) + \delta_n\iota\|_n \\ &\leq (1-\delta_n)\|\hat{f}_n - f_0\|_n + \delta_n,\end{aligned}$$

and since $\delta_n = o(n^{-1/2})$, we can know that with probability tending to 1, $\|\tilde{f}_n(f) - f_0\|_n \leq \rho_n^{-1}$. Then replacing f in (4.10) by $\hat{f}_n$ and $\pi_{r_n}\tilde{f}_n(f)$ as defined in (4.9), we get

$$\begin{aligned}\mathbb{Q}_n(\hat{f}_n) &= \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 - \frac{2}{n}\sum_{i=1}^{n}\epsilon_i(\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) + \|\hat{f}_n - f_0\|_n^2 \\ \mathbb{Q}_n(\pi_{r_n}\tilde{f}_n(f)) &= \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 - \frac{2}{n}\sum_{i=1}^{n}\epsilon_i(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)) + \|\pi_{r_n}\tilde{f}_n(f) - f_0\|_n^2.\end{aligned}$$

Subtracting these two equations yields

$$\mathbb{Q}_n(\hat{f}_n) = \mathbb{Q}_n(\pi_{r_n}\tilde{f}_n(f)) + \frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i) - \hat{f}_n(\boldsymbol{x}_i)\right) + \|\hat{f}_n - f_0\|_n^2 - \|\pi_{r_n}\tilde{f}_n(f) - f_0\|_n^2.$$

Now note that

$$\begin{aligned}
\|\pi_{r_n}\tilde{f}_n(f) - f_0\|_n^2 &= \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f) + \tilde{f}_n(f) - f_0\|_n^2 \\
&= \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f) + (1-\delta_n)\hat{f}_n + \delta_n(f_0+\iota) - f_0\|_n^2 \\
&= \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f) + (1-\delta_n)(\hat{f}_n - f_0) + \delta_n\iota\|_n^2 \\
&= \Big\langle \pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f) + (1-\delta_n)(\hat{f}_n - f_0) + \delta_n\iota, \\
&\qquad\qquad \pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f) + (1-\delta_n)(\hat{f}_n - f_0) + \delta_n\iota \Big\rangle \\
&= \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n^2 + (1-\delta_n)^2\|\hat{f}_n - f_0\|_n^2 + \delta_n^2 \\
&\qquad\qquad + 2(1-\delta_n)\left\langle \pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f), \hat{f}_n - f_0 \right\rangle \\
&\qquad\qquad + 2\delta_n\left\langle \pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f), \iota \right\rangle \\
&\qquad\qquad + 2(1-\delta_n)\delta_n\left\langle \hat{f}_n - f_0, \iota \right\rangle \\
&\le (1-\delta_n)^2\|\hat{f}_n - f_0\|_n^2 + 2(1-\delta_n)\left\langle \hat{f}_n - f_0, \delta_n\iota \right\rangle + \delta_n^2 \\
&\qquad\qquad + 2(1-\delta_n)\|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n\|\hat{f}_n - f_0\|_n \\
&\qquad\qquad + 2\delta_n\|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n + \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n^2,
\end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality and since

$$\begin{aligned}\frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i)-\hat{f}_n(\boldsymbol{x}_i)\right) &= \frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i)-\tilde{f}_n(f)(\boldsymbol{x}_i)+\tilde{f}_n(f)(\boldsymbol{x}_i)-\hat{f}_n(\boldsymbol{x}_i)\right)\\ &= \frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i)-\tilde{f}_n(f)(\boldsymbol{x}_i)\right)\\ &\quad -\frac{2}{n}\delta_n\sum_{i=1}^{n}\epsilon_i\left(\hat{f}_n(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)\right)-\frac{2}{n}\delta_n\sum_{i=1}^{n}\epsilon_i,\end{aligned}$$

we have by the definition of $\hat{f}_n$,

$$\begin{aligned}-\mathcal{O}_p(\delta_n^2) &\leq \inf_{f\in\mathcal{F}_{r_n}}\mathbb{Q}_n(f)-\mathbb{Q}_n(\hat{f}_n)\\ &\leq \mathbb{Q}_n(\pi_{r_n}\tilde{f}_n(f))-\mathbb{Q}_n(\hat{f}_n)\\ &= \|\pi_{r_n}\tilde{f}_n(f)-f_0\|_n^2-\|\hat{f}_n-f_0\|_n^2-\frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i)-\hat{f}_n(\boldsymbol{x}_i)\right)\\ &\leq (1-\delta_n)^2\|\hat{f}_n-f_0\|_n^2-\|\hat{f}_n-f_0\|_n^2+2(1-\delta_n)\delta_n\left\langle\hat{f}_n-f_0,\iota\right\rangle\\ &\qquad +2(1-\delta_n)\|\hat{f}_n-f_0\|_n\|\pi_{r_n}\tilde{f}_n(f)-\tilde{f}_n(f)\|_n\\ &\qquad +2\delta_n\|\pi_{r_n}\tilde{f}_n(f)-\tilde{f}_n(f)\|_n+\|\pi_{r_n}\tilde{f}_n(f)-\tilde{f}_n(f)\|_n^2\\ &\qquad -\frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i)-\tilde{f}_n(f)(\boldsymbol{x}_i)\right)+\frac{2}{n}\delta_n\sum_{i=1}^{n}\epsilon_i\left(\hat{f}_n(\boldsymbol{x}_i)-f_0(\boldsymbol{x}_i)\right)\\ &\qquad +\frac{2}{n}\delta_n\sum_{i=1}^{n}\epsilon_i+\mathcal{O}_p(\delta_n^2)\end{aligned}$$

$$
\begin{aligned}
&= (-2\delta_n + \delta_n^2)\|\hat{f}_n - f_0\|_n^2 + 2(1-\delta_n)\delta_n \left\langle \hat{f}_n - f_0, \iota \right\rangle + 2(1-\delta_n)\|\hat{f}_n - f_0\|_n \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n \\
&\quad + 2\delta_n \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n + \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n^2 \\
&\quad - \frac{2}{n}\sum_{i=1}^{n} \epsilon_i \left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i) - \tilde{f}_n(f)(\boldsymbol{x}_i)\right) + \frac{2}{n}\delta_n \sum_{i=1}^{n} \epsilon_i \left(\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right) \\
&\quad + \frac{2}{n}\delta_n \sum_{i=1}^{n} \epsilon_i + \mathcal{O}_p(\delta_n^2) \\
&\leq \delta_n^2 \|\hat{f}_n - f_0\|_n^2 + 2(1-\delta_n)\delta_n \left\langle \hat{f}_n - f_0, \iota \right\rangle + 2(1-\delta_n)\|\hat{f}_n - f_0\|_n \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n \\
&\quad + 2\delta_n \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n + \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n^2 \\
&\quad - \frac{2}{n}\sum_{i=1}^{n} \epsilon_i \left(\pi_{r_n}\tilde{f}_n(f)(\boldsymbol{x}_i) - \tilde{f}_n(f)(\boldsymbol{x}_i)\right) + \frac{2}{n}\delta_n \sum_{i=1}^{n} \epsilon_i \left(\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right) \\
&\quad + \frac{2}{n}\delta_n \sum_{i=1}^{n} \epsilon_i + \mathcal{O}_p(\delta_n^2), \qquad (4.11)
\end{aligned}
$$

where the last inequality follows by noting that $(1-\delta_n)^2 - 1 = -2\delta_n + \delta_n^2 \leq \delta_n^2$. From the condition (C1), we can get

$$
\begin{aligned}
&[r_n(d+2)+1]V_n^2 \log[r_n V_n(d+2)+1] \\
&\qquad \leq ([r_n(d+2)+1]V_n \log[r_n V_n(d+2)+1])^4 = o(n),
\end{aligned}
$$

combining with Theorem 4.3.1, we obtain that $\|\hat{f}_n - f_0\|_n = o_p(1)$ and hence $\delta_n^2\|\hat{f}_n - f_0\|_n^2 = o_p(\delta_n^2)$. From condition (C3), we have

$$
\begin{aligned}
2(1-\delta_n)\|\hat{f}_n - f_0\|_n \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n &\leq 2\|\hat{f}_n - f_0\|_n \|\pi_{r_n}\tilde{f}_n(f) - \tilde{f}_n(f)\|_n \\
&= \mathcal{O}_p\left(\rho_n^{-1}\rho_n\delta_n^2\right) = \mathcal{O}_p(\delta_n^2).
\end{aligned}
$$

Similarly, since $\rho_n \delta_n = o(1)$, we have

$$2\delta_n \|\pi_{r_n} \tilde{f}_n(f) - \tilde{f}_n(f)\|_n = \mathcal{O}_p(\delta_n \cdot \rho_n \delta_n^2) = o_p(\delta_n^2)$$
$$\|\pi_{r_n} \tilde{f}_n(f) - \tilde{f}_n(f)\|_n^2 = \mathcal{O}_p(\rho_n^2 \delta_n^4) = o_p(\delta_n^2).$$

Based on condition (C4), we also know that

$$\frac{2}{n} \sum_{i=1}^{n} \epsilon_i \left(\pi_{r_n} \tilde{f}_n(f) - \tilde{f}_n(f) \right) = \mathcal{O}_p(\delta_n^2),$$

and from Corollary 4.5.1, we also have

$$\frac{2}{n} \delta_n \sum_{i=1}^{n} \epsilon_i \left(\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i) \right) = o_p(\delta_n \cdot n^{-1/2}).$$

It follows from these observations that

$$-2(1-\delta_n) \left\langle \hat{f}_n - f_0, \delta_n \iota \right\rangle + \frac{2\delta_n}{n} \sum_{i=1}^{n} \epsilon_i \leq \mathcal{O}_p(\delta_n^2) + {}_p(\delta_n^2) + o_p(\delta_n \cdot n^{-1/2}),$$

which implies that

$$-(1-\delta_n) \left\langle \hat{f}_n - f_0, \iota \right\rangle + \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \leq \mathcal{O}_p(\delta_n) + o_p(n^{-1/2}) = o_p(n^{-1/2}).$$

By replacing ι with $-\iota$, we can obtain the same result and hence

$$\begin{aligned}\left|\left\langle \hat{f}_n - f_0, \iota \right\rangle - \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\right| &\leq \left|(1-\delta_n)\left\langle \hat{f}_n - f_0, \iota \right\rangle - \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\right| + \delta_n\left|\left\langle \hat{f}_n - f_0, \iota \right\rangle\right| \\ &\leq o_p(n^{-1/2}) + \delta_n\|\hat{f}_n - f_0\|_n \\ &= o_p(n^{-1/2}).\end{aligned}$$

Therefore,

$$\left\langle \hat{f}_n - f_0, \iota \right\rangle = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i + o_p(n^{-1/2})$$

and the desired result follows from the classical Central Limit Theorem. □

Theorem 4.5.1 can be used directly for hypothesis testing based on the model of neural network with one hidden layer if we know the variance σ^2 of the random error. In practice, this is rarely the case. To perform hypothesis testing when σ^2 is unknown, it is natural to use a good estimator for σ^2 and use a "plug-in" test statistic. A natural estimator for σ^2 is

$$\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}_n(\boldsymbol{x}_i)\right)^2 = \mathbb{Q}_n\left(\hat{f}_n\right).$$

So we also need to establish the asymptotic normality for the statistic $\frac{1}{\hat{\sigma}_n\sqrt{n}}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$.

Theorem 4.5.2 (Asymptotic Normality for Plug-in Statistic)**.** *Suppose that $f_0 \in C(\mathcal{X})$, where $\mathcal{X} \subset \mathbb{R}^d$ is a compact set and $0 \leq \eta_n = o\left(n^{-1}\right)$. Then under condition as in Theorem 4.5.1, we have*

$$\frac{1}{\hat{\sigma}_n\sqrt{n}}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right] \xrightarrow{d} \mathcal{N}(0,1).$$

Proof. Note that

$$\begin{aligned}\hat{\sigma}_n^2 = \mathbb{Q}_n(\hat{f}_n) &= \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}_n(\boldsymbol{x}_i)\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\left(f_0(\boldsymbol{x}_i) + \epsilon_i - \hat{f}_n(\boldsymbol{x}_i)\right)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right)^2 - \frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right) + \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 - \frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right) + \|\hat{f}_n - f_0\|_n^2\end{aligned}$$

Based on the rate of convergence of $\hat{f}_n$ we obtained in Theorem 4.4.1 and condition (C1), we can know that

$$\left\|\hat{f}_n - f_0\right\|_n^2 = \mathcal{O}_p^*\left(\max\left\{\|\pi_{r_n} f_0 - f_0\|_n^2, \frac{r_n(d+2)\log n}{n}\right\}\right)$$

Under (C3), $\|\pi_{r_n} f_0 - f_0\|_n^2 = o\left(\rho_n^2\delta_n^4\right) = o(n^{-1/2})$ and under (C1), we have

$$\begin{aligned}\left(\frac{r_n(d+2)\log n}{n}\right) &\leq o\left(\frac{n^{1/4}\log n}{n}\right) \\ &= o\left(\frac{\log n}{n^{3/4}}\right) = o(n^{-1/2})\end{aligned}$$

which implies that $\left\|\hat{f}_n - f_0\right\|_n^2 = o_p(n^{-1/2})$. Moreover, by the same arguments as in the proof of Theorem 4.5.1, we can show that

$$\frac{2}{n}\sum_{i=1}^{n}\epsilon_i\left(\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right) = o_p(n^{-1/2}).$$

Therefore,

$$\mathbb{Q}_n(\hat{f}_n) = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 + o_p(n^{-1/2}).$$

Based on Weak Law of Large Numbers, we know that $\frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2 = \sigma^2 + o_p(1)$. Therefore,

$$\hat{\sigma}_n^2 = \mathbb{Q}_n(\hat{f}_n) = \sigma^2 + o_p(1)$$

and it follows from Slutsky's Theorem and Theorem 4.5.1, we obtain

$$\frac{1}{\hat{\sigma}_n\sqrt{n}}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right] = \frac{\sigma}{\hat{\sigma}_n}\frac{1}{\sigma\sqrt{n}}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right] \xrightarrow{d} \mathcal{N}(0,1).$$

□

4.6 Simulation Studies

In this section, simulations were conducted to check the validity of the theoretical results obtained in the previous sections. We first used a simple simulation to show that it is hard for the parameter estimators in a neural network with one hidden layer to reach parametric consistency. Then the consistency of the neural network sieve estimators were examined under various simulation setups. Finally, in the last part, we checked the asymptotic normality of the neural network sieve estimators.

4.6.1 Parameter Inconsistency

As we have mentioned in the introduction, due to the loss of identifiability of the parameters, the parameter estimators obtained in a neural network model are unlikely to be consistent. In this simulation, we use empirical results to confirm such observations. We simulated the response through the following model:

$$y_i = f_0(x_i) + \epsilon_i, \quad i = 1, \ldots, n, \tag{4.12}$$

where the total sample size $n = 500$, $x_1, \ldots, x_n \sim$ i.i.d.$\mathcal{N}(0, 1)$, $\epsilon_1, \ldots, \epsilon_n \sim$ i.i.d.$\mathcal{N}(0, 0.1^2)$ and

$$f_0(x_i) = -1 + \sigma(2x_i + 1) - \sigma(-x_i + 1), \tag{4.13}$$

that is, the true model is a single-layer neural network and the number of hidden units is 2. When we conducted the simulation, we also used a single-layer neural network to fit the data. When fitting the neural network, we set the learning rate as 0.1 and performed 3e4 iterations for the back propagation. The cost after 3e4 iterations is 0.0106. Table 4.1 summarizes the estimated values for the parameters in this model.

Table 4.1: Comparison of the true parameters and the estimated parameters in a single-layer neural network with 2 hidden units.

Estimated Values	Weights				Biases		
	γ_1	γ_2	α_1	α_2	$\gamma_{0,1}$	$\gamma_{0,2}$	α_0
True Value	2.00	-1.00	1.00	-1.00	1.00	1.00	-1.00
Estimated Value	0.82	1.30	-0.34	-0.58	-0.03	-0.03	-1.04

Based on the results in Table 4.1, it is clear that the estimators for most of the weights

and biases (except α_0) are far from reaching consistency. On the other hand, if we look at the curve of the true function and the curve of the fitted function as shown in Figure 4.1, we can see that most parts are fitted extremely well except for the tail parts. The approximation error $\|\hat{f} - f_0\|_n$ is almost zero as shown in the Figure. This suggests that we may study the asymptotic properties of the neural network using the estimated function instead of the estimated parameters.

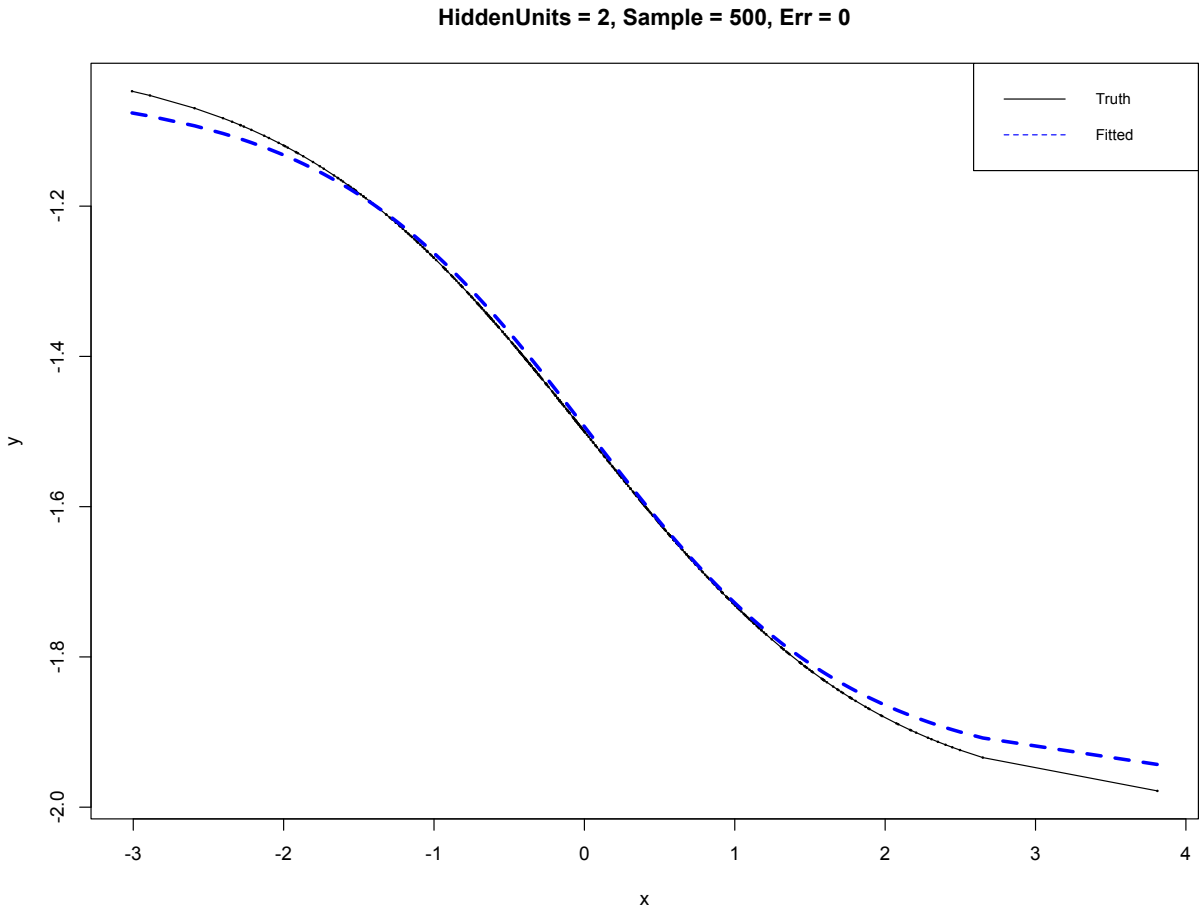

Figure 4.1: Comparison of the true function and fitted function under the simulation model (4.12). The black curve is the true function defined in (4.13) and the blue dashed curve is the fitted curve obtained after fitting the neural network model.

4.6.2 Consistency for Neural Network Sieve Estimators

In this simulation, we are going to check the consistency result obtained in Section **??** and the validity of the assumption made in Theorem 4.3.1. Based on our construction of the neural network sieve estimators, in each sieve space $\mathcal{F}_{r_n}$, there is a constraint on the ℓ_1 norm for $\boldsymbol{\alpha}$: $\sum_{i=0}^{r_n} |\alpha_i| \leq V_n$. So finding the nearly optimal function in $\mathcal{F}_{r_n}$ for $\mathbb{Q}_n(f)$ is in fact a constrained optimization problem. Classical way to conduct this optimization is through introducing a Lagrange multiplier for each constraint. But it is usually hard to find an explicit connection between the Lagrange multiplier and the upper bound in the inequality constraint. Instead, we use the subgradient method as discussed in section 7 in Boyd and Mutapcic (2008)[9]. The basic idea is to update the parameter $\alpha_0, \ldots, \alpha_{r_n}$ through

$$\alpha_i^{(k+1)} = \alpha_i^{(k)} - \delta_k g^{(k)}, \quad i = 0, \ldots, r_n$$

where $\delta_k > 0$ is a step size and δ_k is chosen to be $0.1/\log(e+k)$ throughout the simulation, which is known as a nonsummable diminishing step size rule. $g^{(k)}$ is a subgradient of the objective or the constraint function $\sum_{j=0}^{r_n} |\alpha_j| - V_n$ at $\boldsymbol{\alpha}^{(k)}$. More specifically, we take

$$g^{(k)} \in \begin{cases} \partial_{\boldsymbol{\alpha}^{(k)}} \mathbb{Q}_n(f) & \text{if } \sum_{j=0}^{r_n} |\alpha_j| \leq V_n \\ \partial_{\boldsymbol{\alpha}^{(k)}} \sum_{j=0}^{r_n} |\alpha_j| & \text{if } \sum_{j=0}^{r_n} |\alpha_j| > V_n \end{cases}$$

The updating equation of $\boldsymbol{\gamma}_1, \ldots \boldsymbol{\gamma}_{r_n}, \gamma_{0,1}, \ldots, \gamma_{0,r_n}$ remains the same as in the classical gradient descent algorithm.

We still used equation (4.12) as our simulation model. Instead, we assumed that the random error $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. $\mathcal{N}(0, 0.7^2)$ throughout the simulations. For the true function

$f_0(x)$, we considered the following three functions:

(1) Neural network with single hidden layer and 2 hidden units, which is the same as in equation (4.13).

(2) A trigonometric function:

$$f_0(x) = \sin\left(\frac{\pi}{3}x\right) + \frac{1}{3}\cos\left(\frac{\pi}{4}x + 1\right) \tag{4.14}$$

(3) A continuous function having a non-differential point

$$f_0(x) = \begin{cases} -2x & \text{if } x \leq 0 \\ \sqrt{x}\left(x - \frac{1}{4}\right) & \text{if } x > 0 \end{cases} \tag{4.15}$$

We then trained a neural network using the subgradient method mentioned at the beginning of this subsection and set the number of iterations used for fitting as 20,000. For the growth rate on the number of hidden units r_n and the upper bound for ℓ_1 norm of the weights and bias from the hidden layer to the output layer V_n, we chose $r_n = n^{1/4}$ and $V_n = 10n^{1/4}$. Such choice satisfies the condition mentioned in Remark 4.3.2 and hence satisfies the condition in Theorem 4.3.1. We compared the errors $\|\hat{f}_n - f_0\|_n^2$ and the least square errors $\mathbb{Q}_n(\hat{f}_n)$ under different sample sizes and the results are summarized in Table 4.2.

As we can see from Table 4.2, the errors $\|\hat{f}_n - f_0\|_n^2$ indeed has a decreasing pattern as the sample size increases although there are some cases where the error becomes a little bit larger when the sample sizes increases (e.g. the errors using 500 sample in all scenarios is larger than those errors using 200 sample). One explanation is that the number of hidden units increase from 3 (for 200 sample) to 4 (for 500 sample) under our simulation setup. So

Table 4.2: Comparison of errors $\|\hat{f}_n - f_0\|_n^2$ and the least square errors $\mathbb{Q}_n(\hat{f}_n)$ after 20,000 iterations under different sample sizes.

Sample Sizes	Neural Network		Sine		Piecewise Continuous	
	$\|\hat{f}_n - f_0\|_n^2$	$\mathbb{Q}_n(\hat{f}_n)$	$\|\hat{f}_n - f_0\|_n^2$	$\mathbb{Q}_n(\hat{f}_n)$	$\|\hat{f}_n - f_0\|_n^2$	$\mathbb{Q}_n(\hat{f}_n)$
50	3.33E-2	0.519	6.04E-2	0.513	6.20E-1	1.124
100	2.79E-2	0.552	3.04E-2	0.587	3.20E-1	0.920
200	6.05E-3	0.500	1.05E-2	0.501	2.51E-1	0.786
500	8.15E-3	0.484	1.19E-2	0.499	3.26E-1	0.769
1000	3.02E-3	0.475	1.54E-2	0.480	2.98E-2	0.489
2000	2.88E-3	0.500	9.72E-3	0.506	1.69E-2	0.515

there may be some variations among the estimation performance. Overall, the table shows that the estimated function $\hat{f}_n$ is indeed consistent in the sense that $\|\hat{f}_n - f_0\|_n = o_p^*(1)$. Figure 4.2 illustrates the fitted functions and the true function, from which we can visualize the result more straightforwardly.

4.6.3 Asymptotic Normality for Neural Network Sieve Estimators

The last part of the simulation focus on the asymptotic normality derived in Theorem 4.5.1. We still considered the same three types of true functions as used in section 4.6.2 but the random errors are sampled from the standard normal distribution and we still used the subgradient method to obtain the fitted model. The number of iterations used for fitting was set as 20,000. What is different in the simulation setup from what we did in section 4.6.2 is that the growth rates for r_n and V_n. As mentioned in section 4.5, the growth rates required for asymptotic normality are slower than those required for consistency. Therefore, in the simulation we chose $r_n = n^{1/8}$ and $V_n = 10n^{1/10}$. Such choice satisfies the condition (C1) in Theorem 4.5.1. In order to get the normal Q-Q plot for $n^{-1/2}\sum_{i=1}^n \left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$, we repeated the simulation 200 times.

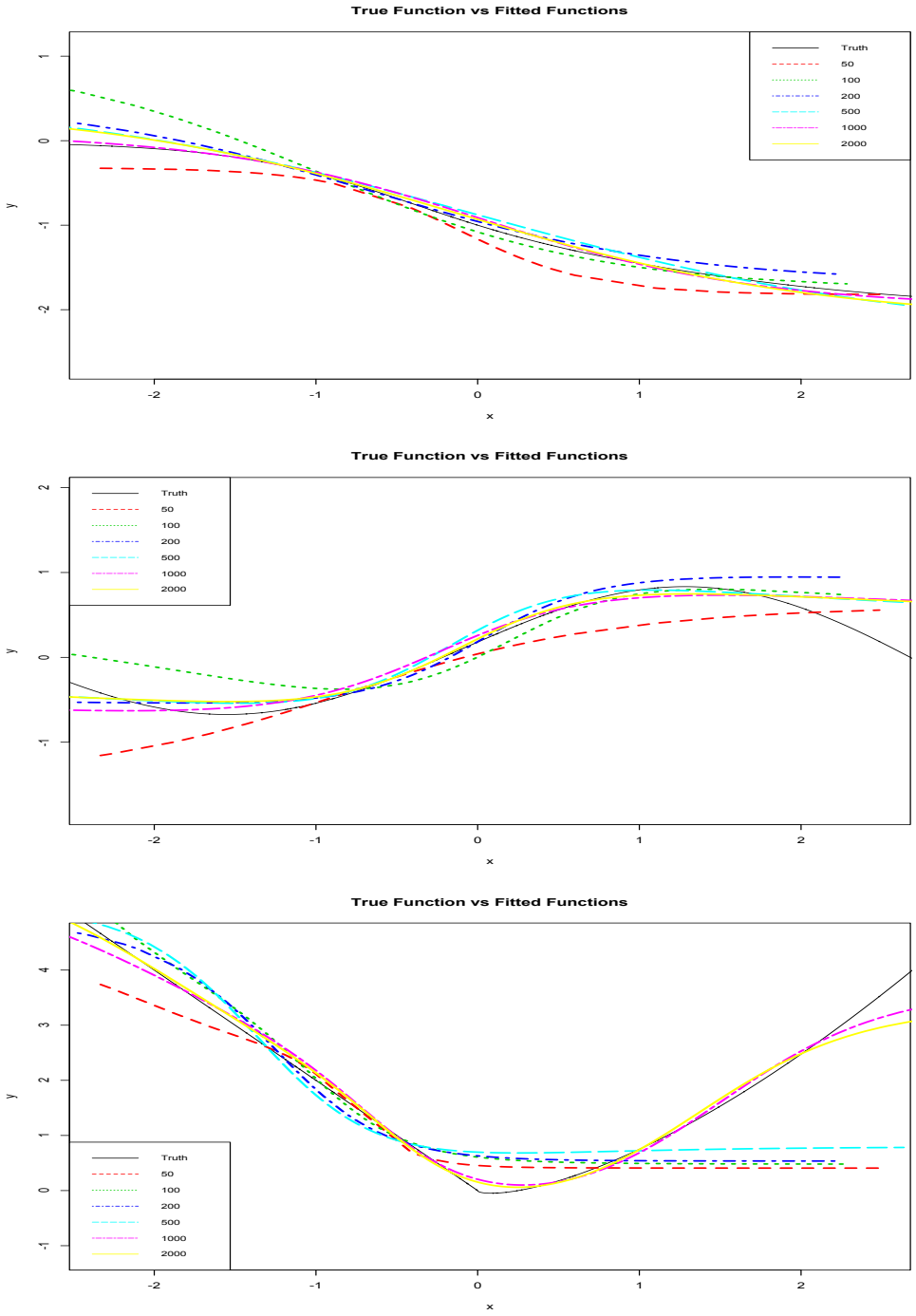

Figure 4.2: Figures on comparison of the true function and the fitted function used in simulations. The top panel shows the scenario when the true function is a single layer neural network; the middle panel shows the scenario when the true function is a sine function and the bottom panel show the scenario when the true function is a continuous function having a non-differentiable point. As we can see from all the cases, the fitted curve becomes closer to the truth as the sample size increases.

Figure 4.3 to Figure 4.5 are the normal Q-Q plots under each simulation setup and various sample sizes. Based on the figures, we can see that the statistic $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ fits the normal distribution pretty well under all simulation models. It is also worth to note that the Q-Q plots looks similar under all simulation models and this is what we should expect since under all scenarios, the limiting distribution for the statistic $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ is $\mathcal{N}(0,1)$. Another implication we can obtain from the Q-Q plots is that the statistic $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ is robust to the choice of f_0. Therefore, as long as the true function f_0 is continuous, $\mathcal{N}(0,1)$ can be served as a good asymptotic distribution for $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ and can be used to conduct hypothesis testing.

Besides the Q-Q plots, we also conducted the normality tests to check whether $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ follows the standard normal distribution. Specifically, we used the Shapiro-Wilks test and Kolmogorov-Smirnov test to perform the normality test. Table 4.3 summarizes the p-values for both normality tests and we can see from the p-values, in all cases, we failed to reject that the distribution of $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ follows the standard normal distribution.

Table 4.3: p-values for Shapiro-Wilks test and Kolmogorov-Smirnov test for nomality test. We use “NN" to denote the true function as a neural network described in (4.13); “TRI" to denote the true function as a trigonometric function described in (4.14) and “ND" to denote the true function as a continuous function having a non-differential point described in (4.15)

Sample Sizes	Shapiro-Wilks Test			Kolmogorov-Smirnov Test		
	NN	TRI	ND	NN	TRI	ND
50	0.878	0.884	0.881	0.584	0.597	0.595
100	0.098	0.095	0.095	0.472	0.508	0.484
200	0.940	0.944	0.944	0.731	0.719	0.708
300	0.884	0.888	0.872	0.976	0.986	0.973
400	0.514	0.525	0.513	0.670	0.754	0.708
500	0.768	0.778	0.768	0.733	0.769	0.733

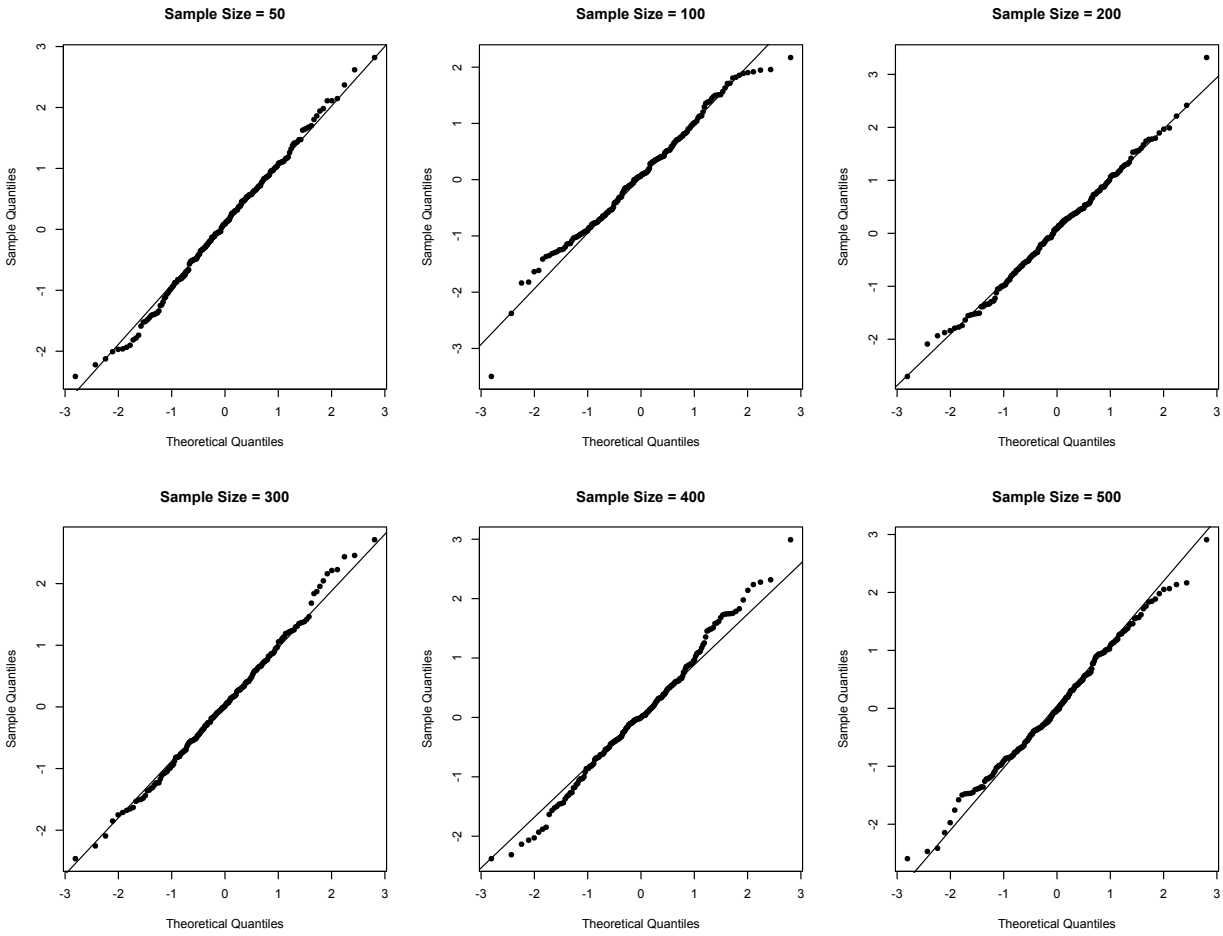

Figure 4.3: Normal Q-Q plot for $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ under 200 iterations and various sample sizes. The true function f_0 is a single-layer neural network with 2 hidden units as defined in (4.13).

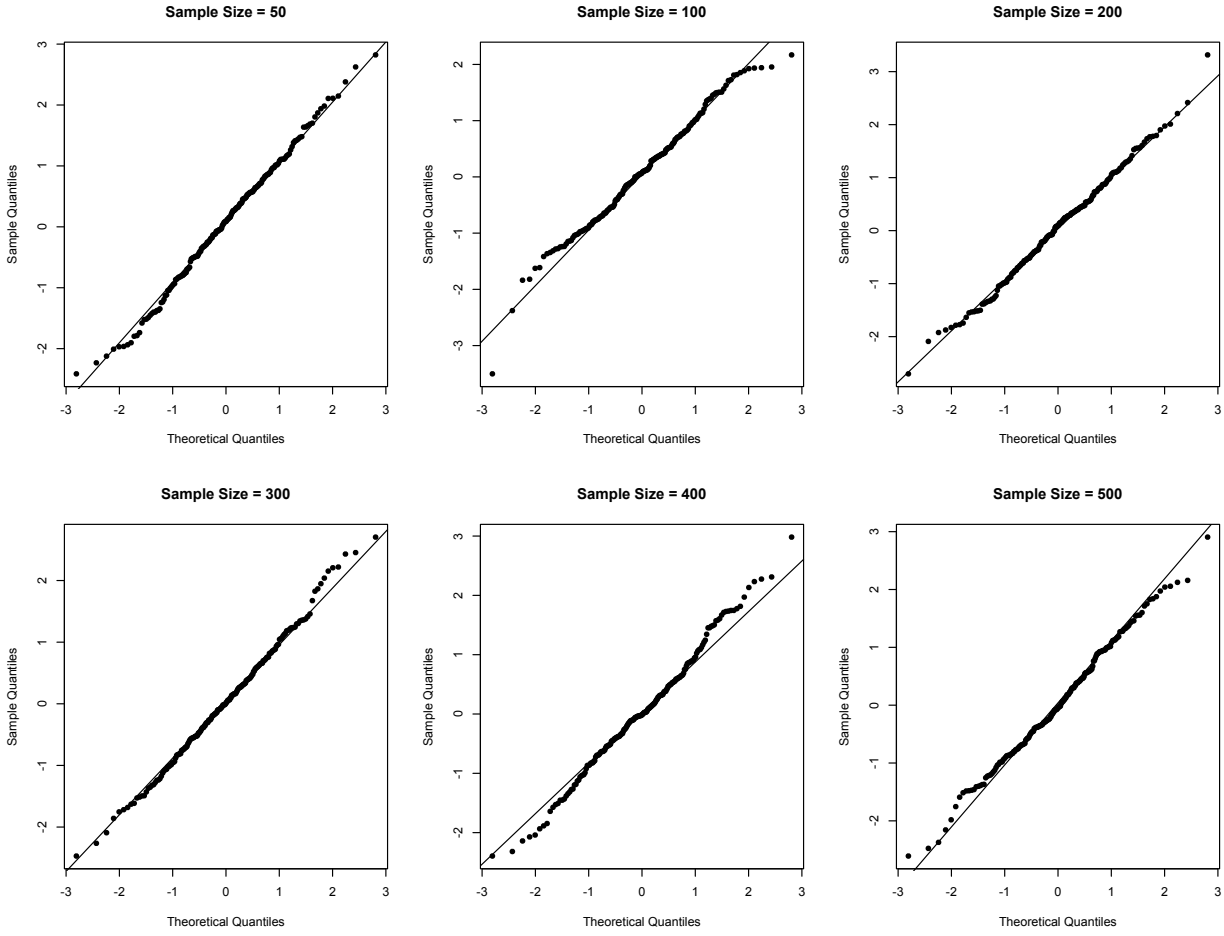

Figure 4.4: Normal Q-Q plot for $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ under 200 iterations and various sample sizes. The true function f_0 is a trigonometric function as defined in (4.14).

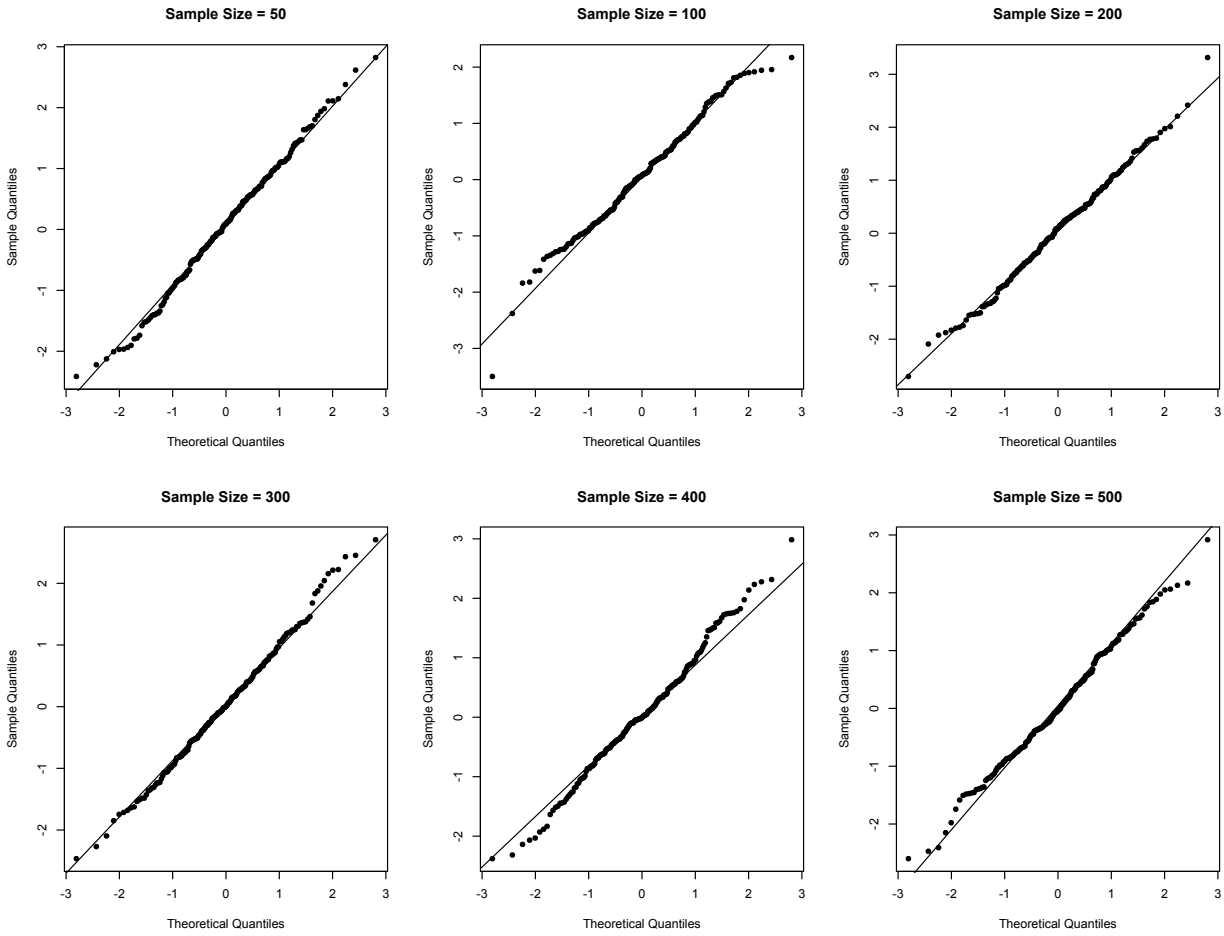

Figure 4.5: Normal Q-Q plot for $n^{-1/2}\sum_{i=1}^{n}\left[\hat{f}_n(\boldsymbol{x}_i) - f_0(\boldsymbol{x}_i)\right]$ under 200 iterations and various sample sizes. The true function f_0 is a continuous function having a non-differential point as defined in (4.15).

4.7 Discussion

We have investigated the asymptotic properties, including the consistency, rate of convergence and asymptotic normality for neural network sieve estimators with one hidden layer. While in practice, the number of hidden unites is often chosen ad hoc, it is important to note that the conditions in the theorems provide theoretical guidelines on choosing the number of hidden units for a neural network with one hidden layer in order to achieve the desired statistical properties. The validity of the conditions made in the theorems has also been checked through simulation results. Theorem 4.5.1 and Theorem 4.5.2 can be served as preliminary work for conducting hypothesis testing on $H_0 : f_0 = h_0$ for a fixed function h_0 according to neural networks. However, currently there is no simple way to check conditions (C3) and (C4) in the theorem, which requires further researches on local entropy numbers for classes of neural networks.

The work conducted in this chapter mainly focus on sieve estimators based on neural networks with one hidden layer and standard sigmoid activation function. There are many extensions on the results presented in this chapter. In fact, the main theorems in this chapter depend heavily on the covering number or the entropy number of the function class consisting of neural network with one hidden layer. Theorem 14.5 in Anthony and Bartlett (2009)[4] provides a general upper bound for the covering number of a function class consisting of deep neural networks with Lipshitz continuous activation functions. Therefore, it is possible to extend our results discussed in this chapter to a deep neural network with Lipshitz continuous activation functions. It is also worthwhile to investigate asymptotic properties of other commonly used deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

When we train a deep neural network, we usually need to fact an overfitting issue and in practice, regularization is frequently used to reduce overfitting. Another natural extension of the results discussed in this chapter is to modify the loss function by involving some regularization terms. By taking regularization into account, we believe the theories may have broader applicability in real world data applications.

Chapter 5

Epilogue

The main goal of this dissertation is to investigate some network-based models with their application to genetic association study and risk prediction. We have seen in Chapter 2 that a conditional autoregressive (CAR) model can achieve higher power when individuals have heterogeneous genetic effects and have robust results under misspecification of weights. The kernel neural network (KNN) model discussed in Chapter 3 provides a novel way to conduct genetic risk prediction and we have shown theoretically and empirically that KNN can reach lower prediction error than classical linear mixed model. Finally, in Chapter 4, we have established the asymptotic properties for neural network sieve estimators, which serves as a foundation for conducting hypothesis testing based on neural networks for later researches. By using the advanced theories in empirical processes, the results were developed based on minimal number of conditions to be checked and as a byproduct, the conditions also provide some guidelines on choosing the number of hidden units in order to achieve desired statistical properties. Here, we will briefly discuss our future work.

In the CAR model discussed in Chapter 2, we mainly focused on the continuous phenotype. In the future, we will extend the CAR model so that it can be applied to discrete dependent variables. Specifically, the CAR model can be extended under the generalized

linear mixed model (GLMM) framework:

$$
\begin{aligned}
y_i|a_i &\sim \text{ind } \exp\left\{\frac{y_i\eta_i - b(\eta_i)}{\phi} + c(y_i,\phi)\right\}, i = 1,\ldots,n \\
a_i|a_j, j \neq i &\sim \mathcal{N}\left(\frac{\gamma}{\sum_{j\neq i} s_{ij}}\sum_{j\neq i} s_{ij}a_j, \frac{\tau}{\sum_{j\neq i} s_{ij}}\right) \\
\eta_i &= g(\mu_i) = \boldsymbol{x}_i^T\boldsymbol{\beta} + a_i,
\end{aligned}
\tag{5.1}
$$

where $g(\cdot)$ is the canonical link function. For example, $g(x) = x$, known as the identity link when $y_i|a_i$ is normally distributed and $g(x) = \log\frac{x}{1-x}$, known as the logit link when $y_i|a_i$ follows a Bernoulli distribution. To test whether genotype-phenotype association exits, it is equivalent to test $H_0 : \tau = 0$ vs $H_1 : \tau > 0$. However, due to the fact that the number of random effects is the same as the sample size, the commonly-used Laplace approximations will not work. So a new framework of approximate inference should be established to conduct the statistical inference.

Some empirical studies reveal that the magnitude difference between the variance component estimates of random effects and the variance component estimate of random error is large, which will tend to make the prediction error small. In future work on KNN, it is necessary to control the magnitude for the variance component estimates of random effects. A commonly used technique to control the magnitude of the parameter in statistics and machine learning is to use regularization. However, direct adding regularization term to loss function of MINQUE is difficult. Instead, we will consider another way to estimate the variance components, known as the variance least square[3]. The basic idea of variance least square is to minimize the matrix distance between sample covariance matrix and population covariance matrix. Specifically, let $\hat{\boldsymbol{\beta}}$ be an OLS estimator for $\boldsymbol{\beta}$ and pretend it was true.

Then

$$\hat{\boldsymbol{E}} = (\boldsymbol{y} - \boldsymbol{Z}\hat{\boldsymbol{\beta}})(\boldsymbol{y} - \boldsymbol{Z}\hat{\boldsymbol{\beta}})^T$$

is clearly an unbiased estimator for the variance-covariance matrix. We know that the marginal covariance matrix of $\boldsymbol{y}$ in KNN is $\text{Var}[\boldsymbol{y}] = \sum_{j=1}^{J} \tau_j \mathbb{E}[\boldsymbol{K}_j(\boldsymbol{U})] + \phi \boldsymbol{I}_n$. So we can estimate the variance component $\boldsymbol{\theta} = [\phi, \boldsymbol{\tau}^T, \boldsymbol{\xi}^T]^T$ based on

$$\hat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}} \text{tr} \left[\left(\hat{\boldsymbol{E}} - \phi \boldsymbol{I}_n - \sum_{j=1}^{J} \tau_j \mathbb{E}[\boldsymbol{K}_j(\boldsymbol{U})] \right)^2 \right]$$

Therefore, to control the magnitude of the variance components, we can consider the penalized estimators via

$$\hat{\boldsymbol{\theta}}_p = \text{argmin}_{\boldsymbol{\theta}} \text{tr} \left[\left(\hat{\boldsymbol{E}} - \phi \boldsymbol{I}_n - \sum_{j=1}^{J} \tau_j \mathbb{E}[\boldsymbol{K}_j(\boldsymbol{U})] \right)^2 \right] + \text{pen}_\lambda(\boldsymbol{\theta}).$$

We will also investigate the statistical properties including the bias and consistency in the futre work.

Having established the basic asymptotic properties for neural networks with one hidden layer, as mentioned in Chapter 4, we will extend the results to deep neural networks, which is a popular tools in deep learning and artificial intelligence, as well as the convolutional neural networks (CNNs) and recurrent neural networks (RNNs). By extending the results to deep neural networks, we expect that some guidelines on selecting the number of hidden units in each layer and the number of layers in the deep network. Considering the loss function with regularization will also be one of the focus in our future research.

APPENDICES

Appendix A

Technical Details and Supplementary Materials for Chapter 2

Proof of Proposition 2.2.1

Proof. Fix $\boldsymbol{a}' = \mathbf{0}$, then based on the first equation in the Brook's Lemma, we have

$$\begin{aligned}
\frac{\pi(\boldsymbol{a})}{\pi(\mathbf{0})} &= \prod_{i=1}^{n} \frac{\pi(a_i|a_1,\ldots,a_{i-1},0,0)}{\pi(0|a_1,\ldots,a_{i-1},0,\ldots,0)} \\
&= \prod_{i=1}^{n} \frac{(2\pi\tau_i^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\tau_i^2}\left(a_i - \sum_{j=1}^{i-1} b_{ij}a_j\right)^2\right\}}{(2\pi\tau_i^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\tau_i^2}\left(-\sum_{j=1}^{i-1} b_{ij}a_j\right)^2\right\}} \\
&= \prod_{i=1}^{n} \exp\left\{-\frac{1}{2\tau_i^2}\left(a_i^2 - 2\sum_{j=1}^{i-1} b_{ij}a_i a_j\right)\right\} \\
&= \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}\frac{a_i^2}{\tau_i^2} + \sum_{i=1}^{n}\frac{1}{\tau_i^2}\sum_{j=1}^{i-1} b_{ij}a_i a_j\right\}
\end{aligned}$$

Similarly, from the second equation in the Brook's Lemma, we have

$$\frac{\pi(\boldsymbol{a})}{\pi(\mathbf{0})} = \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}\frac{a_i^2}{\tau_i^2} + \sum_{i=1}^{n}\frac{1}{\tau_i^2}\sum_{j=i+1}^{n} b_{ij}a_i a_j\right\}$$

By the symmetry of b_{ij}, the density of $\boldsymbol{a}$ can then be expressed as

$$\begin{aligned}
\pi(\boldsymbol{a}) &\propto \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}\frac{a_i^2}{\tau_i^2}+\frac{1}{2}\sum_{i=1}^{n}\frac{1}{\tau_i^2}\sum_{j\neq i}b_{ij}a_ia_j\right\}\\
&= \exp\left\{-\frac{1}{2}\sum_{i=1}^{n}\left[\frac{a_i^2}{\tau_i^2}-\sum_{j\neq i}\frac{b_{ij}}{\tau_i^2}a_ia_j\right]\right\}\\
&= \exp\left\{-\frac{1}{2}\left[\boldsymbol{a}^T\boldsymbol{\Delta}^{-1}\boldsymbol{a}-\boldsymbol{a}^T\boldsymbol{\Delta}^{-1}\boldsymbol{B}\boldsymbol{a}\right]\right\}\\
&= \exp\left\{-\frac{1}{2}\left[\boldsymbol{a}^T\boldsymbol{\Delta}^{-1}(\boldsymbol{I}-\boldsymbol{B})\boldsymbol{a}\right]\right\}
\end{aligned}$$

where

$$\boldsymbol{B}=\begin{bmatrix}0 & b_{12} & b_{13} & \cdots & b_{1n}\\ b_{21} & 0 & b_{23} & \cdots & b_{2n}\\ b_{31} & b_{32} & 0 & \cdots & b_{3n}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ b_{n1} & b_{n2} & b_{n3} & \cdots & 0\end{bmatrix},\quad \boldsymbol{\Delta}=\begin{bmatrix}\tau_1^2 & & & & \\ & \tau_2^2 & & & \\ & & & \ddots & \\ & & & & \tau_n^2\end{bmatrix}.$$

This shows that $\boldsymbol{a}\sim\mathcal{N}_n(\boldsymbol{0},(\boldsymbol{I}-\boldsymbol{B})^{-1}\boldsymbol{\Delta})$ and the proof is finished. □

Proof of Proposition 2.2.2

The proof of this result is based on some facts in matrix analysis. We first provide the definition of a diagonally dominant matrix.

Definition A.0.1 (Diagonally Dominant Matrix)**.** *An $n\times n$ real matrix $\boldsymbol{J}$ is diagonally dominant if*

$$\Delta_i(\boldsymbol{J}):=|\boldsymbol{J}_{ii}|-\sum_{j\neq i}|\boldsymbol{J}_{ij}|\geq 0,\ \textit{for } i=1,\ldots,n. \tag{A.1}$$

If the inequality in (A.1) is a strict inequality, then $\boldsymbol{J}$ *is called a strictly diagonally dominant matrix.*

The following result is a rephrase of Corollary 5.6.17 in Horn and Johnson (1990)[31].

Corollary A.0.1. *Let* $\boldsymbol{A}$ *is an* $n \times n$ *matrix. If* $\boldsymbol{A}$ *is strictly diagonally dominant, then* $\boldsymbol{A}$ *is nonsingular.*

Based on Corollary A.0.1, it is easy to obtain the desired property. The following is the proof for Proposition 2.2.2.

Proof. Based on Corollary A.0.1, it suffices to check that $\boldsymbol{D} - \gamma \boldsymbol{S}$ is strictly diagonally dominant if $|\gamma| < 1$. Note that

$$\boldsymbol{J} := \boldsymbol{D} - \gamma \boldsymbol{S} = \begin{bmatrix} \sum_{j \neq 1} s_{1j} & -\gamma s_{12} & \cdots & -\gamma s_{1n} \\ -\gamma s_{21} & \sum_{j \neq 2} s_{2j} & \cdots & -\gamma s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -\gamma s_{n1} & -\gamma s_{n2} & \cdots & \sum_{j \neq n} s_{nj} \end{bmatrix},$$

then since $s_{ij} \geq 0$ for i, j by the definition of similarity, we have

$$\begin{aligned} \Delta_i(\boldsymbol{J}) &= |\boldsymbol{J}_{ii}| - \sum_{j \neq i} |\boldsymbol{J}_{ij}| \\ &= \left| \sum_{j \neq i} s_{ij} \right| - \sum_{j \neq i} |-\gamma s_{ij}| \\ &= \sum_{j \neq i} s_{ij} - |\gamma| \sum_{j \neq i} s_{ij} \\ &= (1 - |\gamma|) \sum_{j \neq i} s_{ij}. \end{aligned}$$

Hence

$$\Delta_i(\boldsymbol{J}) > 0 \Leftrightarrow |\gamma| < 1,$$

which finishes the proof. □

Top 10 Significant Genes Found by CAR, GGRF and SKAT for Entorhinal, Ventricle and Whole Brain

Table A.1, Table A.2 and Table A.3 summarize the top 10 significant gene selected by the three methods associated with entorhinal, ventricle and whole brain respectively. According to the result, we can find that most of the significant genes associated with entorhinal and whole brain found by GGRF and SKAT are the same. This may be from the fact that both methods are constructed based on the same idea that individuals with similar genotypes tend to have similar phenotypes. Besides, both CAR and SKAT find the gene *G1D76* significant for entorhinal. These two methods also find the gene *G0A21* significant for the whole brain. For ventricle, all the three method discovered that the gene *G14B2* is significant.

Table A.1: Top 10 Genes Detected by GGRF, SKAT and CAR Associated with Entorhinal.

CAR			GGRF			SKAT		
CHR	Gene Name	P-value	CHR	Gene Name	P-value	CHR	Gene Name	P-value
5	*MZB1*	2.43E-05	7	*GPR85*	1.96E-04	7	*MGAM*	6.35E-05
5	*PROB1*	5.11E-05	1	*LRRC40*	2.42E-04	1	*HSD3B1*	1.36E-04
11	*CAT*	5.82E-05	1	*HSD3B1*	2.58E-04	7	*GPR85*	1.41E-04
5	*DNAJC18*	7.46E-05	8	*OXR1*	2.86E-04	1	*LRRC40*	1.55E-04
4	*METTL14*	8.99E-05	1	*CSF1*	4.61E-04	5	*DMGDH*	1.74E-04
1	*OR6F1*	1.34E-04	5	*DMGDH*	4.79E-04	4	*UGT2A3*	2.24E-04
6	*FBXO9*	1.55E-04	4	*UGT2A3*	5.22E-04	2	*HDLBP*	2.54E-04
4	*CAMK2D*	1.95E-04	5	*PTGER4*	5.27E-04	5	*PTGER4*	3.46E-04
19	*FLJ25758*	2.12E-04	10	*C10orf107*	5.67E-04	10	*C10orf107*	3.95E-04
7	*MIR3654*	2.96E-04	15	*MIR4514*	6.33E-04	6	*FBXO9*	4.07E-04

Table A.2: Top 10 Genes Detected by GGRF, SKAT and CAR Associated with Ventricle.

CAR			GGRF			SKAT		
CHR	Gene Name	*P*-value	CHR	Gene Name	*P*-value	CHR	Gene Name	*P*-value
17	*LIG3*	4.57E-05	4	*JCHAIN*	3.23E-05	14	*MEG8*	1.69E-05
18	*CNDP1*	6.21E-05	2	*CYP20A1*	9.99E-05	4	*JCHAIN*	5.47E-05
4	*KIAA0232*	7.89E-05	5	*CEP120*	1.12E-04	19	*CCDC61*	1.76E-04
2	*PGM5P4-AS1*	1.28E-04	1	*TINAGL1*	1.58E-04	1	*MIR4428*	2.76E-04
9	*CTSL*	1.58E-04	19	*CCDC61*	2.29E-04	6	*TREML2*	3.41E-04
19	*ICAM4*	2.15E-04	16	*NPIPB4*	2.94E-04	18	*MIR4743*	3.63E-04
17	*RAD51D*	2.33E-04	2	*RAPH1*	3.07E-04	14	*BCL2L2*	4.73E-04
17	*DNAH17-AS1*	2.42E-04	16	*ABCC11*	4.27E-04	20	*PHACTR3*	4.74E-04
16	*LAT*	2.54E-04	14	*MEG8*	5.01E-04	10	*LOC105378349*	5.13E-04
4	*JCHAIN*	2.73E-04	17	*LINC00670*	5.13E-04	1	*TINAGL1*	5.23E-04

Table A.3: Top 10 Genes Detected by GGRF, SKAT and CAR Associated with Whole Brain.

CAR			GGRF			SKAT		
CHR	Gene Name	*P*-value	CHR	Gene Name	*P*-value	CHR	Gene Name	*P*-value
2	*BCL11A*	1.95E-05	17	*ANKRD13B*	1.18E-04	1	*C1orf194*	1.13E-04
16	*TAOK2*	7.92E-05	20	*B4GALT5*	1.23E-04	5	*BRIX1*	1.29E-04
20	*TRMT6*	1.57E-04	5	*BRIX1*	1.53E-04	17	*ANKRD13B*	1.34E-04
2	*MIR559*	1.63E-04	7	*EEPD1*	1.85E-04	2	*BCL11A*	1.63E-04
16	*TMEM219*	1.76E-04	8	*LOC101929217*	2.18E-04	8	*LOC101929217*	1.83E-04
17	*LOC101559451*	1.78E-04	1	*C1orf194*	2.45E-04	7	*EEPD1*	2.44E-04
6	*LOC101928429*	1.97E-04	17	*C17orf82*	3.42E-04	2	*TMEM163*	3.40E-04
1	*CRNN*	2.62E-04	11	*ME3*	3.45E-04	1	*LINC00467*	4.06E-04
1	*SOAT1*	2.66E-04	2	*TMEM163*	3.76E-04	4	*MAD2L1*	4.45E-04
7	*TRIM24*	2.91E-04	1	*IL6R*	4.87E-04	1	*IL6R*	5.03E-04

Appendix B

Technical Details and Supplementary Materials for Chapter 3

In this appendix, we are going to provide the proofs for the results in Chapter 3. Before we head into the proof, we need some background knowledge on concentration inequalities and matrix analysis.

Some Results from Concentration Inequality and Matrix Analysis

Sub-Gaussian and Sub-Exponential Inequalities

In this part, we present two basic concentration inequalities used in the main text. For more details on concentration inequality, readers may refer to Buldygin and Kozachenko (2000)[11], Boucheron et al. (2004)[8] and Ledoux (2005)[39].

Definition B.0.1. *(i) A random variable X with mean $\mu = \mathbb{E}[X]$ is called sub-Gaussian if there exists a number $\sigma \geq 0$ such that*

$$\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq \exp\left\{\frac{1}{2}\lambda^2\sigma^2\right\}, \quad \forall \lambda \in \mathbb{R}$$

The constant σ is known as the sub-Gaussian parameter.

(ii) A random variable X with mean $\mu = \mathbb{E}[X]$ is called sub-exponential (also called pre-Gaussian) if there exist non-negative constants (β, α) such that

$$\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq \exp\left\{\frac{1}{2}\lambda^2\beta^2\right\}, \quad \forall|\lambda| < \frac{1}{\alpha}.$$

The pair of constants (β, α) is known as the sub-exponential parameter.

Based on the definition of sub-Gaussian random variables, it is clear that if $X \sim \mathcal{N}(\mu, \sigma^2)$, then X is sub-Gaussian with sub-Gaussian parameter σ. The tail probabilities of both sub-Gaussian and sub-exponential random variables can be bounded exponentially. For sub-Gaussian random variables, the result is the famous Hoeffding inequality.

Theorem B.0.1. *[30] Suppose that the random variables $X_1, \ldots, X_n$ are independent and X_i has mean μ_i and sub-Gaussian parameter σ_i. Then for all $t > 0$,*

$$\mathbb{P}\left(\left|\sum_{i=1}^{n}(X_i - \mu_i)\right| > t\right) \leq 2\exp\left\{-\frac{t^2}{2\sum_{i=1}^{n}\sigma_i^2}\right\}.$$

Theorem B.0.2. *Suppose that X is a sub-exponential random variable with mean μ and sub-exponential parameters (β, α). Then for all $t > 0$,*

$$\mathbb{P}\left(|X - \mu| > t\right) \leq \begin{cases} 2\exp\left\{-\frac{t^2}{2\beta^2}\right\}, & \text{if } 0 < t \leq \frac{\beta^2}{\alpha} \\ 2\exp\left\{-\frac{t}{2\alpha}\right\}, & \text{if } t > \frac{\beta^2}{\alpha} \end{cases}.$$

Results on Matrix Analysis

Proposition B.0.1 shows that a inverse map of a matrix is a continuous map, which will be frequently used in later parts when we approximate the average prediction errors.

Proposition B.0.1 (Gentle (2007)[23])**.** *Let $\boldsymbol{\Psi}$ be an arbitrary invertible matrix. Then the map $f : \boldsymbol{\Psi} \mapsto \boldsymbol{\Psi}^{-1}$ is continuous.*

Proof. Since $\boldsymbol{\Psi}^{-1} = |\boldsymbol{\Psi}|^{-1}\text{Adj}(\boldsymbol{\Psi})$, where $\text{Adj}(\boldsymbol{\Psi})$ is the adjugate matrix of $\boldsymbol{\Psi}$, i.e., $\text{Adj}(\boldsymbol{\Psi}) = \boldsymbol{C}^T$ and $\boldsymbol{C}$ is the cofactor matrix of $\boldsymbol{\Psi}$ with $\boldsymbol{C}_{ij} = (-1)^{i+j}\boldsymbol{M}_{ij}$ and $\boldsymbol{M}_{ij}$ is the (i, j) cofactor of $\boldsymbol{\Psi}$. Based on the definition of determinant, it is easy to see that the map $g : \boldsymbol{\Psi} \mapsto |\boldsymbol{\Psi}|$ is continuous and the map $h : \boldsymbol{\Psi} \mapsto \text{Adj}(\boldsymbol{\Psi})$ is continuous as well. Therefore, the map $f : \boldsymbol{\Psi} \mapsto \boldsymbol{\Psi}^{-1}$ is continuous. □

Another result that will be used later is that any two matrix norms are equivalent in the sense that for any given pair of matrix norms $\|\cdot\|_s$ and $\|\cdot\|_t$, there is a finite positive constant C_{st} such that

$$\|\boldsymbol{A}\|_s \leq C_{st}\|\boldsymbol{A}\|_t, \quad \forall \boldsymbol{A} \in \mathcal{M}_n,$$

where $\mathcal{M}_n$ is the collection of all $n \times n$ matrices.

Proof of Lemma 3.2.1

Proof. Note that

$$\mathbb{E}\left[\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m}\right] = \frac{1}{m}\sum_{k=1}^{m}\mathbb{E}\left[\boldsymbol{w}_{ik}\boldsymbol{w}_{jk}\right] = \sigma_{ij}.$$

Now we consider the Taylor expansion of $\boldsymbol{K}_{ij}(\boldsymbol{U})$ around σ_{ij}:

$$\boldsymbol{K}_{ij}(\boldsymbol{U}) = g(\sigma_{ij}) + g'(\sigma_{ij})\left(\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right) + \frac{1}{2}g''(\eta_{ij})\left(\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right)^2,$$

where η_{ij} is between σ_{ij} and $\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m}$. Then the truncation error can be evaluated as follows. For all $\delta > 0$,

$$\begin{aligned}
\mathbb{P}\left(\left|\boldsymbol{K}_{ij}(\boldsymbol{U}) - \hat{\boldsymbol{K}}_{ij}(\boldsymbol{U})\right| > \delta\right) &= \mathbb{P}\left(\frac{1}{2}|g''(\eta_{ij})|\left(\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right)^2 > \delta\right) \\
&\leq \mathbb{P}\left(\frac{1}{2}M\left(\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right)^2 > \delta\right) \\
&= \mathbb{P}\left(\left|\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right| > \sqrt{\frac{2\delta}{M}}\right).
\end{aligned}$$

So it suffices to evaluate the tail probability $\mathbb{P}\left(\left|\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right| > \sqrt{\frac{2\delta}{M}}\right)$. Note that

$$\boldsymbol{w}_j|\boldsymbol{w}_i \sim \mathcal{N}_m\left(\frac{\sigma_{ij}}{\sigma_{ii}}\boldsymbol{w}_i, \left(\sigma_{jj} - \frac{\sigma_{ij}^2}{\sigma_{ii}}\right)\boldsymbol{I}_m\right),$$

we get

$$\boldsymbol{w}_i^T\boldsymbol{w}_j|\boldsymbol{w}_i \sim \mathcal{N}\left(\frac{\sigma_{ij}}{\sigma_{ii}}\boldsymbol{w}_i^T\boldsymbol{w}_i, \left(\sigma_{jj} - \frac{\sigma_{ij}^2}{\sigma_{ii}}\right)\boldsymbol{w}_i^T\boldsymbol{w_i}\right)$$

Therefore, given $\boldsymbol{w}_i$, the random variable $\boldsymbol{w}_i^T\boldsymbol{w}_j$ is sub-Gaussian with sub-Gaussian parameter $\sigma_{ii}^{-1}s_{i,j}\boldsymbol{w}_i^T\boldsymbol{w}_i$, where $s_{ij}=\sigma_{ii}\sigma_{jj}-\sigma_{ij}^2$. Since

$$\begin{aligned}\mathbb{P}\left(\left|\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m}-\sigma_{ij}\right|>\sqrt{\frac{2\delta}{M}}\right)&\leq\mathbb{P}\left(\left|\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m}-\frac{\sigma_{ij}}{\sigma_{ii}}\frac{\boldsymbol{w}_i^T\boldsymbol{w}_i}{m}\right|>\frac{1}{2}\sqrt{\frac{2\delta}{M}}\right)+\\&\qquad\mathbb{P}\left(\left|\frac{\sigma_{ij}}{\sigma_{ii}}\frac{\boldsymbol{w}_i^T\boldsymbol{w}_i}{m}-\sigma_{ij}\right|>\frac{1}{2}\sqrt{\frac{2\delta}{M}}\right)\\&:=(I)+(II),\end{aligned}$$

it suffices to provide bounds for (I) and (II).

For (II), note that $\boldsymbol{w}_i\sim\mathcal{N}_m(\boldsymbol{0},\sigma_{ii}\boldsymbol{I}_m)$, we have $\frac{1}{\sigma_{ii}}\boldsymbol{w}_i^T\boldsymbol{w}_i\sim\chi^2_m$, which implies that

$$\begin{aligned}\mathbb{E}\left[e^{\lambda\left(\frac{\boldsymbol{v}_i^T\boldsymbol{v}_i}{\sigma_{ii}}-m\right)}\right]&=e^{-\lambda m}\mathbb{E}\left[e^{\lambda\frac{\boldsymbol{v}_i^T\boldsymbol{v}_i}{\sigma_{ii}}}\right]=e^{-\lambda m}(1-2\lambda)^{-\frac{m}{2}},\quad\text{for }\lambda<\frac{1}{2}\\&=\left(\frac{e^{-\lambda}}{\sqrt{1-2\lambda}}\right)^m\\&\leq e^{2m\lambda^2},\quad\text{for all }|\lambda|<\frac{1}{4}\\&=e^{\frac{4m\lambda^2}{2}},\quad\text{for all }|\lambda|<\frac{1}{4},\end{aligned}$$

i.e., $\frac{1}{\sigma_{ii}}\boldsymbol{w}_i^T\boldsymbol{w}_i$ is sub-exponential with parameters $(2\sqrt{m},4)$. Hence, for $\sigma_{ij}\neq0$, by Theorem B.0.2, we have

$$\begin{aligned}(II)&=\mathbb{P}\left(\left|\frac{1}{\sigma_{ii}}\boldsymbol{w}_i^T\boldsymbol{w}_i-m\right|>\frac{2m}{|\sigma_{ij}|}\sqrt{\frac{2\delta}{M}}\right)\\&\leq2\exp\left\{-\left(\frac{\delta m}{\sigma_{ij}^2M}\wedge\frac{m}{4|\sigma_{ij}|}\sqrt{\frac{2\delta}{M}}\right)\right\}\end{aligned}\tag{B.1}$$

If $\sigma_{ij}=0$, then $(II)=0$.

For (I), by Hoeffding inequality, we have

$$
\begin{aligned}
(I) &= \mathbb{E}_{\boldsymbol{w}_i}\left[\mathbb{P}\left(\left|\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \frac{\sigma_{ij}}{\sigma_{ii}}\frac{\boldsymbol{w}_i^T\boldsymbol{w}_i}{m}\right| > \frac{1}{2}\sqrt{\frac{2\delta}{M}}\,\middle|\,\boldsymbol{w}_i\right)\right] \\
&= \mathbb{E}_{\boldsymbol{w}_i}\left[\mathbb{P}\left(\left|\boldsymbol{w}_i^T\boldsymbol{w}_j - \frac{\sigma_{ij}}{\sigma_{ii}}\boldsymbol{w}_i^T\boldsymbol{w}_i\right| > \frac{m}{2}\sqrt{\frac{2\delta}{M}}\,\middle|\,\boldsymbol{w}_i\right)\right] \\
&\leq \mathbb{E}_{\boldsymbol{w}_i}\left[2\exp\left\{-\frac{\sigma_{ii}m^2\delta}{4Ms_{ij}\boldsymbol{w}_i^T\boldsymbol{w}_i}\right\}\right] \qquad \text{(B.2)}
\end{aligned}
$$

From Theorem A in Inglot (2010)[34] stated that for a random variable $\chi \sim \chi^2_m$, the $100(1-\alpha)$th percentile is upper bounded by $m + \log(1/\alpha) + 2\sqrt{m\log(1/\alpha)}$, which is of the order $\mathcal{O}(m)$ as $m \to \infty$. Now, since $\sigma_{ii}^{-1}\boldsymbol{w}_i^T\boldsymbol{w}_i \sim \chi^2_m$, we get for any $\alpha \in (0,1)$,

$$
\mathbb{P}\left(\sigma_{ii}^{-1}\boldsymbol{w}_i^T\boldsymbol{w}_i \geq m + 2\log\frac{1}{\alpha} + 2\sqrt{2m\log\frac{1}{\alpha}}\right) = \alpha.
$$

Let $q(\alpha, m) = m + 2\log(1/\alpha) + 2\sqrt{2m\log(1/\alpha)}$. Since the function $\exp\{-a/x\}$ is increasing in x for $a > 0$, we can further bound (B.2) as follows:

$$
\begin{aligned}
(I) &\leq \mathbb{E}_{\boldsymbol{w}_i}\left[2\exp\left\{-\frac{m^2\delta}{4Ms_{ij}\sigma_{ii}^{-1}\boldsymbol{w}_i^T\boldsymbol{w}_i}\right\}\mathbb{I}_{\{\sigma_{ii}^{-1}\boldsymbol{w}_i^T\boldsymbol{w}_i \leq q(\alpha,m)\}}\right] + \\
&\qquad\qquad \mathbb{E}_{\boldsymbol{w}_i}\left[2\exp\left\{-\frac{m^2\delta}{4Ms_{ij}\sigma_{ii}^{-1}\boldsymbol{w}_i^T\boldsymbol{w}_i}\right\}\mathbb{I}_{\{\sigma_{ii}^{-1}\boldsymbol{w}_i^T\boldsymbol{w}_i \geq q(\alpha,m)\}}\right] \\
&\leq 2\exp\left\{-\frac{m^2\delta}{4Ms_{ij}q(\alpha,m)}\right\} + 2\mathbb{P}\left(\sigma_{ii}^{-1}\boldsymbol{w}_i^T\boldsymbol{w}_i \geq q(\alpha,m)\right) \\
&\leq 2\exp\left\{-\frac{m^2\delta}{4Ms_{ij}q(\alpha,m)}\right\} + 2\alpha.
\end{aligned}
$$

By choosing $\alpha = \exp\{-m\}$, we get $q(\alpha, m) = m + 2m + 2\sqrt{m^2} = 5m$ so that

$$\begin{aligned}(I) &\leq 2\exp\left\{-\frac{m^2\delta}{20Mms_{ij}}\right\} + 2\exp\{-m\} \\ &\leq 2\exp\left\{-m\left(1 \wedge \frac{\delta}{20Ms_{ij}}\right)\right\}. \end{aligned} \tag{B.3}$$

Combining (B.3) and (B.1), we obtain for all $\delta > 0$,

$$\begin{aligned}\mathbb{P}\left(\left|\boldsymbol{K}_{ij}(\boldsymbol{U}) - \hat{\boldsymbol{K}}_{ij}(\boldsymbol{U})\right| > \delta\right) &\leq (I) + (II) \\ &\leq 4\exp\left\{-m\left(1 \wedge \frac{\delta}{20Ms_{ij}} \wedge \frac{\delta}{M\sigma_{ij}^2} \wedge \frac{1}{4|\sigma_{ij}|}\sqrt{\frac{2\delta}{M}}\right)\right\}.\end{aligned}$$

□

Remark B.0.1. *The condition (3.12) can be weakened as follows:*

$$g''\left(\lambda\sigma_{ij} + (1-\lambda)\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m}\right) = \mathcal{O}_p(1)$$

for all $\lambda \in [0,1]$. In such case, the evaluation of truncation error can be modified as follows: For all $\delta > 0$, there exists $M_\delta > 0$ such that

$$\mathbb{P}\left(\left|g''(\eta_{ij})\right| > M_\delta\right) < \frac{\delta}{2},$$

where $\eta_{ij} = \lambda\sigma_{ij} + (1-\lambda)\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m}$ *for some* $\lambda \in [0,1]$ *and then*

$$\begin{aligned}
\mathbb{P}\left(\left|\boldsymbol{K}_{ij}(\boldsymbol{U}) - \hat{\boldsymbol{K}}_{ij}(\boldsymbol{U})\right| > \delta\right) &= \mathbb{P}\left(\frac{1}{2}g''(\eta_{ij})\left(\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right)^2 > \delta\right) \\
&\leq \mathbb{P}\left(\left\{\frac{1}{2}g''(\eta_{ij})\left(\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right)^2 > \delta\right\} \cap \left\{|g''(\eta_{ij})| \leq M_\delta\right\}\right) + \\
&\qquad\qquad \mathbb{P}\left(|g''(\eta_{ij})| > M_\delta\right) \\
&\leq \mathbb{P}\left(\left|\frac{\boldsymbol{w}_i^T\boldsymbol{w}_j}{m} - \sigma_{ij}\right| > \sqrt{\frac{2\delta}{M_\delta}}\right) + \frac{\delta}{2} \\
&\leq 4\exp\left\{-m\left(1 \wedge \frac{\delta}{20M_\delta s_{ij}} \wedge \frac{\delta}{M_\delta\sigma_{ij}^2} \wedge \frac{1}{4|\sigma_{ij}|}\sqrt{\frac{2\delta}{M_\delta}}\right)\right\} + \frac{\delta}{2}
\end{aligned}$$

Now, we can choose $m > \left(1 \wedge \frac{\delta}{20M_\delta s_{ij}} \wedge \frac{\delta}{M_\delta\sigma_{ij}^2} \wedge \frac{1}{4|\sigma_{ij}|}\sqrt{\frac{2\delta}{M_\delta}}\right)^{-1} \log\frac{8}{\delta}$ *so that*

$$\mathbb{P}\left(\left|\boldsymbol{K}_{ij}(\boldsymbol{U}) - \hat{\boldsymbol{K}}_{ij}(\boldsymbol{U})\right| > \delta\right) < \frac{\delta}{2} + \frac{\delta}{2} = \delta$$

and hence $\boldsymbol{K}_{ij}(\boldsymbol{U}) = \hat{\boldsymbol{K}}_{ij}(\boldsymbol{U}) + o_p(1)$.

Proof of Lemma 3.3.1

Proof. (i) When $f(x) = x$, we have $\boldsymbol{K}(\boldsymbol{U}) = \frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T$ and since the hidden random vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ are i.i.d, we can know that

$$\boldsymbol{u}_1\boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_m\boldsymbol{u}_m^T \sim \text{ i.i.d. } \mathcal{W}_n\left(1, \sum_{l=1}^{L}\xi_l\boldsymbol{K}_l(\boldsymbol{X})\right),$$

where $\mathcal{W}_n\left(1, \sum_{l=1}^{L} \xi_l \boldsymbol{K}_l(\boldsymbol{X})\right)$ stands for a Wishart distribution with degrees of freedom 1 and covariance matrix $\sum_{l=1}^{L} \xi_l \boldsymbol{K}_l(\boldsymbol{X})$. Therefore, the Strong Law of Large Numbers implies that as $m \to \infty$,

$$\boldsymbol{K}(\boldsymbol{U}) = \frac{1}{m}\boldsymbol{U}\boldsymbol{U}^T = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{u}_i \boldsymbol{u}_i^T \to \mathbb{E}\left[\boldsymbol{u}_1 \boldsymbol{u}_1^T\right] = \sum_{l=1}^{L} \xi_l \boldsymbol{K}_l(\boldsymbol{X}), \quad \text{a.s..}$$

Since the the map $\psi : \boldsymbol{A} \mapsto \boldsymbol{A}^{-1}$ for non-singular matrix $\boldsymbol{A}$ is continuous, then we can obtain the following result by using the Continuous Mapping Theorem.

$$(\tilde{\tau}\boldsymbol{K}(\boldsymbol{U}) + \boldsymbol{I}_n)^{-1} \to \left(\sum_{l=1}^{L} \tilde{\tau}\xi \boldsymbol{K}_l(\boldsymbol{X}) + \boldsymbol{I}_n\right)^{-1}, \quad \text{a.s., as } m \to \infty$$

Let $\tilde{\boldsymbol{A}} = (\tilde{\tau}\boldsymbol{K}(\boldsymbol{U}) + \boldsymbol{I}_n)^{-1}$, we have

$$\begin{aligned}
\max_{1 \le i,j \le n} |\tilde{\boldsymbol{A}}_{ij}| \le \|\tilde{\boldsymbol{A}}\|_\infty &\lesssim \|\tilde{\boldsymbol{A}}\|_{\text{op}} \\
&= \lambda_{\max}\left((\tilde{\tau}\boldsymbol{K}(\boldsymbol{U}) + \boldsymbol{I}_n)^{-1}\right) \\
&= \frac{1}{\tilde{\tau}\lambda_{\min}(\boldsymbol{K}(\boldsymbol{U})) + 1} \le 1 < \infty.
\end{aligned}$$

Therefore, by Bounded Convergence Theorem,

$$\boldsymbol{A} := \mathbb{E}\left[(\tilde{\tau}\boldsymbol{K}(\boldsymbol{U}) + \boldsymbol{I}_n)^{-1}\right] \to \left(\sum_{l=1}^{L} \tilde{\tau}\xi \boldsymbol{K}_l(\boldsymbol{X}) + \boldsymbol{I}_n\right)^{-1}, \quad \text{a.s. as } m \to \infty.$$

Asymptotically, we get as $m \to \infty$.

$$R \simeq \boldsymbol{y}^T \left(\sum_{l=1}^{L} \tilde{\tau}\xi \boldsymbol{K}_l(\boldsymbol{X}) + \boldsymbol{I}_n\right)^{-2} \boldsymbol{y}.$$

(ii) Note that equation (3.13) can be further written as

$$\boldsymbol{K}(\boldsymbol{U}) = f[\boldsymbol{\Sigma}] + o_P(1),$$

or equivalently, $\boldsymbol{K}(\boldsymbol{U}) \xrightarrow{P} f[\boldsymbol{\Sigma}]$ as $m \to \infty$ element-wisely. Similarly, under the assumption of $\|\boldsymbol{K}(\boldsymbol{U})\|_{\text{op}} < \infty$ a.s., we have

$$\mathbb{E}\left[\tilde{\tau}\boldsymbol{K}(\boldsymbol{U}) + \boldsymbol{I}_n\right] \to \tilde{\tau} f[\boldsymbol{\Sigma}] + \boldsymbol{I}_n.$$

Hence by Bounded Convergence Theorem and Continuous Mapping Theorem, we have

$$\boldsymbol{A} = \mathbb{E}\left[(\tilde{\tau}\boldsymbol{K}(\boldsymbol{U}) + \boldsymbol{I}_n)^{-1}\right] \to (\tilde{\tau} f[\boldsymbol{\Sigma}] + \boldsymbol{I}_n)^{-1}, \quad \text{as } m \to \infty,$$

which shows that as $m \to \infty$,

$$R \simeq \boldsymbol{y}^T \left(\tilde{\tau} f\left[\sum_{l=1}^{L} \xi \boldsymbol{K}_l(\boldsymbol{X}) \right] + \boldsymbol{I}_n \right)^{-2} \boldsymbol{y}.$$

□

Proof of Proposition 3.3.2

Proof. The result follows by noting that

$$\begin{aligned}
APELMM &= \mathbb{E}\left[(\boldsymbol{y} - \mathbb{E}[\boldsymbol{a}|\boldsymbol{y}])^T(\boldsymbol{y} - \mathbb{E}[\boldsymbol{a}|\boldsymbol{y}])\right] \\
&= \mathbb{E}\left[\boldsymbol{y}^T\left(\boldsymbol{I}_n - \tilde{\sigma}_R^2\boldsymbol{\Sigma}(\tilde{\sigma}_R^2\boldsymbol{\Sigma} + \boldsymbol{I}_n)^{-1}\right)^T\left(\boldsymbol{I}_n - \tilde{\sigma}_R^2\boldsymbol{\Sigma}(\tilde{\sigma}_R^2\boldsymbol{\Sigma} + \boldsymbol{I}_n)^{-1}\right)\boldsymbol{y}\right] \\
&= \mathbb{E}\left[\boldsymbol{y}^T\left((\tilde{\sigma}_R^2\boldsymbol{\Sigma} + \boldsymbol{I}_n)^{-1}\right)^2\boldsymbol{y}\right] \\
&= \mathrm{tr}\left[\left((\tilde{\sigma}_R^2\boldsymbol{\Sigma} + \boldsymbol{I}_n)^{-1}\right)^2\left(\sigma_R^2\boldsymbol{\Sigma} + \phi\boldsymbol{I}_n\right)\right] \\
&= \phi\mathrm{tr}\left[(\tilde{\sigma}_R^2\boldsymbol{\Sigma} + \boldsymbol{I}_n)^{-1}\right] \\
&= \phi\sum_{i=1}^{n}\left(\tilde{\sigma}_R^2\lambda_i(\boldsymbol{\Sigma}) + 1\right)^{-1}.
\end{aligned}$$

□

More Simulation Results

Figure B.1 demonstrates the performance of LMM and KNN in terms of prediction error when the inverse logistic function and the polynomial function of order 2 are used.

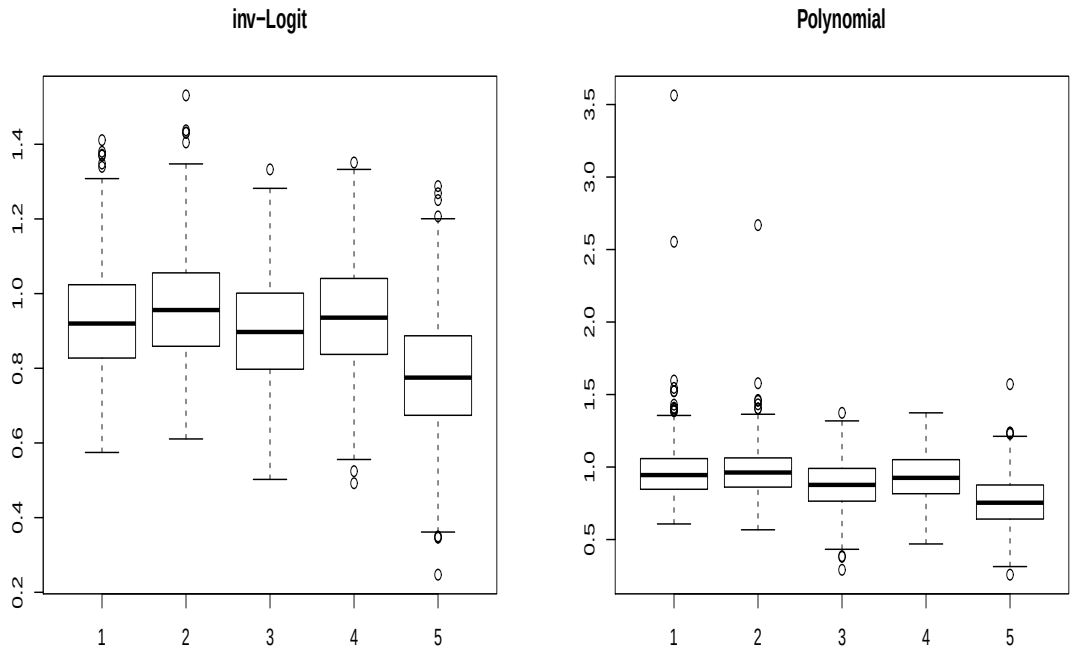

Figure B.1: The boxplots for linear mixed models (LMM) and kernel neural network (KNN) in terms of prediction errors. The left panel shows the results when an inverse logistic function is used and the right panel shows the results when a polynomial function of order 2 is used. In the horizontal axis, "1" corresponds to the LMM; "2" corresponds to the KNN with product input kernel and product output kernel; "3" corresponds to the KNN with product input and polynomial output; "4" corresponds to the KNN with polynomial input and product output and "5" corresponds to the polynomial input and polynomial output.

Appendix C

Technical Details and Supplementary Materials for Chapter 4

In this appendix, we are going to explore some basic properties of the parameter space $(\mathcal{F}, \|\cdot\|_n)$ discussed in Chapter 4.

Proposition C.0.1. *The space $(\mathcal{F}, \|\cdot\|_n)$ is a pseudo-normed space.*

Proof. Note that $\|f\|_n = \left(\frac{1}{n}\sum_{i=1}^n f^2(\boldsymbol{x}_i)\right)^{1/2}$, then

(i) Based on the definition of $\|\cdot\|_n$, it is clear that $\|f\|_n \geq 0$, for any $f \in \mathcal{F}$.

(ii) For any $\lambda \in \mathbb{R}$ and $f \in \mathcal{F}$,

$$\|\lambda f\|_n = \left(\frac{1}{n}\sum_{i=1}^n \lambda^2 f^2(\boldsymbol{x}_i)\right)^{1/2} = |\lambda|\|f\|_n$$

(iii) For any $f, g \in \mathcal{F}$,

$$\begin{aligned}
\|f+g\|_n &= \left(\frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i)+g(\boldsymbol{x}_i))^2\right)^{1/2} = \left(\sum_{i=1}^n \left(\frac{1}{\sqrt{n}}f(\boldsymbol{x}_i)+\frac{1}{\sqrt{n}}g(\boldsymbol{x}_i)\right)^2\right)^{1/2} \\
&\leq \left(\sum_{i=1}^n \left(\frac{1}{\sqrt{n}}f(\boldsymbol{x}_i)\right)^2\right)^{1/2} + \left(\sum_{i=1}^n \left(\frac{1}{\sqrt{n}}g(\boldsymbol{x}_i)\right)^2\right)^{1/2} \\
&= \|f\|_n + \|g\|_n,
\end{aligned}$$

where we have used the triangle inequality for classical Euclidean norm.

Therefore, we can know that $(\mathcal{F}, \|\cdot\|_n)$ is a pseudo-normed space. □

Proposition C.0.2. *There is an pseudo-inner product on $\mathcal{F}$ such that $\|f\|^2 = \langle f, f\rangle$ for any $f \in \mathcal{F}$. Moreover, the pseudo-inner product is given by*

$$\langle f, g\rangle = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)g(\boldsymbol{x}_i), \quad \forall f, g \in \mathcal{F}.$$

Proof. Based on the theorem attributed to Fréchet, von Neumann and Jordan (see for example, Proposition 14.1.2 in Blanchard and Brüning (2015)[7]), to show the existence of the inner product, it suffices to check the parallelogram law of the pseudo-norm and the corresponding pseudo-inner product can be obtained via the polarization identity. To check to validity of the parallelogram law, we note that for any $f, g \in \mathcal{F}$,

$$\begin{aligned}
\|f+g\|_n^2 + \|f-g\|_n^2 &= \frac{1}{n}\sum_{i=1}^{n}\left(f(\boldsymbol{x}_i)+g(\boldsymbol{x}_i)\right)^2 + \frac{1}{n}\sum_{i=1}^{n}\left(f(\boldsymbol{x}_i)-g(\boldsymbol{x}_i)\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} f^2(\boldsymbol{x}_i) + \frac{2}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)g(\boldsymbol{x}_i) + \frac{1}{n}\sum_{i=1}^{n} g^2(\boldsymbol{x}_i) \\
&\qquad + \frac{1}{n}\sum_{i=1}^{n} f^2(\boldsymbol{x}_i) - \frac{2}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)g(\boldsymbol{x}_i) + \frac{1}{n}\sum_{i=1}^{n} g^2(\boldsymbol{x}_i) \\
&= \frac{2}{n}\sum_{i=1}^{n} f^2(\boldsymbol{x}_i) + \frac{2}{n}\sum_{i=1}^{n} g^2(\boldsymbol{x}_i) \\
&= 2\|f\|_n^2 + 2\|g\|_n^2.
\end{aligned}$$

Hence, the parallelogram law is satisfied by the pseudo-norm and hence the pseudo-inner

product does exist and by the polarization identity, we get for any $f, g \in \mathcal{F}$,

$$\begin{aligned}\langle f, g\rangle &= \frac{1}{4}\left(\|f+g\|_n^2 - \|f-g\|_n^2\right) \\ &= \frac{1}{4}\left(\frac{1}{n}\sum_{i=1}^n f^2(\boldsymbol{x}_i) + \frac{2}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)g(\boldsymbol{x}_i) + \frac{1}{n}\sum_{i=1}^n g^2(\boldsymbol{x}_i) - \frac{1}{n}\sum_{i=1}^n f^2(\boldsymbol{x}_i)\right. \\ &\qquad \left. + \frac{2}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)g(\boldsymbol{x}_i) - \frac{1}{n}\sum_{i=1}^n g^2(\boldsymbol{x}_i)\right) \\ &= \frac{1}{n}\sum_{i=1}^n f(\boldsymbol{x}_i)g(\boldsymbol{x}_i).\end{aligned}$$

□

Let

$$\mathcal{G} = \left\{g : \mathbb{R} \to \mathbb{R}, \int \left|g'(z)\right| \mathrm{d}z \leq M\right\}$$

be the class of functions of bounded variation in $\mathbb{R}$ (see Example 9.3.3 in van de Geer (2000)[68]). The following proposition shows that $\mathcal{F}_{r_n} \subset \mathcal{G}$ for fixed n.

Proposition C.0.3. *For fixed n, $\mathcal{F}_{r_n} \subset \mathcal{G}$.*

Proof. For any $f \in \mathcal{F}_{r_n}$, we have

$$f(x) = \alpha_0 + \sum_{j=1}^{r_n} \alpha_j \sigma\left(\gamma_j x + \gamma_{0,j}\right)$$

so that

$$f'(x) = \sum_{j=1}^{r_n} \alpha_j \gamma_j \sigma\left(\gamma_j x + \gamma_{0,j}\right)\left[1 - \sigma\left(\gamma_j x + \gamma_{0,j}\right)\right].$$

Without loss of generality, we assume that $\gamma_j \neq 0$ for $j = 1, \ldots, r_n$. Then note that

$$\begin{aligned}
\int |f'(x)| \mathrm{d}x &= \int \left| \sum_{j=1}^{r_n} \alpha_j \gamma_j \sigma\left(\gamma_j x + \gamma_{0,j}\right) \left[1 - \sigma\left(\gamma_j x + \gamma_{0,j}\right)\right] \right| \mathrm{d}x \\
&\leq \sum_{j=1}^{r_n} |\alpha_j||\gamma_j| \int \sigma(\gamma_j x + \gamma_{0,j})[1 - \sigma(\gamma_j x + \gamma_{0,j})] \mathrm{d}x \\
&\leq \sum_{j=1}^{r_n} |\alpha_j| \frac{|\gamma_j|}{\gamma_j} \int \sigma(u_j)(1 - \sigma(u_j)) \mathrm{d}u_j,
\end{aligned}$$

where in the last inequality, we let $u_j = \gamma_j x + \gamma_{0,j}$. Clearly, $|\gamma_j|/\gamma_j = \text{sign}(\gamma_j)$. Moreover, since

$$\begin{aligned}
\int \sigma(x)(1 - \sigma(x)) \mathrm{d}x &= \int \frac{e^x}{(1 + e^x)^2} \mathrm{d}x \\
&= \int_0^\infty \frac{1}{(1 + u)^2} \mathrm{d}u \quad \text{(by letting } u = e^x\text{)} \\
&= -(1 + u)^{-1} \Big|_0^\infty \\
&= 1,
\end{aligned}$$

we get for fixed n,

$$\int |f'(x)| \mathrm{d}x \leq \sum_{j=1}^{r_n} |\alpha_j| \text{sign}(\gamma_j) \leq \sum_{j=1}^{r_n} |\alpha_j| \leq V_n.$$

Therefore, $f \in \mathcal{G}$ and the desired result follows. □

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Gad Abraham and Michael Inouye. Genomic risk prediction of complex human disease and its clinical application. *Current opinion in genetics & development*, 33:10–16, 2015.

[2] Kenneth S Alexander. Probability inequalities for empirical processes and a law of the iterated logarithm. *The Annals of Probability*, pages 1041–1067, 1984.

[3] Takeshi Amemiya. A note on a heteroscedastic model. *Journal of Econometrics*, 6(3):365–370, 1977.

[4] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

[5] Sudipto Banerjee, Bradley P Carlin, and Alan E Gelfand. *Hierarchical modeling and analysis for spatial data*. Crc Press, 2014.

[6] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[7] Philippe Blanchard and Erwin Brüning. *Mathematical methods in Physics: Distributions, Hilbert space operators, variational methods, and applications in quantum physics*, volume 69. Birkhäuser, 2015.

[8] Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Advanced Lectures on Machine Learning*, pages 208–240. Springer, 2004.

[9] Stephen Boyd and Almir Mutapcic. Subgradient methods (notes for EE364B Winter 2006-07, Stanford University), 2008.

[10] D Brook. On the distinction between the conditional probability and the joint probability approaches in the specification of nearest-neighbour systems. *Biometrika*, 51(3/4):481–483, 1964.

[11] Valeriĭ Vladimirovich Buldygin and IU V Kozachenko. *Metric characterization of random variables and random processes*, volume 188. American Mathematical Soc., 2000.

[12] Xiaohong Chen and Xiaotong Shen. Sieve extremum estimates for weakly dependent data. *Econometrica*, pages 289–314, 1998.

[13] 1000 Genomes Project Consortium et al. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073, 2010.

[14] Wellcome Trust Case Control Consortium et al. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661, 2007.

[15] Robert R Corbeil and Shayle R Searle. Restricted maximum likelihood (reml) estimation of variance components in the mixed model. *Technometrics*, 18(1):31–38, 1976.

[16] Robert B Davies. Algorithm as 155: The distribution of a linear combination of χ 2 random variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):323–333, 1980.

[17] Robert B Davies. Hypothesis testing when a nuisance parameter is present only under the alternatives. *Biometrika*, 74:33–43, 1987.

[18] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[19] Luca Ferrarini, Walter M Palm, Hans Olofsen, Mark A van Buchem, Johan HC Reiber, and Faiza Admiraal-Behloul. Shape differences of the brain ventricles in alzheimer's disease. *NeuroImage*, 32(3):1060–1069, 2006.

[20] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[21] Kenji Fukumizu. A regularity condition of the information matrix of a multilayer perceptron network. *Neural networks*, 9(5):871–879, 1996.

[22] Kenji Fukumizu et al. Likelihood ratio of unidentifiable models and multilayer neural networks. *The Annals of Statistics*, 31(3):833–851, 2003.

[23] James E Gentle. *Matrix algebra: theory, computations, and applications in statistics*. Springer Science & Business Media, 2007.

[24] David B Goldstein, Andrew Allen, Jonathan Keebler, Elliott H Margulies, Steven Petrou, Slavé Petrovski, and Shamil Sunyaev. Sequencing studies in human genetics: design and interpretation. *Nature Reviews Genetics*, 14(7):460, 2013.

[25] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[26] Ulf Grenander. *Abstract Inference*. Wily, New York, 1981.

[27] Zihuai He, Min Zhang, Xiaowei Zhan, and Qing Lu. Modeling and testing for joint association using a genetic random field model. *Biometrics*, 70(3):471–479, 2014.

[28] Charles R Henderson. Sire evaluation and genetic trends. *Journal of Animal Science*, 1973(Symposium):10–41, 1973.

[29] Fumio Hiai. Monotonicity for entrywise functions of matrices. *Linear Algebra and its Applications*, 431(8):1125–1146, 2009.

[30] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[31] Roger A Horn, Roger A Horn, and Charles R Johnson. *Matrix analysis*. Cambridge university press, 1990.

[32] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2012.

[33] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[34] Tadeusz Inglot. Inequalities for quantiles of the chi-square distribution. *Probability and Mathematical Statistics*, 30(2):339–351, 2010.

[35] Luke Jostins and Jeffrey C Barrett. Genetic risk prediction in complex disease. *Human molecular genetics*, 20(R2):R182–R188, 2011.

[36] K Juottonen, MP Laakso, R Insausti, M Lehtovirta, A Pitkänen, K Partanen, and H Soininen. Volumes of the entorhinal and perirhinal cortices in alzheimer's disease. *Neurobiology of aging*, 19(1):15–22, 1998.

[37] Raghu N Kackar and David A Harville. Unbiasedness of two-stage estimation and prediction procedures for mixed linear models. *Communications in statistics-theory and methods*, 10(13):1249–1261, 1981.

[38] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[39] Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2005.

[40] Bingshan Li and Suzanne M Leal. Methods for detecting associations with rare variants for common diseases: application to analysis of sequence data. *The American Journal of Human Genetics*, 83(3):311–321, 2008.

[41] Ming Li, Zihuai He, Min Zhang, Xiaowei Zhan, Changshuai Wei, Robert C Elston, and Qing Lu. A generalized genetic random field method for the genetic association analysis of sequencing data. *Genetic epidemiology*, 38(3):242–253, 2014.

[42] Dawei Liu, Xihong Lin, and Debashis Ghosh. Semiparametric regression of multidimensional genetic pathway data: Least-squares kernel machines and linear mixed models. *Biometrics*, 63(4):1079–1088, 2007.

[43] Xin Liu and Yongzhao Shao. Asymptotics for likelihood ratio tests under loss of identifiability. *The Annals of Statistics*, 31(3):807–832, 2003.

[44] Bo Eskerod Madsen and Sharon R Browning. A groupwise association test for rare mutations using a weighted sum statistic. *PLoS Genet*, 5(2):e1000384, 2009.

[45] Yuly Makovoz. Random approximants and neural networks. *Journal of Approximation Theory*, 85(1):98–109, 1996.

[46] Jon McClellan and Mary-Claire King. Genetic heterogeneity in human disease. *Cell*, 141(2):210–217, 2010.

[47] Charles E. McCulloch, S. R. Searle, and John M. Neuhaus. *Generalized, linear, and mixed models*. Wiley, Hoboken, N.J, 2nd edition, 2008.

[48] Shahar Mendelson. A few notes on statistical learning theory. In *Advanced lectures on machine learning*, pages 1–40. Springer, 2003.

[49] Stephan Morgenthaler and William G Thilly. A strategy to discover genes that carry multi-allelic or mono-allelic risk for common diseases: a cohort allelic sums test (cast). *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 615(1):28–56, 2007.

[50] NE Morton and CJ MacLean. Analysis of family resemblance. 3. complex segregation of quantitative traits. *American journal of human genetics*, 26(4):489, 1974.

[51] Yangling Mu and Fred H Gage. Adult hippocampal neurogenesis and its role in alzheimer's disease. *Molecular neurodegeneration*, 6(1):1, 2011.

[52] Wei Pan, Junghi Kim, Yiwei Zhang, Xiaotong Shen, and Peng Wei. A powerful and adaptive association test for rare variants. *Genetics*, 197(4):1081–1095, 2014.

[53] Yongjin Park and Manolis Kellis. Deep learning for regulatory genomics. *Nature biotechnology*, 33(8):825, 2015.

[54] Long Qu, Tobias Guennel, and Scott L. Marshall. Linear score tests for variance components in linear mixed models and applications to genetic association studies. *Biometrics*, 69(4):883–892, 2013.

[55] C Radhakrishna Rao. Estimation of heteroscedastic variances in linear models. *Journal of the American Statistical Association*, 65(329):161–172, 1970.

[56] C Radhakrishna Rao. Estimation of variance and covariance components—minque theory. *Journal of multivariate analysis*, 1(3):257–275, 1971.

[57] C Radhakrishna Rao. Estimation of variance and covariance components in linear models. *Journal of the American Statistical Association*, 67(337):112–115, 1972.

[58] Havard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

[59] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

[60] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[61] N Schuff, N Woerner, L Boreta, T Kornfield, LM Shaw, JQ Trojanowski, PM Thompson, CR Jack, MW Weiner, Disease Neuroimaging Initiative, et al. Mri of hippocampal volume loss in early alzheimer's disease in relation to apoe genotype and biomarkers. *Brain*, 132(4):1067–1077, 2009.

[62] Laura J Scott, Karen L Mohlke, Lori L Bonnycastle, Cristen J Willer, Yun Li, William L Duren, Michael R Erdos, Heather M Stringham, Peter S Chines, Anne U Jackson, et al. A genome-wide association study of type 2 diabetes in finns detects multiple susceptibility variants. *science*, 316(5829):1341–1345, 2007.

[63] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[64] Xiaotong Shen. On methods of sieves and penalization. *The Annals of Statistics*, pages 2555–2591, 1997.

[65] Xiaotong Shen and Wing Hung Wong. Convergence rate of sieve estimates. *The Annals of Statistics*, pages 580–615, 1994.

[66] Steven E Stern and Alan H Welsh. Likelihood inference for small variance components. *Canadian Journal of Statistics*, 28(3):517–532, 2000.

[67] Madhav Thambisetty, Andrew Simmons, Abdul Hye, James Campbell, Eric Westman, Yi Zhang, Lars-Olof Wahlund, Anna Kinsey, Mirsada Causevic, Richard Killick, et al. Plasma biomarkers of brain atrophy in alzheimer's disease. *PloS one*, 6(12):e28527, 2011.

[68] Sara van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

[69] Aad W van der Vaart and Jon A Wellner. *Weak convergence and empirical processes*. Springer, 1996.

[70] Vladimir Vapnik. *Statistical learning theory. 1998*, volume 3. Wiley, New York, 1998.

[71] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 6:18962, 2016.

[72] Yalu Wen, Xiaoxi Shen, and Qing Lu. Genetic risk prediction using a spatial autoregressive model with adaptive lasso. *Statistics in medicine*, 37(26):3764–3775, 2018.

[73] Halbert White. Learning in artificial neural networks: A statistical perspective. *Neural computation*, 1(4):425–464, 1989.

[74] Halbert White. Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. *Neural networks*, 3(5):535–549, 1990.

[75] Halbert White and Jeffrey Racine. Statistical inference, the bootstrap, and neural-network modeling with application to foreign exchange rates. *IEEE Transactions on Neural Networks*, 12(4):657–673, 2001.

[76] Halbert White and J Wooldridge. Some results on sieve estimation with dependent observations. In W.A. Barnett, J. Powell, and G. Tauchen, editors, *Nonparametric and Semiparametric Methods in Economics*, pages 459–493. Cambridge University Press New York, 1991.

[77] Chien-Fu Wu. Asymptotic theory of nonlinear least squares estimation. *The Annals of Statistics*, pages 501–513, 1981.

[78] Michael C Wu, Seunggeun Lee, Tianxi Cai, Yun Li, Michael Boehnke, and Xihong Lin. Rare-variant association testing for sequencing data with the sequence kernel association test. *The American Journal of Human Genetics*, 89(1):82–93, 2011.

[79] Jian Yang, S Hong Lee, Michael E Goddard, and Peter M Visscher. Gcta: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82, 2011.

[80] Hongtu Zhu and Heping Zhang. Asymptotics for estimation and testing procedures under loss of identifiability. *Journal of Multivariate Analysis*, 97(1):19–45, 2006.