University of California
Santa Barbara

Nonparametric Learning Methods for Graphical Models

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

Doctor of Philosophy
in
Statistics and Applied Probability

by

Hao Dong

Committee in charge:

Professor Yuedong Wang, Chair
Professor Sang-Yun Oh
Professor Wendy Meiring

June 2022

The Dissertation of Hao Dong is approved.

Professor Sang-Yun Oh

Professor Wendy Meiring

Professor Yuedong Wang, Committee Chair

May 2022

Nonparametric Learning Methods for Graphical Models

by

Hao Dong

To My Family

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Yuedong Wang, for his continued guidance, inspiration in research, and encouragement throughout my Ph.D. study. I am also sincerely thankful for him paying attention to my personal growth and giving me great support and help in my career development. His sage advice on academics and his way of thinking will be a lifetime treasure for me.

I also would like to thank Professor Wendy Meiring and Professor Sang-Yun Oh for being on the committee for my dissertation work and giving me insightful advice and encouragement along the way.

Finally, I want to thank my family for their unconditional love and support. They give me confidence, encourage me to pursue the life I love, and become my eternal backing. I also want to express my thanks to my roommates and friends for their company during my four years in Santa Barbara. They enrich my Ph.D. life and make me a better person.

Curriculum Vitæ

Hao Dong

Education

2022	Ph.D. in Statistics and Applied Probability (Expected), University of California, Santa Barbara.
2018	M.S. in Applied Mathematics, Texas A&M University.
2017	B.S. in Mathematics, Beihang University

Experience

2018 - 2022	Teaching Assistant, Department of Statistics and Applied Probability, University of California, Santa Barbara.
2021	Data Science Intern, Mutual of Omaha.

Abstract

Nonparametric Learning Methods for Graphical Models

by

Hao Dong

Graphical models reveal the conditional dependence structure between random variables. By estimating the joint density or conditional density, we can detect edges and recover the structure of a graphical model. We propose new nonparametric methods to learn edges for graphical models under a consolidated framework of smoothing spline ANOVA (SS ANOVA) decomposition.

We first develop an automatic nonparametric edge detection method by estimating the joint density function with an L_1 penalty to interactions in the SS ANOVA decomposition. In the second project, we work directly on the conditional dependence structure and develop a fully nonparametric neighborhood selection method. We detect edges by applying an L_1 regularization to interactions in the SS ANOVA decomposition of conditional density functions. These two methods are flexible and contain many existing models as special cases. They also provide a unified framework without any restrictions on the type of each random variable. The joint density approach requires a large computer memory and is thus computationally feasible only when the dimension is small. The neighborhood selection approach overcomes this disadvantage and is more computationally efficient.

We propose iterative procedures to compute the estimates and establish the convergence rates for both the joint and conditional density as well as interactions. Simulations indicate that both joint and neighborhood selection methods perform well under Gaussian and non-Gaussian settings. We illustrate the proposed methods using real data

examples.

Contents

Chapter 1

Introduction

1.1 Graphical Models

Discovering the conditional independence among random variables is an important task in statistics. Undirected probabilistic graphical models play a significant role in characterizing conditional independence and have been utilized in a wide range of scientific and engineering domains, including statistical physics, computer vision, machine learning, and computational biology (Koller and Friedman [22]). A graphical model is constructed based on an undirected graph $G = (V, E)$ with node set $V = \{1, \cdots, p\}$ representing p random variables $X_1, \cdots, X_p$ and edge set $E \subseteq V \times V$ describing the conditional independence among $X_1, \cdots, X_p$. Denote $\mathcal{X}_j$ as the domain of X_j, and $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$. Let $\boldsymbol{X} = (X_1, \cdots, X_p)$ and $\boldsymbol{X}_{\setminus\{i_1,\cdots,i_k\}}$ be the sub-vector of $\boldsymbol{X}$ without elements $\{i_1, \cdots, i_k\}$. Then, $\{i, j\} \notin E$ corresponds to the conditional independence between X_i and X_j given other variables in $\boldsymbol{X}$, which is denoted as $X_i \perp X_j | \boldsymbol{X}_{\setminus\{i,j\}}$.

Many parametric and semi-parametric graphical models have been studied in the literature with assumptions on the joint density or conditional density. We will review some existing joint density and neighborhood selection approaches in Section 1.2 and

Section 1.3, respectively. We review the nonparametric SS ANOVA model for joint density and conditional density estimation in Section 1.4 and Section 1.5, respectively. In Section 1.6, we review two SS ANOVA model-based edge detection methods.

1.2 Joint Density Approach for Graphical Model Estimation

As joint density ultimately determines the conditional relationship, methods for edge detection based on estimating the joint density have been proposed. The key idea of using the joint density approach for edge detection is to represent the joint distribution as a product of clique-wise compatibility functions. For any given graph, each of these compatibility functions depends only on a subset of variables within any clique of the underlying graph. Let $\mathcal{C}$ be a set of cliques (fully-connected subgraphs) of the graph G and let $\{\phi_c(\boldsymbol{X}_c)\}_{c\in\mathcal{C}}$ be a set of clique-wise basis functions and $\boldsymbol{X}_c$ contains all variables in the clique c. By the Hammersley-Clifford theorem (Dobruschin [7]), the joint density function represented by the graph G takes the form:

$$f(\boldsymbol{x}) \propto \exp\left\{\sum_{c\in\mathcal{C}} \theta_c \phi_c(\boldsymbol{x}_c)\right\},$$

where $\{\theta_c\}$ are parameters over the basis functions. In this dissertation, we will consider an important special case, pairwise graphical models, where all cliques have size no more than two. For a pairwise graphical model, the joint density distribution has the form of

$$f(\boldsymbol{x}) \propto \exp\left\{\sum_{j=1}^{p} \theta_j \phi_j(x_j) + \sum_{1\le j<k\le p} \theta_{jk}\phi_{jk}(x_j, x_k)\right\}. \tag{1.1}$$

Then $X_j \perp X_k | \boldsymbol{X}_{\setminus\{j,k\}}$ if and only if $\theta_{jk} = 0$. We want to perform edge detection to reveal conditional dependence structure and discover all pairwise cliques. We note that the proposed methods in this dissertation may be extended to more general density (graphical) models with high-order interactions. There exists a close connection between density estimation and graphical model estimation. The graph is completely decided by the joint density function. On the other hand, the graph structure can be used to simplify the density estimation. Therefore, learning structures in the density and graph are two sides of the same coin.

In parametric graphical models, ϕ_j and ϕ_{jk} are known up to some parameters. The functional form of ϕ_j and ϕ_{jk} are either decided by specific distributions or expressed using well-chosen basis functions.

Univariate exponential family distributions are used to model different types of data including skewed continuous data and count data. Some special multivariate pairwise graphical model distributions have been derived from univariate exponential family distributions, such as Normal, Poisson, and exponential distributions. The assumption is that the distribution of each variable conditioned on the other variables has an exponential family form (Suggala et al. [37]), which leads to the following joint density

$$f(\boldsymbol{x}) = \exp\left\{\sum_{j=1}^{p} \theta_j B(x_j) + \sum_{1\le j<k\le p} \theta_{jk} B(x_j)B(x_k) + \sum_{j=1}^{p} C(x_j) - A(\boldsymbol{\theta})\right\}, \tag{1.2}$$

where $B(\cdot)$ is a basis function, $C(\cdot)$ is a base measure and $A(\boldsymbol{\theta}) < \infty$ is the log-partition function defined as

$$A(\boldsymbol{\theta}) := \log \int_{\mathcal{X}} \exp\left\{\sum_{j=1}^{p} \theta_j B(x_j) + \sum_{1\le j<k\le p} \theta_{jk} B(x_j)B(x_k) + \sum_{j=1}^{p} C(x_j)\right\} d\boldsymbol{x}.$$

Since $B(\cdot)$ and $C(\cdot)$ are known, the exponential family graphical models are parametric pairwise graphical models with multiplicative interactions. The conditional independence $X_j \perp X_k | \boldsymbol{X}_{\setminus\{j,k\}}$ holds if and only if $\theta_{jk} = 0$. We now list some examples of multivariate exponential graphical model distributions with linear functions $B(x_j) = x_j$. More details can be found in Yang et al. [44].

The popular Gaussian graphical model can be derived from univariate Gaussian distribution with basis function $B(x_j) = \frac{x_j}{\sigma_j}$ and base measure $C(x_j) = -\frac{x_j^2}{2\sigma_j^2}$. Some well-developed methods for the Gaussian graphical model are to estimate the precision matrix (Banerjee et al. [4], Friedman et al. [10], Yuan and Lin [47]). The jkth element in the precision matrix equals zero if and only if $\theta_{jk} = 0$. The Ising model can be derived from the Bernoulli distribution, where $B(x_j) = x_j$ and base measure $C(x_i) = 0$ with variables taking values in the set $\mathcal{X}_j = \{0, 1\}$ for $j = 1, \cdots, p$. Poisson graphical models have the Poisson distribution as the univariate exponential family distribution with $B(x_j) = x_j$ and $C(x_j) = -\log(x_j!)$ with variables taking values in the set $\mathcal{X}_j = \{0, 1, 2, \cdots\}$ for $j = 1, \cdots, p$. Exponential graphical model distribution has $B(x_j) = -x_j$ and $C(x_j) = 0$ with variables taking values in $\mathcal{X}_j = [0, \infty)$.

As extensions of the exponential family, more flexible pairwise graphical models have been studied in the literature. Yuan et al. [48] proposed a way to model ϕ_j and ϕ_{jk} parametrically using basis functions. They assume the formulations of ϕ_j and ϕ_{jk} are unknown but admit linear representations over two sets of pre-fixed basis functions $\{\varphi_t(\cdot), t = 1, 2, \cdots, s\}$ and $\{\psi_l(\cdot, \cdot), l = 1, 2, \cdots, r\}$ respectively, that is

$$\phi_j(x_j) = \sum_{t=1}^{s} \theta_{j,t} \varphi_t(x_j), \quad \phi_{jk}(x_j, x_k) = \sum_{l=1}^{r} \theta_{jk,l} \psi_l(x_j, x_k),$$

where s and r are the truncation order parameters. In this formulation, the choice of basis and their sizes is flexible and task-dependent.

Suggala et al. [37] assumed the distribution of each variable conditioned on the other variables has a non-parametric exponential family form which leads to a consistent joint density of (1.1) with $\phi_j(x_j) = \theta_j B_j(x_j)$ and $\phi_{jk}(x_j, x_k) = \theta_{j,k} B_j(x_j) B_k(x_k)$. For estimation, $B_j(\cdot)$ is over a uniformly bounded, orthonormal basis $\{\varphi_l(\cdot)\}_{l=0}^{\infty}$ with $\varphi_0(\cdot) \propto 1$:

$$B_j(x_j) = \sum_{l=1}^{m} \alpha_{j,l}\varphi_l(x_j) + \rho_{j,m}(x_j) \quad \text{where} \quad \rho_{j,m}(x_j) = \alpha_{j,0}\varphi_0(x_j) + \sum_{l=m+1}^{\infty} \alpha_{j,l}\varphi_l(x_j).$$

Suggala et al. [37] imposes additional constraints on its parameters and require $B_j(x_j)$ to satisfy $\int_{\mathcal{X}_j} B_j(x)dx = 0$, which is equivalent to $\alpha_{j,0} = 0$. To convert the infinite dimensional optimization problem into a finite dimensional problem, they truncate $B_j(x_j)$ to the first m terms and drop the remainder term $\rho_{j,m}(x_j)$.

Yang et al. [45] proposed a general semiparametric model with unspecified base measure functions for each node. They modeled ϕ_j nonparametrically and ϕ_{jk} parametrically using $\eta_{jk} = \theta_{jk} x_j x_k$. The joint probability distribution has the density in the form of

$$f(\boldsymbol{x}) = \exp\left\{\sum_{j=1}^{p} \eta_j(x_j) + \sum_{1 \le j < k \le p} \theta_{jk} x_j x_k - A(\boldsymbol{\theta}, \boldsymbol{\eta})\right\},$$

where $\boldsymbol{\eta} = (\eta_1, ..., \eta_p)$ and $\eta_j(\cdot)$ is an unknown base measure function and $A(\cdot)$ is still the log-partition function given by

$$A(\boldsymbol{\theta}, \boldsymbol{\eta}) := \log\left\{\int_{\chi} \exp[\sum_{j=1}^{p} \eta_j(x_j) + \sum_{1 \le j < k \le p} \theta_{jk} x_j x_k] d\boldsymbol{x}\right\}.$$

For edge detection, Yang et al. [45] treat θ_{jk} as the parameter of interest and the base functions $\eta_j(\cdot)$ as nuisance parameter. They exploited a pseudo-likelihood loss function to eliminate the presence of $\eta_j(\cdot)$ and estimate θ_{jk}.

1.3 Neighborhood Selection Approach

The neighborhood selection approach is usually more computationally efficient by working on conditional densities instead of the joint density. By the conditional independence properties of undirected graphical models, for any node $\alpha \in V$, X_α only depends on other variables in its neighborhood set $nb_G(\alpha)$, where $nb_G(\alpha) = \{k \in V | \{\alpha, k\} \in E\}$. Consequently, the conditional independence structure of graph G can be constructed by estimating all of its neighborhoods $nb_G(\alpha)$ for $\alpha = 1, \cdots, p$. The goal of neighborhood selection is to determine a minimal set of variables in $nb_G(\alpha)$ that X_α depends on for each node $\alpha \in V$.

Many neighborhood selection methods have been developed based on the conditional likelihood or pseudo-likelihood for learning sparse graphical models (Hastie et al. [18], Drton and Maathuis [8]). Flexible models were proposed for discrete data (Höfling and Tibshirani [19], Ravikumar et al. [33]). For the continuous type, methods are usually based on modeling the conditional mean (Meinshausen and Bühlmann [30], Voorman et al. [40]) or conditional quantiles (Ali et al. [2]). For example, Meinshausen and Bühlmann [30] considered a linear model for the conditional mean while Voorman et al. [40] considered an additive model for the conditional mean. It is worth noting that the conditional mean approach seems distribution-free since no specific distributional assumption is made for the regression errors. However, the joint distribution must be multivariate Gaussian under mild assumptions if the conditional relationships are linear (Voorman et al. [40]). In other words, the restriction of Gaussianity has not been removed as it appears. For the mixed type of data, Lee and Hastie [26] and Cheng et al. [6] both proposed to fit a conditional Gaussian model for continuous variables. Lee and Hastie [26] considered discrete variables and each conditional distribution given the rest is multinomial. Then, a regularized multi-class logistic regression problem was optimized. Cheng et al. [6] con-

sidered binary variables and logistic regression models are fitted for each binary variable given all other variables. For discrete responses and a covariate in a generic domain, Gu and Ma [14] proposed to estimate the conditional density nonparametrically in a functional ANOVA decomposition way and use the Kullback-Leibler projection to identify the conditional independence structures. However, analyzing the mixed types of data is still very challenging. Most well-developed methods are parametric or semi-parametric and Gaussian assumption is used for parameter estimation. Gu and Ma [14] introduced a nonparametric method but the dimension of continuous variables is just one in both simulation and real data. Therefore, there is still no nonparametric neighborhood selection method for high-dimensional continuous variables to the best of our knowledge.

1.4 SS ANOVA Models for Joint Density

SS ANOVA models are extensions of the classical ANOVA models from discrete domains to general domains. It decomposes the logistic transformation of a joint density function into a summation of main effects and interactions. Let $f(\boldsymbol{x})$ be the joint density function of $\boldsymbol{X}$, and consider the transformation $f(\boldsymbol{x}) = e^{\eta(\boldsymbol{x})}/\int e^{\eta(\boldsymbol{x})}d\boldsymbol{x}$ to enforce the conditions of $f > 0$ and $\int f = 1$. The function $\eta(\boldsymbol{x})$ is referred as the logistic transformation of f. The function $\eta(\boldsymbol{x})$ can be decomposed as a summation of a constant term, main effects and interactions:

$$\eta(x_1, \cdots, x_p) = c + \sum_{j=1}^{p} \eta_j(x_j) + \sum_{1\le j<k\le p} \eta_{jk}(x_j, x_k) + \cdots + \eta_{1\cdots p}(x_1, \cdots, x_p). \qquad (1.3)$$

The identifiability of the terms in (1.3) is ensured by side conditions through averaging operators (Gu [12], Wang [41]).

Let $\mathcal{H}^{(j)}$ be a reproducing kernel Hilbert space (RKHS) on $\mathcal{X}_j$ and $\mathcal{H}^{(j)} = \{1_{(j)}\} \oplus$

$\mathcal{H}_{(j)}$, where $\{1_{(j)}\}$ is the space of constant functions on $\mathcal{X}_j$ and $\mathcal{H}_{(j)}$ is its orthogonal complement. The decomposition in equation (1.3) for the logistic transformation of the joint density corresponds to the following SS ANOVA decomposition of the tensor product space $\mathcal{H}$ on $\mathcal{X}$

$$\begin{aligned}\mathcal{H} &= \bigotimes_{j=1}^{p} \mathcal{H}^{(j)} = \bigotimes_{j=1}^{p} \left\{\{1_{(j)}\} \oplus \mathcal{H}_{(j)}\right\} \\ &= \{1\} \oplus \left\{\bigoplus_{j=1}^{p} \mathcal{H}_{(j)}\right\} \oplus \left\{\bigoplus_{1\le j<k\le p} [\mathcal{H}_{(j)} \otimes \mathcal{H}_{(k)}]\right\} \oplus \cdots \oplus \left\{\mathcal{H}_{(1)} \otimes \cdots \otimes \mathcal{H}_{(p)}\right\}. \quad (1.4)\end{aligned}$$

The expansion in (1.3) is usually truncated in some manner to overcome the curse of dimensionality. A common and simple truncated model is a pairwise model:

$$\eta(x_1, \cdots, x_p) = c + \sum_{j=1}^{p} \eta_j(x_j) + \sum_{1\le j<k\le p} \eta_{jk}(x_j, x_k), \quad (1.5)$$

where interactions of order higher than 2 are removed.

For an SS ANOVA model in (1.3), Gu and Qiu [15] proposed to estimate η via minimizing the penalized likelihood

$$-\frac{1}{n}\sum_{i=1}^{n} \eta(\boldsymbol{x}_i) + \log \int_{\mathcal{X}} e^{\eta(\boldsymbol{x})} d\boldsymbol{x} + \frac{\lambda}{2} J(\eta), \quad (1.6)$$

where the first two terms correspond to the negative logarithm of the likelihood function, and $J(\eta)$ is a quadratic roughness functional, and the smoothing parameters λ controls the trade-off between the smoothness of η and its fidelity to the data. The computation of this minimization problem with cross-validated λ has been studied in Gu and Qiu [15] and Gu and Wang [16].

This penalized likelihood method is infeasible when p is large since it needs to compute the multivariate integral $\int_{\mathcal{X}} e^{\eta(\boldsymbol{x})} d\boldsymbol{x}$. To avoid calculating the integration of high

dimensional functions, Jeon and Lin [21] proposed a penalized pseudo-likelihood method by estimating η as the minimizer of

$$\frac{1}{n}\sum_{i=1}^{n} e^{-\eta(\boldsymbol{x}_i)} + \int_{\chi} \eta(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} + \frac{\lambda}{2}J(\eta), \tag{1.7}$$

where $\rho(\boldsymbol{x})$ is some known density. The resulting estimate can be calculated as $\hat{f}(\boldsymbol{x}) \propto e^{\hat{\eta}(\boldsymbol{x})}\rho(\boldsymbol{x})$ where $\hat{\eta}(\boldsymbol{x})$ is the minimizer from penalized pseudo-likelihood method. With a proper selection of $\rho(\boldsymbol{x})$, the integral $\int_{\mathcal{X}} \eta(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}$ can be decomposed into products of univariate integrals. Specifically, one may choose $\rho(\boldsymbol{x}) = \prod_{j=1}^{p} \rho_j(x_j)$ as the product of marginal density estimates which can be modeled parametrically or nonparametrically. Suppose that $J(\eta)$ annihilates constant and the RKHS $\mathcal{H}$ for η can be decomposed into $\mathcal{H} = \{1\} \oplus \mathcal{G}$, where $\{1\}$ is the constant space, and $\mathcal{G}$ is its orthogonal complement. Then one can write $\eta = l + g$ with $l \in \{1\}$ and $g \in \mathcal{G}$. The penalized pseudo-likelihood (1.7) becomes

$$\frac{1}{n}\sum_{i=1}^{n} e^{-g(\boldsymbol{x}_i)-l} + \int_{\mathcal{X}} (g(\boldsymbol{x}) + l)\rho(\boldsymbol{x})d\boldsymbol{x} + \frac{\lambda}{2}J(g).$$

One can take derivative with respect to l and set it to be zero. Then $e^l = \frac{1}{n}\sum_{i=1}^{n} e^{-g(\boldsymbol{x}_i)}$. Plugging this term back and drop terms not involving g, the profile penalized pseudo-likelihood can be written as:

$$\log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-g(\boldsymbol{x}_i)}\right\} + \int_{\chi} g(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} + \frac{\lambda}{2}J(g). \tag{1.8}$$

The solution to (1.7) do not fall in finite dimensional spaces. Gu [12] proposed to approximate the solution in a space

$$\mathcal{H}^* = \mathcal{N}_J \oplus \operatorname{span}\left\{R_J(\boldsymbol{u}_j, \cdot), j = 1, \cdots, q\right\}, \tag{1.9}$$

where $\mathcal{N}_J = \{f : J(f) = 0\}$, and $\{\boldsymbol{u}_j\}$ is a random subset of $\{\boldsymbol{x}_i\}$. Denote $\phi_1, \cdots, \phi_m$ as a basis function of $\mathcal{N}_J \ominus \{1\}$, and $\xi_j(\cdot) = R_J(\boldsymbol{u}_j, \cdot)$ for $j = 1, \cdots, q$. Then, an approximate estimate of g can be expressed as

$$g(\boldsymbol{x}) = \sum_{v=1}^{m} d_v \phi_v(\boldsymbol{x}) + \sum_{j=1}^{q} c_j R_J(\boldsymbol{u}_j, \boldsymbol{x}) = \boldsymbol{\phi}^T \boldsymbol{d} + \boldsymbol{\xi}^T \boldsymbol{c}. \tag{1.10}$$

Plugging (1.10) into (1.8), one can solve coefficients $\boldsymbol{c}, \boldsymbol{d}$ using Newton method and select the tuning parameter λ. More details can be found in Chapter 10.1 in Gu [12].

1.5 SS ANOVA Models for Conditional Density

We are interested in estimating the conditional density $f(x_\alpha | \boldsymbol{x}_{\setminus\{\alpha\}})$ for αth variable given $\boldsymbol{x}_{\setminus\{\alpha\}} = (x_1, \cdots, x_{\alpha-1}, x_{\alpha+1}, \cdots, x_p)$, and consider the logistic density transformation of f as

$$f(x_\alpha | \boldsymbol{x}_{\setminus\{\alpha\}}) = \frac{e^{\eta(\boldsymbol{x})}}{\int_{\mathcal{X}_\alpha} e^{\eta(\boldsymbol{x})} dx_\alpha}. \tag{1.11}$$

An SS ANOVA model for η in (1.11) may contain any subset of components in the SS ANOVA decomposition (1.4). For simplicity, we will consider a model with main effects and two-way interactions only. We note that the SS ANOVA model (1.3) with two-way interaction only is a pairwise graphical model which is commonly assumed in the existing literature. Our methods developed in Chapter 6 can be easily extended to include higher order interactions. Denote the model space for η as

$$\mathcal{M}_\alpha = \{1\} \oplus \left\{ \bigoplus_{j=1}^{p} \mathcal{H}_{(j)} \right\} \oplus \left\{ \bigoplus_{k \neq \alpha} [\mathcal{H}_{(\alpha)} \otimes \mathcal{H}_{(k)}] \right\}. \tag{1.12}$$

A function $\eta \in \mathcal{M}_\alpha$ can be decomposed as follows:

$$\eta(\boldsymbol{x}) = \varsigma + \sum_{j=1}^{p} \eta_j(x_j) + \sum_{k \neq \alpha} \eta_{\alpha k}(x_\alpha, x_k), \tag{1.13}$$

where each functional component in (1.13) belongs to the corresponding subspace in (1.12). Two ways to estimate η were proposed in Gu [12]: one is penalized likelihood estimation, and the other one is penalized pseudo-likelihood estimation.

Denote $\boldsymbol{X}_i = (X_{i,1}, \cdots, X_{i,p})$ and $\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,p})$ for $i = 1, \cdots, n$ as n i.i.d. random vectors and their realizations. Let $\boldsymbol{x}_{\setminus\{\alpha\}} = (x_1, \cdots, x_{\alpha-1}, x_{\alpha+1}, \cdots, x_p)$, $\boldsymbol{x}_{i,\setminus\{\alpha\}} = (x_{i,1}, \cdots, x_{i,\alpha-1}, x_{i,\alpha+1}, \cdots, x_{i,p})$ be the ith realization of $\boldsymbol{x}_{\setminus\{\alpha\}}$ and $\boldsymbol{x}_i^\alpha = (\boldsymbol{x}_{i,\setminus\{\alpha\}}, x_\alpha) = (x_{i,1}, \cdots, x_{i,\alpha-1}, x_\alpha, x_{i,\alpha+1}, \cdots, x_{i,p})$, where x_α is still a variable. The penalized likelihood estimation is to minimize

$$-\frac{1}{n} \sum_{i=1}^{n} \left\{ \eta(\boldsymbol{x}_i) - \log \int_{\mathcal{X}_\alpha} e^{\eta(\boldsymbol{x}_i^\alpha)} dx_\alpha \right\} + \frac{\lambda}{2} J(\eta), \tag{1.14}$$

where the first two terms are the negative logarithm of the conditional density function, and $J(\eta)$ is a quadratic roughness functional. The computation of this minimization problem with cross-validated λ has been stuided in Gu [12]. However, the calculation of the integral $\int_{\mathcal{X}_\alpha} e^{\eta(\boldsymbol{x}_i^\alpha)} dx_\alpha$ could be computationally intensive. Penalized pseudo-likelihood estimation is developed in Gu [12] to avoid repeated numerical integrations and gain numerical efficiency.

For each node $\alpha \in V$, we assume that $\eta(\boldsymbol{x}) \in \mathcal{M}_\alpha$ where $\mathcal{M}_\alpha$ is given in (1.12). For an SS ANOVA model in (1.13), the penalized pseudo-likelihood estimation approach is to minimize

$$\frac{1}{n} \sum_{i=1}^{n} e^{-\eta(\boldsymbol{x}_i)} + \int_{\mathcal{X}_\alpha} \eta(\boldsymbol{x}_i^\alpha) \rho(\boldsymbol{x}_i^\alpha) dx_\alpha + \frac{\lambda}{2} J(\eta), \tag{1.15}$$

where $\rho(\cdot)$ is a known density of X_α conditional on $\boldsymbol{X}_{\setminus\{\alpha\}} = \boldsymbol{x}_{i,\setminus\{\alpha\}}$. A simple choice of ρ is $e^{\eta_\alpha(x_\alpha)}/\int_{\mathcal{X}_\alpha} e^{\eta_\alpha(x_\alpha)}dx_\alpha$, an estimate of the marginal density on $\mathcal{X}_\alpha$. Alternatively, we can choose ρ as

$$\rho(x_\alpha, \boldsymbol{x}_{\setminus\{\alpha\}}) = \frac{\phi((x_\alpha - \mu(\boldsymbol{x}_{\setminus\{\alpha\}}))/\sigma)}{\Phi((1-\mu(\boldsymbol{x}_{\setminus\{\alpha\}}))/\sigma) - \Phi((-\mu(\boldsymbol{x}_{\setminus\{\alpha\}}))/\sigma)}, \tag{1.16}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the standard normal density and CDF, and $\mu(\cdot)$ and σ are estimated by fitting a nonparametric regression model in model space (1.12) with covariates $\boldsymbol{x}_{\setminus\{\alpha\}}$. More estimation details can be found in Chapter 3 of Gu [12].

Similar to (1.8) in Section 1.4, by supposing that $J(\eta)$ annihilates constant and rewriting η in (1.13) as $\eta(\boldsymbol{x}) = \varsigma + g(\boldsymbol{x})$ where $g(\boldsymbol{x}) = \sum_{j=1}^{p} g_j(x_j) + \sum_{k\neq\alpha} g_{\alpha k}(x_\alpha, x_k) \in \mathcal{M}_\alpha \ominus \{1\}$, $g_j = \eta_j$, and $g_{\alpha k} = \eta_{\alpha k}$, then (1.15) becomes

$$\frac{1}{n}\sum_{i=1}^{n}\left\{e^{-g(\boldsymbol{x}_i)-\varsigma} + \int_{\mathcal{X}_\alpha}(g(\boldsymbol{x}_i^\alpha)+\varsigma)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha\right\} + \frac{\lambda}{2}J(g). \tag{1.17}$$

Setting the derivative of (1.17) with respect to ς to zero, we get $e^\varsigma = n^{-1}\sum_{i=1}^{n} e^{-g(\boldsymbol{x}_i)}$. Plugging back to (1.17), we have the profile penalized pseudo-likelihood:

$$\log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-g(\boldsymbol{x}_i)}\right\} + \frac{1}{n}\sum_{i=1}^{n}\int_{\mathcal{X}_\alpha} g(\boldsymbol{x}_i^\alpha)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + \frac{\lambda}{2}J(g). \tag{1.18}$$

Similar to Section 1.4, an approximate estimate of g can be expressed as

$$g(\boldsymbol{x}) = \sum_{v=1}^{m} d_v\phi_v(\boldsymbol{x}) + \sum_{j=1}^{q} c_j R_J(\boldsymbol{u}_j, \boldsymbol{x}) = \boldsymbol{\phi}^T\boldsymbol{d} + \boldsymbol{\xi}^T\boldsymbol{c}. \tag{1.19}$$

Plugging (1.19) into (1.18), one can solve coefficients $\boldsymbol{c}, \boldsymbol{d}$ using Newton method and select the tuning parameter λ. More details can be found in Chapter 10.3 in Gu [12].

1.6 Edge Detection via SS ANOVA Models

1.6.1 L_1 Penalty to Both Main Effects and Interactions

SS ANOVA decomposition provides a consolidated framework for edge detection. Jeon and Lin [21] relabelled the subspaces in (1.4), and denoted it as $\mathcal{H} = \{1\} \oplus \{\bigoplus_{\alpha=1}^{m} \mathcal{G}^{(\alpha)}\}$. For edge detection, they considered $J(g)$ in (1.8) as the sum of functional component norms or L_1 penalty to encourage sparsity, instead of sum of squared norms employed in (1.6). Specifically, they set $J(g) = \sum_{\alpha=1}^{m} ||P^{\alpha} g||$, where P^{α} projects g onto $\mathcal{G}^{(\alpha)}$ for $\alpha = 1, \cdots, m$. In the pairwise graphical model, there are p main effect spaces and $p(p-1)/2$ two-way interaction spaces, thus $m = p(p+1)/2$. They considered an equivalent form

$$\log \left\{ \frac{1}{n} \sum_{i=1}^{n} e^{-g(\boldsymbol{x}_i)} \right\} + \int_{\mathcal{X}} g(\boldsymbol{x}) \rho(\boldsymbol{x}) d\boldsymbol{x} + \lambda_0 \sum_{\alpha=1}^{m} \theta_{\alpha}^{-1} ||P^{\alpha} g||^2 + \lambda \sum_{\alpha=1}^{m} \theta_{\alpha}, \tag{1.20}$$

subject to $\theta_{\alpha} \geq 0$ and $\sum_{\alpha=1}^{m} \theta_{\alpha} \leq M$ for some constant M.

For fixed λ, θ_{α}'s, Jeon and Lin solved (1.20) as a smoothing spline problem and approximated the solution with form $g(\boldsymbol{x}) = \sum_{i=1}^{n} c_i R_{\theta}(\boldsymbol{x}_i, \boldsymbol{x}) = \sum_{\alpha=1}^{m} \theta_{\alpha} \sum_{i=1}^{n} c_i R_{\alpha}(\boldsymbol{x}_i, \boldsymbol{x})$, where R_{α} is the RK in each $\mathcal{G}^{(\alpha)}$, and R_{θ} is the RK in $\bigoplus_{\alpha=1}^{m} \mathcal{G}^{(\alpha)}$. The Newton-Raphson iteration was applied to solve for c_i's. For fixed $\boldsymbol{c} = (c_1, \cdots, c_n)^T$, the iteration for updating $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_m)^T$ is via solving a quadratic programming subject to $\theta_{\alpha} \geq 0$ and $\sum_{\alpha=1}^{m} \theta_{\alpha} \leq M$.

The L_1 penalty $\sum_{\alpha=1}^{m} ||P^{\alpha} g||$ penalizes both main effects and interactions. Consequently, it selects both nodes and edges. For graphical models, the nodes are usually given and the goal is to detect edges. The L_1 penalty on main effects may cause undesired effects on edge selection.

1.6.2 Squared Error Projection

Gu et al. [13] proposed the squared error projection approach to assess the practical significance of interaction terms. This approach was based on the Kullback-Leibler geometry and was developed in Gu [11]. Let $\mathcal{H} = \mathcal{H}^0 \oplus \mathcal{H}^1$, where $\mathcal{H}^1$ is a functional space for which the practical significance is to be assessed. Consider the functional

$$\tilde{V}(f,g) = \int_{\mathcal{X}} f(\boldsymbol{x})g(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} - \left\{\int_{\mathcal{X}} f(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\right\}\left\{\int_{\mathcal{X}} g(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\right\}$$

and denote $\tilde{V}(f)$ for $\tilde{V}(f,f)$. One has

$$\tilde{V}(\hat{g}-g) = \int_{\mathcal{X}} (\hat{g}-g)^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} - \left\{\int_{\mathcal{X}} (\hat{g}-g)(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\right\}^2 \tag{1.21}$$

for $\hat{g} \in \mathcal{H}^0 \oplus \mathcal{H}^1$. $\tilde{V}(\hat{g}-g)$ can be treated as a proxy of the symmetrized Kullback-Leibler distance $\mathrm{KL}(\hat{g},g) + \mathrm{KL}(g,\hat{g})$. Then the squared error projection of $\hat{g}$ in $\mathcal{H}^0$ is defined as

$$\tilde{g} = \arg\min_{g\in\mathcal{H}^0} \left\{\tilde{V}(\hat{g}-g)\right\}.$$

Gu et al. [13] introduced a functional $A_{\tilde{g},h}(a) = \tilde{V}(\hat{g} - (\tilde{g} + ah))$ for $h \in \mathcal{H}^0$. Since the derivative of $A_{\tilde{g},h}(a)$ with respect to a evaluating at $a = 0$ equals to zero, they have $\tilde{V}(\hat{g}-\tilde{g},h) = 0$, $\forall h \in \mathcal{H}^0$. Let $g_u = -\log\rho(\boldsymbol{x})$. When $g_u \in \mathcal{H}^0$, $\tilde{V}(\hat{g}-\tilde{g},\tilde{g}-g_u) = 0$, so $\tilde{V}(\hat{g}-g_u) = \tilde{V}(\hat{g}-\tilde{g}) + \tilde{V}(\tilde{g}-g_u)$. Gu et al. [13] proposed to cut out the subspace $\mathcal{H}^1$ when the ratio $\tilde{V}(\hat{g}-\tilde{g})/\tilde{V}(\hat{g}-g_u)$ is small, say $2\% - 3\%$. The choice of the cut-off percentage is ad hoc and there is no overall criterion.

1.7 Dissertation Outline

In this dissertation, we propose two nonparametric methods for edge detection using the SS ANOVA framework. Part 1 presents the first proposed method via joint density estimation with L_1 penalty. In Chapter 2, we propose the joint density method for edge detection, which is a modification of the method in Section 1.6.1. We also present estimation and computation methods in this chapter. Chapter 3 provides the theoretical analysis of convergence rates for both joint density estimate and interactions. Chapter 4 and 5 present the simulation results and two real data applications.

The second method is the neighborhood selection approach through L_1 penalty, which is covered in Part 2. Chapter 6 develops this method and its computational algorithm. We give the convergence rates for both conditional density estimate and its interactions in Chapter 7. Chapters 8 and 9 show the simulation results and the estimated graphs on two real data sets.

In Part 3, we first introduce an `R` package named `edgeSelection` for our two method in Chapter 10. Chapter 11 provides the conclusions of two methods.

Part 1

Joint Density Approach

Chapter 2

Edge Detection via SS ANOVA Model Selection

2.1 Edge Detection Through L_1 Penalty

In this section, we introduce our joint density method with an L_1 penalty on interactions. Let $\boldsymbol{X} = (X_1, \cdots, X_p)$, $\mathcal{X}_j$ be the domain of X_j, and $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$. The domains $\mathcal{X}_j$ are arbitrary sets. Therefore, the proposed method can deal with mixture of data. Let $f(\boldsymbol{x})$ be the joint density function of $\boldsymbol{X}$, and consider the transformation $f(\boldsymbol{x}) = e^{g(\boldsymbol{x})} / \int e^{g(\boldsymbol{x})} d\boldsymbol{x}$ to enforce the conditions of $f > 0$ and $\int f = 1$. The function $g(\boldsymbol{x})$ can be decomposed as a summation of a constant term, main effects and interactions:

$$g(x_1, \cdots, x_p) = c + \sum_{j=1}^{p} g_j(x_j) + \sum_{1 \le j < k \le p} g_{jk}(x_j, x_k) + \cdots + g_{1 \cdots p}(x_1, \cdots, x_p). \qquad (2.1)$$

The decomposition in equation (2.1) for the logistic transformation of the joint density corresponds to the following SS ANOVA decomposition of the tensor product space $\mathcal{H}$

on $\mathcal{X}$

$$\mathcal{H} = \{1\} \oplus \left\{ \bigoplus_{j=1}^{p} \mathcal{H}_{(j)} \right\} \oplus \left\{ \bigoplus_{1 \le j < k \le p} [\mathcal{H}_{(j)} \otimes \mathcal{H}_{(k)}] \right\} \oplus \cdots \oplus \left\{ \mathcal{H}_{(1)} \otimes \cdots \otimes \mathcal{H}_{(p)} \right\}. \quad (2.2)$$

The expansion in (2.1) is usually truncated in some manner to overcome the curse of dimensionality. A pairwise model is used for our joint density method:

$$g(x_1, \cdots, x_p) = \sum_{j=1}^{p} g_j(x_j) + \sum_{1 \le j < k \le p} g_{jk}(x_j, x_k), \quad (2.3)$$

where interactions of order higher than 2 are removed and the constant is also removed for identifiability.

$\mathcal{H}_{(j)}$ is an RKHS of functions on $\mathcal{X}_j$ of the form $\mathcal{H}_{(j)} = \mathcal{H}^0_{(j)} \oplus \mathcal{H}^1_{(j)}$, where $\mathcal{H}^0_{(j)} = \text{span}\{\phi_{j1}, \cdots, \phi_{jm_j}\}$, and $\mathcal{H}^1_{(j)}$ is the orthogonal complement of $\mathcal{H}^0_{(j)}$ with RK R_j. Then,

$$\begin{aligned} \mathcal{H}_{(jk)} &:= \mathcal{H}_{(j)} \otimes \mathcal{H}_{(k)} = (\mathcal{H}^0_{(j)} \oplus \mathcal{H}^1_{(j)}) \otimes (\mathcal{H}^0_{(k)} \oplus \mathcal{H}^1_{(k)}) \\ &= (\mathcal{H}^0_{(j)} \otimes \mathcal{H}^0_{(k)}) \oplus (\mathcal{H}^0_{(j)} \otimes \mathcal{H}^1_{(k)}) \oplus (\mathcal{H}^1_{(j)} \otimes \mathcal{H}^0_{(k)}) \oplus (\mathcal{H}^1_{(j)} \otimes \mathcal{H}^1_{(k)}) = \mathcal{H}^0_{(jk)} \oplus \mathcal{H}^1_{(jk)}, \end{aligned}$$

where $\mathcal{H}^0_{(jk)} = \mathcal{H}^{(j)}_0 \otimes \mathcal{H}^{(k)}_0 = \text{span}\{\psi_{jk1}, \cdots, \psi_{jkm_{jk}}\}$, $\psi_{jk(m_k(u-1)+v)} = \phi_{ju}\phi_{kv}$, for $u = 1, ..., m_j$, $v = 1, ..., m_k$, and $m_{jk} = m_j m_k$, $\mathcal{H}^1_{(jk)} = \mathcal{H}^{(1)}_{(jk)} \oplus \mathcal{H}^{(2)}_{(jk)} \oplus \mathcal{H}^{(3)}_{(jk)}$, $\mathcal{H}^{(1)}_{(jk)} = \mathcal{H}^0_{(j)} \otimes \mathcal{H}^1_{(k)}$, $\mathcal{H}^{(2)}_{(jk)} = \mathcal{H}^1_{(j)} \otimes \mathcal{H}^0_{(k)}$, and $\mathcal{H}^{(3)}_{(jk)} = \mathcal{H}^1_{(j)} \otimes \mathcal{H}^1_{(k)}$. We denote R_{jk1}, R_{jk2} and R_{jk3} as the RKs for $\mathcal{H}^{(1)}_{(jk)}$, $\mathcal{H}^{(2)}_{(jk)}$ and $\mathcal{H}^{(3)}_{(jk)}$. The RK of $\mathcal{H}^0_{(jk)}$ is $R_{jk0}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{v=1}^{m_{jk}} \psi_{jkv}(x_j, x_k)\psi_{jkv}(z_j, z_k)$. In this case, we have $\mathcal{H}_{(jk)} = \mathcal{H}^0_{(jk)} \oplus \mathcal{H}^{(1)}_{(jk)} \oplus \mathcal{H}^{(2)}_{(jk)} \oplus \mathcal{H}^{(3)}_{(jk)}$ and the corresponding RK of this functional space is $R_{jk} = R_{jk0} + R_{jk1} + R_{jk2} + R_{jk3}$.

In our joint density method, we consider the L_2 penalty for main effects for smoothness and L_1 penalty for interactions for sparsity. Specifically, we propose the following

penalized pseudo-likelihood:

$$\log\left\{\frac{1}{n}\sum_{i=1}^{n}e^{-g(\boldsymbol{x}_i)}\right\}+\int_{\mathcal{X}}g(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}+\frac{\lambda_1}{2}J_1(g)+\tau_1 J_2(g), \tag{2.4}$$

where $J_1(g)=\sum_{j=1}^{p}\theta_j^{-1}||P_jg_j||^2$, P_j is the projection operator in $\mathcal{H}_{(j)}$ onto $\mathcal{H}^1_{(j)}$, $\lambda_1\theta_j^{-1}$ for $j=1,\cdots,p$ are smoothing parameters, $J_2(g)=\sum_{1\le j<k\le p}w_{jk}||g_{jk}(x_j,x_k)||$ is an L_1 penalty for interaction terms, and $0\le w_{jk}<\infty$ are pre-specified weights.

Similar to Lin and Zhang [27], we will minimize the following equivalent but more convenient form

$$\begin{aligned}&\log\left\{\frac{1}{n}\sum_{i=1}^{n}e^{-g(\boldsymbol{x}_i)}\right\}+\int_{\mathcal{X}}g(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\\&+\frac{\lambda_1}{2}\left\{\sum_{j=1}^{p}\theta_j^{-1}||P_jg_j||^2+\sum_{1\le j<k\le p}w_{jk}\theta_{jk}^{-1}||g_{jk}(x_j,x_k)||^2\right\}+\lambda_2\sum_{1\le j<k\le p}w_{jk}\theta_{jk},\end{aligned} \tag{2.5}$$

subject to $\theta_{jk}\ge 0$ for $1\le j<k\le p$, where $\lambda_1\theta_j^{-1}$ for $j=1,\cdots,p$ are smoothing parameters and λ_2 is a tuning parameter. Let $\boldsymbol{\theta}_1=(\theta_1,\cdots,\theta_p)^T$, $\boldsymbol{\theta}_2=(\theta_{12},\cdots,\theta_{(p-1)p})^T$, and $\boldsymbol{w}=(w_{12},\cdots,w_{(p-1)p})^T$. The equivalence is given by the following lemma.

Lemma 2.1 *Set $\lambda_2=\tau_1^2/2\lambda_1$. If $\hat{g}$ minimizes* (2.4)*, set $\hat{\theta}_{jk}=\lambda_1^{1/2}\lambda_2^{-1/2}||\hat{g}_{jk}||/\sqrt{2}$, then the pair $(\hat{\boldsymbol{\theta}}_2,\hat{g})$ minimizes* (2.5)*. On the other hand, if a pair $(\hat{\boldsymbol{\theta}}_2,\hat{g})$ minimizes* (2.5)*, then $\hat{g}$ minimizes* (2.4)*.*

Proof: Denote the functional in (2.4) by $A(g)$ and the functional in (2.5) by $B(\boldsymbol{\theta}_2,g)$. We have $\frac{\lambda_1}{2}\theta_{jk}^{-1}||g_{jk}||^2+\lambda_2\theta_{jk}\ge\sqrt{2}\lambda_1^{1/2}\lambda_2^{1/2}||g_{jk}||=\tau_1||g_{jk}||$, for any $\theta_{jk}\ge 0$ and $g\in\mathcal{H}$, and the equality holds if and only if $\theta_{jk}=\lambda_1^{1/2}\lambda_2^{-1/2}||g_{jk}||/\sqrt{2}$. Therefore, $B(\boldsymbol{\theta}_2,g)\ge A(g)$ for any $\theta_{jk}\ge 0$ and $g\in\mathcal{H}$, and the equality holds if and only if $\theta_{jk}=\lambda_1^{1/2}\lambda_2^{-1/2}||g_{jk}||/\sqrt{2}$ for $1\le j<k\le p$. The conclusion of lemma follows. □

2.2 Computation and Algorithm

In this section, we derive our algorithm for solving (2.5). Since in general the minimization problem (2.5) does not have a solution in a finite dimensional space, as in Gu [12], we approximate the solution by a subset of representers. Specifically, let $\{\tilde{\boldsymbol{x}}_u = (\tilde{x}_{u,1}, \cdots, \tilde{x}_{u,p}),\ u = 1, \cdots, q\}$ be a subset of all observations $\{\boldsymbol{x}_i,\ i = 1, \cdots, n\}$. We collect all basis functions ϕ_{jk} for $j = 1, \cdots, p$ and $k = 1, \cdots, m_j$ and denote them as $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_m)^T$, a vector of functions of $\boldsymbol{x}$ with dimension $m = \sum_{j=1}^{p} m_j$. Let $\xi_{ju}(x_j) = R_j(\tilde{x}_{u,j}, x_j)$, $\xi_{\boldsymbol{\theta_1},u}(\boldsymbol{x}) = \sum_{j=1}^{p} \theta_j \xi_{ju}(x_j)$, $\xi_{jku}(x_j, x_k) = R_{jk}((\tilde{x}_{u,j}, \tilde{x}_{u,k}), (x_j, x_k))$, and $\xi_{\boldsymbol{\theta_2},u}(\boldsymbol{x}) = \sum_{1 \le j < k \le p} w_{jk}^{-1} \theta_{jk} \xi_{jku}(x_j, x_k)$ for $u = 1, \cdots, q$. Let $\boldsymbol{\xi}_{\boldsymbol{\theta}_1}(\boldsymbol{x}) = (\xi_{\boldsymbol{\theta_1},1}, \cdots, \xi_{\boldsymbol{\theta_1},q})^T$, $\boldsymbol{\xi}_{\boldsymbol{\theta}_2}(\boldsymbol{x}) = (\xi_{\boldsymbol{\theta_2},1}, \cdots, \xi_{\boldsymbol{\theta_2},q})^T$, and $\boldsymbol{\xi}(\boldsymbol{x}) = \boldsymbol{\xi}_{\boldsymbol{\theta}_1}(\boldsymbol{x}) + \boldsymbol{\xi}_{\boldsymbol{\theta}_2}(\boldsymbol{x})$. The approximate solution can be represented as

$$\begin{aligned} \hat{g}(\boldsymbol{x}) &= \sum_{v=1}^{m} d_v \phi_v(\boldsymbol{x}) + \sum_{u=1}^{q} c_u \left\{ \sum_{j=1}^{p} \theta_j \xi_{ju}(x_j) + \sum_{1 \le j < k \le p} w_{jk}^{-1} \theta_{jk} \xi_{jku}(x_j, x_k) \right\} \\ &= \boldsymbol{\phi}^T(\boldsymbol{x}) \boldsymbol{d} + \boldsymbol{\xi}^T(\boldsymbol{x}) \boldsymbol{c}, \end{aligned} \tag{2.6}$$

where $\boldsymbol{c} = (c_1, \cdots, c_q)^T$ and $\boldsymbol{d} = (d_1, \cdots, d_m)^T$ are coefficients. Plugging $\hat{g}(\boldsymbol{x}_i)$ in (2.6) into (2.5), we need to compute $\boldsymbol{c}$, $\boldsymbol{d}$, and $\boldsymbol{\theta}_2$ as minimizers of

$$\log \left\{ \frac{1}{n} \sum_{i=1}^{n} e^{-\boldsymbol{\phi}_i^T \boldsymbol{d} - \boldsymbol{\xi}_i^T \boldsymbol{c}} \right\} + \boldsymbol{b}_{\boldsymbol{\phi}}^T \boldsymbol{d} + \boldsymbol{b}_{\boldsymbol{\xi}}^T \boldsymbol{c} + \frac{\lambda_1}{2} \boldsymbol{c}^T Q \boldsymbol{c} + \lambda_2 \boldsymbol{w}^T \boldsymbol{\theta}_2 \tag{2.7}$$

subject to $\boldsymbol{\theta}_2 \ge 0$ where $\boldsymbol{\phi}_i = \boldsymbol{\phi}(\boldsymbol{x}_i)$, $\boldsymbol{\xi}_i = \boldsymbol{\xi}(\boldsymbol{x}_i)$, $\boldsymbol{b}_{\boldsymbol{\phi}} = \int_{\mathcal{X}} \boldsymbol{\phi}(\boldsymbol{x}) \rho(\boldsymbol{x}) d\boldsymbol{x}$, $\boldsymbol{b}_{\boldsymbol{\xi}} = \int_{\mathcal{X}} \boldsymbol{\xi}(\boldsymbol{x}) \rho(\boldsymbol{x}) d\boldsymbol{x}$,
$Q_1 = \left\{ \sum_{j=1}^{p} \theta_j R_j(\tilde{x}_{u,j}, \tilde{x}_{v,j}) \right\}_{u,v=1}^{q}$, $Q_{jk} = \left\{ R_{jk}((\tilde{x}_{u,j}, \tilde{x}_{u,k}), (\tilde{x}_{v,j}, \tilde{x}_{v,k})) \right\}_{u,v=1}^{q}$,
$Q_2 = \sum_{1 \le j < k \le p} w_{jk}^{-1} \theta_{jk} Q_{jk}$, and $Q = Q_1 + Q_2$.

In the following, we propose a computational procedure that solves (2.7) iteratively. We first fix $\boldsymbol{\theta}_2$ and update $\boldsymbol{c}$ and $\boldsymbol{d}$ using the Newton-Raphson algorithm. With fixed

$\boldsymbol{\theta}_2$, dropping the last term that does not depend on $\boldsymbol{c}$ and $\boldsymbol{d}$, we update $\boldsymbol{c}$ and $\boldsymbol{d}$ by minimizing

$$A_1(\boldsymbol{d},\boldsymbol{c}) = \log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-\boldsymbol{\phi}_i^T\boldsymbol{d}-\boldsymbol{\xi}_i^T\boldsymbol{c}}\right\} + \boldsymbol{b}_{\boldsymbol{\phi}}^T\boldsymbol{d} + \boldsymbol{b}_{\boldsymbol{\xi}}^T\boldsymbol{c} + \frac{\lambda_1}{2}\boldsymbol{c}^T Q\boldsymbol{c}. \tag{2.8}$$

Taking derivatives of $A_1(\boldsymbol{d},\boldsymbol{c})$ in (2.8) with respect to $\boldsymbol{d}$ and $\boldsymbol{c}$ at $\tilde{g} = \boldsymbol{\phi}^T\tilde{\boldsymbol{d}} + \boldsymbol{\xi}^T\tilde{\boldsymbol{c}}$, one has gradient vectors and Hessian matrices

$$\begin{aligned}
\frac{\partial A_1}{\partial \boldsymbol{d}} &= -\mu_{\tilde{g}}(\boldsymbol{\phi}) + \boldsymbol{b}_{\boldsymbol{\phi}} = -\mu_{\boldsymbol{\phi}} + \boldsymbol{b}_{\boldsymbol{\phi}},\\
\frac{\partial A_1}{\partial \boldsymbol{c}} &= -\mu_{\tilde{g}}(\boldsymbol{\xi}) + \boldsymbol{b}_{\boldsymbol{\xi}} + \lambda_1 Q\tilde{\boldsymbol{c}} = -\mu_{\boldsymbol{\xi}} + \boldsymbol{b}_{\boldsymbol{\xi}} + \lambda_1 Q\tilde{\boldsymbol{c}},\\
\frac{\partial^2 A_1}{\partial \boldsymbol{d}\partial \boldsymbol{d}^T} &= V_{\tilde{g}}(\boldsymbol{\phi},\boldsymbol{\phi}^T) = V_{\boldsymbol{\phi},\boldsymbol{\phi}},\\
\frac{\partial^2 A_1}{\partial \boldsymbol{c}\partial \boldsymbol{c}^T} &= V_{\tilde{g}}(\boldsymbol{\xi},\boldsymbol{\xi}^T) + \lambda_1 Q = V_{\boldsymbol{\xi},\boldsymbol{\xi}} + \lambda_1 Q,\\
\frac{\partial^2 A_1}{\partial \boldsymbol{d}\partial \boldsymbol{c}^T} &= V_{\tilde{g}}(\boldsymbol{\phi},\boldsymbol{\xi}^T) = V_{\boldsymbol{\phi},\boldsymbol{\xi}},
\end{aligned}$$

where $\mu_g(f) = \sum_{i=1}^{n} e^{-g(\boldsymbol{X}_i)} f(\boldsymbol{X}_i) / \sum_{i=1}^{n} e^{-g(\boldsymbol{X}_i)}$ and $V_g(f_1,f_2) = \mu_g(f_1 f_2) - \mu_g(f_1)\mu_g(f_2)$. Then, the Newton updating equation becomes

$$\begin{pmatrix} V_{\boldsymbol{\phi},\boldsymbol{\phi}} & V_{\boldsymbol{\phi},\boldsymbol{\xi}} \\ V_{\boldsymbol{\xi},\boldsymbol{\phi}} & V_{\boldsymbol{\xi},\boldsymbol{\xi}} + \lambda_1 Q \end{pmatrix} \begin{pmatrix} \boldsymbol{d} - \tilde{\boldsymbol{d}} \\ \boldsymbol{c} - \tilde{\boldsymbol{c}} \end{pmatrix} = \begin{pmatrix} \mu_{\boldsymbol{\phi}} - \boldsymbol{b}_{\boldsymbol{\phi}} \\ \mu_{\boldsymbol{\xi}} - \boldsymbol{b}_{\boldsymbol{\xi}} - \lambda_1 Q\tilde{\boldsymbol{c}} \end{pmatrix}.$$

After arranging terms we get,

$$\begin{pmatrix} V_{\boldsymbol{\phi},\boldsymbol{\phi}} & V_{\boldsymbol{\phi},\boldsymbol{\xi}} \\ V_{\boldsymbol{\xi},\boldsymbol{\phi}} & V_{\boldsymbol{\xi},\boldsymbol{\xi}} + \lambda_1 Q \end{pmatrix} \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{c} \end{pmatrix} = \begin{pmatrix} \mu_{\boldsymbol{\phi}} - \boldsymbol{b}_{\boldsymbol{\phi}} + V_{\tilde{g}}(\boldsymbol{\phi},\tilde{g}) \\ \mu_{\boldsymbol{\xi}} - \boldsymbol{b}_{\boldsymbol{\xi}} - \lambda_1 Q\tilde{\boldsymbol{c}} + V_{\tilde{g}}(\boldsymbol{\xi},\tilde{g}) \end{pmatrix}. \tag{2.9}$$

We solve (2.9) using a modified version of the `ssden1` function in the `gss` package, and select λ_1 and $\boldsymbol{\theta}_1$ by the approximate cross-validation (ACV) method (Gu [12]). Details

will be given in Section 2.3.1.

With fixed $\boldsymbol{c}$ and $\boldsymbol{d}$, we update $\boldsymbol{\theta}_2$ using the quadratic programming method. Let $\psi_j(\boldsymbol{x}) = \sum_{u=1}^{q} c_u \xi_{ju}(x_j)$ for $j = 1, \cdots, p$, $\boldsymbol{\psi}_1(\boldsymbol{x}) = (\psi_1, \cdots, \psi_p)^T$, $\psi_{jk}(\boldsymbol{x}) = w_{jk}^{-1} \sum_{u=1}^{q} c_u \xi_{jku}(x_j, x_k)$ for $1 \le j < k \le p$, and $\boldsymbol{\psi}_2(\boldsymbol{x}) = (\psi_{12}, \cdots, \psi_{(p-1)p})^T$. We rewrite $\hat{g}$ in (2.6) as $\hat{g}(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{d} + \boldsymbol{\psi}_1^T(\boldsymbol{x})\boldsymbol{\theta}_1 + \boldsymbol{\psi}_2^T(\boldsymbol{x})\boldsymbol{\theta}_2$. Plugging $\hat{g}(\boldsymbol{x}_i)$ into (2.7) and keeping terms involving $\boldsymbol{\theta}_2$ only, (2.7) reduces to

$$\log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-\boldsymbol{\phi}_i^T\boldsymbol{d} - \boldsymbol{\psi}_{1i}^T\boldsymbol{\theta}_1 - \boldsymbol{\psi}_{2i}^T\boldsymbol{\theta}_2}\right\} + \boldsymbol{b}_{\boldsymbol{\psi}_2}^T\boldsymbol{\theta}_2 + \frac{\lambda_1}{2}\boldsymbol{c}^T Q_2 \boldsymbol{c} + \lambda_2 \boldsymbol{w}^T\boldsymbol{\theta}_2 \tag{2.10}$$

subject to $\boldsymbol{\theta}_2 \ge 0$, where $\boldsymbol{\psi}_{1i} = \boldsymbol{\psi}_1(\boldsymbol{x}_i)$, $\boldsymbol{\psi}_{2i} = \boldsymbol{\psi}_2(\boldsymbol{x}_i)$, and $\boldsymbol{b}_{\boldsymbol{\psi}_2} = \int_{\mathcal{X}} \boldsymbol{\psi}_2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}$. Furthermore, the constraint minimization problem (2.10) is equivalent to

$$A_2(\boldsymbol{\theta}_2) = \log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-\boldsymbol{\phi}_i^T\boldsymbol{d} - \boldsymbol{\psi}_{1i}^T\boldsymbol{\theta}_1 - \boldsymbol{\psi}_{2i}^T\boldsymbol{\theta}_2}\right\} + \boldsymbol{b}_{\boldsymbol{\psi}_2}^T\boldsymbol{\theta}_2 + \frac{\lambda_1}{2}\boldsymbol{c}^T Q_2 \boldsymbol{c} \tag{2.11}$$

subject to $\boldsymbol{\theta}_2 \ge 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \le M$ for some constant M, where M controls the sparsity in $\boldsymbol{\theta}_2$. Note that $A_2(\boldsymbol{\theta}_2)$ is a convex function of $\boldsymbol{\theta}_2$. To prove the convexity, we now show that the Hessian matrix $H_A(\boldsymbol{\theta}_2)$ of $A_2(\boldsymbol{\theta}_2)$ is positive semi-definite. For any vector $\boldsymbol{\nu} \ne \boldsymbol{0}$, let $s_i = e^{-\tilde{g}(\boldsymbol{x}_i)}$ and $t_i = \boldsymbol{\nu}^T\boldsymbol{\psi}_2(\boldsymbol{x}_i)$, we have

$$\boldsymbol{\nu}^T H_A(\boldsymbol{\theta}_2)\boldsymbol{\nu} = \frac{\left(\sum_{i=1}^{n} s_i t_i^2\right)\left(\sum_{i=1}^{n} s_i\right) - \left(\sum_{i=1}^{n} t_i s_i\right)^2}{\left(\sum_{i=1}^{n} s_i\right)^2} \ge 0, \tag{2.12}$$

by the Cauchy-Schwartz inequality.

We solve (2.11) iteratively using the quadratic programming. Denote the current estimate of $\boldsymbol{\theta}_2$ as $\tilde{\boldsymbol{\theta}}_2$ and $\tilde{g}(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{d} + \boldsymbol{\psi}_1^T(\boldsymbol{x})\boldsymbol{\theta}_1 + \boldsymbol{\psi}_2^T(\boldsymbol{x})\tilde{\boldsymbol{\theta}}_2$. We update $\boldsymbol{\theta}_2$ by minimizing the following second order Taylor approximation of $A_2(\boldsymbol{\theta}_2)$ (some constants

independent of $\boldsymbol{\theta}_2$ have been removed):

$$\frac{1}{2}\boldsymbol{\theta}_2^T H_A(\tilde{\boldsymbol{\theta}}_2)\boldsymbol{\theta}_2 + \boldsymbol{\theta}_2^T \left\{ G_A(\tilde{\boldsymbol{\theta}}_2) - H_A(\tilde{\boldsymbol{\theta}}_2)\tilde{\boldsymbol{\theta}}_2 \right\} \tag{2.13}$$

subject to $\boldsymbol{\theta}_2 \geq 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$ for some constant M, where $G_A(\tilde{\boldsymbol{\theta}}_2) = -\mu_{\tilde{g}}(\boldsymbol{\psi}_2) + \boldsymbol{b}_{\boldsymbol{\psi}_2} + \lambda_1 \boldsymbol{q}_2/2$ is the gradient, $H_A(\tilde{\boldsymbol{\theta}}_2) = V_{\tilde{g}}(\boldsymbol{\psi}_2, \boldsymbol{\psi}_2^T)$ is the Hessian, $\boldsymbol{q}_2 = (w_{12}^{-1}\boldsymbol{c}^T Q_{12}\boldsymbol{c}, \cdots, w_{(p-1)p}^{-1}\boldsymbol{c}^T Q_{(p-1)p}\boldsymbol{c})^T$, and $Q_{jk} = \left\{ R_{jk}((\tilde{x}_{u,j}, \tilde{x}_{u,k}), (\tilde{x}_{v,j}, \tilde{x}_{v,k})) \right\}_{u,v=1}^{q}$ for $1 \leq j < k \leq p$.

We apply the quadratic programming to solve (2.13) and k-fold cross-validation to select M. The iterative procedure for updating $\boldsymbol{\theta}_2$ may be stopped after a fixed number of steps or until convergence. We summarize the whole algorithm as follows.

Algorithm for the joint density approach:

1. *Initialize*: $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_2^0$.

2. *Cycle until convergence*: Update $\boldsymbol{c}$, $\boldsymbol{d}$ and $\boldsymbol{\theta}_2$ sequentially:

   (a) Fix $\boldsymbol{\theta}_2$ at the current estimate, update $\boldsymbol{c}$ and $\boldsymbol{d}$ by solving (2.9) with tuning parameters λ_1, $\boldsymbol{\theta}_1$ selected by the ACV method.

   (b) Fix $\boldsymbol{d}$, $\boldsymbol{c}$, λ_1 and $\boldsymbol{\theta}_1$ at the current estimates, update $\boldsymbol{\theta}_2$ by applying quadratic programming to iteratively solve the quadratic approximations (2.13) subject to $\boldsymbol{\theta}_2 \geq 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$ where the tuning parameter M is selected by the k-fold cross-validation.

2.3 Implementation of the Algorithm

In this section, we provide details about the implementation of the proposed algorithm using existing R packages. Specifically, we implement Step 2.(a) in the algorithm using

a modification of the `ssden1` function in the `gss` package (Gu et al. [17]) and Step 2.(b) using the R function `solve.QP` in the `quadprog` package (Turlach and Weingessel [38]).

2.3.1 Implementation of the Newton-Raphson Method

Given current value of $\boldsymbol{\theta}_2$, we update $\boldsymbol{c}$ and $\boldsymbol{d}$ by minimizing (2.8) using the Newton-Raphson method. We implement by modifying the function `ssden1` in the `gss` package since (2.8) has the same form as (10.6) in Gu [12] with different penalties. By definition, $\mathcal{H}_{(jk)} = \mathcal{H}_{(j)} \otimes \mathcal{H}_{(k)} = (\mathcal{H}^0_{(j)} \oplus \mathcal{H}^1_{(j)}) \otimes (\mathcal{H}^0_{(k)} \oplus \mathcal{H}^1_{(k)}) = (\mathcal{H}^0_{(j)} \otimes \mathcal{H}^0_{(k)}) \oplus (\mathcal{H}^0_{(j)} \otimes \mathcal{H}^1_{(k)}) \oplus (\mathcal{H}^1_{(j)} \otimes \mathcal{H}^0_{(k)}) \oplus (\mathcal{H}^1_{(j)} \otimes \mathcal{H}^1_{(k)}) = \mathcal{H}^{(0)}_{(jk)} \oplus \mathcal{H}^{(1)}_{(jk)} \oplus \mathcal{H}^{(2)}_{(jk)} \oplus \mathcal{H}^{(3)}_{(jk)}$ where $\mathcal{H}^{(0)}_{(jk)} = \mathcal{H}^0_{(j)} \otimes \mathcal{H}^0_{(k)}$, $\mathcal{H}^{(1)}_{(jk)} = \mathcal{H}^0_{(j)} \otimes \mathcal{H}^1_{(k)}$, $\mathcal{H}^{(2)}_{(jk)} = \mathcal{H}^1_{(j)} \otimes \mathcal{H}^0_{(k)}$, and $\mathcal{H}^{(3)}_{(jk)} = \mathcal{H}^1_{(j)} \otimes \mathcal{H}^1_{(k)}$. For density estimation, the penalized likelihood method in Gu [12] does not penalize functions in the parametric component space $\mathcal{H}^0_{(jk)}$ and has different smoothing parameters for components in the nonparametric component spaces $\mathcal{H}^{(1)}_{(jk)}$, $\mathcal{H}^{(2)}_{(jk)}$, and $\mathcal{H}^{(3)}_{(jk)}$. Our goal is edge detection by detecting non-zero interactions. Therefore, we penalize the combined interaction $g_{jk} \in \mathcal{H}_{(jk)}$ as a whole with a smoothing parameter θ_{jk} for $1 \le j < k \le p$. The interaction g_{jk} collects parametric and nonparametric interaction components in $\mathcal{H}^{(0)}_{(jk)}$, $\mathcal{H}^{(1)}_{(jk)}$, $\mathcal{H}^{(2)}_{(jk)}$, and $\mathcal{H}^{(3)}_{(jk)}$. Note that $\boldsymbol{\theta}_2 = (\theta_{12}, \cdots, \theta_{(p-1)p})^T$ is fixed at this step. We modified the function `ssden1` to solve (2.8) with smoothing parameters λ_1 and $\boldsymbol{\theta}_1$ estimated by an approximated cross-validation estimate of the Kullback-Leibler (KL) divergence. More details can be found in Gu [12] and Gu et al. [13].

2.3.2 Implementation of Quadratic Programming

With $\boldsymbol{c}$ and $\boldsymbol{d}$ being fixed at their current values, we need to update $\boldsymbol{\theta}_2$ iteratively by applying the quadratic programming algorithm to minimize

$$\frac{1}{2}\boldsymbol{\theta}_2^T H_A(\tilde{\boldsymbol{\theta}}_2)\boldsymbol{\theta}_2 + \boldsymbol{\theta}_2^T \left\{ G_A(\tilde{\boldsymbol{\theta}}_2) - H_A(\tilde{\boldsymbol{\theta}}_2)\tilde{\boldsymbol{\theta}}_2 \right\} \tag{2.14}$$

subject to $\theta_{jk} \geq 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$ for some M, where M is a tuning parameter. We use the R function `solve.QP` to solve (2.14). We estimate the tuning parameter M by minimizing a k-fold cross-validation derived as follows.

Let $I_j = \{i_1, \cdots, i_j\}$ be the sample indexes in j-th fold. We consider the following loss function

$$V(M) = \log\left\{ \frac{1}{n}\sum_{j=1}^{k}\sum_{i\in I_j} e^{-g_M^{(-j)}(\boldsymbol{x}_i)} \right\} + \int_{\mathcal{X}} g_M^{(-j)}(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}, \tag{2.15}$$

where $g_M^{(-j)}$ denotes the estimate of g_0 without the observations in j-th fold, and the subscript M denotes the estimate is based on each fixed M. Let $\boldsymbol{\theta}_{2,M}^{(-j)}$ be the estimate of $\boldsymbol{\theta}_2$ without the observations in j-th fold. Then, $g_M^{(-j)} = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{d} + \boldsymbol{\psi}_1^T(\boldsymbol{x})\boldsymbol{\theta}_1 + \boldsymbol{\psi}_2^T(\boldsymbol{x})\boldsymbol{\theta}_{2,M}^{(-j)}$. We minimize the following version of (2.5) with respect to $\boldsymbol{\theta}_2$

$$\log\left\{ \frac{1}{n}\sum_{i\notin I_j} e^{-\boldsymbol{\phi}_i^T\boldsymbol{d} - \boldsymbol{\psi}_{1i}^T\boldsymbol{\theta}_1 - \boldsymbol{\psi}_{2i}^T\boldsymbol{\theta}_2} \right\} + \boldsymbol{b}_{\boldsymbol{\psi}_2}^T\boldsymbol{\theta}_2 + \frac{\lambda_1}{2}\boldsymbol{c}^T Q_2 \boldsymbol{c}, \tag{2.16}$$

subject to $\boldsymbol{\theta}_2 \geq 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$.

Define $\mu_{\tilde{g}}^{(-j)}(h) = \sum_{i\notin I_j} e^{-\tilde{g}(\boldsymbol{x}_i)}h(\boldsymbol{x}_i) / \sum_{i\notin I_j} e^{-\tilde{g}(\boldsymbol{x}_i)}$. Then, we solve $\boldsymbol{\theta}_2$ by minimizing the following second order Taylor approximation of (2.16) (some constants independent of

$\boldsymbol{\theta}_2$ have been removed):

$$\frac{1}{2}\boldsymbol{\theta}_2^T H_A^{(-j)}(\tilde{\boldsymbol{\theta}}_2)\boldsymbol{\theta}_2 + \boldsymbol{\theta}_2^T \left\{ G_A^{(-j)}(\tilde{\boldsymbol{\theta}}_2) - H_A^{(-j)}(\tilde{\boldsymbol{\theta}}_2)\tilde{\boldsymbol{\theta}}_2 \right\} \tag{2.17}$$

subject to $\boldsymbol{\theta}_2 \geq 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$, where $G_A^{(-j)}(\tilde{\boldsymbol{\theta}}_2) = -\mu_{\tilde{g}}^{(-j)}(\boldsymbol{\psi}_2) + \boldsymbol{b}_{\boldsymbol{\psi}_2} + \lambda_1 \boldsymbol{q}_2/2$ is the gradient, $H_A^{(-j)}(\tilde{\boldsymbol{\theta}}_2) = V_{\tilde{g}}(\boldsymbol{\psi}_2, \boldsymbol{\psi}_2^T)$ is the Hessian, $V_{\tilde{g}}(\boldsymbol{\psi}_2, \boldsymbol{\psi}_2^T) = \mu_{\tilde{g}}^{(-j)}(\boldsymbol{\psi}_2\boldsymbol{\psi}_2^T) - \mu_{\tilde{g}}^{(-j)}(\boldsymbol{\psi}_2)\mu_{\tilde{g}}^{(-j)}(\boldsymbol{\psi}_2^T)$, $\boldsymbol{q}_2$ and $\boldsymbol{b}_{\boldsymbol{\psi}_2}$ are defined as before. By minimizing (2.17), we can obtain the estimate $\boldsymbol{\theta}_{2,M}^{(-j)}$ for $\boldsymbol{\theta}_2$ without the observations in j-th fold. Then, we obtain $g_M^{(-j)}$ using $\boldsymbol{\theta}_{2_M}^{(-j)}$ and plug it into (2.15). We select the M that minimizes the k-fold cross-validation score in (2.15).

2.3.3 Initial Values and Convergence Criterion

To get a good initial value $\boldsymbol{\theta}_2^0$, we first estimate $g(\boldsymbol{x})$ with $\tau_1 \sum_{1\leq j<k\leq p} w_{jk}||g_{jk}||$ in (2.4) being replaced by $(\lambda_1/2) \sum_{1\leq j<k\leq p} \theta_{jk}^{-1}||g_{jk}||^2$. We modified the `ssden1` function in the `gss` package to estimate g and denote the estimate of g_{jk} as $\check{g}_{jk}$. Since $\theta_{jk} = 0$ in $\boldsymbol{\theta}_2$ iff $g_{jk} = 0$, the magnitude of $\check{g}_{jk}$ provides one way to initialize θ_{jk}. Specifically, we set $\theta_{jk}^0 = \{\sum_{i=1}^n \check{g}_{jk}^2(\boldsymbol{x}_i)\}^{1/2}$.

The convergence criterion in Step 2 in the algorithm is $||\boldsymbol{\theta}_2 - \tilde{\boldsymbol{\theta}}_2||_2/(||\tilde{\boldsymbol{\theta}}_2||_2 + 10^{-6}) \leq \varepsilon$ or the number of zeros in $\boldsymbol{\theta}_2$ stops increasing for fixed number of steps, where $\boldsymbol{\theta}_2$ and $\tilde{\boldsymbol{\theta}}_2$ are the updated and previous estimates, respectively, $||\cdot||_2$ is the Euclidean norm, and ε a threshold. We set $\varepsilon = 0.001$ in simulation and real data examples.

Chapter 3

Theoretical Analysis

In this chapter, we study the theoretical properties of the proposed joint approach. Following similar steps and under same regularity conditions as Gu [12], we derive convergence rate for the joint density estimate $\hat{g}$ subject to both L_1 and L_2 penalties. In addition, we derive the convergence rate for interactions in the SS ANOVA decomposition, which is new and important for edge detection.

3.1 Notations

Let $f_0(\boldsymbol{x}) = e^{g_0(\boldsymbol{x})}\rho(\boldsymbol{x})$ be the density to be estimated. Let $g = g^{(1)} + g^{(2)} = \sum_{j=1}^{p} g_j + \sum_{1\le j<k\le p} g_{jk}$. Let $\hat{g}$ be the minimizer of (2.4). Define

$$
\begin{aligned}
V^*(h_1, h_2) &= \int_{\mathcal{X}} h_1(\boldsymbol{x}) h_2(\boldsymbol{x}) \rho(\boldsymbol{x}) d\boldsymbol{x} \\
J_1(h_1, h_2) &= \sum_{j=1}^{p} \theta_j^{-1} \int_{\mathcal{X}_j} (P_j h_1)(P_j h_2) dx_j \\
J_2(h_1, h_2) &= \sum_{1\le j<k\le p} w_{jk} \Big(\int_{\mathcal{X}_j} \int_{\mathcal{X}_k} |(P_j h_1)(P_j h_2)| dx_j dx_k \Big)^{1/2} \\
J_2^*(h_1, h_2) &= \sum_{1\le j<k\le p} \theta_{jk}^{-1} \int_{\mathcal{X}_j} \int_{\mathcal{X}_k} (P_j h_1)(P_j h_2) dx_j dx_k
\end{aligned}
$$

for any functions $h_1, h_2 \in \mathcal{H}$. We denote $||h|| = (\int_{\mathcal{X}_j} h^2 dx_j)^{1/2}$ as the L_2 norm for any $h \in \mathcal{H}^{(j)}$. Then, we have $V^*(g) = V^*(g, g)$. Denote $V_1(g^{(1)}) = V^*(g^{(1)})$, $V_2(g^{(2)}) = [V^*(g^{(2)})]^{1/2}$, $J_1(g) = J_1(g, g) = \sum_{j=1}^{p} \theta_j^{-1} ||P_j g_j||^2$, $J_2(g) = J_2(g, g) = \sum_{1\le j<k\le p} w_{jk} ||g_{jk}||$, and $J_2^*(g) = J_2^*(g, g) = \sum_{1\le j<k\le p} \theta_{jk}^{-1} ||g_{jk}||^2$. Without loss of generality, we assume $w_{jk} = 1$ in the proof, simulations and real applications. Furthermore, we let $V(g) = V_1(g^{(1)}) + V_2(g^{(2)})$, $J = J_1 + J_2$, and $J^*(g) = J_1(g) + J_2^*(g)$.

We derive convergence rates under metrics $V^* + \lambda_1 J^*$ and $V + \lambda_1 J$. For a sequence of random variables, $\{A_n\}$, and a sequence of constants, $\{a_n\}$, the notation $A_n = O_p(a_n)$, means that $\{A_n/a_n\}$ is stochastically bounded (or bounded in probability). That is, for any $\tau > 0$, there exit a constant $K(\tau)$ and an integer $n(\tau)$ such that if $n \ge n(\tau)$, we have

$$P(|A_n/a_n| \le K(\tau)) \ge 1 - \tau.$$

The notation $A_n = o_p(a_n)$ denotes that for $\forall \epsilon > 0$

$$\lim_{n\to\infty} P(|A_n/a_n - 0| \leq \epsilon) = 1.$$

More details and examples of O_p, o_p notations can be found in Section 1.2 of Serfling [35]. Furthermore, let S_n be a sequence of random variables. We denote $S_n \leq_p C$ if $P(\limsup_{n\to\infty} S_n \leq C) = 1$ for a fixed constant $C < \infty$, and denote $S_n \xrightarrow{a.s.} C$ if $P(\lim_{n\to\infty} |S_n - C| \leq \epsilon) = 1$ for $\forall \epsilon > 0$.

3.2 Convergence Rates

We start this section by introducing conditions and lemmas that are needed for theoretical analysis.

Condition 3.1 *V^* is completely continuous with respect to J^*.*

From Theorem 3.1 of Weinberger [42], there exists eigenvalues γ_v of J^* with respect to V^* and the associated eigenfunctions ζ_v such that

$$V^*(\zeta_v, \zeta_u) = \delta_{v,u}, \qquad J^*(\zeta_v, \zeta_u) = \gamma_v \delta_{v,u},$$

where $0 \leq \gamma_v \uparrow \infty$ and $\delta_{v,u}$ is the Kronecker delta. We refer to γ_v as the eigenvalues of J^* with respect to V^* and to ζ_v as the associated eigenfunctions. Functions satisfying $J^*(g) < \infty$ can be expressed as a Fourier series expansion $g = \sum_v a_v \zeta_v$, where $a_v = V^*(g, \zeta_v)$ are the Fourier coefficients.

Condition 3.2 *For v sufficiently large and some $\varphi > 0$, the eigenvalues γ_v of J^* with respect to V^* satisfy $\gamma_v > \varphi v^r$ where $r > 1$.*

Consider the quadratic functional

$$\frac{1}{n}\sum_{i=1}^{n} -e^{-g_0(\boldsymbol{X}_i)}g(\boldsymbol{X}_i) + \int_{\mathcal{X}} g(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} + \frac{1}{2}V^*(g-g_0) + \frac{\lambda_1}{2}J^*(g), \tag{3.1}$$

and denote the minimizer of (3.1) as $\tilde{g}$. Plugging the Fourier series expansions $g = \sum_v a_v\zeta_v$ and $g_0 = \sum_v a_{v,0}\zeta_v$ into (3.1), $\tilde{g}$ has Fourier coefficients $\tilde{a}_v = (\kappa_v + a_{v,0})/(1+\lambda_1\gamma_v)$, where $\kappa_v = n^{-1}\sum_{i=1}^{n}\{e^{-g_0(\boldsymbol{X}_i)}\zeta_v(\boldsymbol{X}_i) - \int_{\mathcal{X}}\zeta_v(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\}$. It is not difficult to verify that $E(\kappa_v) = 0$ and $E(\kappa_v^2) \le n^{-1}\int_{\mathcal{X}}\zeta_v^2(\boldsymbol{x})e^{-g_0(\boldsymbol{x})}\rho(\boldsymbol{x})d\boldsymbol{x}$.

Condition 3.3 *For some $c_1 < \infty$, $e^{-g_0} < c_1$.*

Under Condition 3.3, noting that $V^*(\zeta_v) = \int_{\mathcal{X}}\zeta_v^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} = 1$ by the definition of V^* and ζ_v, we have $E(\kappa_v^2) \le c_1/n$.

By the Fourier series expansions of $\tilde{g}$ and g_0, we can show that

$$\begin{aligned} V^*(\tilde{g}-g_0) &= \sum_v(\tilde{a}_v - a_{v,0})^2 = \sum_v \frac{\kappa_v^2 - 2\kappa_v\lambda_1\gamma_v a_{v,0} + \lambda_1^2\gamma_v^2 a_{v,0}^2}{(1+\lambda_1\gamma_v)^2}, \\ \lambda_1 J^*(\tilde{g}-g_0) &= \sum_v \lambda_1\gamma_v(\tilde{a}_v - a_{v,0})^2 = \sum_v \lambda_1\gamma_v\frac{\kappa_v^2 - 2\kappa_v\lambda_1\gamma_v a_{v,0} + \lambda_1^2\gamma_v^2 a_{v,0}^2}{(1+\lambda_1\gamma_v)^2}. \end{aligned}$$

Since $E(\kappa_v) = 0$ and $E(\kappa_v^2) \le c_1/n$, we have

$$\begin{aligned} E[V^*(\tilde{g}-g_0)] &\le \frac{c_1}{n}\sum_v\frac{1}{(1+\lambda_1\gamma_v)^2} + \lambda_1\sum_v\frac{\lambda_1\gamma_v}{(1+\lambda_1\gamma_v)^2}\gamma_v a_{v,0}^2, \\ E[\lambda_1 J^*(\tilde{g}-g_0)] &\le \frac{c_1}{n}\sum_v\frac{\lambda_1\gamma_v}{(1+\lambda_1\gamma_v)^2} + \lambda_1\sum_v\frac{(\lambda_1\gamma_v)^2}{(1+\lambda_1\gamma_v)^2}\gamma_v a_{v,0}^2. \end{aligned} \tag{3.2}$$

We can further bound these two quantities by using the following lemma.

Lemma 3.1 *Under Condition 3.2, as $\lambda \to 0$, one has*

$$\sum_v \frac{\lambda\gamma_v}{(1+\lambda\gamma_v)^2} = O(\lambda^{-1/r}), \quad \sum_v \frac{1}{(1+\lambda\gamma_v)^2} = O(\lambda^{-1/r}), \quad \sum_v \frac{\lambda\gamma_v}{1+\lambda\gamma_v} = O(\lambda^{-1/r}).$$

Proof: We prove the first equation.

$$\begin{aligned}
\sum_v \frac{\lambda\gamma_v}{(1+\lambda\gamma_v)^2} &= \left\{ \sum_{v<\lambda^{-1/r}} + \sum_{v\geq\lambda^{-1/r}} \right\} \frac{\lambda\gamma_v}{(1+\lambda\gamma_v)^2} \\
&= O(\lambda^{-1/r}) + O\Big(\int_{\lambda^{-1/r}}^{\infty} \frac{\lambda x^r}{(1+\lambda x^r)^2} dx \Big) \\
&= O(\lambda^{-1/r}) + \lambda^{-1/r} O\Big(\int_{1}^{\infty} \frac{x^r}{(1+x^r)^2} dx \Big) \\
&= O(\lambda^{-1/r}).
\end{aligned}$$

The other two follow similar arguments. □

Theorem 3.1 *Assume $J^*(g_0) < \infty$. Under Conditions 3.1–3.3, as $\lambda_1 \to 0$ and $n \to \infty$,*

$$(V^* + \lambda_1 J^*)(\tilde{g} - g_0) = O(n^{-1}\lambda_1^{-1/r} + \lambda_1).$$

Proof: Note that $\sum_v \rho_v g_{v,0}^2 = J^*(g_0) < \infty$. The theorem follows from (3.2) and Lemma 3.1. □

As in Gu [12], when g_0 is "supersmooth", in the sense that $\sum_v \gamma_v^l a_{v,0}^2 < \infty$ for some $1 < l \leq 2$ which is assumed in Theorem 3.2, the rates can be improved to $O(n^{-1}\lambda_1^{-1/r} + \lambda_1^l)$.

Now we want to bound the approximation error $\hat{g} - \tilde{g}$. Define

$$A_{f,h}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} e^{-(f+\alpha h)(\boldsymbol{X}_i)} + \int_{\mathcal{X}} (f+\alpha h)\rho + \frac{\lambda_1}{2} J^*(f+\alpha h) + \lambda_2 \sum_{1\le j<k\le p} \theta_{jk}$$

$$B_{f,h}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} -e^{-g_0(\boldsymbol{X}_i)}(f+\alpha h)(\boldsymbol{X}_i) + \int_{\mathcal{X}} (f+\alpha h)\rho + \frac{1}{2} V^*(f+\alpha h - g_0) + \frac{\lambda_1}{2} J^*(f+\alpha h).$$

We take derivative for $A_{f,h}, B_{f,h}$ with respect to α evaluated at $\alpha = 0$. Then we obtain

$$\dot{A}_{f,h}(0) = \frac{1}{n}\sum_{i=1}^{n} -e^{-f(\boldsymbol{X}_i)} h(\boldsymbol{X}_i) + \int_{\mathcal{X}} h\rho + \lambda_1 J^*(f,h), \tag{3.3}$$

$$\dot{B}_{f,h}(0) = \frac{1}{n}\sum_{i=1}^{n} -e^{-g_0(\boldsymbol{X}_i)} h(\boldsymbol{X}_i) + \int_{\mathcal{X}} h\rho + V^*(f-g_0,h) + \lambda_1 J^*(f,h). \tag{3.4}$$

Plugging $f = \hat{g}$ and $h = \hat{g} - \tilde{g}$ into (3.3), we have

$$\frac{1}{n}\sum_{i=1}^{n} -e^{-\hat{g}(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) + \int_{\mathcal{X}} (\hat{g}-\tilde{g})\rho + \lambda_1 J^*(\hat{g}, \hat{g}-\tilde{g}) = 0. \tag{3.5}$$

Setting $f = \tilde{g}$ and $h = \hat{g} - \tilde{g}$ in (3.4), we obtain

$$\frac{1}{n}\sum_{i=1}^{n} -e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) + \int_{\mathcal{X}} (\hat{g}-\tilde{g})\rho + V^*(\tilde{g}-g_0, \hat{g}-\tilde{g}) + \lambda_1 J^*(\tilde{g}, \hat{g}-\tilde{g}) = 0. \tag{3.6}$$

Subtracting (3.6) from (3.5), we have

$$\begin{aligned} &\lambda_1 J^*(\hat{g}-\tilde{g}) - \frac{1}{n}\sum_{i=1}^{n} \{e^{-\hat{g}(\boldsymbol{x}_i)} - e^{-\tilde{g}(\boldsymbol{x}_i)}\}(\hat{g}-\tilde{g})(\boldsymbol{x}_i) \\ =&\frac{1}{n}\sum_{i=1}^{n} \{e^{-\tilde{g}(\boldsymbol{x}_i)} - e^{-g_0(\boldsymbol{x}_i)}\}(\hat{g}-\tilde{g})(\boldsymbol{x}_i) + V^*(\hat{g}-\tilde{g}, \tilde{g}-g_0). \end{aligned} \tag{3.7}$$

Condition 3.4 *For g in a convex set B_0 around g_0 containing $\hat{g}$ and $\tilde{g}$, $c_2 < e^{g_0 - g} < c_3$ holds uniformly for some $0 < c_2 < c_3 < \infty$.*

Condition 3.5 *For any $u, v = 1, 2, \cdots$, $\int_{\mathcal{X}} \zeta_v^2 \zeta_u^2 e^{-g_0} \rho(\boldsymbol{x}) d\boldsymbol{x} < c_4$ for some $c_4 < \infty$.*

Applying the mean value theorem, we have $e^{-\hat{g}(\boldsymbol{X}_i)} - e^{-\tilde{g}(\boldsymbol{X}_i)} = -e^{-(\tilde{g}+\tau_i(\hat{g}-\tilde{g}))(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})(\boldsymbol{X}_i)$ where $\tau_i \in [0, 1]$. Since $\hat{g}$ and $\tilde{g}$ belongs to B_0 which is a convex set around g_0, under Condition 3.4, there exists a $b_0^{(i)} \in (c_2, c_3)$ such that $-e^{-(\tilde{g}+\tau_i(\hat{g}-\tilde{g}))(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) = -b_0^{(i)} e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})(\boldsymbol{X}_i)$. Then

$$\begin{aligned} -\frac{1}{n}\sum_{i=1}^{n} \left\{e^{-\hat{g}(\boldsymbol{X}_i)} - e^{-\tilde{g}(\boldsymbol{X}_i)}\right\}(\hat{g} - \tilde{g})(\boldsymbol{X}_i) =& \frac{1}{n}\sum_{i=1}^{n} b_0^{(i)} e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_i) \\ \geq& \frac{c_2}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_i). \end{aligned} \tag{3.8}$$

By the same argument, there exists a $c_0^{(i)} \in (c_2, c_3)$ such that

$$\frac{1}{n}\sum_{i=1}^{n} \left\{e^{-\tilde{g}(\boldsymbol{X}_i)} - e^{-g_0(\boldsymbol{X}_i)}\right\}(\hat{g} - \tilde{g})(\boldsymbol{X}_i) = -\frac{1}{n}\sum_{i=1}^{n} c_0^{(i)} e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})(\boldsymbol{X}_i)(\tilde{g} - g_0)(\boldsymbol{X}_i). \tag{3.9}$$

Lemma 3.2 *Under Conditions 3.1, 3.2 and 3.5, suppose h_1 and h_2 are functions satisfying $J^*(h_1) < \infty, J^*(h_2) < \infty$, as $\lambda_1 \to 0$ and $n\lambda_1^{2/r} \to \infty$, one has*

$$\left|\frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)} h_1(\boldsymbol{X}_i) h_2(\boldsymbol{X}_i) - V^*(h_1, h_2)\right| = o_p\Big(\{(V^* + \lambda_1 J^*)(h_1)(V^* + \lambda_1 J^*)(h_2)\}^{1/2}\Big). \tag{3.10}$$

Proof: Since $J^*(h_1) < \infty, J^*(h_2) < \infty$, then h_1 and h_2 can be expressed as Fourier series

$h_1 = \sum_v h_{1,v}\zeta_v$ and $h_2 = \sum_v h_{2,v}\zeta_v$. Let

$$U_i = \zeta_v(\boldsymbol{X}_i)\zeta_u(\boldsymbol{X}_i)e^{-g_0(\boldsymbol{X}_i)} - \int_{\mathcal{X}} \zeta_v(\boldsymbol{x})\zeta_u(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}.$$

Note that U_i are i.i.d. random variables with $\mathrm{E}(U_i) = 0$. Then under Condition 3.5, we have

$$E\left(\frac{1}{n}\sum_{i=1}^n U_i\right)^2 = \frac{1}{n}\mathrm{Var}\left(\zeta_v(\boldsymbol{X}_1)\zeta_u(\boldsymbol{X}_1)e^{-g_0(\boldsymbol{X}_1)}\right) < \frac{c_4}{n}.$$

Furthermore,

$$\begin{aligned}
&\Big|\frac{1}{n}\sum_{i=1}^n e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i) - V^*(h_1,h_2)\Big| \\
=&\Big|\sum_v\sum_u h_{1,v}h_{2,u}\frac{1}{n}\sum_{i=1}^n U_i\Big| \\
\le&\left\{\sum_v\sum_u \frac{1}{1+\lambda_1\gamma_v}\frac{1}{1+\lambda_1\gamma_u}\left(\frac{1}{n}\sum_{i=1}^n U_i\right)^2\right\}^{1/2}\left\{\sum_v\sum_u(1+\lambda_1\gamma_v)(1+\lambda_1\gamma_u)h_{1,v}^2h_{2,u}^2\right\}^{1/2} \\
=&O_p\Big(n^{-1/2}\lambda_1^{-1/r}\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2}\Big) \\
=&o_p\Big(\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2}\Big),
\end{aligned}$$

where the second equality holds because of $\sum_v \frac{1}{1+\lambda_1\gamma_v} = O(\lambda_1^{-1/r})$ and the strong law of large numbers. □

Lemma 3.3 *Under Conditions 3.1, 3.2 and 3.5, suppose h_1 and h_2 are functions satisfying $V^*(h_1) < \infty, V^*(h_2) < \infty$, as $\lambda_1 \to 0$ and $n\lambda_1^{2/r} \to \infty$, one has*

$$\begin{aligned}
&\Big|\frac{1}{n}\sum_{i=1}^n e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i) - \frac{1}{n}\sum_{i=1}^n c_0^{(i)}e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)\Big| \\
\le_p& c_0\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2}, \qquad (3.11)
\end{aligned}$$

where $c_0 = max\{|c_2 - 1|, |c_3 - 1|\}$.

Proof: Note that for each $\boldsymbol{X}_i$,

$$\begin{aligned}
&\mathrm{E}|e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| = \int_{\mathcal{X}} |h_1(\boldsymbol{x})h_2(\boldsymbol{x})|\rho(\boldsymbol{x})d\boldsymbol{x} \\
\leq &\Big\{ \big(\int_{\mathcal{X}} h_1^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} \big) \big(\int_{\mathcal{X}} h_2^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} \big) \Big\}^{1/2} \\
= &\{V^*(h_1)V^*(h_2)\}^{1/2} \leq \{(V^* + \lambda_1 J^*)(h_1)(V^* + \lambda_1 J^*)(h_2)\}^{1/2},
\end{aligned}$$

where the first inequality follows Cauchy-Schwartz inequality. Since $\boldsymbol{X}_i$'s are independent, we have $|e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)|$'s are i.i.d. random variables with mean $\mathrm{E}|e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \leq \big\{V^*(h_1)V^*(h_2)\big\}^{1/2} < \infty$. Therefore, by the strong law of large numbers, $\frac{1}{n}\sum_{i=1}^{n} |e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \overset{a.s.}{\to} \mathrm{E}|e^{-g_0(\boldsymbol{X}_1)}h_1(\boldsymbol{X}_1)h_2(\boldsymbol{X}_1)|$ as $n \to \infty$. Then, we have

$$\begin{aligned}
&\Big|\frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i) - \frac{1}{n}\sum_{i=1}^{n} c_0^{(i)} e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)\Big| \\
= &\Big|\frac{1}{n}\sum_{i=1}^{n} (1 - c_0^{(i)}) e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)\Big| \\
\leq &\frac{1}{n}\sum_{i=1}^{n} |(1 - c_0^{(i)})||e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \\
\leq &\frac{c_0}{n}\sum_{i=1}^{n} |e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \\
\overset{a.s.}{\to} &c_0 \mathrm{E}|e^{-g_0(\boldsymbol{X}_1)}h_1(\boldsymbol{X}_1)h_2(\boldsymbol{X}_1)| \qquad\qquad (\lambda_1 \to 0, n\lambda_1^{2/r} \to \infty) \\
\leq &c_0\{(V^* + \lambda_1 J^*)(h_1)(V^* + \lambda_1 J^*)(h_2)\}^{1/2}.
\end{aligned}$$

□

Conditions 3.1-3.5 are common assumptions for convergence rate analysis of the SS ANOVA estimates, which were also made in Gu [12]. Condition 3.2 states that the growth

rate of the eigenvalues γ_v is at v^r, which controls how fast λ_1 approaches zero. Condition 3.4 bounds e^{g_0-g} at g in a convex set B_0 around g_0. Condition 3.5 requires bounded fourth moment of ζ_v. Now, we introduce the main theorems for the convergence rate.

Theorem 3.2 *Assume* $\sum_v \gamma_v^l a_{v,0}^2 < \infty$ *for some* $l \in [1,2]$. *Under Conditions 3.1-3.5, suppose* $V^*(\hat{g}-\tilde{g}) < \infty$, *for some* $r > 1$, *as* $\lambda_1 \to 0$ *and* $n\lambda_1^{2/r} \to \infty$,

$$(V^* + \lambda_1 J^*)(\hat{g} - g_0) = O_p(n^{-1}\lambda_1^{-1/r} + \lambda_1^l).$$

Proof: Note that for each $\boldsymbol{X}_i$, $\mathrm{E}\{e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_i)\} = \int_{\mathcal{X}}(\hat{g}-\tilde{g})^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x} = V^*(\hat{g}-\tilde{g}) < \infty$. Since $\boldsymbol{X}_i$'s are independent, we have $e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_i)$'s are i.i.d. random variables with mean $\mathrm{E}\{e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_i)\} = V^*(\hat{g}-\tilde{g}) < \infty$. Therefore, by the strong law of large numbers, $\frac{1}{n}\sum_{i=1}^n e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_i) \xrightarrow{a.s.} \mathrm{E}\{e^{-g_0(\boldsymbol{X}_1)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_1)\}$ as $n \to \infty$. Substituting (3.8) into the left-hand side of (3.7), we have

$$\begin{aligned}
&\lambda_1 J^*(\hat{g}-\tilde{g}) - \frac{1}{n}\sum_{i=1}^n \left\{e^{-\hat{g}(\boldsymbol{X}_i)} - e^{-\tilde{g}(\boldsymbol{X}_i)}\right\}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) \\
\geq &\frac{c_2}{n}\sum_{i=1}^n e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_i) + \lambda_1 J^*(\hat{g}-\tilde{g}) \\
\xrightarrow{a.s.} &c_2\mathrm{E}\{e^{-g_0(\boldsymbol{X}_1)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_1)\} + \lambda_1 J^*(\hat{g}-\tilde{g}) \qquad (\lambda_1 \to 0, n\lambda_1^{2/r} \to \infty) \\
= &c_2 V^*(\hat{g}-\tilde{g}) + \lambda_1 J^*(\hat{g}-\tilde{g}). \qquad (3.12)
\end{aligned}$$

Substituting (3.10) and (3.11) into the right-hand side of (3.7) and let $h_1 = \hat{g} - \tilde{g}$,

$h_2 = \tilde{g} - g_0$, as $\lambda_1 \to 0$ and $n\lambda_1^{2/r} \to \infty$, we have

$$\begin{aligned}
&\Big|\frac{1}{n}\sum_{i=1}^{n}\left\{e^{-\tilde{g}(\boldsymbol{X}_i)} - e^{-g_0(\boldsymbol{X}_i)}\right\}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) + V^*(\hat{g}-\tilde{g}, \tilde{g}-g_0)\Big| \\
\leq &\Big|V^*(\hat{g}-\tilde{g}, \tilde{g}-g_0) - \frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)(\tilde{g}-g_0)(\boldsymbol{X}_i)\Big| \\
&+ \Big|\frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)(\tilde{g}-g_0)(\boldsymbol{X}_i) - \frac{1}{n}\sum_{i=1}^{n} c_0^{(i)} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)(\tilde{g}-g_0)(\boldsymbol{X}_i)\Big| \\
\leq_p &(o_p(1)+c_0)\{(V^*+\lambda_1 J^*)(\hat{g}-\tilde{g})(V^*+\lambda_1 J^*)(\tilde{g}-g_0)\}^{1/2}, \qquad (3.13)
\end{aligned}$$

where the first inequality follows (3.9) and the second inequality directly follows Lemma 3.2 and 3.3. Combining (3.7), (3.12), and (3.13), we obtain

$$(c_2 V^* + \lambda_1 J^*)(\hat{g}-\tilde{g}) \leq_p (o_p(1)+c_0)\{(V^*+\lambda_1 J^*)(\hat{g}-\tilde{g})(V^*+\lambda_1 J^*)(\tilde{g}-g_0)\}^{1/2}. \qquad (3.14)$$

Combining (3.14) with Lemma 3.1, as $\lambda_1 \to 0$ and $n\lambda_1^{2/r} \to \infty$, we have $(V^* + \lambda_1 J^*)(\hat{g} - \tilde{g}) = O_p(n^{-1}\lambda_1^{-1/r} + \lambda_1^l)$ and Theorem 3.2 holds. □

Theorem 3.3 *Under the conditions in Theorem 3.2,*

$$(V + \lambda_1 J)(\hat{g} - g_0) = O_p(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}).$$

Proof: We know

$$\sum_{1\leq j<k\leq p} ||g_{jk}(x_j, x_k)||^2 \leq \left\{\sum_{1\leq j<k\leq p} ||g_{jk}(x_j, x_k)||\right\}^2 \leq \frac{(p-1)p}{2}\sum_{1\leq j<k\leq p} ||g_{\alpha k}(x_\alpha, x_k)||^2. \qquad (3.15)$$

Then, there exists some constant $C \in [1, \sqrt{\frac{(p-1)p}{2}}]$ such that $C\{\sum_{1\le j<k\le p}||g_{jk}(x_j,x_k)||^2\}^{1/2} = \sum_{1\le j<k\le p}||g_{jk}(x_j,x_k)||$. Since $\sum_{1\le j<k\le p}\theta_{jk}$ is bounded by a fixed $M<\infty$, we can scale λ_1,λ_2 such that $\theta_{jk}\le 1$. Since $J_2^*(g) = \sum_{1\le j<k\le p}\theta_{jk}^{-1}||g_{jk}(x_j,x_k)||^2 = \boldsymbol{c}^T(\sum_{1\le j<k\le p}\theta_{jk}Q_{jk})\boldsymbol{c}$, $\sum_{1\le j<k\le p}||g_{jk}(x_j,x_k)||^2 = \boldsymbol{c}^T(\sum_{1\le j<k\le p}\theta_{jk}^2Q_{jk})\boldsymbol{c}$, we have $J_2^2(g) = C^2\sum_{1\le j<k\le p}||g_{jk}(x_j,x_k)||^2 \le C^2J_2^*(g)$ and consequently $J_2\le C(J^*)^{1/2}$.

Furthermore, since $V_2^2(g^{(2)}) = \int_{\mathcal{X}}\left\{g^{(2)}(\boldsymbol{x})\right\}^2\rho(\boldsymbol{x})d\boldsymbol{x} = V^*(g^{(2)})$, we have $V_2(g^{(2)}) = [V^*(g^{(2)})]^{1/2}$. Therefore,

$$(V_2+\lambda_1J_2)(g^{(2)}) = ((V^*)^{1/2}+C\sqrt{\lambda_1}(\lambda_1J^*)^{1/2})(g^{(2)}) \le (1+C^2\lambda_1)^{1/2}(V^*+\lambda_1J^*)^{1/2}(g^{(2)})$$

by the Cauchy-Schwarz inequality. Finally,

$$\begin{aligned}(V+\lambda_1J)(\hat{g}-g^0) &= (V_1+\lambda_1J_1)(\hat{g}^{(1)}-g_0^{(1)}) + (V_2+\lambda_1J_2)(\hat{g}^{(2)}-g_0^{(2)})\\ &\le (V^*+\lambda_1J^*)(\hat{g}^{(1)}-g_0^{(1)}) + (1+C^2\lambda_1)^{1/2}(V^*+\lambda_1J^*)^{1/2}(\hat{g}^{(2)}-g_0^{(2)})\\ &= O_p(n^{-1}\lambda_1^{-1/r}+\lambda_1^l) + O(n^{-1/2}\lambda_1^{-1/2r}+\lambda_1^{l/2})\\ &= O_p(n^{-1/2}\lambda_1^{-1/2r}+\lambda_1^{l/2}). \qquad (3.16)\end{aligned}$$

□

Corollary 3.1 *Assume conditions in Theorem 3.3 hold,* $0<c_5<\rho(\boldsymbol{x})<c_6$ *for some positive constants* c_5,c_6*, we have*

$$||\hat{g}_{jk}-g_{0jk}|| = O_p(n^{-1/2}\lambda_1^{-1/2r}+\lambda_1^{l/2}), \quad 1\le j<k\le p,$$

where g_{0jk} *are two-way interactions in the SS ANOVA decomposition of* g_0*.*

Proof: By definition of $V(\cdot)$, $V(\hat{g}-g^0) = V_1(\hat{g}^{(1)}-g_0^{(1)}) + V_2(\hat{g}^{(2)}-g_0^{(2)}) = V^*(\hat{g}^{(1)}-$

$g_0^{(1)}) + [V^*(\hat{g}^{(2)} - g_0^{(2)})]^{1/2}$. Following (3.16),

$$[V^*(\hat{g}^{(2)} - g_0^{(2)})]^{1/2} = O_p(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}).$$

Following Lin et al. [28], under the condition $0 < c_5 < \rho(\boldsymbol{x}) < c_6$ for some positive constants c_5, c_6, $[V^*(g)]^{1/2}$ is equivalent to the L_2 norm. Specifically,

$$V^*(g) \sim ||g||^2 = \sum_{j=1}^{p} ||g_j||^2 + \sum_{1\le j<k\le p} ||g_{jk}(x_j, x_k)||^2,$$

where $\sim$ means equivalence, $||\cdot||$ is the L_2 norm, and $V^*(g^{(1)}) \sim \sum_{j=1}^{p} ||g_j||^2$, $V^*(g^{(2)}) \sim \sum_{1\le j<k\le p} ||g_{jk}(x_j, x_k)||^2$, respectively. By definition, $V(g^{(2)}) = [V^*(g^{(2)})]^{1/2}$ $\sim (\sum_{1\le j<k\le p} ||g_{jk}(x_j, x_k)||^2)^{1/2}$. Consequently, two-way interactions under L_2 norm have the same convergence rate as $[V^*(g^{(2)})]^{1/2}$,

$$||\hat{g}_{jk} - g_{0jk}|| = O_p(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}), \quad 1 \le j < k \le p.$$

□

The convergence rate in Theorem 3.3 is the square root of the rate in Theorem 3.2. This is because $V^* + \lambda_1 J^*$ is associated with the square of L_2 norm while the L_2 norm was used in $V + \lambda_1 J$. Corollary 3.1 holds because V_2 and J_2 associated with two-way interactions are equivalent to L_2 norm. Consequently, two-way interactions under L_2 norm have the same convergence rate as Theorem 3.3. We only show convergence rate for interactions in Corollary 3.1 since we are mainly interested in edge selection.

Chapter 4

Simulation Studies

In this chapter, we evaluate the performance of our method (referred to as NEW) in various settings, and compare it with Jeon and Lin [21]'s method (referred to as OLD) for edge detection. In the implementation of our method, we estimate the joint density with each variable on the data range and transform the data into $[0, 1]$. We construct an SS ANOVA model using tensor product of cubic spline models. Specifically, let $\mathcal{H}^{(j)} = W_2^2[0, 1]$ where

$$W_2^2[0,1] = \left\{ f : f, f' \text{ are absolutely continuous}, \int_0^1 (f'')^2 dx < \infty \right\} \tag{4.1}$$

is the Sobolev space for cubic spline models. Each $\mathcal{H}^{(j)}$ can be decomposed as $\mathcal{H}^{(j)} = \{1_{(j)}\} \oplus \mathcal{H}_{(j)}$ and $\mathcal{H}_{(j)} = \mathcal{H}^0_{(j)} \oplus \mathcal{H}^1_{(j)}$ where $\mathcal{H}^0_{(j)}$ and $\mathcal{H}^1_{(j)}$ are RKHS's with RKs $R^0_j(x, z) = k_1(x)k_1(z)$ and $R^1_j(x, z) = k_2(x)k_2(z) - k_4(|x - z|)$ respectively, $k_1(x) = x - 0.5$, $k_2(x) = \frac{1}{2}(k_1^2(x) - \frac{1}{12})$, and $k_4(x) = \frac{1}{24}(k_1^4(x) - \frac{k_1^2(x)}{2} + \frac{7}{240})$. SS ANOVA decomposition of $\bigotimes_{j=1}^{p} \mathcal{H}^{(j)}$ can then be constructed based on these decompositions. More details can be found in Wang [41]. In all simulations and real data applications, we select the tuning parameter M using the 5-fold cross-validation method as we described in Section 2.3.2. For the fair

comparison between our method and Jeon and Lin's method, we use the same subset of representers $\{\tilde{\boldsymbol{x}}_u = (\tilde{x}_{u,1}, \cdots, \tilde{x}_{u,p}),\ u = 1, \cdots, q\}$ in the fitting procedure with $q = \max\{40, 12n^{2/9}\}$, which is the default value used in `ssden1` function in the `gss` package.

For edge detection, we look into four different numerical experiments on domains of dimensions three or five. We simulate data with $n = 200, 400$ and 600 for each case. The simulation is repeated 100 times under each simulation setting. We record the frequencies of appearance of each two-way interaction term in 100 runs. Also, to evaluate the performance of edge detection, we compute three criteria: specificity (SPE), sensitivity (SEN), and F_1 scores, which are defined as follows:

$$\text{SPE} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \qquad \text{SEN} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \qquad \text{F}_1 = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}},$$

where TP, TN, FP and FN are the numbers of true positives, true negatives, false positives and false negatives.

To further compare overall performance, we create a search grid of tuning parameter M and plot the average of the ROC curves under different M for two methods. Specifically, the search grid contains 25 different values: nine values from 10 to 90 equally, ten values between 100 and 1000 equally, four values from 1100 to 1700 equally, and $2500, 5000$. We fix the sample size as $n = 600$ and run 100 simulations in total. For each simulation, we record the true positive rate (TPR) and false positive rate (FPR) for each M. Then, we take the average of TPRs and FPRs of 100 simulations for each M and plot the average of the ROC curves.

Both Gaussian and non-Gaussian settings are considered in the simulation. A trivariate simulation is studied in Section 4.1, which was an example used in Gu et al. [13]. Section 4.2 is a 5-dimensional multivariate Gaussian distribution with specified mean and covariance matrix. We study a 5-dimensional skewed Gaussian distribution (Azzalini and

Valle [3]) in Section 4.3. Lastly, a mixture model is studied in Section 4.4, and its setting was also used in Gu et al. [13].

4.1 Trivariate Simulation on $[0,1]^3$

In this section, we consider a three variables simulation. Samples are taken from

$$f_3(x_1, x_2, x_3) \propto f_1(x_1 - 0.3x_3 + 0.1) f_1(x_2 - 0.2x_3 + 0.1) e^{-12.5(x_3 - 0.5)^2}, \tag{4.2}$$

where $f_1(x) \propto e^{-50(y-0.3)^2} + 2e^{-50(y-0.7)^2}$ is the $1:2$ mixture of $\mathcal{N}(0.3, 0.1^2)$ and $\mathcal{N}(0.7, 0.1^2)$. The joint observation (X_1, X_2, X_3) is truncated to $\mathcal{X} = [0,1]^3$. Note that $X_1 \perp X_2 | X_3$. Then, the correct model has log density of form $g(x_1, x_2, x_3) = g_1 + g_2 + g_3 + g_{1,3} + g_{2,3}$. The true model has edges in (X_1, X_3), and (X_2, X_3).

Table 4.1 presents the frequencies of appearance of the two-way interactions. The numbers in the Interactions row represent the corresponding edges between variables, for example, $(1,2)$ represents the interaction between X_1 and X_2. In the Ground Truth row, 1s denote the presence of a two-way interaction while 0s denote the absence of a two-way interaction. We notice that Jeon and Lin's method missed the existing edge between X_2 and X_3 quite often. Table 4.2 displays the averages and standard deviations of SPE, SEN, and F_1 score for two methods. Both methods have small SPEs. As the sample size increases, our method can better specify the false edge but also missed the edge between X_2 and X_3 in some cases. Overall, our method performs much better in terms of the sensitivity and F_1 score.

We also plot the average of the ROC curves for both methods in Figure 4.1. Overall, the average of the ROC curves of our method is above the average of Jeon and Lin's method, which indicates that our method has better overall performance in edge

detection.

	NEW			OLD		
Edge Set	(1,2)	(1,3)	(2,3)	(1,2)	(1,3)	(2,3)
Ground Truth	0	1	1	0	1	1
n=200	43	99	96	45	92	68
n=400	56	99	99	45	98	51
n=600	43	99	96	43	97	63

Table 4.1: The frequencies of selected edges in 100 runs.

	NEW			OLD		
	SPE	SEN	F_1	SPE	SEN	F_1
n=200	0.570	0.975	0.898	0.550	0.800	0.771
	(0.498)	(0.110)	(0.120)	(0.500)	(0.275)	(0.216)
n=400	0.440	0.990	0.881	0.550	0.745	0.741
	(0.499)	(0.070)	(0.103)	(0.500)	(0.271)	(0.213)
n=600	0.690	0.885	0.861	0.570	0.800	0.784
	(0.465)	(0.211)	(0.138)	(0.498)	(0.256)	(0.205)

Table 4.2: Averages and standard deviations (in parentheses) of specificity (SPE), sensitivity (SEN), and F_1 score for trivariate simulation.

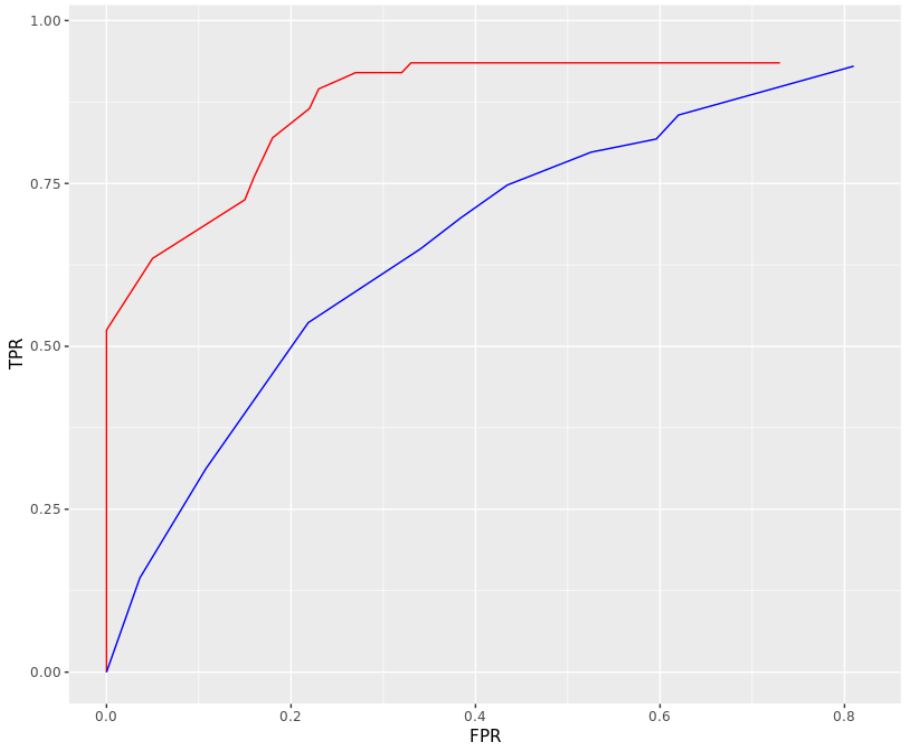

Figure 4.1: Averages of the ROC curves from our method (red) and Jeon and Lin's method (blue).

4.2 Multivariate Gaussian Distribution

In this section, we consider a 5-dimensional multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu} = (0.5, 0.5, 0.5, 0.5, 0.5)^T$ and

$$\Sigma^{-1} = \begin{pmatrix} 62 & -20 & 0 & 0 & -20 \\ -20 & 62 & -10 & 0 & 0 \\ 0 & -10 & 62 & 10 & 0 \\ 0 & 0 & 10 & 62 & -15 \\ -20 & 0 & 0 & -15 & 62 \end{pmatrix}.$$

Note that the edge set $E = \{(1,2),(1,5),(2,3),(3,4),(4,5)\}$, and the correct model has log density of form $g(x_1,x_2,x_3,x_4,x_5) = g_1+g_2+g_3+g_4+g_5+g_{1,2}+g_{1,5}+g_{2,3}+g_{3,4}+g_{4,5}$. Table 4.3 presents the frequencies of selected edges in 100 runs. Table 4.4 displays the averages and standard deviations of SPE, SEN and F_1 score for two methods. Based on these two tables, our method has better SPE, F_1 score for all three different sample sizes and has better SEN for $n = 200, 400$. In Figure 4.2, the average of the ROC curves of our method is above the average of Jeon and Lin's method, which can indicate that our method has better overall performance in edge detection.

	NEW										OLD									
Edge Set	12	13	14	15	23	24	25	34	35	45	12	13	14	15	23	24	25	34	35	45
Ground Truth	1	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0	0	1	0	1
n=200	100	14	8	97	49	7	7	53	9	89	98	17	24	91	52	28	23	56	23	74
n=400	100	0	1	100	57	3	4	53	2	97	93	22	25	96	56	23	18	68	20	87
n=600	100	4	2	100	60	1	5	58	4	96	98	23	27	99	70	26	24	72	29	93

Table 4.3: The frequencies of selected edges in 100 runs.

	NEW			OLD		
	SPE	SEN	F_1	SPE	SEN	F_1
n=200	0.910	0.776	0.826	0.770	0.742	0.748
	(0.176)	(0.174)	(0.116)	(0.202)	(0.191)	(0.141)
n=400	0.980	0.814	0.879	0.784	0.800	0.789
	(0.060)	(0.164)	(0.101)	(0.179)	(0.178)	(0.134)
n=600	0.968	0.828	0.884	0.742	0.864	0.814
	(0.123)	(0.158)	(0.102)	(0.195)	(0.161)	(0.126)

Table 4.4: Averages and standard deviations (in parentheses) of specificity (SPE), sensitivity (SEN), and F_1 score for multivariate Gaussian simulation.

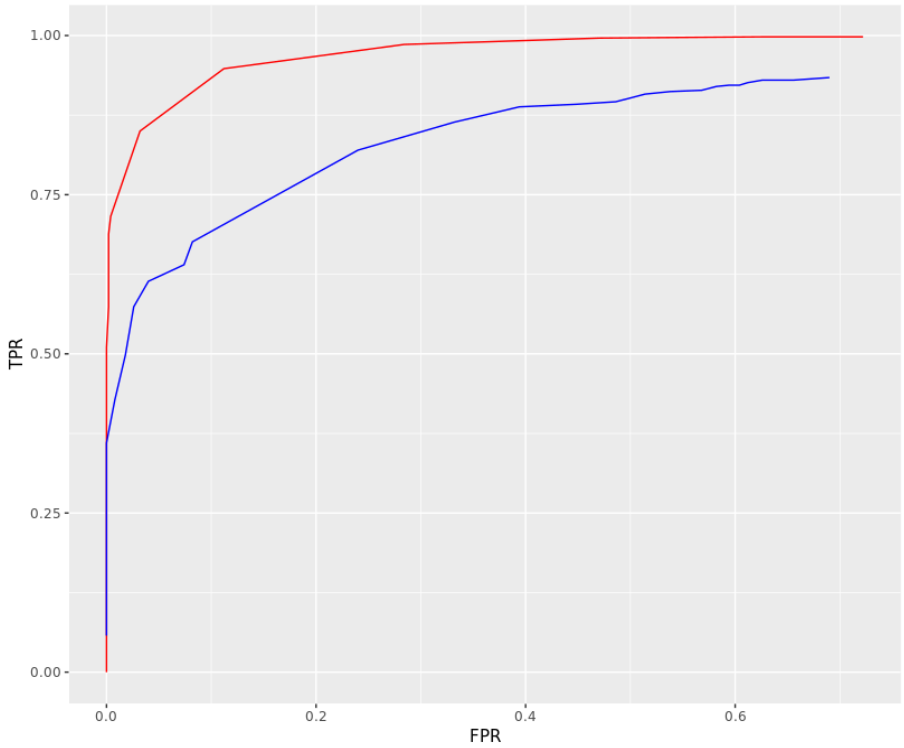

Figure 4.2: Averages of the ROC curves from our method (red) and Jeon and Lin's method (blue).

4.3 Multivariate Skewed Gaussian Distribution

In this section, we consider the simulation setting when $\boldsymbol{X}$ follows a multivariate skewed Gaussian distribution with density function (Azzalini and Valle [3])

$$f(\boldsymbol{x}) = 2\phi_p(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)\Phi(\boldsymbol{\alpha}^T\boldsymbol{x}),$$

where $\phi_p(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma)$ is the p-dimensional normal density with mean $\boldsymbol{\mu}$ and covariance matrix Σ, $\Phi(\cdot)$ is the CDF of the standard Gaussian distribution, and $\boldsymbol{\alpha}$ is a p-dimensional vector that controls the skewness of the multivariate Gaussian distribution. When $\boldsymbol{\alpha} = \mathbf{0}$, the distribution reduces to the multivariate Gaussian distribution. We set

$\boldsymbol{\mu} = (0.5, 0.5, 0.5, 0.5, 0.5)^T$ and

$$\Sigma^{-1} = \begin{pmatrix} 62 & -30 & 0 & 0 & -30 \\ -30 & 62 & -15 & 0 & 0 \\ 0 & -15 & 62 & 13 & 0 \\ 0 & 0 & 13 & 62 & -19 \\ -30 & 0 & 0 & -19 & 62 \end{pmatrix}.$$

Note that the correct model has the same form as the one in Section 4.2 and has the edge set $E = \{(1,2), (1,5), (2,3), (3,4), (4,5)\}$.

Table 4.5 presents the frequencies of selected edges in 100 runs. Table 4.6 displays the averages and standard deviations of SPE, SEN, and F_1 score for two methods. From these two tables, we can see that our method has better SPE, F_1 for all three different sample sizes, and has better SEN for $n = 200, 600$. In Figure 4.3, the average of the ROC curves of our method is above the average of Jeon and Lin's method. Two tables and Figure 4.3 can indicate that our method has better overall performance in edge detection.

	NEW										OLD									
Edge Set	12	13	14	15	23	24	25	34	35	45	12	13	14	15	23	24	25	34	35	45
Ground Truth	1	0	0	1	1	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1
n=200	98	10	9	100	20	30	11	98	32	47	89	26	25	91	41	36	24	79	40	54
n=400	100	11	9	100	24	19	10	99	28	66	94	31	29	98	55	31	29	85	37	77
n=600	100	12	10	100	31	28	15	100	26	75	96	25	25	98	43	34	26	84	33	70

Table 4.5: The frequencies of selected edges in 100 runs.

	NEW			OLD		
	SPE	SEN	F_1	SPE	SEN	F_1
n=200	0.816	0.726	0.760	0.698	0.708	0.690
	(0.192)	(0.135)	(0.099)	(0.200)	(0.240)	(0.177)
n=400	0.846	0.778	0.806	0.686	0.818	0.765
	(0.192)	(0.133)	(0.099)	(0.213)	(0.168)	(0.123)
n=600	0.818	0.812	0.815	0.714	0.782	0.752
	(0.205)	(0.142)	(0.092)	(0.228)	(0.191)	(0.138)

Table 4.6: Averages and standard deviations (in parentheses) of specificity (SPE), sensitivity (SEN), and F_1 score for multivariate skewed Gaussian simulation.

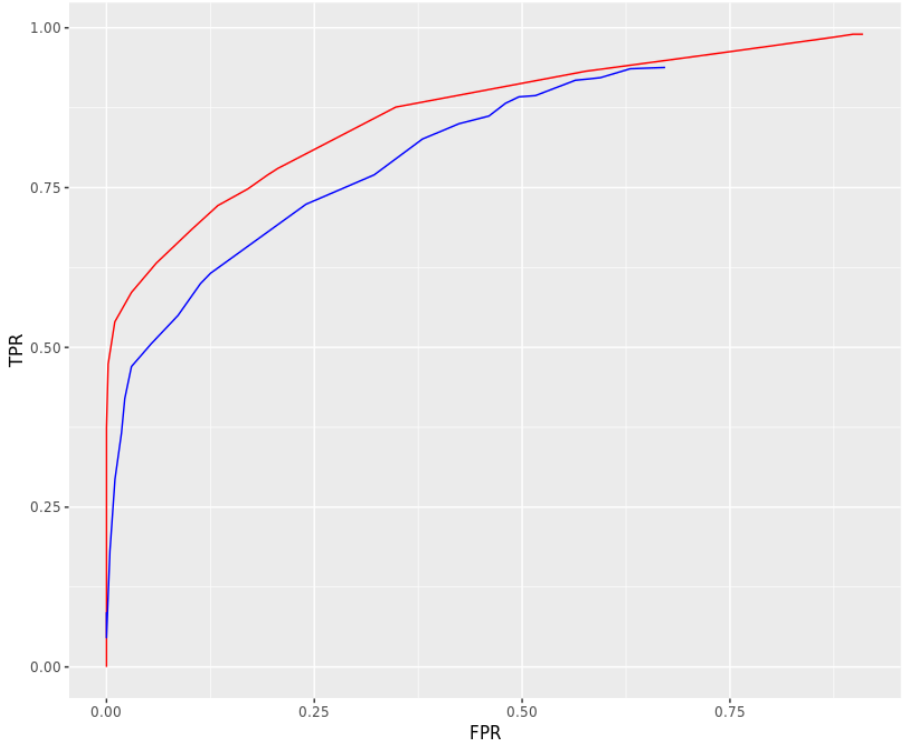

Figure 4.3: Averages of the ROC curves from our method (red) and Jeon and Lin's method (blue).

4.4 Mixture Model Simulation

In this section, we conduct another 5-dimensional simulation from mixture distributions. We consider $(X_2, X_3, X_4)^T \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu} = (0.5, 0.5, 0.5)^T$ and

$$\Sigma^{-1} = \begin{pmatrix} 62 & -15 & 0 \\ -15 & 62 & -30 \\ 0 & -30 & 62 \end{pmatrix},$$

$X_1 = Y_1 - 0.4X_2 - 0.1$, and $X_5 = Y_2 + 0.3X_4 - 0.1$, where Y_1 and Y_2 are independently generated from f_1 in Section 4.1. Note that the edge set $E = \{(1,2),(2,3),(3,4),(4,5)\}$, and the correct model has log density of form $g(x_1, x_2, x_3, x_4, x_5) = g_1 + g_2 + g_3 + g_4 + g_5 + g_{1,2} + g_{2,3} + g_{3,4} + g_{4,5}$.

Table 4.7 presents the frequencies of selected edges in 100 runs. Table 4.8 displays the averages and standard deviations of SPE, SEN, and F_1 score for two methods. It is clear that our method has better performance in SPE and F_1 score and outperforms Jeon and Lin's method in SEN when $n = 400, 600$. In Figure 4.4, the average of the ROC curves of our method is above the average of Jeon and Lin's method, which can also indicate that our method has better overall performance in edge detection.

	NEW										OLD									
Edge Set	12	13	14	15	23	24	25	34	35	45	12	13	14	15	23	24	25	34	35	45
Ground Truth	1	0	0	0	1	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1
n=200	75	13	14	13	63	5	10	100	11	57	77	20	26	33	67	11	26	98	24	63
n=400	86	9	7	7	87	3	6	100	8	74	84	29	31	43	81	18	38	100	41	74
n=600	98	4	2	3	91	2	2	100	4	73	95	33	32	32	75	16	31	100	38	76

Table 4.7: The frequencies of selected edges in 100 runs.

	NEW			OLD		
	SPE	SEN	F_1	SPE	SEN	F_1
n=200	0.890	0.738	0.768	0.767	0.762	0.718
	(0.184)	(0.217)	(0.150)	(0.180)	(0.217)	(0.167)
n=400	0.933	0.868	0.879	0.667	0.848	0.726
	(0.144)	(0.179)	(0.128)	(0.212)	(0.177)	(0.138)
n=600	0.972	0.905	0.925	0.697	0.865	0.749
	(0.089)	(0.154)	(0.108)	(0.194)	(0.172)	(0.132)

Table 4.8: Averages and standard deviations (in parentheses) of specificity (SPE), sensitivity (SEN), and F_1 score for mixture simulation.

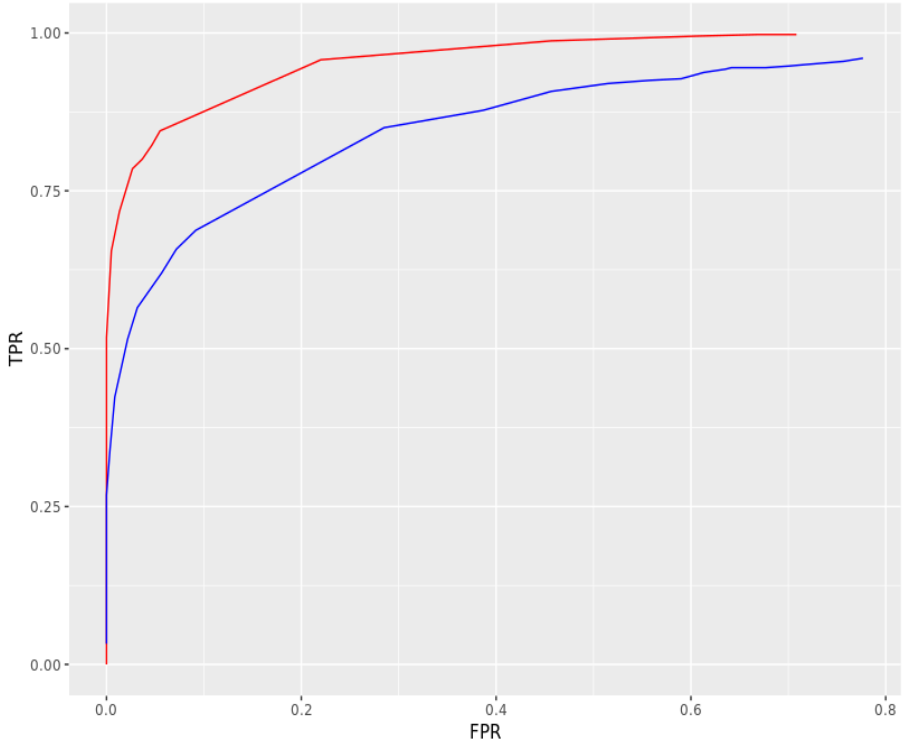

Figure 4.4: Averages of the ROC curves from our method (red) and Jeon and Lin's method (blue).

4.5 Discussions

Overall, our method performs well in specificity and F_1 score and can sometimes outperform Jeon and Lin's method in sensitivity. In general, Jeon and Lin's method

selects more false edges and misses true edges quite often. Based on four averages of the ROC curve plots, our method's curves are above those curves of Jeon and Lin's method, which indicates our method has a better overall performance in edge detection.

Chapter 5

Real Data Examples

In this chapter, we consider three different data sets: air pollution data, transcription factor association, and cellular signaling networks. We plot the estimated graph and compare the estimated graph structures with those from Jeon and Lin's method. To visualize the difference, we also add the symmetric difference plots. We use red and blue edges to denote those edges selected by our method only and by Jeon and Lin's method only.

5.1 Air Pollution and Road Traffic

We first investigate the data set used in Jeon and Lin [21], which studied the relationship between air pollution on a road, traffic volume, and meteorological variables. It contains a subset of 500 observations from Alnabru in Oslo, Norway, between October 2001 and August 2003. This data set can be found in `gss` package as a data frame `NO2`, which contains six variables: `no2` (the concentration of N02), `cars` (the number of cars per hour), `temp` (temperature 2 meter above ground), `wind` (wind speed), `temp2` (temperature difference between 25 and 2 meters above ground), and `wind2` (wind direction).

Figure 5.1 shows the estimated graph from our method and Jeon and Lin's method and the symmetric difference between them. These two methods both select edges between `no2 - cars, cars - temp, cars - temp2, cars - wind, temp - wind2`.

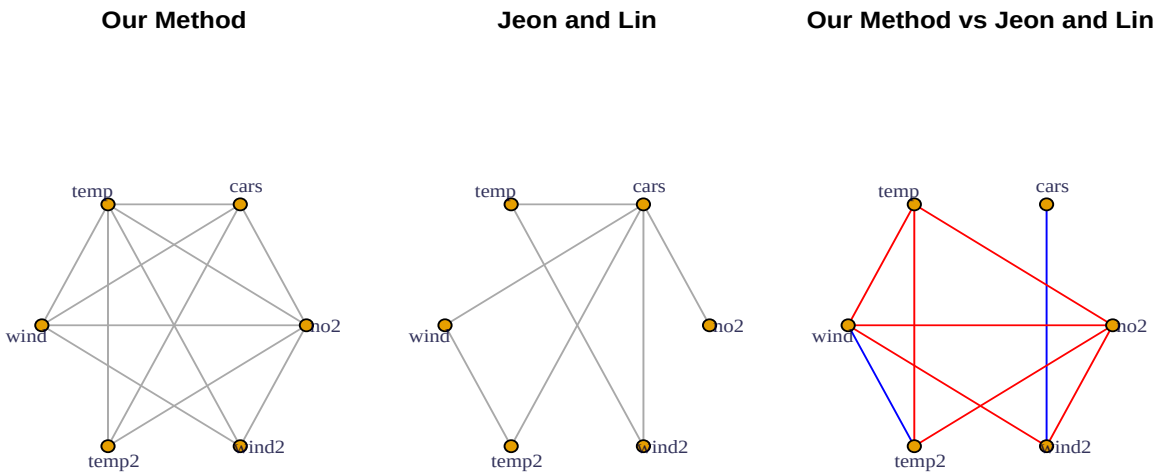

Figure 5.1: The estimated graph for air pollution data and symmetric difference plot between two methods. Red and blue edges are selected by our method only and by Jeon and Lin's method only.

In the symmetric difference plot, red edges are selected by our method only. Our method gets a denser graph. Jeon and Lin's method misses some connections related to `temp2`, `no2` and `temp`. The additional edges detected by our method may indicate that air pollution has a more complicated relationship with traffic volume and meteorological variables.

5.2 Transcription Factor Association

Transcription factors are significant in the research of gene expression. Ouyang et al. [31] studied 12 transcription factors on 18936 genes and this data set can be download from `https://www.pnas.org/doi/10.1073/pnas.0904863106`. They found that a remarkably high proportion of variation in gene expression (65%) can be explained by the binding signals of these 12 transcription factors. In their paper, they identified two groups of transcription factors. The first group (E2f1, Myc, Mycn, and Zfx) played the role of activators in general. The second group (Klf4, Oct4, Nanog, Sox2, Smad1, Stat3, Tcfcp2l1, and Esrrb) might act as either activators or repressors depending on the target. Their results showed that these two groups might cooperate tightly to activate genes.

Figure 5.2 shows the estimated graph from our method and Jeon and Lin's method and the symmetric difference between them. We notice that Jeon and Lin's method finds two connections from two different groups but misses the relationship within either the first or second group. Our method identifies connections within both the first group and the second group. Also, connections between two groups are detected by our method. Overall, our method finds interrelationships in two groups and their connections, which may provide new interpretations in the study of gene expression.

5.3 Cellular Signaling Networks

Studying causal signaling pathways is an important task in biological field. Measurements of 11 phosphorylated protein and phospholipid components in 7466 individual primary human immune system cells have been studied by Sachs et al. [34] and they used Bayesian network computational methods to report signaling relationships. This data set is included in the `gss` package named `Sachs` where measured molecules are `praf`, `pmek`,

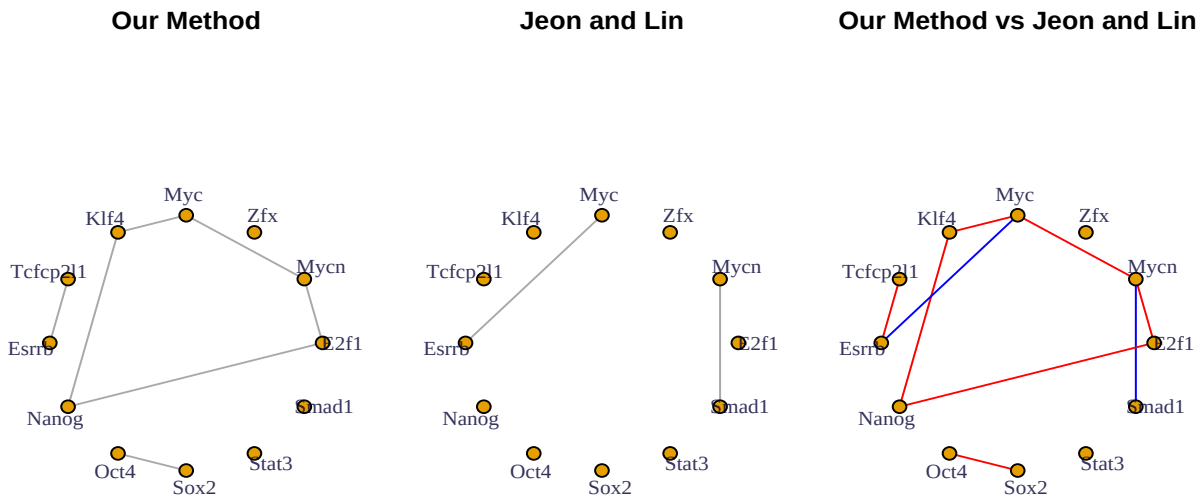

Figure 5.2: The estimated graph for transcription data and symmetric difference plot between two methods. Red and blue edges are selected by our method only and by Jeon and Lin's method only.

`plcg`, `pip2`, `pip3`, `p44.42`, `pakts473`, `pka`, `pkc`, `p38`, and `pjnk`. We use the undirected graph in Figure 5.3 (*left*) to represent the causal relationships found by Sachs et al. [34] and plot estimated graphs using our method (*middle*) and Jeon and Lin's method (*right*) for comparison.

Compared to the graph in Sachs et al. [34], our method and Jeon and Lin's method only select a subset of edges, including two common edges `pip2 - plag` and `p44.42 - pakts473`. As shown in Sachs et al. [34], there are some intermediate molecules to help build connections. Since some of them are not measured in this data set, it is possible that our method and Jeon and Lin's method miss those connections. For all edges identified by our method, only edge `plcg - pakts473` is not in Sachs et al. [34]'s graph, which can indicate our method has good accuracy in edge detection. On the contrary, Jeon and

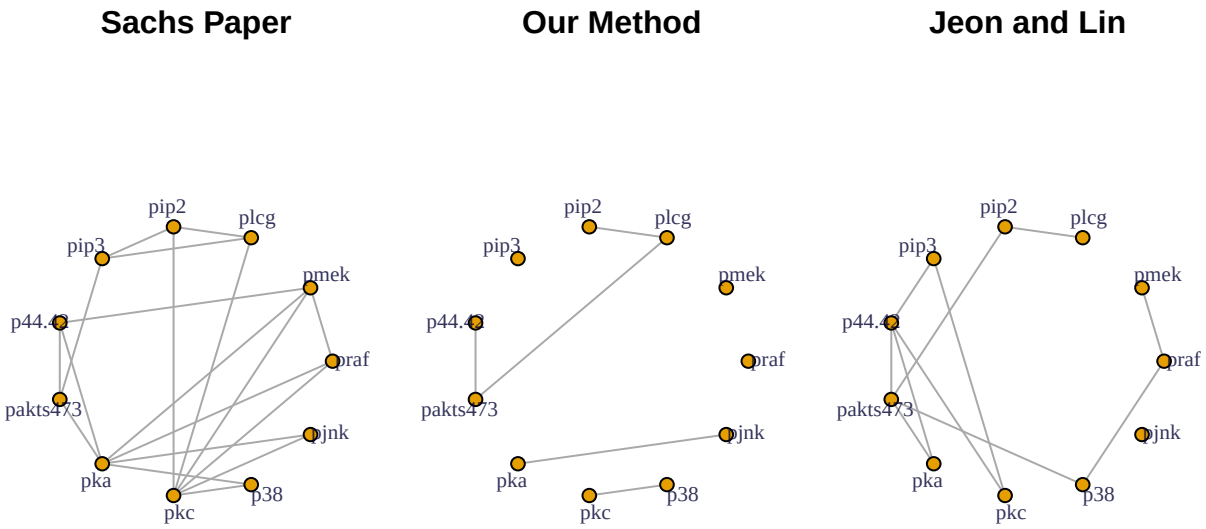

Figure 5.3: The estimated graph for cellular signaling data. The left one is from Sachs et al. [34]. The middle and right plots are from our method and Jeon and Lin's method.

Lin's method identifies some edges that are not reflected in Sachs et al. [34], for example, `pip3 - pkc`, `pakts473 - p38`, and `praf - p38`. Edges detected by our method may show that some molecules have direct connections instead of indirect influences found in Sachs et al. [34], which may provide new directions for researchers to consider causal signaling relationships.

Part 2

Neighborhood Selection Approach

Chapter 6

Neighborhood Selection Through Conditional Density Estimation

6.1 Neighborhood Selection with L_1 Penalty

In this section, we introduce our neighborhood selection method for edge detection. Let $\boldsymbol{X} = (X_1, \cdots, X_p)$ and $\boldsymbol{X}_{\setminus\{i_1,\cdots,i_k\}}$ be the sub-vector of $\boldsymbol{X}$ without elements $\{i_1, \cdots, i_k\}$. Denote $\mathcal{X}_j$ be the domain of X_j, and $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_p$. We are interested in estimating the conditional density $f(x_\alpha|\boldsymbol{x}_{\setminus\{\alpha\}})$ for αth variable given $\boldsymbol{x}_{\setminus\{\alpha\}} = (x_1, \cdots, x_{\alpha-1}, x_{\alpha+1}, \cdots, x_p)$, and consider the logistic density transformation of f as $f(x_\alpha|\boldsymbol{x}_{\setminus\{\alpha\}}) = e^{g(\boldsymbol{x})}/\int_{\mathcal{X}_\alpha} e^{g(\boldsymbol{x})} dx_\alpha$. Denote the model space for g as

$$\mathcal{M}_\alpha = \{1\} \oplus \left\{ \bigoplus_{j=1}^{p} \mathcal{H}_{(j)} \right\} \oplus \left\{ \bigoplus_{k \neq \alpha} [\mathcal{H}_{(\alpha)} \otimes \mathcal{H}_{(k)}] \right\}. \tag{6.1}$$

We further decompose $\mathcal{H}_{(j)}$ as $\mathcal{H}_{(j)} = \mathcal{H}^0_{(j)} \oplus \mathcal{H}^1_{(j)}$, where $\mathcal{H}^0_{(j)}$ is a finite dimensional space containing functions that are not subject to penalty. A function $g \in \mathcal{M}_\alpha$ can be

decomposed as follows:

$$g(\boldsymbol{x}) = \varsigma + \sum_{j=1}^{p} g_j(x_j) + \sum_{k \neq \alpha} g_{\alpha k}(x_\alpha, x_k), \tag{6.2}$$

where each functional component in (6.2) belongs to the corresponding subspace in (6.1). In general, we remove the constant ς for identifiability.

Note that model (6.2) contains many parametric models as special cases. Specifically, the Gaussian graphical model is a special case with $\mathcal{X}_j = \mathbb{R}$, $g_j(x_j) = \beta_j x_j - x_j^2/2$ for $j = \alpha$ and 0 otherwise, and $g_{\alpha k}(x_\alpha, x_k) = \beta_{\alpha k} x_\alpha x_k$ for some constants β_j and $\beta_{\alpha k}$. The Ising model for binary data is a special case with $\mathcal{X}_j = \{0, 1\}$, $g_j(x_j) = x_j$ for $j = \alpha$ and 0 otherwise, and $g_{\alpha k}(x_\alpha, x_k) = \beta_{\alpha k} x_\alpha x_k$. The Poisson graphical model for discrete data is a special case with $\mathcal{X}_j = \{0, 1, 2, \cdots\}$, $g_j(x_j) = x_j - \log(x_j!)$ for $j = \alpha$ and 0 otherwise, and $g_{\alpha k}(x_\alpha, x_k) = \beta_{\alpha k} x_\alpha x_k$. The exponential family model proposed by Suggala et al. [37],

$$\log f(x_\alpha | \boldsymbol{x}_{\setminus\{\alpha\}}) \propto \left\{ \beta_\alpha B_\alpha(x_\alpha) + \sum_{\{\alpha,k\} \in E} \beta_{\alpha k} B_\alpha(x_\alpha) B_k(x_k) + C_\alpha(x_\alpha) \right\}, \tag{6.3}$$

is also a special case with $g_j(x_j) = \beta_j B_j(x_j) + C_j(x_j)$ for $j = \alpha$ and 0 otherwise, and $g_{\alpha k}(x_\alpha, x_k) = \beta_{\alpha k} B_\alpha(x_\alpha) B_k(x_k)$. Note that many existing exponential family models including (6.3) assume a multiplicative interaction while model (6.2) does not assume any specific interaction. Therefore, the proposed neighborhood selection approach is more general.

Denote $\boldsymbol{X}_i = (X_{i,1}, \cdots, X_{i,p})$ and $\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,p})$ for $i = 1, \cdots, n$ as n i.i.d. random vectors and their realizations. Let $\boldsymbol{x}_{\setminus\{\alpha\}} = (x_1, \cdots, x_{\alpha-1}, x_{\alpha+1}, \cdots, x_p)$, $\boldsymbol{x}_{i,\setminus\{\alpha\}} = (x_{i,1}, \cdots, x_{i,\alpha-1}, x_{i,\alpha+1}, \cdots, x_{i,p})$ be the ith realization of $\boldsymbol{x}_{\setminus\{\alpha\}}$ and $\boldsymbol{x}_i^\alpha = (\boldsymbol{x}_{i,\setminus\{\alpha\}}, x_\alpha) = (x_{i,1}, \cdots, x_{i,\alpha-1}, x_\alpha, x_{i,\alpha+1}, \cdots, x_{i,p})$, where x_α is still a variable. We estimate g by mini-

mizing the following profile penalized pseudo-likelihood:

$$L_\alpha + \frac{\lambda_1}{2}\sum_{j=1}^{p}\theta_j^{-1}||P_j g_j||^2 + \tau_1\sum_{k\neq\alpha} w_{\alpha k}||g_{\alpha k}||, \tag{6.4}$$

where $L_\alpha = \log\{n^{-1}\sum_{i=1}^{n} e^{-g(\boldsymbol{x}_i)}\} + n^{-1}\sum_{i=1}^{n}\int_{\mathcal{X}_\alpha} g(\boldsymbol{x}_i^\alpha)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha$, $\rho(\cdot)$ is a known density of X_α conditional on $\boldsymbol{X}_{\setminus\{\alpha\}} = \boldsymbol{x}_{i,\setminus\{\alpha\}}$, P_j is the projection operator onto $\mathcal{H}^1_{(j)}$, λ_1, τ_1, and θ_j's are tuning parameters, and $0 \leq w_{\alpha k} < \infty$ are pre-specified weights. We consider the pseudo-likelihood since the integral can be computed easily with a proper choice of ρ (Gu [12]). L_α measures the goodness-of-fit. The second element in (6.4) is the roughness L_2 penalty on main effects. The third element in (6.4) is the L_1 penalty for selecting the neighborhood $nb_G(\alpha)$. We allow different weights in the L_1 penalty for flexibility. Gu [12] applied the L_2 penalty to both main effects and interactions for the purpose of density estimation. Jeon and Lin [21] applied the L_1 penalty to both main effects and interactions for edge selection. Consequently, it selects both nodes and edges. In practice, the nodes are usually given and the goal is to detect edges. Therefore, we consider the smoothness promoting L_2 penalty to main effects and the the sparsity promoting L_1 penalty to interactions. Note that Jeon and Lin [21]'s approach is a global method that estimates the joint density, thus is computationally intensive and can only handle very small dimensions.

The resulting estimate for the conditional density is $\hat{f}(x_\alpha|\boldsymbol{x}_{\setminus\{\alpha\}}) \propto e^{\hat{g}(\boldsymbol{x})}\rho(\boldsymbol{x})$ where $\hat{g}$ is the minimizer of (6.4). Notice that the minimization problem (6.4) involves $p-1$ two-way interaction terms. Solving (6.4) for all $\alpha = 1, \cdots, p$ leads to two estimates for each of the two-way interaction, denoted as $\hat{\eta}_{\alpha\kappa}$ and $\hat{\eta}_{\kappa\alpha}$ for $\alpha, \kappa = 1, \cdots, p$ and $\alpha \neq \kappa$. To deal with the possible discrepancy, there are two commonly used rules: AND-rule ($\{\alpha, \kappa\} \in E$ iff $\hat{\eta}_{\alpha\kappa} \neq 0$ and $\hat{\eta}_{\kappa\alpha} \neq 0$) or OR-rule ($\{\alpha, \kappa\} \in E$ iff $\hat{\eta}_{\alpha\kappa} \neq 0$ or $\hat{\eta}_{\kappa\alpha} \neq 0$) [18]. As discussed in Section 4.2 in [5], when the αth and kth nodes are of the same type (same

marginal distribution) or they are both non-Gaussian, there is no clear reason to prefer one edge estimate over the other. In our simulations, all nodes are of the same type. Also, there is no node having obvious normal marginal distribution in real applications. So, either AND-rule or OR-rule can be used. Specifically, we adopt the AND-rule to control false positives. for $\alpha, k = 1, \cdots, p$ and $\alpha \neq k$. We adopt the commonly used AND-rule ($\{\alpha, k\} \in E$ iff $\hat{g}_{\alpha k} \neq 0$ and $\hat{g}_{k\alpha} \neq 0$) or OR-rule ($\{\alpha, k\} \in E$ iff $\hat{g}_{\alpha k} \neq 0$ or $\hat{g}_{k\alpha} \neq 0$) (Hastie et al. [18]).

Similar to Lin and Zhang [27], instead of (6.4), we will minimize the following equivalent but more convenient form

$$L_\alpha + \frac{\lambda_1}{2}\Big(\sum_{j=1}^{p} \theta_j^{-1}||P_j g_j||^2 + \sum_{k\neq\alpha} w_{\alpha k}\theta_{\alpha k}^{-1}||g_{\alpha k}||^2\Big) + \lambda_2 \sum_{k\neq\alpha} w_{\alpha k}\theta_{\alpha k}, \tag{6.5}$$

subject to $\theta_{\alpha k} \geq 0$ for $k = 1, \cdots, p$ and $k \neq \alpha$. The proof of equivalence is the same as Lemma 2.1 and is omitted.

6.2 Computation and Algorithm

In this section, we derive the algorithm to solve (6.5). Note that $g_j \in \mathcal{H}_{(j)} = \mathcal{H}_{(j)}^0 \oplus \mathcal{H}_{(j)}^1$ and $g_{\alpha k} \in \mathcal{H}_{(\alpha k)} = \mathcal{H}_{(\alpha)} \otimes \mathcal{H}_{(k)}$. Denote $\phi_{j1}, \cdots, \phi_{jm_j}$ as basis functions of $\mathcal{H}_{(j)}^0$, and R_j^1, R_j, and $R_{\alpha k}$ as reproducing kernels of $\mathcal{H}_{(j)}^1$, $\mathcal{H}_{(j)}$, and $\mathcal{H}_{(\alpha k)}$ respectively. Since in general the minimization problem (6.5) does not have a solution in a finite dimensional space, as in Gu [12], we approximate the solution by a subset of representers. Specifically, let $\{\tilde{\boldsymbol{x}}_u = (\tilde{x}_{u,1}, \cdots, \tilde{x}_{u,p}),\ u = 1, \cdots, q\}$ be a subset of all observations $\{\boldsymbol{x}_i,\ i = 1, \cdots, n\}$. We collect all basis functions ϕ_{jk} for $j = 1, \cdots, p$ and $k = 1, \cdots, m_j$ and denote them as $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_m)^T$, a vector of functions of $\boldsymbol{x}$ with dimension $m = \sum_{j=1}^{p} m_j$. Denote $\boldsymbol{\theta}_1 = (\theta_1, \cdots, \theta_p)^T$ and $\boldsymbol{\theta}_2 = (\theta_{\alpha 1}, \cdots, \theta_{\alpha(\alpha-1)}, \theta_{\alpha(\alpha+1)}, \cdots, \theta_{\alpha p})^T$. Let

$\xi_{1ju}(x_j) = R_j^1(\tilde{x}_{u,j}, x_j)$, $\xi_{\boldsymbol{\theta_1},u}(\boldsymbol{x}) = \sum_{j=1}^{p} \theta_j \xi_{1ju}(x_j)$, $\xi_{\alpha ku}(x_\alpha, x_k) = R_{\alpha k}((\tilde{x}_{u,\alpha}, \tilde{x}_{u,k}), (x_\alpha, x_k))$, and $\xi_{\boldsymbol{\theta_2},u}(\boldsymbol{x}) = \sum_{k=1,k\neq\alpha}^{p} w_{\alpha k}^{-1} \theta_{\alpha k} \xi_{\alpha ku}(x_\alpha, x_k)$ for $u = 1, \cdots, q$, $k = 1, \cdots, p$, and $k \neq \alpha$. Let $\boldsymbol{\xi}_{\boldsymbol{\theta}_1}(\boldsymbol{x}) = (\xi_{\boldsymbol{\theta_1},1}, \cdots, \xi_{\boldsymbol{\theta_1},q})^T$, $\boldsymbol{\xi}_{\boldsymbol{\theta}_2}(\boldsymbol{x}) = (\xi_{\boldsymbol{\theta_2},1}, \cdots, \xi_{\boldsymbol{\theta_2},q})^T$, and $\boldsymbol{\xi}(\boldsymbol{x}) = \boldsymbol{\xi}_{\boldsymbol{\theta}_1}(\boldsymbol{x}) + \boldsymbol{\xi}_{\boldsymbol{\theta}_2}(\boldsymbol{x})$. The approximate solution can be represented as

$$\begin{aligned} \hat{g}(\boldsymbol{x}) &= \sum_{v=1}^{m} d_v \phi_v(\boldsymbol{x}) + \sum_{u=1}^{q} c_u \left\{ \sum_{j=1}^{p} \theta_j \xi_{1ju}(x_j) + \sum_{k=1,k\neq\alpha}^{p} w_{\alpha k}^{-1} \theta_{\alpha k} \xi_{\alpha ku}(x_\alpha, x_k) \right\} \\ &= \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{d} + \boldsymbol{\xi}^T(\boldsymbol{x})\boldsymbol{c}, \end{aligned} \tag{6.6}$$

where $\boldsymbol{c} = (c_1, \cdots, c_q)^T$ and $\boldsymbol{d} = (d_1, \cdots, d_m)^T$ are coefficients. Plugging $\hat{g}(\boldsymbol{x}_i)$ in (6.6) into (6.5), we need to compute $\boldsymbol{c}$, $\boldsymbol{d}$, and $\boldsymbol{\theta}_2$ as minimizers of

$$\log \left\{ \frac{1}{n} \sum_{i=1}^{n} e^{-\boldsymbol{\phi}_i^T \boldsymbol{d} - \boldsymbol{\xi}_i^T \boldsymbol{c}} \right\} + \boldsymbol{b}_{\boldsymbol{\phi}}^T \boldsymbol{d} + \boldsymbol{b}_{\boldsymbol{\xi}}^T \boldsymbol{c} + \frac{\lambda_1}{2} \boldsymbol{c}^T Q \boldsymbol{c} + \lambda_2 \boldsymbol{w}^T \boldsymbol{\theta}_2 \tag{6.7}$$

subject to $\boldsymbol{\theta}_2 \geq 0$ where $\boldsymbol{\phi}_i = \boldsymbol{\phi}(\boldsymbol{x}_i)$, $\boldsymbol{\xi}_i = \boldsymbol{\xi}(\boldsymbol{x}_i)$, $\boldsymbol{b}_{\boldsymbol{\phi}} = n^{-1} \sum_{i=1}^{n} \int_{\mathcal{X}_\alpha} \boldsymbol{\phi}(\boldsymbol{x}_i^\alpha) \rho(\boldsymbol{x}_i^\alpha) dx_\alpha$, $\boldsymbol{b}_{\boldsymbol{\xi}} = n^{-1} \sum_{i=1}^{n} \int_{\mathcal{X}_\alpha} \boldsymbol{\xi}(\boldsymbol{x}_i^\alpha) \rho(\boldsymbol{x}_i^\alpha) dx_\alpha$, $Q_1 = \left\{ \sum_{j=1}^{p} \theta_j R_j^1(\tilde{x}_{u,j}, \tilde{x}_{v,j}) \right\}_{u,v=1}^{q}$, $Q_{\alpha k} = \Big\{ R_{\alpha k}((\tilde{x}_{u,\alpha}, \tilde{x}_{u,k}), (\tilde{x}_{v,\alpha}, \tilde{x}_{v,k})) \Big\}_{u,v=1}^{q}$, $Q_2 = \sum_{k=1,k\neq\alpha}^{p} w_{\alpha k}^{-1} \theta_{\alpha k} Q_{\alpha k}$, and $Q = Q_1 + Q_2$.

In the following we propose a computational procedure that solves (6.7) iteratively. We first fix $\boldsymbol{\theta}_2$ and update $\boldsymbol{c}$ and $\boldsymbol{d}$ using the Newton-Raphson algorithm. With fixed $\boldsymbol{\theta}_2$, dropping the last term that does not depend on $\boldsymbol{c}$ and $\boldsymbol{d}$, we update $\boldsymbol{c}$ and $\boldsymbol{d}$ by minimizing

$$A_1(\boldsymbol{d}, \boldsymbol{c}) = \log \left\{ \frac{1}{n} \sum_{i=1}^{n} e^{-\boldsymbol{\phi}_i^T \boldsymbol{d} - \boldsymbol{\xi}_i^T \boldsymbol{c}} \right\} + \boldsymbol{b}_{\boldsymbol{\phi}}^T \boldsymbol{d} + \boldsymbol{b}_{\boldsymbol{\xi}}^T \boldsymbol{c} + \frac{\lambda_1}{2} \boldsymbol{c}^T Q \boldsymbol{c}. \tag{6.8}$$

Note that (6.8) has the same form as (10.31) in Gu [12]. Therefore, we can solve (6.8) using the Newton-Raphson procedure with λ_1 and $\boldsymbol{\theta}_1$ selected by the approximate

cross-validation (ACV) method (Gu [12]).

With fixed $\boldsymbol{c}$ and $\boldsymbol{d}$, we update $\boldsymbol{\theta}_2$ using the quadratic programming method. Let $\psi_{1j}(\boldsymbol{x}) = \sum_{u=1}^{q} c_u \xi_{1ju}(x_j)$ for $j = 1, \cdots, p$, $\boldsymbol{\psi}_1(\boldsymbol{x}) = (\psi_{11}, \cdots, \psi_{1p})^T$, $\psi_{2k}(\boldsymbol{x}) = w_{\alpha k}^{-1} \sum_{u=1}^{q} c_u \xi_{\alpha k u}(x_\alpha, x_k)$ for $k = 1, \cdots, p$ and $k \neq \alpha$, and $\boldsymbol{\psi}_2(\boldsymbol{x}) = (\psi_{21}, \cdots, \psi_{2(\alpha-1)}, \psi_{2(\alpha+1)}, \cdots, \psi_{2p})^T$. We rewrite $\hat{g}$ in (6.6) as $\hat{g}(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{d} + \boldsymbol{\psi}_1^T(\boldsymbol{x})\boldsymbol{\theta}_1 + \boldsymbol{\psi}_2^T(\boldsymbol{x})\boldsymbol{\theta}_2$. Plugging $\hat{g}(\boldsymbol{x}_i)$ into (6.5) and keeping terms involving $\boldsymbol{\theta}_2$ only, (6.7) reduces to

$$\log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-\boldsymbol{\phi}_i^T \boldsymbol{d} - \boldsymbol{\psi}_{1i}^T \boldsymbol{\theta}_1 - \boldsymbol{\psi}_{2i}^T \boldsymbol{\theta}_2}\right\} + \boldsymbol{b}_{\boldsymbol{\psi}_2}^T \boldsymbol{\theta}_2 + \frac{\lambda_1}{2}\boldsymbol{c}^T Q_2 \boldsymbol{c} + \lambda_2 \boldsymbol{w}^T \boldsymbol{\theta}_2 \tag{6.9}$$

subject to $\boldsymbol{\theta}_2 \geq 0$, where $\boldsymbol{\psi}_{1i} = \boldsymbol{\psi}_1(\boldsymbol{x}_i)$, $\boldsymbol{\psi}_{2i} = \boldsymbol{\psi}_2(\boldsymbol{x}_i)$, and $\boldsymbol{b}_{\boldsymbol{\psi}_2} = \frac{1}{n}\sum_{i=1}^{n} \int_{\mathcal{X}_\alpha} \boldsymbol{\psi}_2(\boldsymbol{x}_i^\alpha)\rho(\boldsymbol{x}_i^\alpha) dx_\alpha$. Furthermore, the constraint minimization problem (6.9) is equivalent to

$$A_2(\boldsymbol{\theta}_2) = \log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-\boldsymbol{\phi}_i^T \boldsymbol{d} - \boldsymbol{\psi}_{1i}^T \boldsymbol{\theta}_1 - \boldsymbol{\psi}_{2i}^T \boldsymbol{\theta}_2}\right\} + \boldsymbol{b}_{\boldsymbol{\psi}_2}^T \boldsymbol{\theta}_2 + \frac{\lambda_1}{2}\boldsymbol{c}^T Q_2 \boldsymbol{c} \tag{6.10}$$

subject to $\boldsymbol{\theta}_2 \geq 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$ for some constant M, where M controls the sparsity in $\boldsymbol{\theta}_2$. Note that $A_2(\boldsymbol{\theta}_2)$ is a convex function of $\boldsymbol{\theta}_2$ (The proof is the same as (2.12)). We solve (6.10) iteratively using the quadratic programming. Denote the current estimate of $\boldsymbol{\theta}_2$ as $\tilde{\boldsymbol{\theta}}_2$ and $\tilde{g}(\boldsymbol{x}) = \boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{d} + \boldsymbol{\psi}_1^T(\boldsymbol{x})\boldsymbol{\theta}_1 + \boldsymbol{\psi}_2^T(\boldsymbol{x})\tilde{\boldsymbol{\theta}}_2$. Define $\mu_{\tilde{g}}(\boldsymbol{h}) = \sum_{i=1}^{n} e^{-\tilde{g}(\boldsymbol{x}_i)}\boldsymbol{h}(\boldsymbol{x}_i) / \sum_{i=1}^{n} e^{-\tilde{g}(\boldsymbol{x}_i)}$, $\mu_{\tilde{g}}(\boldsymbol{h}_1\boldsymbol{h}_2^T) = \sum_{i=1}^{n} e^{-\tilde{g}(\boldsymbol{x}_i)}\big(\boldsymbol{h}_1(\boldsymbol{x}_i)\boldsymbol{h}_2^T(\boldsymbol{x}_i)\big) / \sum_{i=1}^{n} e^{-\tilde{g}(\boldsymbol{x}_i)}$ for any functions $\boldsymbol{h}, \boldsymbol{h}_1, \boldsymbol{h}_2$. We update $\boldsymbol{\theta}_2$ by minimizing the following second order Taylor approximation of $A_2(\boldsymbol{\theta}_2)$ (some constants independent of $\boldsymbol{\theta}_2$ have been removed):

$$\frac{1}{2}\boldsymbol{\theta}_2^T H_A(\tilde{\boldsymbol{\theta}}_2)\boldsymbol{\theta}_2 + \boldsymbol{\theta}_2^T\left\{G_A(\tilde{\boldsymbol{\theta}}_2) - H_A(\tilde{\boldsymbol{\theta}}_2)\tilde{\boldsymbol{\theta}}_2\right\} \tag{6.11}$$

subject to $\boldsymbol{\theta}_2 \geq 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$ for some constant M, where $G_A(\tilde{\boldsymbol{\theta}}_2) = -\mu_{\tilde{g}}(\boldsymbol{\psi}_2) + \boldsymbol{b}_{\boldsymbol{\psi}_2} + \lambda_1\boldsymbol{q}_2/2$ is the gradient, $H_A(\tilde{\boldsymbol{\theta}}_2) = V_{\tilde{g}}(\boldsymbol{\psi}_2, \boldsymbol{\psi}_2^T)$ is the Hessian, $V_{\tilde{g}}(\boldsymbol{\psi}_2, \boldsymbol{\psi}_2^T) = \mu_{\tilde{g}}(\boldsymbol{\psi}_2\boldsymbol{\psi}_2^T) -$

$\mu_{\tilde{g}}(\boldsymbol{\psi}_2)\mu_{\tilde{g}}(\boldsymbol{\psi}_2^T)$, $\boldsymbol{q}_2 = (w_{\alpha 1}^{-1}\boldsymbol{c}^T Q_{\alpha 1}\boldsymbol{c}, \cdots, w_{\alpha(\alpha-1)}^{-1}\boldsymbol{c}^T Q_{\alpha(\alpha-1)}\boldsymbol{c}, w_{\alpha(\alpha+1)}^{-1}\boldsymbol{c}^T Q_{\alpha(\alpha+1)}\boldsymbol{c}, \cdots,$
$w_{\alpha p}^{-1}\boldsymbol{c}^T Q_{\alpha p}\boldsymbol{c})^T$, and $Q_{\alpha k} = \left\{R_{\alpha k}((\tilde{x}_{u,\alpha}, \tilde{x}_{u,k}), (\tilde{x}_{v,\alpha}, \tilde{x}_{v,k}))\right\}_{u,v=1}^{q}$ for $k = 1, \cdots, p$ and $k \neq \alpha$.

We apply the quadratic programming to solve (6.11) and k-fold cross-validation or BIC method to select M. The iterative procedure for updating $\boldsymbol{\theta}_2$ may be stopped after a fixed number of steps or until convergence. We summarize our complete algorithm as follows.

Algorithm for the neighborhood selection approach:

1. *Initialize*: $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_2^0$.

2. *Cycle until convergence*: Update $\boldsymbol{c}$, $\boldsymbol{d}$ and $\boldsymbol{\theta}_2$ sequentially:

   (a) Fix $\boldsymbol{\theta}_2$ at the current estimate, update $\boldsymbol{c}$ and $\boldsymbol{d}$ by solving (6.8) with tuning parameters λ_1 and $\boldsymbol{\theta}_1$ selected by the ACV method.

   (b) Fix $\boldsymbol{d}$, $\boldsymbol{c}$, λ_1 and $\boldsymbol{\theta}_1$ at the current estimates, update $\boldsymbol{\theta}_2$ by applying quadratic programming to iteratively solve the quadratic approximations (6.11) subject to $\boldsymbol{\theta}_2 \geq 0$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$ where the tuning parameter M is selected by the k-fold cross-validation or the BIC method.

6.3 Algorithm Implementation

In this section, we provide details about the implementation of the proposed algorithm using existing R packages. Specifically, we implement Step 2.(a) in the algorithm using a modification of the `sscden1` function in the `gss` package (Gu et al. [17]) and Step 2.(b) using the R function `solve.QP` in the `quadprog` package (Turlach and Weingessel [38]).

6.3.1 Implementation of the Newton-Raphson Method

Given current value of $\boldsymbol{\theta}_2$, we update $\boldsymbol{c}$ and $\boldsymbol{d}$ by minimizing (6.8) using the Newton-Raphson method. We implement by modifying the function `sscden1` in the `gss` package since (6.8) has the same form as (10.31) in Gu [12] with different penalties. By definition, $\mathcal{H}_{(\alpha k)} = \mathcal{H}_{(\alpha)} \otimes \mathcal{H}_{(k)} = (\mathcal{H}^0_{(\alpha)} \oplus \mathcal{H}^1_{(\alpha)}) \otimes (\mathcal{H}^0_{(k)} \oplus \mathcal{H}^1_{(k)}) = (\mathcal{H}^0_{(\alpha)} \otimes \mathcal{H}^0_{(k)}) \oplus (\mathcal{H}^0_{(\alpha)} \otimes \mathcal{H}^1_{(k)}) \oplus (\mathcal{H}^1_{(\alpha)} \otimes \mathcal{H}^0_{(k)}) \oplus (\mathcal{H}^1_{(\alpha)} \otimes \mathcal{H}^1_{(k)}) = \mathcal{H}^{(0)}_{(\alpha k)} \oplus \mathcal{H}^{(1)}_{(\alpha k)} \oplus \mathcal{H}^{(2)}_{(\alpha k)} \oplus \mathcal{H}^{(3)}_{(\alpha k)}$ where $\mathcal{H}^{(0)}_{(\alpha k)} = \mathcal{H}^0_{(\alpha)} \otimes \mathcal{H}^0_{(k)}$, $\mathcal{H}^{(1)}_{(\alpha k)} = \mathcal{H}^0_{(\alpha)} \otimes \mathcal{H}^1_{(k)}$, $\mathcal{H}^{(2)}_{(\alpha k)} = \mathcal{H}^1_{(\alpha)} \otimes \mathcal{H}^0_{(k)}$, and $\mathcal{H}^{(3)}_{(\alpha k)} = \mathcal{H}^1_{(\alpha)} \otimes \mathcal{H}^1_{(k)}$. For density estimation, the penalized likelihood method in Gu [12] does not penalize functions in the parametric component space $\mathcal{H}^0_{(\alpha k)}$ and has different smoothing parameters for components in the nonparametric component spaces $\mathcal{H}^{(1)}_{(\alpha k)}$, $\mathcal{H}^{(2)}_{(\alpha k)}$, and $\mathcal{H}^{(3)}_{(\alpha k)}$. Our goal is edge detection by detecting non-zero interactions. Therefore, we penalize the combined interaction $g_{\alpha k} \in \mathcal{H}_{(\alpha k)}$ as a whole with a smoothing parameter $\theta_{\alpha k}$ for $k = 1, \cdots, p$ and $k \neq \alpha$. The interaction $g_{\alpha k}$ collects parametric and nonparametric interaction components in $\mathcal{H}^{(0)}_{(\alpha k)}$, $\mathcal{H}^{(1)}_{(\alpha k)}$, $\mathcal{H}^{(2)}_{(\alpha k)}$, and $\mathcal{H}^{(3)}_{(\alpha k)}$. Note that $\boldsymbol{\theta}_2 = (\theta_{\alpha 1}, \cdots, \theta_{\alpha(\alpha-1)}, \theta_{\alpha(\alpha+1)}, \cdots, \theta_{\alpha p})^T$ is fixed at this step. We modified the function `sscden1` to solve (6.8) with smoothing parameters λ_1 and $\boldsymbol{\theta}_1$ estimated by the approximated cross-validation method.

6.3.2 Implementation of Quadratic Programming

With $\boldsymbol{c}$ and $\boldsymbol{d}$ being fixed at their current values, we need to update $\boldsymbol{\theta}_2$ iteratively by applying the quadratic programming algorithm to minimize

$$\frac{1}{2}\boldsymbol{\theta}_2^T H_A(\tilde{\boldsymbol{\theta}}_2)\boldsymbol{\theta}_2 + \boldsymbol{\theta}_2^T \left\{ G_A(\tilde{\boldsymbol{\theta}}_2) - H_A(\tilde{\boldsymbol{\theta}}_2)\tilde{\boldsymbol{\theta}}_2 \right\} \tag{6.12}$$

subject to $\theta_{\alpha k} \geq 0$ for $k \neq \alpha$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$ for some M, where M is a tuning parameter. We use the R function `solve.QP` to solve (6.12). We estimate the tuning parameter M

by minimizing a k-fold cross-validation or the BIC score defined as follows. Let $I_1, \cdots, I_k$ be k randomly partitioned subsamples of the original data, $n_j = |I_j|$, and $n_{(-j)} = n - n_j$. Denote $g_M^{(-j)}$ as the estimate without observations in the subset I_j which minimizes the following function with respect to $\boldsymbol{\theta}_2$:

$$\log\left\{\frac{1}{n_{(-j)}}\sum_{i\notin I_j} e^{-g(\boldsymbol{x}_i^\alpha)}\right\} + \frac{1}{n_{(-j)}}\sum_{i\notin I_j}\int_{\mathcal{X}_\alpha} g(\boldsymbol{x}_i^\alpha)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + \lambda_1\sum_{k\neq\alpha} w_{\alpha k}\theta_{\alpha k}^{-1}||g_{\alpha k}||^2 \quad (6.13)$$

subject to $\theta_{\alpha k} \geq 0$ for $k \neq \alpha$ and $\boldsymbol{w}^T\boldsymbol{\theta}_2 \leq M$. The k-fold cross-validation score is defined as

$$\mathrm{CV}(M) = \log\left\{\frac{1}{n}\sum_{j=1}^{k}\sum_{i\in I_j} e^{-g_M^{(-j)}(\boldsymbol{x}_i^\alpha)}\right\} + \frac{1}{n}\sum_{j=1}^{k}\sum_{i\in I_j}\int_{\mathcal{X}_\alpha} g_M^{(-j)}(\boldsymbol{x}_i^\alpha)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha. \quad (6.14)$$

The BIC score is defined as

$$\mathrm{BIC}(M) = \log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-g_M(\boldsymbol{x}_i^\alpha)}\right\} + \frac{1}{n}\sum_{i=1}^{n}\int_{\mathcal{X}_\alpha} g_M(\boldsymbol{x}_i^\alpha)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + \log(nk_n), \quad (6.15)$$

where g_M expresses the dependence of the estimate on M explicitly and k_n is the number of non-zero elements in the estimate of $\boldsymbol{\theta}_2$. We applied the k-fold cross-validation method in all simulations with $k = 5$ and the BIC method in real data applications to get sparser graphs.

6.3.3 Initial Values and Convergence Criterion

To get a good initial value $\boldsymbol{\theta}_2^0$, we first estimate the conditional density $f(x_\alpha|\boldsymbol{x}_{\setminus\{\alpha\}}) \propto e^{g(\boldsymbol{x})}\rho(\boldsymbol{x})$ with $\tau_1\sum_{k\neq\alpha} w_{\alpha k}||g_{\alpha k}||$ being replaced by $(\lambda_1/2)\sum_{k\neq\alpha}\theta_{\alpha k}^{-1}||g_{\alpha k}||^2$. We modified the `sscden` function in the `gss` package to estimate the conditional density and denote the estimate of $\eta_{\alpha k}$ as $\check{\eta}_{\alpha k}$. Since $\theta_{\alpha k} = 0$ in $\boldsymbol{\theta}_2$ if and only if $\eta_{\alpha k} = 0$, the magnitude of $\check{\eta}_{\alpha k}$

provides one way to initialize $\theta_{\alpha k}$. Specifically, we set $\theta^0_{\alpha k} = \{\sum_{i=1}^n \check{\eta}^2_{\alpha k}(\boldsymbol{x}_i)\}^{1/2}$.

The convergence criterion in Step 2 in the algorithm is $||\boldsymbol{\theta}_2 - \tilde{\boldsymbol{\theta}}_2||_2/(||\tilde{\boldsymbol{\theta}}_2||_2 + 10^{-6}) \leq \varepsilon$ or the number of zeros in $\boldsymbol{\theta}_2$ stops increasing for fixed number of steps, where $\boldsymbol{\theta}_2$ and $\tilde{\boldsymbol{\theta}}_2$ are the updated and previous estimates, respectively, $||\cdot||_2$ is the Euclidean norm, and ε a threshold. We set $\varepsilon = 0.001$ in simulation and real data examples.

Chapter 7

Theoretical Analysis

In this chapter, we study the theoretical properties of the proposed method. Following similar steps and under same regularity conditions as Gu [12], we derive convergence rate for the conditional density estimate $\hat{g}$ subject to both L_1 and L_2 penalties. In addition, we derive the convergence rates for interactions in the SS ANOVA decomposition, which is new and important for edge detection.

7.1 Notations

Let $f_0(x_\alpha|\boldsymbol{x}_{\setminus\{\alpha\}}) = e^{g_0(\boldsymbol{x})}\rho(\boldsymbol{x})$ be the true conditional density to be estimated. Let $g = g^{(1)} + g^{(2)} = \sum_{j=1}^{p} g_j + \sum_{k\neq\alpha} g_{\alpha k}$, and $\hat{g}$ be the minimizer of (6.4). Define $V^*(h_1, h_2) = \int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} h_1(\boldsymbol{x})h_2(\boldsymbol{x})\rho(\boldsymbol{x})dx_\alpha d\boldsymbol{x}_{\setminus\{\alpha\}}$, $J_1(h_1, h_2) = \sum_{j=1}^{p} \theta_j^{-1} \int_{\mathcal{X}_j} h_1 h_2 dx_j$, $J_2(h_1, h_2) = \sum_{k\neq\alpha} w_{\alpha k}(\int_{\mathcal{X}_\alpha}\int_{\mathcal{X}_k} |h_1 h_2| dx_\alpha dx_k)^{1/2}$, and $J_2^*(h_1, h_2) = \sum_{k\neq\alpha} \theta_{\alpha k}^{-1} \int_{\mathcal{X}_\alpha}\int_{\mathcal{X}_k} h_1 h_2 dx_\alpha dx_k$ for any functions h_1, h_2, where $f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}})$ is the density of $\boldsymbol{X}_{\setminus\{\alpha\}}$ on $\mathcal{X}_{\setminus\{\alpha\}} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_{\alpha-1} \times \mathcal{X}_{\alpha+1} \times \cdots \times \mathcal{X}_p$. We denote $||h|| = (\int_{\mathcal{X}_j} h^2 dx_j)^{1/2}$ as the L_2 norm for any $h \in \mathcal{H}^{(j)}$. Then, we have $V^*(g) = V^*(g, g)$, $V_1(g^{(1)}) = V^*(g^{(1)})$, $V_2(g^{(2)}) = [V^*(g^{(2)})]^{1/2}$,

$J_1(g) = J_1(g,g) = \sum_{j=1}^{p} \theta_j^{-1}||P_j g_j||^2$, $J_2(g) = J_2(g,g) = \sum_{k\neq\alpha} w_{\alpha k}||g_{\alpha k}||$, and $J_2^*(g) = J_2^*(g,g) = \sum_{k\neq\alpha} \theta_{\alpha k}^{-1}||g_{\alpha k}||^2$.

Without loss of generality, we assume $w_{\alpha k} = 1$ in the proof, simulations and real applications. Furthermore, we let $V(g) = V_1(g^{(1)}) + V_2(g^{(2)})$, $J = J_1 + J_2$, and $J^*(g) = J_1(g) + J_2^*(g)$. Notations O_p, o_p, $\leq_p$ and $\stackrel{a.s.}{\to}$ are defined the same as Section 3.1.

7.2 Convergence Rates

We start this section by introducing conditions and lemmas that are needed for theoretical analysis.

Condition 7.1 *V^* is completely continuous with respect to J^*.*

From Theorem 3.1 of Weinberger [42], there exists eigenvalues γ_v of J^* with respect to V^* and the associated eigenfunctions ζ_v such that

$$V^*(\zeta_v, \zeta_u) = \delta_{v,u}, \qquad J^*(\zeta_v, \zeta_u) = \gamma_v \delta_{v,u},$$

where $0 \leq \gamma_v \uparrow \infty$ and $\delta_{v,u}$ is the Kronecker delta. Functions satisfying $J^*(g) < \infty$ can be expressed as a Fourier series expansion $g = \sum_v a_v \zeta_v$, where $a_v = V^*(g, \zeta_v)$ are the Fourier coefficients.

Condition 7.2 *For v sufficiently large and some $\varphi > 0$, the eigenvalues γ_v of J^* with respect to V^* satisfy $\gamma_v > \varphi v^r$ where $r > 1$.*

Consider the quadratic functional

$$\frac{1}{n}\sum_{i=1}^{n} -e^{-g_0(\boldsymbol{X}_i)} g(\boldsymbol{X}_i) + \frac{1}{n}\sum_{i=1}^{n}\int_{\mathcal{X}_\alpha} g(\boldsymbol{x}_i^\alpha)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + \frac{1}{2}V^*(g - g_0) + \frac{\lambda_1}{2}J^*(g), \qquad (7.1)$$

and denote the minimizer of (7.1) as $\tilde{g}$. Plugging the Fourier series expansions $g = \sum_v a_v \zeta_v$ and $g_0 = \sum_v a_{v,0}\zeta_v$ into (7.1), $\tilde{g}$ has Fourier coefficients $\tilde{a}_v = (\kappa_v + a_{v,0})/(1+\lambda_1\gamma_v)$, where $\kappa_v = n^{-1}\sum_{i=1}^{n}\{e^{-g_0(\boldsymbol{X}_i)}\zeta_v(\boldsymbol{X}_i) - \int_{\mathcal{X}_\alpha}\zeta_v(\boldsymbol{x})\rho(\boldsymbol{x})dx_\alpha\}$. It is not difficult to verify that $E(\kappa_v) = 0$ and $E(\kappa_v^2) \le n^{-1}\int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}})\int_{\mathcal{X}_\alpha}\zeta_v^2(\boldsymbol{x})e^{-g_0(\boldsymbol{x})}\rho(\boldsymbol{x})dx_\alpha d\boldsymbol{x}_{\setminus\{\alpha\}}$.

Condition 7.3 *For some $c_1 < \infty$, $e^{-g_0} < c_1$.*

Under Condition 7.3, noting that $V^*(\zeta_v) = \int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}})\int_{\mathcal{X}_\alpha}\zeta_v^2(\boldsymbol{x})\rho(\boldsymbol{x})dx_\alpha d\boldsymbol{x}_{\setminus\{\alpha\}} = 1$ by the definition of V^* and ζ_v, we have $E(\kappa_v^2) \le n^{-1}c_1$.

Lemma 7.1 *Assume $J^*(g_0) < \infty$. Under Conditions 7.1–7.3, as $\lambda_1 \to 0$ and $n \to \infty$,*

$$(V^* + \lambda_1 J^*)(\tilde{g} - g_0) = O_p(n^{-1}\lambda_1^{-1/r} + \lambda_1).$$

Proof: By the Fourier series expansions of $\tilde{g}$ and g_0, we have

$$\begin{aligned} V^*(\tilde{g} - g_0) &= \sum_v (\tilde{a}_v - a_{v,0})^2 = \sum_v \frac{\kappa_v^2 - 2\kappa_v\lambda_1\gamma_v a_{v,0} + \lambda_1^2\gamma_v^2 a_{v,0}^2}{(1+\lambda_1\gamma_v)^2}, \\ \lambda_1 J^*(\tilde{g} - g_0) &= \sum_v \lambda_1\gamma_v(\tilde{a}_v - a_{v,0})^2 = \sum_v \lambda_1\gamma_v\frac{\kappa_v^2 - 2\kappa_v\lambda_1\gamma_v a_{v,0} + \lambda_1^2\gamma_v^2 a_{v,0}^2}{(1+\lambda_1\gamma_v)^2}. \end{aligned}$$

Since $E(\kappa_v) = 0$ and $E(\kappa_v^2) \le c_1/n$, we have

$$\begin{aligned} E[V^*(\tilde{g} - g_0)] &\le \frac{c_1}{n}\sum_v \frac{1}{(1+\lambda_1\gamma_v)^2} + \lambda_1\sum_v \frac{\lambda_1\gamma_v}{(1+\lambda_1\gamma_v)^2}\gamma_v a_{v,0}^2, \\ E[\lambda_1 J^*(\tilde{g} - g_0)] &\le \frac{c_1}{n}\sum_v \frac{\lambda_1\gamma_v}{(1+\lambda_1\gamma_v)^2} + \lambda_1\sum_v \frac{(\lambda_1\gamma_v)^2}{(1+\lambda_1\gamma_v)^2}\gamma_v a_{v,0}^2. \end{aligned} \tag{7.2}$$

Following similar arguments in the proof of Lemma 9.1 in Gu [12], we have

$$\sum_v \frac{\lambda_1\gamma_v}{(1+\lambda_1\gamma_v)^2} = O(\lambda_1^{-1/r}), \quad \sum_v \frac{1}{(1+\lambda_1\gamma_v)^2} = O(\lambda_1^{-1/r}), \quad \sum_v \frac{1}{1+\lambda_1\gamma_v} = O(\lambda_1^{-1/r}).$$

The lemma follows from (7.2) and the fact that $\sum_v \gamma_v a_{v,0}^2 = J^*(g_0) < \infty$. $\square$

As in Gu [12], when g_0 is "supersmooth", in the sense that $\sum_v \gamma_v^l a_{v,0}^2 < \infty$ for some $1 < l \leq 2$ which is assumed in Theorem 7.1, the rates can be improved to $O(n^{-1}\lambda_1^{-1/r} + \lambda_1^l)$.

Now we want to bound the approximation error $\hat{g} - \tilde{g}$. Define

$$\begin{aligned}
A_{h_1,h_2}(\tau) =& \frac{1}{n}\sum_{i=1}^n e^{-(h_1+\tau h_2)(\boldsymbol{X}_i)} + \frac{1}{n}\sum_{i=1}^n \int_{\mathcal{X}_\alpha} (h_1+\tau h_2)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + \frac{\lambda_1}{2}J^*(h_1+\tau h_2) \\
&+ \lambda_2 \sum_{k\neq\alpha} \theta_{\alpha k}, \\
B_{h_1,h_2}(\tau) =& \frac{1}{n}\sum_{i=1}^n -e^{-g_0(\boldsymbol{X}_i)}(h_1+\tau h_2)(\boldsymbol{X}_i) + \frac{1}{n}\sum_{i=1}^n \int_{\mathcal{X}_\alpha} (h_1+\tau h_2)\rho(\boldsymbol{x}_i^\alpha)dx_\alpha \\
&+ \frac{1}{2}V^*(h_1+\tau h_2 - g_0) + \frac{\lambda_1}{2}J^*(h_1+\tau h_2).
\end{aligned}$$

Taking derivative of A_{h_1,h_2} and B_{h_1,h_2} with respect to τ evaluated at $\tau = 0$, we obtain

$$\dot{A}_{h_1,h_2}(0) = -\frac{1}{n}\sum_{i=1}^n e^{-h_1(\boldsymbol{X}_i)}h_2(\boldsymbol{X}_i) + \frac{1}{n}\sum_{i=1}^n \int_{\mathcal{X}_\alpha} h_2\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + \lambda_1 J^*(h_1,h_2), \tag{7.3}$$

$$\begin{aligned}
\dot{B}_{h_1,h_2}(0) =& -\frac{1}{n}\sum_{i=1}^n e^{-g_0(\boldsymbol{X}_i)}h_2(\boldsymbol{X}_i) + \frac{1}{n}\sum_{i=1}^n \int_{\mathcal{X}_\alpha} h_2\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + V^*(h_1-g_0,h_2) \\
&+ \lambda_1 J^*(h_1,h_2).
\end{aligned} \tag{7.4}$$

Setting $h_1 = \hat{g}$ and $h_2 = \hat{g} - \tilde{g}$ in (7.3), we have

$$-\frac{1}{n}\sum_{i=1}^n e^{-\hat{g}(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) + \frac{1}{n}\sum_{i=1}^n \int_{\mathcal{X}_\alpha} (\hat{g}-\tilde{g})\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + \lambda_1 J^*(\hat{g},\hat{g}-\tilde{g}) = 0. \tag{7.5}$$

Setting $h_1 = \tilde{g}$ and $h_2 = \hat{g} - \tilde{g}$ in (7.4), we have

$$-\frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) + \frac{1}{n}\sum_{i=1}^{n}\int_{\mathcal{X}_\alpha}(\hat{g}-\tilde{g})\rho(\boldsymbol{x}_i^\alpha)dx_\alpha + V^*(\tilde{g}-g_0, \hat{g}-\tilde{g})$$
$$+ \lambda_1 J^*(\tilde{g}, \hat{g}-\tilde{g}) = 0. \tag{7.6}$$

Subtracting (7.6) from (7.5), we obtain

$$\begin{aligned}&\lambda_1 J^*(\hat{g}-\tilde{g}) - \frac{1}{n}\sum_{i=1}^{n}\left\{e^{-\hat{g}(\boldsymbol{X}_i)} - e^{-\tilde{g}(\boldsymbol{X}_i)}\right\}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)\\ =&\frac{1}{n}\sum_{i=1}^{n}\left\{e^{-\tilde{g}(\boldsymbol{X}_i)} - e^{-g_0(\boldsymbol{X}_i)}\right\}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) + V^*(\hat{g}-\tilde{g}, \tilde{g}-g_0).\end{aligned} \tag{7.7}$$

Condition 7.4 *For g in a convex set B_0 around g_0 containing $\hat{g}$ and $\tilde{g}$, $c_2 < e^{g_0-g} < c_3$ holds uniformly for some $0 < c_2 < c_3 < \infty$.*

Condition 7.5 *For any $u, v = 1, 2, \cdots$, $\int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} \zeta_v^2\zeta_u^2 e^{-g_0}\rho(\boldsymbol{x})dx_\alpha d\boldsymbol{x}_{\setminus\{\alpha\}} < c_4$ for some $c_4 < \infty$.*

Applying the mean value theorem, we have $e^{-\hat{g}(\boldsymbol{X}_i)} - e^{-\tilde{g}(\boldsymbol{X}_i)} = -e^{-(\tilde{g}+\tau_i(\hat{g}-\tilde{g}))(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)$ where $\tau_i \in [0,1]$. Since $\hat{g}$ and $\tilde{g}$ belongs to B_0 which is a convex set around g_0, under Condition 7.4, there exists a $b_0^{(i)} \in (c_2, c_3)$ such that $-e^{-(\tilde{g}+\tau_i(\hat{g}-\tilde{g}))(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) = -b_0^{(i)}e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)$. Then

$$\begin{aligned}-\frac{1}{n}\sum_{i=1}^{n}\left\{e^{-\hat{g}(\boldsymbol{X}_i)} - e^{-\tilde{g}(\boldsymbol{X}_i)}\right\}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) =&\frac{1}{n}\sum_{i=1}^{n} b_0^{(i)}e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_i)\\ \geq&\frac{c_2}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})^2(\boldsymbol{X}_i).\end{aligned} \tag{7.8}$$

By the same argument, there exists a $c_0^{(i)} \in (c_2, c_3)$ such that

$$\frac{1}{n}\sum_{i=1}^{n}\left\{e^{-\tilde{g}(\boldsymbol{X}_i)} - e^{-g_0(\boldsymbol{X}_i)}\right\}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) = -\frac{1}{n}\sum_{i=1}^{n} c_0^{(i)} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)(\tilde{g}-g_0)(\boldsymbol{X}_i). \tag{7.9}$$

Lemma 7.2 *Under Conditions 7.1, 7.2 and 7.5, suppose h_1, h_2 are functions satisfying $J^*(h_1) < \infty, J^*(h_2) < \infty$, as $\lambda_1 \to 0$ and $n\lambda_1^{2/r} \to \infty$, one has*

$$\left|\frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)} h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i) - V^*(h_1,h_2)\right| = o_p\Big(\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2}\Big). \tag{7.10}$$

Proof: Since $J^*(h_1) < \infty, J^*(h_2) < \infty$, then h_1 and h_2 can be expressed as Fourier series $h_1 = \sum_v h_{1,v}\zeta_v$ and $h_2 = \sum_v h_{2,v}\zeta_v$. Let

$$U_i = \zeta_v(\boldsymbol{X}_i)\zeta_u(\boldsymbol{X}_i)e^{-g_0(\boldsymbol{X}_i)} - \int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} \zeta_v(\boldsymbol{x})\zeta_u(\boldsymbol{x})\rho(\boldsymbol{x}) dx_\alpha d\boldsymbol{x}_{\setminus\{\alpha\}}.$$

Note that U_i are i.i.d. random variables with $\mathrm{E}(U_i) = 0$. Then under Condition 7.5, we have

$$E\left(\frac{1}{n}\sum_{i=1}^{n} U_i\right)^2 = \frac{1}{n}\mathrm{Var}\left(\zeta_v(\boldsymbol{X}_1)\zeta_u(\boldsymbol{X}_1)e^{-g_0(\boldsymbol{X}_1)}\right) < \frac{c_4}{n}.$$

Furthermore,

$$
\begin{aligned}
&\Big|\frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)} h_1(\boldsymbol{X}_i) h_2(\boldsymbol{X}_i) - V^*(h_1, h_2)\Big| \\
=&\Big|\sum_v \sum_u h_{1,v} h_{2,u} \frac{1}{n}\sum_{i=1}^{n} U_i\Big| \\
\le&\left\{\sum_v \sum_u \frac{1}{1+\lambda_1\gamma_v}\frac{1}{1+\lambda_1\gamma_u}\left(\frac{1}{n}\sum_{i=1}^{n} U_i\right)^2\right\}^{1/2}\left\{\sum_v\sum_u (1+\lambda_1\gamma_v)(1+\lambda_1\gamma_u) h_{1,v}^2 h_{2,u}^2\right\}^{1/2} \\
=&O_p\Big(n^{-1/2}\lambda_1^{-1/r}\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2}\Big) \\
=&o_p\Big(\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2}\Big),
\end{aligned}
$$

where the second equality holds because of $\sum_v \frac{1}{1+\lambda_1\gamma_v} = O(\lambda_1^{-1/r})$ and the strong law of large numbers. □

Lemma 7.3 *Under Conditions 7.1, 7.2 and 7.5, suppose h_1 and h_2 are functions satisfying $V^*(h_1) < \infty, V^*(h_2) < \infty$, as $\lambda_1 \to 0$ and $n\lambda_1^{2/r} \to \infty$, one has*

$$
\begin{aligned}
&\Big|\frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)} h_1(\boldsymbol{X}_i) h_2(\boldsymbol{X}_i) - \frac{1}{n}\sum_{i=1}^{n} c_0^{(i)} e^{-g_0(\boldsymbol{X}_i)} h_1(\boldsymbol{X}_i) h_2(\boldsymbol{X}_i)\Big| \\
\le_p\, & c_0\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2}, \qquad (7.11)
\end{aligned}
$$

where $c_0 = max\{|c_2 - 1|, |c_3 - 1|\}$.

Proof: Note that for each $\boldsymbol{X}_i$,

$$
\begin{aligned}
&\mathrm{E}|e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \\
=&\int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} |h_1(\boldsymbol{x})h_2(\boldsymbol{x})|\rho(\boldsymbol{x})d\boldsymbol{x} \\
\leq&\big\{\big(\int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} h_1^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\big)\big(\int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} h_2^2(\boldsymbol{x})\rho(\boldsymbol{x})d\boldsymbol{x}\big)\big\}^{1/2} \\
=&\{V^*(h_1)V^*(h_2)\}^{1/2} \\
\leq&\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2},
\end{aligned}
$$

where the first inequality follows Cauchy-Schwartz inequality. Since $\boldsymbol{X}_i$'s are independent, we have $|e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)|$'s are i.i.d. random variables with mean $\mathrm{E}|e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \leq \big\{V^*(h_1)V^*(h_2)\big\}^{1/2} < \infty$. Therefore, by the strong law of large numbers, $\frac{1}{n}\sum_{i=1}^n |e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \xrightarrow{a.s.} \mathrm{E}|e^{-g_0(\boldsymbol{X}_1)}h_1(\boldsymbol{X}_1)h_2(\boldsymbol{X}_1)|$ as $n \to \infty$. Then, we have

$$
\begin{aligned}
&\Big|\frac{1}{n}\sum_{i=1}^n e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i) - \frac{1}{n}\sum_{i=1}^n c_0^{(i)} e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)\Big| \\
=&\Big|\frac{1}{n}\sum_{i=1}^n (1-c_0^{(i)}) e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)\Big| \\
\leq&\frac{1}{n}\sum_{i=1}^n |(1-c_0^{(i)})||e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \\
\leq&c_0\frac{1}{n}\sum_{i=1}^n |e^{-g_0(\boldsymbol{X}_i)}h_1(\boldsymbol{X}_i)h_2(\boldsymbol{X}_i)| \\
\xrightarrow{a.s.}&c_0\mathrm{E}|e^{-g_0(\boldsymbol{X}_1)}h_1(\boldsymbol{X}_1)h_2(\boldsymbol{X}_1)| \qquad\qquad (\lambda_1 \to 0, n\lambda_1^{2/r} \to \infty) \\
\leq&c_0\{(V^*+\lambda_1 J^*)(h_1)(V^*+\lambda_1 J^*)(h_2)\}^{1/2}.
\end{aligned}
$$

□

Note that conditions 7.1-7.5 are common assumptions for convergence rate analysis

of the SS ANOVA estimates, which were also made in Gu [12]. Condition 7.2 states that the growth rate of the eigenvalues γ_v is at v^r, which controls how fast λ_1 approaches zero. Condition 7.4 bounds e^{g_0-g} at g in a convex set B_0 around g_0. Condition 7.5 requires bounded fourth moment of ζ_v. We consider metrics $V^* + \lambda_1 J^*$ and $V + \lambda_1 J$.

Theorem 7.1 *Assume* $\sum_v \gamma_v^l a_{v,0}^2 < \infty$ *for some* $l \in [1, 2]$. *Under Conditions 7.1-7.5, suppose* $V^*(\hat{g} - \tilde{g}) < \infty$, *for some* $r > 1$, *as* $\lambda_1 \to 0$ *and* $n\lambda_1^{2/r} \to \infty$,

$$(V^* + \lambda_1 J^*)(\hat{g} - g_0) = O_p(n^{-1}\lambda_1^{-1/r} + \lambda_1^l).$$

Proof: Note that for each $\boldsymbol{X}_i$, $\mathrm{E}\{e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_i)\} = \int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} (\hat{g} - \tilde{g})^2(\boldsymbol{x})\rho(\boldsymbol{x}) dx_\alpha d\boldsymbol{x}_{\setminus\{\alpha\}} = V^*(\hat{g} - \tilde{g}) < \infty$. Since $\boldsymbol{X}_i$'s are independent, we have $e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_i)$'s are i.i.d. random variables with mean $\mathrm{E}\{e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_i)\} = V^*(\hat{g} - \tilde{g}) < \infty$. Therefore, by the strong law of large numbers, $\frac{1}{n}\sum_{i=1}^n e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_i) \stackrel{a.s.}{\to} \mathrm{E}\{e^{-g_0(\boldsymbol{X}_1)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_1)\}$ as $n \to \infty$. Substituting (7.8) into the left-hand side of (7.7), we have

$$\begin{aligned}
&\lambda_1 J^*(\hat{g} - \tilde{g}) - \frac{1}{n}\sum_{i=1}^n \left\{e^{-\hat{g}(\boldsymbol{X}_i)} - e^{-\tilde{g}(\boldsymbol{X}_i)}\right\}(\hat{g} - \tilde{g})(\boldsymbol{X}_i) \\
\geq &\frac{c_2}{n}\sum_{i=1}^n e^{-g_0(\boldsymbol{X}_i)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_i) + \lambda_1 J^*(\hat{g} - \tilde{g}) \\
\stackrel{a.s.}{\to} &c_2 \mathrm{E}\{e^{-g_0(\boldsymbol{X}_1)}(\hat{g} - \tilde{g})^2(\boldsymbol{X}_1)\} + \lambda_1 J^*(\hat{g} - \tilde{g}) \qquad (\lambda_1 \to 0, n\lambda_1^{2/r} \to \infty) \\
= &c_2 V^*(\hat{g} - \tilde{g}) + \lambda_1 J^*(\hat{g} - \tilde{g}). \qquad (7.12)
\end{aligned}$$

Substituting (7.10) and (7.11) into the right-hand side of (7.7) and let $h_1 = \hat{g} - \tilde{g}$,

$h_2 = \tilde{g} - g_0$, as $\lambda_1 \to 0$ and $n\lambda_1^{2/r} \to \infty$, we have

$$\begin{aligned}
&\Big|\frac{1}{n}\sum_{i=1}^{n}\left\{e^{-\tilde{g}(\boldsymbol{X}_i)} - e^{-g_0(\boldsymbol{X}_i)}\right\}(\hat{g}-\tilde{g})(\boldsymbol{X}_i) + V^*(\hat{g}-\tilde{g}, \tilde{g}-g_0)\Big| \\
\leq &\Big|V^*(\hat{g}-\tilde{g}, \tilde{g}-g_0) - \frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)(\tilde{g}-g_0)(\boldsymbol{X}_i)\Big| \\
&+ \Big|\frac{1}{n}\sum_{i=1}^{n} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)(\tilde{g}-g_0)(\boldsymbol{X}_i) - \frac{1}{n}\sum_{i=1}^{n} c_0^{(i)} e^{-g_0(\boldsymbol{X}_i)}(\hat{g}-\tilde{g})(\boldsymbol{X}_i)(\tilde{g}-g_0)(\boldsymbol{X}_i)\Big| \\
\leq_p &(o_p(1)+c_0)\{(V^*+\lambda_1 J^*)(\hat{g}-\tilde{g})(V^*+\lambda_1 J^*)(\tilde{g}-g_0)\}^{1/2}, \qquad (7.13)
\end{aligned}$$

where the first inequality follows (7.9) and the second inequality directly follows Lemma 7.2 and 7.3. Combining (7.7), (7.12), and (7.13), we obtain

$$(c_2 V^* + \lambda_1 J^*)(\hat{g}-\tilde{g}) \leq_p (o_p(1)+c_0)\{(V^*+\lambda_1 J^*)(\hat{g}-\tilde{g})(V^*+\lambda_1 J^*)(\tilde{g}-g_0)\}^{1/2}. \qquad (7.14)$$

Combining (7.14) with Lemma 7.1, as $\lambda_1 \to 0$ and $n\lambda_1^{2/r} \to \infty$, we have $(V^* + \lambda_1 J^*)(\hat{g} - \tilde{g}) = O_p(n^{-1}\lambda_1^{-1/r} + \lambda_1^l)$ and Theorem 7.1 holds. □

Theorem 7.2 *Under the conditions in Theorem 7.1,*

$$(V + \lambda_1 J)(\hat{g} - g_0) = O_p(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}).$$

Proof: We know

$$\sum_{k\neq\alpha} ||g_{\alpha k}(x_\alpha, x_k)||^2 \leq \left\{\sum_{k\neq\alpha} ||g_{\alpha k}(x_j, x_k)||\right\}^2 \leq (p-1)\sum_{k\neq\alpha} ||g_{\alpha k}(x_\alpha, x_k)||^2. \qquad (7.15)$$

Then, there exists some constant $C \in [1, \sqrt{p-1}]$ such that $C\{\sum_{k\neq\alpha} ||g_{\alpha k}(x_\alpha, x_k)||^2\}^{1/2} = \sum_{k\neq\alpha} ||g_{\alpha k}(x_\alpha, x_k)||$. Since $\sum_{k\neq\alpha} \theta_{\alpha k}$ is bounded by a fixed $M < \infty$, we can scale λ_1, λ_2 such

that $\theta_{\alpha k} \leq 1$. Since $J_2^*(g) = \sum_{k \neq \alpha} \theta_{\alpha k}^{-1} ||g_{\alpha k}(x_\alpha, x_k)||^2 = \boldsymbol{c}^T (\sum_{k \neq \alpha} \theta_{\alpha k} Q_{\alpha k}) \boldsymbol{c}$, $\sum_{k \neq \alpha} ||g_{\alpha k}(x_\alpha, x_k)||^2 = \boldsymbol{c}^T (\sum_{k \neq \alpha} \theta_{\alpha k}^2 Q_{\alpha k}) \boldsymbol{c}$, we have $J_2^2(g) = C^2 \sum_{k \neq \alpha} ||g_{\alpha k}(x_\alpha, x_k)||^2 \leq C^2 J_2^*(g)$ and consequently $J_2 \leq C(J^*)^{1/2}$. Furthermore, $V_2^2(g^{(2)}) = \int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} \left\{g^{(2)}(\boldsymbol{x})\right\}^2 \rho(\boldsymbol{x}) dx_\alpha d\boldsymbol{x}_{\setminus\{\alpha\}}$ $= V^*(g^{(2)})$, we have $V_2(g^{(2)}) = [V^*(g^{(2)})]^{1/2}$. Therefore,

$$(V_2 + \lambda_1 J_2)(g^{(2)}) = ((V^*)^{1/2} + C\sqrt{\lambda_1}(\lambda_1 J^*)^{1/2})(g^{(2)}) \leq (1 + C^2\lambda_1)^{1/2}(V^* + \lambda_1 J^*)^{1/2}(g^{(2)})$$

by the Cauchy-Schwarz inequality. Finally,

$$\begin{aligned}
(V + \lambda_1 J)(\hat{g} - g^0) &= (V_1 + \lambda_1 J_1)(\hat{g}^{(1)} - g_0^{(1)}) + (V_2 + \lambda_1 J_2)(\hat{g}^{(2)} - g_0^{(2)}) \\
&\leq (V^* + \lambda_1 J^*)(\hat{g}^{(1)} - g_0^{(1)}) + (1 + C^2\lambda_1)^{1/2}(V^* + \lambda_1 J^*)^{1/2}(\hat{g}^{(2)} - g_0^{(2)}) \\
&= O_p(n^{-1}\lambda_1^{-1/r} + \lambda_1^l) + O(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}) \\
&= O_p(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}). \qquad (7.16)
\end{aligned}$$

□

Corollary 7.1 *Assume conditions in Theorem 7.2 hold,* $0 < c_5 < \rho(\boldsymbol{x}) < c_6$ *and* $0 < c_7 < f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) < c_8$ *for some positive constants* c_5, c_6, c_7, c_8*, we have*

$$||\hat{g}_{\alpha k} - g_{0\alpha k}|| = O_p(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}), \quad k \neq \alpha,\ k = 1, \cdots, p,$$

where $g_{0\alpha k}$ *are two-way interactions in the true function* g_0*.*

Proof: By definition of $V(\cdot)$, $V(\hat{g} - g^0) = V_1(\hat{g}^{(1)} - g_0^{(1)}) + V_2(\hat{g}^{(2)} - g_0^{(2)}) = V^*(\hat{g}^{(1)} - g_0^{(1)}) + [V^*(\hat{g}^{(2)} - g_0^{(2)})]^{1/2}$. Following (7.16),

$$[V^*(\hat{g}^{(2)} - g_0^{(2)})]^{1/2} = O_p(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}).$$

Following Lin et al. [28], under the condition $0 < c_5 < \rho(\boldsymbol{x}) < c_6$ and $0 < c_7 < f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) < c_8$ for some positive constants c_5, c_6, c_7, c_8, $[V^*(g)]^{1/2}$ is equivalent to the L_2 norm. Specifically,

$$V^*(g) \sim ||g||^2 = \sum_{j=1}^{p} ||g_j||^2 + \sum_{k \neq \alpha} ||g_{\alpha k}(x_\alpha, x_k)||^2,$$

where $\sim$ means equivalence, $||\cdot||$ is the L_2 norm, and $V^*(g^{(1)}) \sim \sum_{j=1}^{p} ||g_j||^2$, $V^*(g^{(2)}) \sim \sum_{k \neq \alpha} ||g_{\alpha k}(x_\alpha, x_k)||^2$, respectively.

By definition, $V(g^{(2)}) = [V^*(g^{(2)})]^{1/2} \sim (\sum_{k \neq \alpha} ||g_{\alpha k}(x_\alpha, x_k)||^2)^{1/2}$. Consequently, two-way interactions under L_2 norm have the same convergence rate as $[V^*(g^{(2)})]^{1/2}$,

$$||\hat{g}_{\alpha k} - g_{0\alpha k}|| = O_p(n^{-1/2}\lambda_1^{-1/2r} + \lambda_1^{l/2}), \quad k \neq \alpha,\ k = 1, \cdots, p.$$

□

The convergence rate in Theorem 7.2 is the square root of the rate in Theorem 7.1. This is because $V^* + \lambda_1 J^*$ is associated with the square of L_2 norm while the L_2 norm was used in $V + \lambda_1 J$. Corollary 7.1 holds because V_2 and J_2 associated with two-way interactions are equivalent to L_2 norm. Consequently, two-way interactions under L_2 norm have the same convergence rate as Theorem 7.2. We only show convergence rate for interactions in Corollary 7.1 since we are mainly interested in edge selection.

Chapter 8

Simulation Results

In this chapter, we conduct simulations to evaluate the performance of the proposed method. We consider continuous random variables only such that we can compare them with some existing methods. In our implementation, we estimate the conditional density for each variable on the data range and transform the data into $[0, 1]$. We construct an SS ANOVA model using the tensor product of cubic spline models. Specifically, let $\mathcal{H}^{(j)} = W_2^2[0, 1]$ where

$$W_2^2[0,1] = \left\{ f : f, f' \text{ are absolutely continuous}, \int_0^1 (f'')^2 dx < \infty \right\} \tag{8.1}$$

is the Sobolev space for cubic spline models. Each $\mathcal{H}^{(j)}$ can be decomposed as $\mathcal{H}^{(j)} = \{1_{(j)}\} \oplus \mathcal{H}_{(j)}$ and $\mathcal{H}_{(j)} = \mathcal{H}^0_{(j)} \oplus \mathcal{H}^1_{(j)}$ where $\mathcal{H}^0_{(j)}$ and $\mathcal{H}^1_{(j)}$ are RKHS's with RKs $R^0_j(x, z) = k_1(x)k_1(z)$ and $R^1_j(x, z) = k_2(x)k_2(z) - k_4(|x - z|)$ respectively, $k_1(x) = x - 0.5$, $k_2(x) = \frac{1}{2}(k_1^2(x) - \frac{1}{12})$, and $k_4(x) = \frac{1}{24}(k_1^4(x) - \frac{k_1^2(x)}{2} + \frac{7}{240})$. SS ANOVA decomposition of $\bigotimes_{j=1}^{p} \mathcal{H}^{(j)}$ can then be constructed based on these decompositions. More details can be found in Wang [41]. In all simulations and real data applications, when using the pseudo-likelihood

method, we choose ρ as

$$\rho(x_\alpha, \boldsymbol{x}_{\setminus\{\alpha\}}) = \frac{\phi((x_\alpha - \mu(\boldsymbol{x}_{\setminus\{\alpha\}}))/\sigma)}{\Phi((1 - \mu(\boldsymbol{x}_{\setminus\{\alpha\}}))/\sigma) - \Phi((-\mu(\boldsymbol{x}_{\setminus\{\alpha\}}))/\sigma)}, \tag{8.2}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the standard normal density and CDF, and $\mu(\cdot)$ and σ are estimated by fitting a nonparametric regression model in model space (1.12) with covariates $\boldsymbol{x}_{\setminus\{\alpha\}}$. More estimation details can be found in Chapter 3 of Gu [12]. We select the tuning parameter M using the 5-fold cross-validation method.

We compare the proposed method with four existing parametric and semiparametric methods: `space` (Sparse PArtial Correlation Estimation) (Peng et al. [32]), `QUIC` (QUadratic Inverse Covariance estimation) (Hsieh et al. [20]), nonparanormal (NPN) (Liu et al. [29]), and SpaCE JAM (Voorman et al. [40]). Due to computational constraints, we will not compare the proposed method with the nonparametric joint density estimation method in Gu et al. [13]. Based on our experience, the joint method becomes computationally infeasible when p is large due to memory restrictions.

The `space` method assumes that $\mathrm{E}(\boldsymbol{X}) = \mathbf{0}$ and $\mathrm{Cov}(\boldsymbol{X}) = \Sigma$. Denote the precision matrix $\Omega = \Sigma^{-1} = (\sigma^{ij})_{p\times p}$ and $\rho^{ij} = -\sigma^{ij}/\sqrt{\sigma^{ii}\sigma^{jj}}$ as the partial correlation between X_i and X_j. Denote $\boldsymbol{x}_{(i)} = (x_{1,i}, \cdots, x_{n,i})^T$ as the vector of n observations on the ith variable, $i = 1, \cdots, p$. Peng et al. [32] solved the following regularization problem for edge selection

$$\frac{1}{2}\Big(\sum_{i=1}^{p} w_i ||\boldsymbol{x}_{(i)} - \sum_{j\neq i} \rho^{ij}\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}\boldsymbol{x}_{(j)}||^2\Big) + \lambda \sum_{1\leq i<j\leq p} |\rho^{ij}|, \tag{8.3}$$

where $\boldsymbol{w} = \{w_i\}_{i=1}^p$ are non-negative weights. This method is implemented with R package `space`. We select the tuning parameter λ using the 5-fold cross-validation method (Lafit et al. [25]).

The QUIC method assumes that $\boldsymbol{X}$ is multivariate Gaussian and learns the precision matrix Ω by solving the following penalized likelihood

$$-\log\det(\Omega) + \mathrm{tr}(S\Omega) + \lambda||\Omega||_1, \tag{8.4}$$

where $||\cdot||_1$ is the L_1 penalty, S is the sample covariance matrix, and λ is the tuning parameter which is selected by minimizing the BIC score. The method is implemented with R package `QUIC`.

The NPN method assumes that there exists some monotone functions $f_1, \cdots, f_p$ such that $f(\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ where $f(\boldsymbol{X}) = (f_1(X_1), \cdots, f_p(X_p))^T$. The NPN is a semiparametric model since it consists of parameters $\boldsymbol{\mu}$ and Σ and nonparametric transformations f's. The graphical lasso is applied to the transformed data to estimate the undirected graph, and the tuning parameter is selected by the extended BIC score (Foygel and Drton [9]). Estimation details were given in Liu et al. [29]. We use R package `huge` to implement the NPN method.

The SpaCE JAM method models the conditional mean nonparametrically using additive models: $\mathrm{E}(X_j|\boldsymbol{X}_{\setminus\{j\}}) = \sum_{k\neq j} f_{jk}(X_k)$ where $f_{jk}(\cdot)$ belongs to a functional space $\mathcal{F}$ (Voorman et al. [40]). The functions f_{jk} are estimated as the minimizers of the following least squares with a group lasso type penalty:

$$\mathrm{argmin}_{f_{jk}\in\mathcal{F}} \left\{ \frac{1}{2n}\sum_{j=1}^{p} ||\boldsymbol{x}_{(j)} - \sum_{k\neq j} \boldsymbol{s}_{jk}||_2^2 + \lambda \sum_{k>j} \left(||\boldsymbol{s}_{jk}||_2^2 + ||\boldsymbol{s}_{kj}||_2^2\right)^{1/2} \right\}, \tag{8.5}$$

where $\boldsymbol{s}_{jk} = (f_{jk}(x_{1,k}), \cdots, f_{jk}(x_{n,k}))^T$ and $\boldsymbol{s}_{kj} = (f_{kj}(x_{1,j}), \cdots, f_{kj}(x_{n,j}))^T$. The SpaCE JAM method is implemented with R package `spacejam` (Voorman et al. [40]) and cubic basis function is used to allow non-linear conditional relationships among variables. The tuning parameter λ is selected by the BIC method. Note that `space` is a neighborhood

selection method while `QUIC`, `NPN`, and `SpaCE JAM` are global methods.

To evaluate the performance of edge detection, we compute three criteria: specificity (SPE), sensitivity (SEN), and F_1 scores, which are defined the same way as Chapter 4. We consider both Gaussian and non-Gaussian simulation settings to evaluate the performance of edge detection. We set dimension $p = 20$ and consider two sample sizes $n = 150$ and $n = 300$. All simulations are repeated 100 times.

8.1 Gaussian Simulation

We first use `huge.generator` function to randomly generate a $p \times p$ sparse precision matrix Ω, where the probability p_{off} of the off-diagonal elements being non-zero is equal to 0.2 or 0.4. Then we generate n i.i.d. samples $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n$ from $\mathcal{N}(\mathbf{0}, \Omega^{-1})$. We apply the proposed method, `space`, `QUIC`, `NPN`, and `SpaCE JAM` methods to select edges.

Tables 8.1, 8.2 presents averages and standard deviations of sensitivity, specificity, and F_1 score. Different methods have different trade-offs between sensitivity and specificity. In terms of F_1 score, the proposed method has slightly better performance than `QUIC`, `space`, and `SpaCE JAM` in most cases. This indicates that the proposed nonparametric method is as efficient as the parametric `QUIC` method when the Gaussian assumption holds. The NPN method has worse performance than other methods.

8.2 Non-Gaussian Simulation

In general, it is difficult to construct a flexible multivariate nonparametric distribution as discussed in Section 2 in Voorman et al. [40]. To overcome this problem, we use the same approach in Voorman et al. [40] to generate a graphical model using a directed

	Proposed Method			`space`			QUIC		
	SPE	SEN	F_1	SPE	SEN	F_1	SPE	SEN	F_1
$p_{\text{off}} = 0.2$									
n=150	0.912	0.968	0.834	0.986	0.654	0.762	0.768	0.964	0.666
	(0.028)	(0.037)	(0.050)	(0.011)	(0.096)	(0.075)	(0.036)	(0.034)	(0.042)
n=300	0.929	0.998	0.870	0.989	0.869	0.907	0.813	0.982	0.712
	(0.026)	(0.008)	(0.046)	(0.01)	(0.059)	(0.04)	(0.032)	(0.025)	(0.04)
$p_{\text{off}} = 0.4$									
n=150	0.866	0.815	0.807	0.969	0.461	0.599	0.668	0.797	0.691
	(0.040)	(0.066)	(0.046)	(0.013)	(0.062)	(0.058)	(0.047)	(0.058)	(0.030)
n=300	0.883	0.968	0.903	0.961	0.564	0.681	0.689	0.827	0.717
	(0.042)	(0.027)	(0.028)	(0.015)	(0.058)	(0.049)	(0.043)	(0.046)	(0.026)

Table 8.1: Averages and standard deviations (in parentheses) of specificity (SPE), sensitivity (SEN), and F_1 score for the proposed method, `space`, QUIC in Gaussian simulation.

	NPN			SpaCE JAM		
	SPE	SEN	F_1	SPE	SEN	F_1
$p_{\text{off}} = 0.2$						
n=150	0.820	0.751	0.521	0.939	0.798	0.776
	(0.108)	(0.402)	(0.281)	(0.035)	(0.147)	(0.068)
n=300	0.762	0.995	0.668	0.945	0.954	0.875
	(0.042)	(0.016)	(0.042)	(0.022)	(0.041)	(0.037)
$p_{\text{off}} = 0.4$						
n=150	0.793	0.617	0.474	0.945	0.508	0.608
	(0.142)	(0.411)	(0.312)	(0.025)	(0.135)	(0.126)
n=300	0.673	0.951	0.706	0.905	0.704	0.727
	(0.039)	(0.036)	(0.021)	(0.031)	(0.128)	(0.076)

Table 8.2: Averages and standard deviations (in parentheses) of specificity (SPE), sensitivity (SEN), and F_1 score for NPN, SpaCE JAM in Gaussian simulation.

acyclic graph (DAG) and conditional distributions. We use the `rdag` function in the `spacejam` package to generate a DAG of $\boldsymbol{X}$ and denote E_D as the directed edge set. The conditional relationships among variables can be created via $\mathrm{E}(X_j|\boldsymbol{X}_{\setminus\{j\}}) = \sum_{k\neq j} f_{jk}(X_k)$. The distribution of $\boldsymbol{X}$ is usually not a well-known multivariate distribution except for the particular case when all f_{jk}s are linear associated with a multivariate Gaussian distribution.

We decompose $\boldsymbol{X}^T = (\boldsymbol{Y}^T, \boldsymbol{Z}^T)$ where $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are random vectors of dimensions 5 and 15 respectively. We first generate a DAG with $p = 20$ nodes and m edges selected at random from all possible $p(p-1)/2$ possible edges. We consider two choices of m: $m = 20$ and $m = 40$. Given a DAG, we generate data as follows:

$$Z_j|\{Z_k, Y_s : \{k,j\}, \{s,j\} \in E_D\} = \sum_{\{k,j\}\in E_D} f_{jk}^{(1)}(Z_k) + \sum_{\{s,j\}\in E_D} f_{js}^{(1)}(Y_s) + \epsilon_j$$
$$Y_j|\{Y_k : \{k,j\} \in E_D\} = \sum_{\{k,j\}\in E_D} f_{jk}^{(2)}(Y_k) + \epsilon_j,$$

where ϵ_j's are i.i.d. random noises from the standard normal distribution, $f_{jk}^{(1)}(t) = b_{jk,1}^{(1)} t$ with $b_{jk,1}^{(1)}$ generated from the standard Gaussian distribution, and $f_{jk}^{(2)}(t) = b_{jk,1}^{(2)} t + b_{jk,2}^{(2)} t^2 + b_{jk,3}^{(2)} t^3$ with $b_{jk,1}^{(2)}, b_{jk,2}^{(2)}$ and $b_{jk,3}^{(2)}$ independently generated from the Gaussian distributions with mean zero and variances 1, 0.3, and 0.1, respectively.

Simulation results are shown in TableS 8.3, 8.4. Since data are generated according to a model assumed by the `SpaCE JAM` method, as expected, the `SpaCE JAM` performs better than the `space`, `QUIC`, and `NPN` methods. Remarkably, the proposed method has larger F_1 scores than `SpaCE JAM` in all cases. In conclusion, the proposed method is efficient in edge detection and performs better than some existing methods.

	Proposed Method			`space`			QUIC		
	SPE	SEN	F_1	SPE	SEN	F_1	SPE	SEN	F_1
$m = 20$									
$n = 150$	0.97	0.835	0.840	0.997	0.588	0.730	0.808	0.838	0.588
	(0.020)	(0.074)	(0.066)	(0.004)	(0.084)	(0.068)	(0.034)	(0.079)	(0.050)
$n = 300$	0.984	0.917	0.915	0.998	0.631	0.767	0.859	0.854	0.660
	(0.013)	(0.064)	(0.050)	(0.003)	(0.075)	(0.058)	(0.035)	(0.079)	(0.055)
$m = 40$									
$n = 150$	0.970	0.598	0.725	0.984	0.359	0.517	0.705	0.671	0.627
	(0.020)	(0.064)	(0.05)	(0.012)	(0.039)	(0.041)	(0.039)	(0.053)	(0.031)
$n = 300$	0.985	0.685	0.800	0.982	0.430	0.587	0.740	0.673	0.645
	(0.014)	(0.067)	(0.049)	(0.013)	(0.054)	(0.050)	(0.046)	(0.049)	(0.036)

Table 8.3: Averages and standard deviations (in parentheses) of specificity (SPE), sensitivity (SEN), and F_1 score for the proposed method, `space`, QUIC in nonparametric distribution simulation.

	NPN			SpaCE JAM		
	SPE	SEN	F_1	SPE	SEN	F_1
$m = 20$						
$n = 150$	0.83	0.791	0.588	0.963	0.697	0.736
	(0.066)	(0.125)	(0.054)	(0.019)	(0.079)	(0.062)
$n = 300$	0.818	0.894	0.629	0.979	0.716	0.786
	(0.041)	(0.082)	(0.052)	(0.057)	(0.089)	(0.072)
$m = 40$						
$n = 150$	0.671	0.708	0.634	0.690	0.710	0.643
	(0.042)	(0.060)	(0.031)	(0.111)	(0.112)	(0.044)
$n = 300$	0.707	0.724	0.662	0.654	0.814	0.692
	(0.045)	(0.048)	(0.038)	(0.07)	(0.051)	(0.031)

Table 8.4: Averages and standard deviations (in parentheses) of specificity (SPE), sensitivity (SEN), and F_1 score for NPN, SpaCE JAM in nonparametric distribution simulation.

Chapter 9

Real Data Examples

In this chapter, We illustrate our neighborhood selection method using two real datasets. In Section 9.1, we apply our method to Arabidopsis Thaliana gene expression data and compare the estimated graph with those from `space`, `QUIC`, `NPN`, and `SpaCE JAM`. In addition, we present a diagnostic procedure for some existing methods. In Section 9.2, we illustrate our method using real data with a mixed data type.

9.1 Isoprenoid Gene Network in Arabidopsis Thaliana

In this section, we consider the gene expression data for *Arabidopsis thaliana*, which is an important plant species in molecular biology and genetics studies. There are $n = 118$ observations of Affymetrix GeneChip microarrays in the dataset, where a subset of $p = 39$ genes from the isoprenoid pathway is selected for analysis. The dataset was introduced in Wille et al. [43] and can be downloaded from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC545783/`. Lafferty et al. [24] also analyzed this dataset using the nonparanormal method.

All observations are preprocessed by log-transformation and standardization as in

Lafferty et al. [24]. We build a graph for all 39 gene expression levels using the proposed method and compare its structure with those from `space`, `QUIC`, `NPN`, and `SpaCE JAM`. Wille et al. [43] stated that the GGM selection using the BIC method usually detects too many edges for biologically relevant analysis. Therefore, we limit the number of edges in the graph by controlling the regularization parameters as in Lafferty et al. [24]. Specifically, we tune M such that the number of edges $|E| = 52$. Similarly, by tuning the regularization parameters in `space`, `QUIC`, `NPN`, and `SpaCE JAM`, we select the graphs with the same number of edges $|E| = 52$.

Figure 9.1 presents graphs with $|E| = 52$ for all four methods. These five graphs have some common edges, for example, edges `1-27`, `1-33`, `2-28`, `2-30`, `2-34`, `2-35`, `3-32`, `3-33`, `3-39`, `5-37`, `10-26`, `10-33`, `10-39`, `11-36`, `12-29`, `12-30`, `12-34`, `12-35`, `22-39`, `23-33`, `25-37`, `28-34`, `34-35`, and `37-38`. There are also some interesting differences. For instance, only our proposed method detects the edge `16-21`. We now describe a general diagnostic procedure to explain reasons why other methods miss this edge.

We first extend the squared error projection in Gu [12] for diagnostics on any subspaces of $\mathcal{M}_\alpha$. Let

$$\tilde{V}(\hat{g} - g) = \int_{\mathcal{X}_{\setminus\{\alpha\}}} f_{\setminus\{\alpha\}}(\boldsymbol{x}_{\setminus\{\alpha\}}) \int_{\mathcal{X}_\alpha} \left\{(\hat{g} - g)(\boldsymbol{x}) - \int_{\mathcal{X}_\alpha} (\hat{g} - g)(\boldsymbol{x})\rho(\boldsymbol{x})\right\}^2 \rho(\boldsymbol{x}) dx_\alpha d\boldsymbol{x}_{\setminus\{\alpha\}} \tag{9.1}$$

where $\hat{g} \in \mathcal{M}_\alpha \ominus \{1\}$. We remove the constant functions from the model space since they are not relevant to the diagnostics on interactions. $\tilde{V}(\hat{g} - g)$ can be treated as a proxy of the symmetrized Kullback-Leibler distance (Gu [12]). For any decomposition $\mathcal{M}_\alpha \ominus \{1\} = \mathcal{M}_\alpha^0 \oplus \mathcal{M}_\alpha^1$, the squared error projection of $\hat{g}$ in $\mathcal{M}_\alpha^0$ is defined as $\tilde{g} = \underset{g \in \mathcal{M}_\alpha^0}{\arg\min} \left\{\tilde{V}(\hat{g} - g)\right\}$. It can be shown that $\tilde{V}(\hat{g} - g_u) = \tilde{V}(\hat{g} - \tilde{g}) + \tilde{V}(\tilde{g} - g_u)$ when $g_u = -\log \rho(\mathbf{x}) \in \mathcal{M}_\alpha^0$. The ratio $\tilde{V}(\hat{g} - \tilde{g})/\tilde{V}(\hat{g} - g_u)$ represents the contribution of

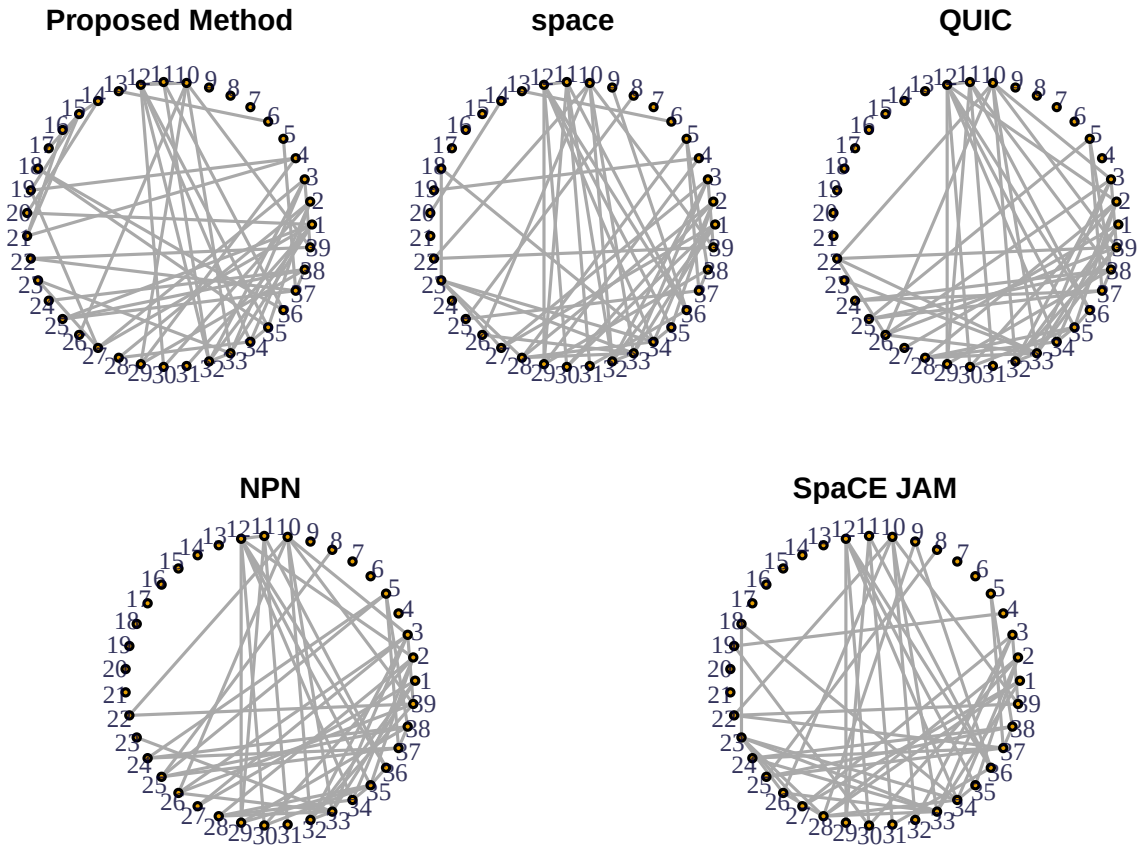

Figure 9.1: The estimated graph with 52 edges from the proposed method (*top left*), the `space` (*top middle*), the QUIC (*top right*), the NPN (*bottom left*), the SpaCE JAM (*bottom right*).

functions in the subspace $\mathcal{M}_\alpha^1$ which can be dropped when the ratio is small (Gu et al. [13]).

Now we apply the diagnostic procedure to explain why our proposed method detects the edge `16-21` which is missed by other methods. Note that each interaction space $\mathcal{H}_{(\alpha k)} = \mathcal{H}_{(\alpha k)}^{(0)} \oplus \mathcal{H}_{(\alpha k)}^{(1)} \oplus \mathcal{H}_{(\alpha k)}^{(2)} \oplus \mathcal{H}_{(\alpha k)}^{(3)}$ where $\mathcal{H}_{(\alpha k)}^{(0)} = \mathcal{H}_{(\alpha)}^{0} \otimes \mathcal{H}_{(k)}^{0}$, $\mathcal{H}_{(\alpha k)}^{(1)} = \mathcal{H}_{(\alpha)}^{0} \otimes \mathcal{H}_{(k)}^{1}$, $\mathcal{H}_{(\alpha k)}^{(2)} = \mathcal{H}_{(\alpha)}^{1} \otimes \mathcal{H}_{(k)}^{0}$, and $\mathcal{H}_{(\alpha k)}^{(3)} = \mathcal{H}_{(\alpha)}^{1} \otimes \mathcal{H}_{(k)}^{1}$ correspond to linear-linear, linear-smooth, smooth-linear, and smooth-smooth interactions (Wang [41]). The `QUIC` and

`space` are special cases with $g_{\alpha k} \in \mathcal{H}^{(0)}_{(\alpha k)}$, and the `SpaCE JAM` is a special cases with $g_{\alpha k} \in \mathcal{H}^{(0)}_{(\alpha k)} \oplus \mathcal{H}^{(1)}_{(\alpha k)}$. Therefore, for diagnostics of `QUIC` and `SpaCE JAM` methods, we consider the contribution of $\mathcal{M}^1_\alpha = \mathcal{M}_\alpha \ominus \{1\} \ominus \mathcal{H}^{(0)}_{(\alpha k)}$ and the contribution of $\mathcal{M}^1_\alpha = \mathcal{M}_\alpha \ominus \{1\} \ominus \mathcal{H}^{(0)}_{(\alpha k)} \ominus \mathcal{H}^{(1)}_{(\alpha k)}$, respectively. For the edge `16-21`, we have $\tilde{V}(\hat{g} - \tilde{g})/\tilde{V}(\hat{g} - g_u) = 0.352$ for `QUIC` and $\tilde{V}(\hat{g} - \tilde{g})/\tilde{V}(\hat{g} - g_u) = 0.340$ for `SpaCE JAM`, respective. These non-ignorable contributions suggest that the assumptions of the `QUIC` and `SpaCE JAM` methods are likely violated.

9.2 Conditional Dependence Among Demographic, Clinical, Laboratory and Treatment Variables of Hemodialysis Patients

In this section, we illustrate the application of the proposed methods to mixed binary and continuous variables using a data set collected from hemodialysis patients. The data include patients who received dialysis treatments from 2010 to 2014 and stayed at the Fresenius Medical Care - North America throughout their treatments. To reduce heterogeneity, we include $n = 2932$ non-diabetic and non-Hispanic patients who used arteriovenous fistula for dialysis access and survived longer than two years. We use the averages of measurements in the second year of dialysis for analysis. We consider the following 23 variables: demographic variables including `race` (white and non-white) and `gender` (male and female); clinical variables including `height` (cm), `weight` (kg), `sbp` (systolic blood pressure, mmHg), `dbp` (diastolic blood pressure, mmHg), and `temp` (temperature, Celsius); laboratory variables including `albumin` (g/dL), `ferritin` (ng/mL), `hgb` (hemoglobin, g/dL), `lymphocytes` (%), `neutrophils` (%), `nlr` (neutrophils to lymphocytes ratio, unitless), `sna` (serum sodium concentration, mEq/L), `wbc` (white blood

cell, 1000/mc); and treatment variables including `qb` (blood flow, mL/min), `qd` (dialysis flow, mL/min), `saline` (mL), `olc` (on-line clearance, unitless), `idwg` (interdialytic weight gain, kg), `ufv` (ultrafiltration volume, L), `ufr` (ultrafiltration rate, mL/hr/kg), and `epodose` (erythropoietin dose, unit).

We have 2 binary variables, `race` and `male`, and 21 continuous variables. We apply the logistic regression approach described in the Supplement to estimate the conditional density of each binary variable, and the pseudo-likelihood to estimate the conditional density of each continuous variable. We apply the BIC method to select the tuning parameter M and the AND rule to decide edges. Figure 9.2 shows the estimated graph which contains some of the expected dependences between variables such as `gender` and `height`, `weight` and `height`, and `sbp` and `dbp`. The link between `ufv` and `idwg` is also well-known (Uduagbamen et al. [39]). Many other edges corroborate with existing literature. For example, anemia is a common complication of dialysis patients, and its management is a major challenge. A central aim of anemia management is to maintain patients' hemoglobin level consistently within a target range. Erythropoietin has been used to raise hemoglobin level, which is revealed by the edge between `epodose` and `hgb`. Serum albumin has been found strongly associated with erythropoietin sensitivity (Agarwal et al. [1]), which is corroborated by the edge between `epodose` and `albumin`. It has been found that black patients receive greater doses of erythropoietin than white patients (Lacson et al. [23]), which is corroborated by the edge between `epodose` and `race`. The graph in Figure 9.2 provides a holistic view of complex relationships between demographic, clinical, laboratory and treatment variables and help building new theories to be tested in future studies.

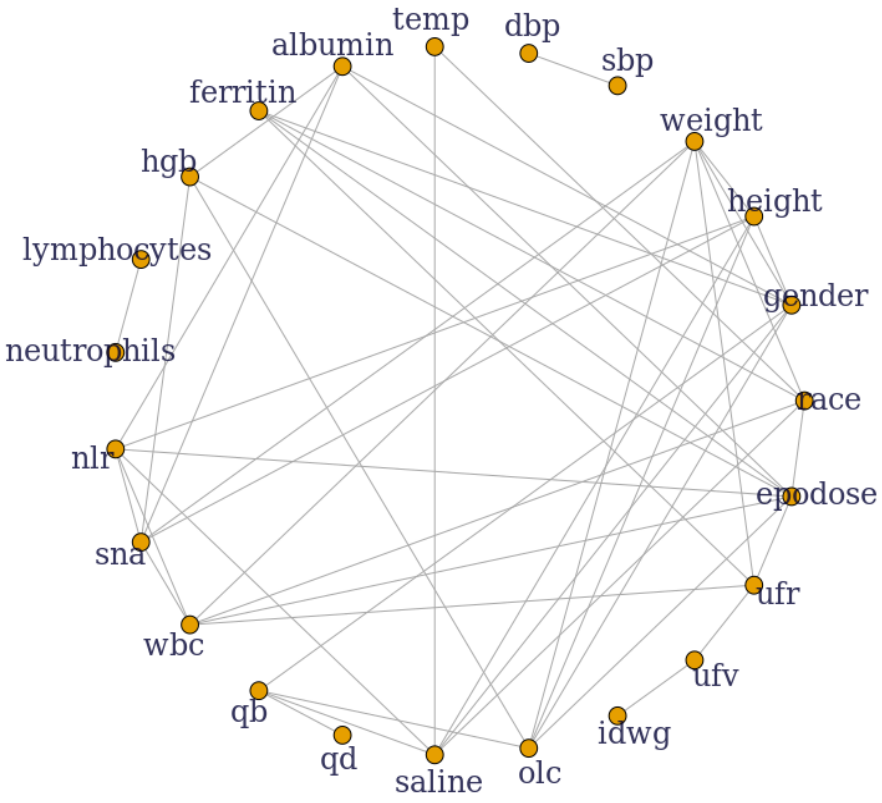

Figure 9.2: The estimated graph for dialysis data.

Part 3

R Package `edgeSelection` and Conclusions

Chapter 10

Package Description

We create an `R` package named `edgeSelection`, which can be installed from GitHub. Detailed codes and descriptions can be found in `https://github.com/haodongucsb/edgeSelection`. In this chapter, we provide a brief introduction to this package with some examples to illustrate how to perform edge selection using these two methods.

10.1 Introduction

The `edgeSelection` package unifies two edge selection methods into one main function named `edge.selection`, which is a wrapper of two functions, `selection.joint` and `selection.neighborhood`, specified by the argument `method= "joint"` or `method = "neighborhood"`. Specifically, the main function can be called with the following syntax:

```
edge.selection(data, method = c("joint", "neighborhood"), ...)
```

The required arguments include:

- `data` Data Frame containing all variables.
- `method` Which method should be used for edge selection.

Other optional arguments are in ..., which can be specified based on the user's needs. Some of those arguments are shared by these two methods, including `type, alpha, subset, na.action, seed, prec, maxiter, id.basis, nbasis`. They work the same way as in `gss` package (Gu et al. [17]). The remaining optional arguments are specifically related to the joint density approach or the neighborhood selection approach, and we provide those details in Section 10.2 and 10.3.

The returned value is `edgeMatrix`, which is the estimated graph structure in a $p \times p$ matrix, where p is the dimension of data. The jkth element in `edgeMatrix` is an estimate of θ_{jk}. Non-zero jkth and kjth elements indicate that there is an edge between X_j and X_k.

10.2 Joint Density Approach

`selection.joint` function select edges using the joint density approach. Syntax of this function is:

```
selection.joint(data, ...)
```

where `data` is the data frame from `edge.selection`. Besides those common arguments shared by two methods, other optional arguments in `...` include `domain, quad, w`. `domain, quad` work the same way as in the function `ssden1` in `gss` package. `w` is an optional vector to specify pre-defined weights of two-way interactions with default values as all ones. This function returns `edgeMatrix` as the edge selection result.

`selection.joint` is only feasible when the dimension p is small. Our experience indicates that the joint method becomes almost infeasible when the dimension is large because of memory restriction.

10.3 Neighborhood Selection Approach

`selection.neighborhood` function select edges using the neighborhood selection approach. This function is called in the following syntax:

```
selection.neighborhood(data, ...)
```

where `data` is the data frame from `edge.selection`. Other optional arguments in `...` include `rho, ydomain, yquad, skip.iter, W, neighborhoodMethod`. `rho, ydomain, yquad, skip.iter` work the same way as in the function `sscden1` in `gss` package. `W` is an optional matrix to specify pre-defined weights for two-way interactions in each conditional density with default values as all ones. `neighborhoodMethod` can specify which method to select tuning parameter M. It can be `"cv"` or `"BIC"`. This function also returns `edgeMatrix` as the edge selection result.

`selection.neighborhood` can be used for high-dimensional data, and a parallel backend can be set up for parallel computation. To use parallel computation, packages `doMC, foreach` usually need to be installed and loaded first, and multiple cores are necessary for the machine. Section 10.4.2 gives one example for the set up of parallel computation. If a parallel backend is not applicable, the usual `for` loop is used in the `selection.neighborhood` function.

10.4 Examples

10.4.1 `edge.selection` for Joint Density Approach

We first simulate a 5-dimensional data, which follows multivariate normal distribution as in Section 4.2 and apply the joint density approach to perform edge selection.

```
library(MASS)
simu5 <- function(n) {
```

```
   inverseSigma <- matrix(c(62, -20, 0, 0, -20, -20, 62, -10, 0, 0, 0, -10,
62, 10, 0, 0, 0, 10, 62, -15, -20, 0, 0, -15, 62), 5)
   Sigma <- solve(inverseSigma)
   data5 <- MASS::mvrnorm(n, rep(0.5, 5), Sigma)
   data5
}
set.seed(5732)
n <- 600
data <- as.data.frame(simu5(n))
edgeMatrix <- edge.selection(data = data, method = "joint")
```

This `edgeMatrix` selects all true edges (larger than zero in the off-diagonals) and exclude all false edges (zeros in the off-diagonals).

We also apply the joint density approach for a real data set `NO2` in the `gss` package, which has been introduced in Section 5.1.

```
library(gss)
data(NO2)
edgeMatrix <- edge.selection(data = NO2, method = "joint",nbasis = 100)
```

where we specify `nbasis`, the number of observations to fit smoothing spline model as 100. The plot of the estimated graph is shown in Figure 5.1.

10.4.2 `edge.selection` for Neighborhood Selection Approach

In the first example, we provide details to set up the parallel computation. If a parallel backend is not applicable, `edge.selection` still works by using a `for` loop inside. We simulate multivariate normal distribution using the `huge` package as in Section 8.1 and apply the neighborhood selection approach to select two-way interactions.

```
library(doMC)
library(foreach)
library(huge)
registerDoMC(20)
n <- 200; p <- 20
set.seed(5732)
z <- huge.generator(n, d = p, graph = "random", prob = .2, verbose =
FALSE, vis = FALSE, v = .65) data <- data.frame(z$data)
edgeMatrix <- edge.selection(data = data, method = "neighborhood")
trueEdges <- as.matrix(z$theta)
```

To see the accuracy of edge selection, one can compare the estimated graph structure in `edgeMatrix` with the ground truth in `trueEdges`.

We also look at the edge detection on the Arabidopsis Thaliana gene expression data in Section 9.1, and this data set has been included in our package.

```
data(Gene)
edgeMatrix <- edge.selection(data = Gene, method = "neighborhood",
neighborhoodMethod = "BIC")
```

where we specify `neighborhoodMethod`, the method to select tuning parameter M as `"BIC"`. The plot of the estimated graph is shown in Figure 9.1.

Chapter 11

Conclusions

11.1 Joint Approach

In the first part of this dissertation, we present a nonparametric method to learn edges for pairwise graphical models under the SS ANOVA decomposition. The proposed method provides a unified framework without restrictions on data types for each variable. The joint density function is estimated via a penalized pseudo-likelihood method with L_2 penalty on main effects and L_1 penalty on two-way interactions. We propose an iterative procedure to compute the estimates. We establish convergence rates of joint density function estimate and interaction components in the SS ANOVA decomposition. In simulation studies, we compare our method with Jeon and Lin's method (details in Section 1.6.1), which applied L_1 penalty on both main effects and two-way interactions. Simulation results showed our method has better overall performance in both the F_1 score and the ROC curve. In real applications, the estimated graphs are compared among our method, Jeon and Lin's method, and Gu's method (details in Section 1.6.2). As shown in Section 5.1, 5.2, our method finds new connections among variables, which may provide a new perspective in the corresponding area.

11.2 Neighborhood Selection Approach

In the second part of this dissertation, we develop a fully nonparametric method for neighborhood selection in pairwise graphical models. Since the range of each random variable is an arbitrary set, the proposed method provides a unified framework for mixed data types. The proposed SS ANOVA models are more general than existing parametric and semiparametric models. We developed penalized likelihood and pseudo-likelihood methods with L_1 penalty to select edges. As illustrated in Section 9.1, in addition to providing more flexible alternatives, the proposed method also serves as a new diagnostic tool for existing graphical models. We establish convergence rates of conditional density function estimate and interaction components in the SS ANOVA decomposition. Simulation results showed that the proposed method is efficient in edge detection and performs well under Gaussian and non-Gaussian situations. Applications to real data indicated the proposed method could detect edges that may provide new perspectives for researchers.

11.3 Comparison and Future Work

These two approaches are developed under a consolidated framework of SS ANOVA decomposition. They are nonparametric methods and more flexible than existing parametric and semi-parametric methods. They also provide a unified framework without any restrictions on the type of each random variable. However, the joint approach becomes computationally infeasible when the dimension is large due to memory restrictions. The neighborhood selection approach overcomes this disadvantage and is more computationally efficient by working on conditional densities.

We note that the proposed methods can be easily extended to select variables in nonparametric conditional density estimation, which has not been studied to the best of our

knowledge. The proposed method can also be extended to incorporate prior knowledge of the conditional density of a node using a model-based penalty or a semiparametric model (Shi et al. [36], Yu et al. [46]). For example, it may be known that the conditional density of X_α is close to, but not necessarily a Gaussian distribution. We may consider a quintic thin-plate spline space for $\mathcal{H}^{(\alpha)}$ with a tensor sum decomposition $\mathcal{H}^{(\alpha)} = \mathcal{H}^0_{(\alpha)} \oplus \mathcal{H}^1_{(\alpha)}$ where $\mathcal{H}^0_{(\alpha)} = \{1_{(\alpha)}, x_{(\alpha)}, x^2_{(\alpha)}\}$ corresponds to the space for logistic density of a Gaussian distribution.

Bibliography

[1] Agarwal, R., Davis, J. L. and Smith, L. [2008]. Serum albumin is strongly associated with erythropoietin sensitivity in hemodialysis patients, *Clin J Am Soc Nephrol.* **3**(1): 98–104.

[2] Ali, A., Kolter, J. Z. and Tibshirani, R. J. [2016]. The multiple quantile graphical model, *arXiv preprint arXiv:1607.00515* .

[3] Azzalini, A. and Valle, A. D. [1996]. The multivariate skew-normal distribution, *Biometrika* **83**(4): 715–726.

[4] Banerjee, O., El Ghaoui, L. and d'Aspremont, A. [2008]. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data, *The Journal of Machine Learning Research* **9**: 485–516.

[5] Chen, S., Witten, D. M. and Shojaie, A. [2015]. Selection and estimation for mixed graphical models, *Biometrika* **102**(1): 47–64.

[6] Cheng, J., Li, T., Levina, E. and Zhu, J. [2017]. High-dimensional mixed graphical models, *Journal of Computational and Graphical Statistics* **26**(2): 367–378.

[7] Dobruschin, P. [1968]. The description of a random field by means of conditional probabilities and conditions of its regularity, *Theory of Probability & Its Applications* **13**(2): 197–224.

[8] Drton, M. and Maathuis, M. H. [2017]. Structure learning in graphical modeling, *Annual Review of Statistics and Its Application* **4**: 365–393.

[9] Foygel, R. and Drton, M. [2010]. Extended bayesian information criteria for gaussian graphical models, *arXiv preprint arXiv:1011.6640* .

[10] Friedman, J., Hastie, T. and Tibshirani, R. [2008]. Sparse inverse covariance estimation with the graphical lasso, *Biostatistics* **9**(3): 432–441.

[11] Gu, C. [2004]. Model diagnostics for smoothing spline anova models, *Canadian Journal of Statistics* **32**(4): 347–358.

[12] Gu, C. [2013]. *Smoothing spline ANOVA models*, Vol. 297, Springer Science & Business Media.

[13] Gu, C., Jeon, Y. and Lin, Y. [2013]. Nonparametric density estimation in high-dimensions, *Statistica Sinica* pp. 1131–1153.

[14] Gu, C. and Ma, P. [2011]. Nonparametric regression with cross-classified responses, *Canadian Journal of Statistics* **39**(4): 591–609.

[15] Gu, C. and Qiu, C. [1993]. Smoothing spline density estimation: Theory, *The Annals of Statistics* pp. 217–234.

[16] Gu, C. and Wang, J. [2003]. Penalized likelihood density estimation: Direct cross-validation and scalable approximation, *Statistica Sinica* pp. 811–826.

[17] Gu, C. et al. [2014]. Smoothing spline anova models: R package gss, *Journal of Statistical Software* **58**(5): 1–25.

[18] Hastie, T., Tibshirani, R. and Wainwright, M. [2015]. *Statistical learning with sparsity: the lasso and generalizations*, CRC press.

[19] Höfling, H. and Tibshirani, R. [2009]. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods., *Journal of Machine Learning Research* **10**(4).

[20] Hsieh, C.-J., Dhillon, I. S., Ravikumar, P. K. and Sustik, M. A. [2011]. Sparse inverse covariance matrix estimation using quadratic approximation, *Advances in neural information processing systems*, pp. 2330–2338.

[21] Jeon, Y. and Lin, Y. [2006]. An effective method for high-dimensional log-density anova estimation, with application to nonparametric graphical model building, *Statistica Sinica* pp. 353–374.

[22] Koller, D. and Friedman, N. [2009]. *Probabilistic graphical models: principles and techniques*, MIT press.

[23] Lacson, E., Rogus, J., Teng, M., Lazarus, M. and Hakim, R. [2008]. The association of race with erythropoietin dose in patients on long-term hemodialysis, *American Journal of Kidney Diseases* **52**(6): 1104–1114.

[24] Lafferty, J., Liu, H., Wasserman, L. et al. [2012]. Sparse nonparametric graphical models, *Statistical Science* **27**(4): 519–537.

[25] Lafit, G., Tuerlinckx, F., Myin-Germeys, I. and Ceulemans, E. [2019]. A partial correlation screening approach for controlling the false positive rate in sparse gaussian graphical models, *Scientific Reports* **9**(1): 1–24.

[26] Lee, J. D. and Hastie, T. J. [2015]. Learning the structure of mixed graphical models, *Journal of Computational and Graphical Statistics* **24**(1): 230–253.

[27] Lin, Y. and Zhang, H. H. [2006]. Component selection and smoothing in multivariate nonparametric regression, *The Annals of Statistics* **34**(5): 2272–2297.

[28] Lin, Y. et al. [2000]. Tensor product space anova models, *The Annals of Statistics* **28**(3): 734–755.

[29] Liu, H., Lafferty, J. and Wasserman, L. [2009]. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs, *Journal of Machine Learning Research* **10**(Oct): 2295–2328.

[30] Meinshausen, N. and Bühlmann, P. [2006]. High-dimensional graphs and variable selection with the lasso, *Annals of Statistics* **34**(3): 1436–1462.

[31] Ouyang, Z., Zhou, Q. and Wong, W. H. [2009]. Chip-seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells, *Proceedings of the National Academy of Sciences* **106**(51): 21521–21526.

[32] Peng, J., Wang, P., Zhou, N. and Zhu, J. [2009]. Partial correlation estimation by joint sparse regression models, *Journal of the American Statistical Association* **104**(486): 735–746.

[33] Ravikumar, P., Wainwright, M. J., Lafferty, J. D. et al. [2010]. High-dimensional ising model selection using 1-regularized logistic regression, *The Annals of Statistics* **38**(3): 1287–1319.

[34] Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A. and Nolan, G. P. [2005]. Causal protein-signaling networks derived from multiparameter single-cell data, *Science* **308**(5721): 523–529.

[35] Serfling, R. J. [1980]. *Approximation Theorems of Mathematical Statistics*, John Wiley & Sons.

[36] Shi, J., Liu, A. and Wang, Y. [2019]. Spline density estimation and inference with model-based penalties, *Journal of Nonparametric Statistics* **31**: 596–611.

[37] Suggala, A., Kolar, M. and Ravikumar, P. K. [2017]. The expxorcist: nonparametric graphical models via conditional exponential densities, *Advances in neural information processing systems*, pp. 4446–4456.

[38] Turlach, B. A. and Weingessel, A. [2007]. quadprog: Functions to solve quadratic programming problems, *CRAN-Package quadprog* .

[39] Uduagbamen, P., Ogunkoya, J., Nwogbe, C., Eigbe, S. and Timothy, O. [2021]. Ultrafiltration vol-ume: Surrogate marker of the extraction ratio, determinants, clinical correlates and relationship with the dialysis dose, *J Clin Nephrol Ren Care* **7**: 068.

[40] Voorman, A., Shojaie, A. and Witten, D. [2014]. Graph estimation with joint additive models, *Biometrika* **101**(1): 85–101.

[41] Wang, Y. [2011]. *Smoothing splines: methods and applications*, CRC Press.

[42] Weinberger, H. F. [1974]. *Variational methods for eigenvalue approximation*, SIAM.

[43] Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., Prelić, A., von Rohr, P., Thiele, L. et al. [2004]. Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana, *Genome biology* **5**(11): 1–13.

[44] Yang, E., Ravikumar, P., Allen, G. I. and Liu, Z. [2015]. Graphical models via univariate exponential family distributions, *The Journal of Machine Learning Research* **16**(1): 3813–3847.

[45] Yang, Z., Ning, Y. and Liu, H. [2018]. On semiparametric exponential family graphical models, *The Journal of Machine Learning Research* **19**(1): 2314–2372.

[46] Yu, J., Shi, J., Liu, A. and Wang, Y. [2020]. Smoothing spline semiparametric density models, *Journal of the American Statistical Association* .

[47] Yuan, M. and Lin, Y. [2007]. Model selection and estimation in the gaussian graphical model, *Biometrika* **94**(1): 19–35.

[48] Yuan, X., Li, P., Zhang, T., Liu, Q. and Liu, G. [2016]. Learning additive exponential family graphical models via $l_{2,1}$-norm regularized m-estimation, *Advances in Neural Information Processing Systems*, pp. 4367–4375.