Statistical Analysis of Structured High-dimensional Data

Yizhi Sun

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Statistics

Hongxiao Zhu, Chair

Xinwei Deng

Pang Du

Inyoung Kim

Wednesday 19th September, 2018

Blacksburg, Virginia

Key Words: Bayesian Variable Selection; Functional Data Analysis; Ising Prior; Network Data Analysis; Portfolio Theory.

Statistical Analysis of Structured High-dimensional Data

Yizhi Sun

Abstract

High-dimensional data such as multi-modal neuroimaging data and large-scale networks carry excessive amount of information, and can be used to test various scientific hypotheses or discover important patterns in complicated systems. While considerable efforts have been made to analyze high-dimensional data, existing approaches often rely on simple summaries which could miss important information, and many challenges on modeling complex structures in data remain unaddressed. In this proposal, we focus on analyzing structured high-dimensional data, including functional data with important local regions and network data with community structures.

The first part of this dissertation concerns the detection of "important" regions in functional data. We propose a novel Bayesian approach that enables region selection in the functional data regression framework. The selection of regions is achieved through encouraging sparse estimation of the regression coefficient, where nonzero regions correspond to regions that are selected. To achieve sparse estimation, we adopt compactly supported and potentially overcomplete basis to capture local features of the regression coefficient function, and assume a spike-slab prior to the coefficients of the bases functions. To encourage continuous shrinkage of nearby regions, we assume an Ising hyper-prior which takes into account the neighboring structure of the bases functions. This neighboring structure is represented by an undirected graph. We perform posterior sampling through Markov chain Monte Carlo algorithms. The practical performance of the proposed approach is demonstrated through simulations as well as near-infrared and sonar data.

The second part of this dissertation focuses on constructing diversified portfolios using stock return data in the Center for Research in Security Prices (CRSP) database maintained

by the University of Chicago. Diversification is a risk management strategy that involves mixing a variety of financial assets in a portfolio. This strategy helps reduce the overall risk of the investment and improve performance of the portfolio. To construct portfolios that effectively diversify risks, we first construct a co-movement network using the correlations between stock returns over a training time period. Correlation characterizes the synchrony among stock returns thus helps us understand whether two or multiple stocks have common risk attributes. Based on the co-movement network, we apply multiple network community detection algorithms to detect groups of stocks with common co-movement patterns. Stocks within the same community tend to be highly correlated, while stocks across different communities tend to be less correlated. A portfolio is then constructed by selecting stocks from different communities. The average return of the constructed portfolio over a testing time period is finally compared with the S&P 500 market index. Our constructed portfolios demonstrate outstanding performance during a non-crisis period (2004-2006) and good performance during a financial crisis period (2008-2010).

Statistical Analysis of Structured High-dimensional Data

Yizhi Sun

General Abstract

High dimensional data, which are composed by data points with a tremendous number of features (a.k.a. attributes, independent variables, explanatory variables), brings challenges to statistical analysis due to their "high-dimensionality" and complicated structure. In this dissertation work, I consider two types of high-dimension data. The first type is functional data in which each observation is a function. The second type is network data whose internal structure can be described as a network. I aim to detect "important" regions in functional data by using a novel statistical model, and I treat stock market data as network data to construct quality portfolios efficiently.

Contents

1 Introduction and Background **1**

1.1 Functional Data Analysis 1

1.1.1 Basis Representation 1

1.1.2 Functional Regression 4

1.2 Network Data 8

1.2.1 Network Representation 9

1.2.2 Properties of a Network 11

1.2.3 Community Detection 12

1.3 Outline of the Dissertation 14

2 Bayesian Region Selection in Functional Data Regression **15**

2.1 Introduction 15

2.2 The Bayesian Region Selection Model 18

2.2.1 Functional Linear Regression and Basis Representation 18

2.2.2 Prior Setups 20

2.2.3 Posterior Inference 21

2.2.4 Extension to the Probit Model Case 24

2.3 Simulation Study 26

2.3.1 Results for the Continuous Response Case 27

2.3.2 Results for the Binary Response Case 30

2.4 Real Data Application 32

2.4.1 Application to the Tecator Data 32

2.4.2 Application to the Sonar Data 35

2.5 Hyperparameter Setups 36

2.6 Discussion 40

3 Constructing Diversified Portfolios by Detecting Communities in Stock Co-movement Network 42

3.1 Introduction 42

3.2 Network Construction 44

3.3 Community Detection 45

3.4 Stock Selection Using Importance Measures 52

3.5 A Workflow for Portfolio Construction and Evaluation 52

3.6 Results of the CRSP Data Analysis 53

3.6.1 Results for Period I (Year 2003-2006) 56

3.6.2 Results for Period II (Year 2007-2010) 58

3.6.3 Conclusion 59

3.7 A Study of Some Related Issues 61

3.7.1 Correlation Measures 62

3.7.2 Weight Assignments 63

3.7.3 Association Between Community Assignment and the Industrial Sector 68

3.7.4 Tuning of the Thresholding Parameter θ 69

4 Conclusion 71

References 77

List of Figures

1.1 A simple network example. 10

2.1 Estimates of $\beta(t)$ using BRS, compared with FLiRTI, SLoS, FLMBBR and FLMFPC. 29

2.2 Estimates of $\beta(t)$ using BRS, compared with FGLMBR and FGLMFPC. . . . 31

2.3 The near infrared absorbance spectrum. 33

2.4 Estimates of $\beta(t)$ using BRS, compared with FLiRTI, SLoS, FLMBBR and FLMFPC for the Tecator data. 33

2.5 Estimates of $\beta(t)$ using BRS, compared with FGLMBR and FGLMFPC for the Sonar data. 36

2.6 Maximal conditional prior probability for $\gamma_j = 1$. 38

2.7 Average number of 1's in $\boldsymbol{\gamma}$ for different a and d. 39

2.8 Computational scalability. 41

3.1 The 10 smallest eigenvalues. 50

3.2 The proposed workflow to construct portfolios. 53

3.3 The correlation pattern of weekly return for the period 2003-2006. 54

3.4 Plot for the portfolios' performances in year 2004-2006, $\theta = 0.5$. 58

3.5 Plot for the portfolios' performances in year 2008-2010, $\theta = 0.5$. 59

3.6 The performance of the constructed portfolios in 2004 by using three correlation measures with $\theta = 0.5$. 64

3.7 The performance of the constructed portfolios in 2008 by using three correlation measures with $\theta = 0.5$. 64

3.8 The performance of the OPWMI-based portfolio and the EW-based portfolios in 2004-2006. 67

3.9 The performance of the OPWMI-based portfolio and the EW-based portfolios in 2008-2010. 68

List of Tables

2.1 Simulation results: a comparison of BRS with existing methods. 30

2.2 The average RMSE for prediction by using 10-fold CV for the five methods. . 34

2.3 The AUC of prediction by using the 10-fold CV. 36

3.1 Summary statistics for period I (2004-2006). 57

3.2 Summary statistics for period II (2008-2010). 60

3.3 Performance of portfolio made by using the 2004-2006 and 2008-2010 weekly return data, with the Louvain method and three different threshold values of θ. 69

Acknowledgements

I would like to express my sincere gratitude and appreciation to my advisor, Dr. Hongxiao Zhu, for her mentorship with great patience, invaluable support, valuable advice and warm encouragement throughout the journey of my Ph.D. study. I feel blessed to have such a brilliant and conscientious advisor who has not only addressed my research questions promptly but also provided me with countless advice on career and life. I can hardly imagine having a better mentor like her.

I am enormously grateful to Dr. Fengrong Wei. The second project would not have been possible without the dataset and relevant explanations kindly offered by her.

Apart from my advisor and Dr. Wei, I would like to thank the other members of my dissertation committee: Dr. Inyoung Kim, Dr. Peng Du, and Dr. Xinwei Deng, for the insightful comments and the helpful suggestions, making my defense an enjoyable experience.

Finally, special thanks to my parents for their endless emotional and physical support since the first day of my life.

Chapter 1 Introduction and Background

The ever-growing high-throughput technology enables automatic collection of high-dimensional data, such as multi-platform genomic data, medical images, large-scale social networks, etc. This poses enormous challenges on statistical analysis. We consider two types of high-dimensional data with complex structures—functional data whose local regions may be associated to a variable of interest and network data which demonstrate community structures.

1.1 Functional Data Analysis

Functional data is a type of high-dimensional data with basic observational units being curves or surfaces measured over fine grids. Functional data are intrinsically infinite dimensional, containing tremendous amount of information, yet also bringing big challenges to both theory and computation. We often represent functional data using a set of stochastic processes, denoted by $X_1(t), \ldots, X_n(t)$. These data are realizations of stochastic processes in the Hilbert space $L^2(\mathcal{T})$ and often share a compact support $\mathcal{T} \subset \mathbb{R}^d$.

In reality, the true processes are often latent and cannot be observed directly. The data are often collected over a discrete grid of $\mathcal{T}$. For each of the n objects, a general assumption is that the functional data are collected over time grid $\mathbf{T} = \{\mathbf{t_1}, \ldots, \mathbf{t_p}\}$.

In sections 1.1.1 and 1.1.2, we introduce some important concepts related to functional data analysis (FDA), including basis representation and functional data regression.

1.1.1 Basis Representation

Basis functions are basic building blocks for representing functional data. Here, we review four most prevalent basis representation approaches, including B-splines, Fourier Series, Wavlets, and Functional Principal Components. One commonly used functional data representation is the B-spline basis representation. B-splines (De Boor et al., 1978) are created by joining

polynomial pieces at various values of t which are called knots. They can be computed recursively for any degree of polynomials without too much difficulties once the knots are given. There are several general properties of a degree-q B-spline. First, it has $q+1$ polynomial pieces of degree q; second, it is $q-1$ order continuous at the q inner knots; third, the B-spline is positive over the support spanned by $q+2$ knots and zero elsewhere; fourth, each non-boundary B-splines basis function overlaps with $2q$ neighboring basis functions; fifth, at a particular value of t there are $q+1$ B-spline bases that are nonzero. B-splines are popular choices of basis for one-dimensional functions because most smooth curves can be generated by linear combinations of B-splines. Denote $B_j(t;q)$ as the jth q degree B-spline at value t. The data pair $\{t, X(t)\}$ can be fitted as $\hat{X}(t) = \sum_j \hat{\alpha}_j B_j(t;q)$. De Boor et al. (1978) proposed an algorithm to compute B-splines of any degree by using those with lower degrees. The algorithm works for both equal-distant knots and knots that are arbitrarily distributed.

Besides B-splines, another popularly basis representation approach is the functional principal components (FPC). Principal Components Analysis (PCA), a key tool for the dimension reduction of high dimensional data, has been extended to functional data, and is termed Functional Principal Components Analysis (FPCA) (Dauxois et al., 1982). Functional data can be parsimoniously represented with the help of FPCA. To be specific, each of the independent real-valued functions $X_1(t), \ldots, X_n(t)$ can be modeled as a collection of principal component coefficients. We assume that $X_i(t)$ has unknown mean function $EX(t) = \mu(t)$. The covariance function is denoted as $\text{cov}(X(s), X(t)) = G(s,t)$. The domain of $X(t)$, $\mathcal{T}$ is assumed to be bounded and closed. The Mercer's theorem claims that G has the decomposition $G(s,t) = \sum_{k=1}^{\infty} \lambda_k \phi_k(s)\phi_k(t)$ under mild assumptions, where λ_k's are the eigenvalues in non-increasing order and ϕ_k's are the corresponding orthogonal eigen-fucntions. Moreover, the ith random curve $X_i(t)$ has the following FPCA expansion

$$X_i(t) = \mu(t) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(t), \quad t \in \mathcal{T}, \tag{1.1}$$

where $\xi_{ik} = \int_{\mathcal{T}} (X_i(t) - \mu(t))\phi(t)dt$ is the kth functional principal component of $X_i(t)$, sati-

fying $\mathrm{E}[\xi_{ij}] = 0$ and $\mathrm{Var}(\xi_{ik}) = \lambda_k$ for $\sum_k \lambda_k < \infty$ and $\lambda_1 \geq \lambda_2 \geq \ldots$ The Equation (1.1) enables dimensional reduction by retaining the first K terms if these terms provide a decent approximation of $X_i(t)$. Under this approximation, the K-dimension vector $\xi_i = (\xi_{i1}, \ldots, \xi_{iK})$ retains the majority of the information in $X_i(t)$. The approximation can be written as

$$X_i(t) \approx \hat{X}_i(t) = \mu(t) + \sum_{k=1}^{K} \xi_{ik}\phi_k(t), \quad t \in \mathcal{T}.$$

In addition to B-splines and FPCs, another classical multi-scale functional representation approach is wavelets representation. Using wavelets, a general function $y(t)$ can be represented by basis functions at various scales and positions. Let j denote the scale and k denote the location. The wavelet basis $\{\psi_{j,k}, j, k \in \mathbb{Z}\}$ is an orthonormal basis of $L^2(\mathcal{T})$ with $\psi_{j,k} = 2^{j/2}\psi(2^j x - k)$. Using wavelets, a function $y(t)$ can be represented by

$$y(t) = \sum_{j,k\in\mathbb{Z}} d_{jk}\psi_{jk}(t), \quad t \in \mathcal{T},$$

where $d_{jk} = \int_{\mathcal{T}} y(t)\psi_{jk}(t)dt$, $t \in \mathcal{T}$. According to multiresolution analysis, the function space $L^2(\mathcal{T})$ can be decomposed into a sequence of linear and closed subspaces $\{V_j, j \in \mathbb{Z}\}$. Denote W_j the orthogonal complement of V_j in V_{j+1}, we can decompose $L^2(\mathcal{T})$ as

$$L^2(\mathcal{T}) = V_0 \oplus \bigoplus_{j\geq 0} W_j,$$

where the $\oplus$ denotes the direct sum. Accordingly, $y(t)$ can be written as the linear combinations of the orthonormal basis at each subspace, i.e,

$$X(t) = \sum_{k\in\mathbb{Z}} d_{0k}\psi_{0k}(t) + \sum_{j>0}\sum_{k\in\mathbb{Z}} d_{jk}\psi_{jk}(t), \quad t \in \mathcal{T}, \tag{1.2}$$

where $\{\psi_{0k}, k \in \mathbb{Z}\}$ is the set of father wavelets that span V_0 and $\{\psi_{jk}, j > 0, k \in \mathbb{Z}\}$ is the set of mother wavelets that span W_j. In the right hand side of equation (1.2), the first term

is the orthogonal projection of $y(t)$ onto V_0, which gives an approximation for the function $y(t)$. The second term is the summation of the projections of $y(t)$ onto W_j for $j > 0$, which provides extra terms that add finer details to the approximation.

1.1.2 Functional Regression

One of the most intensively studied topics in FDA is functional regression. Functional regression can be considered as an extension of regular regression to the functional data case. One typical type of functional regression is that the responses are scalars and the predictors are functions (scalar-on-function regression). In this section, we overview the classical scalar-on-function regression models and the commonly used regularization approaches when fitting these models.

Models of scalar-on-function regression The scalar-on-function regression is perhaps the simplest form of functional regression. Ramsay and Dalzell (1991) introduced functional linear model (FLM) for the scalar-on-function regression. Hastie and Mallows (1993) gave the commonly used form

$$y_i = \beta_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt + \epsilon_i, \quad t \in \mathcal{T},$$

where $\{y_i, i = 1, \ldots, N\}$ is continuous scalar variables. Each y_i is related with a functional predictor $X_i(t)$ through a linear integral equation. Here, β_0 is the intercept and $\epsilon_i \sim N(0, \sigma^2)$ is the residual errors.

If the response has non-Gaussian distribution, the scalar-on-function regression becomes generalized FLM (GFLM). Marx and Eilers (1999) considers GFLM with responses in exponential family distribution. The general form of GFLM is

$$g(E(y_i)) = \beta_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt, \quad t \in \mathcal{T},$$

where $g(\cdot)$ is a link function. The GFLMs are widely used for functional classification when the responses y_i's are binary. A classical framework is the functional logistic regression (James,

2002), which takes the form

$$\log\left[\frac{P(y_i=1 \mid X_i(t))}{P(y_i=0 \mid X_i(t))}\right] = \beta_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt, \quad t \in \mathcal{T}.$$

An alternative is the GFLM with a probit link called functional probit model, in which an univariate latent variable Z_i is introduced to link the response y_i with the functional predictor $X_i(t)$ by:

$$y_i = \begin{cases} 1, & \text{if } Z_i < 0, \\ 0, & \text{if } Z_i \geq 0. \end{cases} \tag{1.3}$$

$$Z_i = \beta_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt + \epsilon_i, \quad \epsilon_i \sim N(0,1), \quad t \in \mathcal{T}. \tag{1.4}$$

This framework is convenient for Bayesian modeling. It has been used by Zhu et al. (2007) and Zhu et al. (2010) for binary classification.

Regularization on Functional Regression Regularization is a way of adding a constraint to high-dimensional problems, so that the ill-posed problems can be addressed. For example, in a high-dimensional regression, when the sample size n is less than the number of predictors p, the problem cannot be solved due to the lack of a unique solution. However, with regularization (e.g., Lasso penalty), the problem is tractable. In functional regression problems, the regularization task can be accomplished via applying one of the three regularization approaches, truncation, roughness penalty, or adding sparsity assumptions, to the coefficients of the function's functional basis. We overview these regularization approaches as follows.

- **Truncation** In basis representation, truncation means discarding all terms after a certain threshold. This regularization technique is commonly seen in FPCA in which the first several FPCs are usually sufficient to explain most variability of $X(t)$. For example, Cardot et al. (1999) tackled the FLM by using the FPCA. They used a set of eigen-functions $\{\phi_k(t)\}$ to expand the observed functions $X_i(t) = \sum_{k=0}^{\infty} \xi_{ik}\phi_k(t)$ and the functional regression coefficient $\beta(t) = \sum_{k=0}^{\infty} \beta_k\phi_k(t)$. Here, $\xi_{ik} = \int_{\mathcal{T}} X_i(t)\phi_{ik}(t)dt$ and

$\beta_k = \int_{\mathcal{T}} \beta(t)\phi_k(t)dt$, $t \in \mathcal{T}$. If the basis set $\{\phi_k(t)\}$ is truncated after $k > K$, the FLM can be approximated by

$$\hat{y} = \int_{\mathcal{T}} X(t)\beta(t)dt \approx \sum_{k=1}^{K} \xi_{ik}\hat{\beta}_k.$$

Let $(\xi_{ij})_{i=1,\ldots,n,j=1,\ldots,K}$ be an $n \times K$ matrix, then the estimate of β_j is

$$\hat{\beta}_j = \frac{\xi_j^T y}{n\lambda_j}, \quad j = 1, \ldots, K,$$

where ξ_j is the jth column of the N by K matrix formed by $\{\xi_{ij}\}$. The $\beta(t)$ can be estimated by $\hat{\beta}(t) = \sum_{j=1}^{K} \hat{\beta}_j\phi_j(t)$.

- **Roughness Penalty** Roughness penalty has been extensively used in functional data's smoothing and regression (Cardot et al., 2003; Eilers and Marx, 1996; Ramsay and Silverman, 1997; Ruppert et al., 2003; Yuan and Cai, 2010). In the case of roughness penalty, the integration of the squared second derivatives of the function will be penalized via applying L2 penalty on the basis coefficients (Eilers and Marx, 1996; Ruppert et al., 2003). By adding a tuning parameter λ, a global smoothing mechanism will be induced. For example, Ramsay and Silverman (1997) considered FLM by using two separated basis sets $\{\psi_k\}_{k=1,\ldots,K_x}$ and $\{\phi_k\}_{k=1,\ldots,K_\beta}$ to expand the observed functions $X(t)$ and the regression coefficient function $\beta(t)$. In particular, $X_i(t) = \sum_{k=1}^{K_x} c_{ik}\psi_k(t) = c_i^T\psi(t)$, where $\{c_{ik}\}$ is the basis coefficients for $X_i(t)$, c_i is a column vector with elements being $\{c_{ik}\}_{k=1,\ldots,K_x}$ and $\psi(t)$ is a column vector with entries being $\psi_k(t)$. Furthermore, $\beta(t) =$

$\sum_{k=1}^{K_\beta} b_k\phi_k = \phi^T(t)b$, where b_k is the basis coefficients for $\beta(t)$. So

$$\begin{aligned} y_i &= \int_{\mathcal{T}} X_i(t)\beta(t) + \epsilon_i \\ &= \left(\int_{\mathcal{T}} c_i^T\psi(t)\phi^T(t)bdt\right) + \epsilon_i \\ &= \left(c_i^T\int_{\mathcal{T}}\psi(t)\phi^T(t)dt\right)b + \epsilon_i \\ &= \tilde{x}_i^T b + \epsilon_i. \end{aligned} \tag{1.5}$$

where $c_i^T\int_{\mathcal{T}}\psi(t)\phi^T(t)dt$ is denoted by a row vector $\tilde{x}_i^T$. Rewrite (1.5) in matrix form, we get $y = \tilde{X}b + \epsilon$ where $\tilde{X}$ is a matrix formed by stacking the row vectors in $\{\tilde{x}_i^T\}_{i=1,\ldots,n}$.

In order to decrease the model complexity and get a finer control of the estimated regression coefficients, Ramsay and Silverman (1997) and Cardot et al. (2003) incorporated a tuning parameter λ and a matrix $R = \int[D\beta(t)]^2dt$ into residual sum of squares where D is some linear differential operator. The objective function takes the form

$$\text{PEN}_\lambda(\beta) = \|y - \tilde{X}b\| + \lambda b^T Rb \tag{1.6}$$

By minimizing (1.6) w.r.t. b, the estimate of b can be obtained by $\hat{b}_\lambda = (\tilde{X}^T\tilde{X} + \lambda R_0)\tilde{X}^T y$, where R_0 is the penalty matrix derived from the matrix R. Finally, the estimated regression coefficient is $\hat{\beta}(t) = \phi^T(t)\hat{b}_\lambda$. Yuan and Cai (2010) proposed a representer theorem which makes the implementation of the estimators of these regularized coefficients easy.

- **Sparsity Assumptions** The sparsity of the basis coefficients can be achieved by using either L1 penalty such as LASSO (Tibshirani, 1994) or via Bayesian modeling by introducing sparse priors like SSVS (George and McCulloch, 1993). Zhu et al. (2007) proposed a Bayesian probit model with variable selection for functional data regression. The model setting is the same as that given by equation (1.3) and equation (1.4). By using a set of truncated orthonormal basis $\{\phi_k\}_{k=1}^{\infty}$, the observed functions $X_i(t)$ and

the functional regression coefficient can be approximated by $X_i(t) \approx \sum_{k=1}^{p} c_{ik}\phi_k(t)$ and $\beta(t) \approx \sum_{k=1}^{p} b_k\phi_k(t)$.

The equation (1.4) can be re-written as:

$$\begin{aligned} Z_i &= \beta_0 + \int_{\mathcal{T}} X_i(t)\beta(t) + \epsilon_i \\ &= \beta_0 + \int_{\mathcal{T}} \left(\sum_{k=1}^{p} c_{ik}\phi_k(t)\right)\left(\sum_{k=1}^{p} b_k\phi_k(t)\right) + \epsilon_i \\ &= \beta_0 + \sum_{k=1}^{p} c_{ik}b_k + \epsilon_i \\ &= \beta_0 + c_i^T b + \epsilon_i, \end{aligned}$$

where $c_i = (c_{i1}, \ldots, c_{ip})^T$ and $b = (b_1, \ldots, b_p)^T$. A hyperparameter $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)^T$ is introduced to conduct variable selection via a spike-and-slab prior for b

$$b_j \mid \gamma_j \sim \gamma_j N(0, v_{1j}^2) + (1-\gamma_j)N(0, v_{0j}^2), \quad j = 1, \ldots, p,$$

where v_{1j} and v_{0j} are some positive numbers satisfying $v_{1j} \gg v_{0j} > 0$. The prior of γ_j is set as Bernoulli(w). With the above setting and the likelihood function $p(Z_i \mid b, \gamma, y_i)$, the joint posterior distribution $p(b, \gamma \mid \{Z_i\}, \{y_i\})$ can be obtained. Based on the joint posterior distribution, a gibbs sampling procedure can be performed for the variable selection and estimation of b_j. With the estimated basis coefficient $\hat{b}_j$, the estimate for the functional regression coefficient can be calculated by $\hat{\beta}(t) = \sum_{\{\hat{b}_j \neq 0\}} \hat{b}_j\phi_j(t)$.

1.2 Network Data

Network data is ubiquitous in today's society. People send text messages to each other, make phone calls, follow or un-follow friends on social media, transfer money and trade goods via e-commerce platforms, and take public transportations such as bus, subway, or flight. Networks can be used to describe activities in all there cases. Similarly, networks exist in many major

phenomena. Financial crisis impacts the whole financial systems through the web of banks and cooperations. Epidemic diseases such as flu spread all over the world via the transportation network. Human activities cause the distinction of species, which breaks the balance of the eco-system. In all these situations, we need to deal with large sets of units which can be individuals, banks, cooperations, social media accounts, cell phones, airports, or species in the ecosystem. The inter-relationship between these units form a network. Studying network helps us understand the functional mechanism of a complex system.

In this section, we review the basic network representation, statistics that summarize the properties of a network, as well as the community detection.

1.2.1 Network Representation

We use $V = \{1, \ldots, v\}$ to denote a set of units in a network system. The units are called vertices, with equivalent names being "nod", "individual", "agent" or "player". These vertices may stand for different objects such as individual persons, organizations, countries, depending on the problem of interest.

The classical form of a network is an undirected graph, in which two vertices are either linked or not. The edges in an undirected graph is suitable to describe whether certain relationship exists between a pair of vertices. For example, undirected graphs can be used to represent social relationship such as friends and co-workers, whereas for relationships such as friends following each other on social media, directed graphs are often used in which the edges have directions. An undirected network, denoted by $G(V, A)$, consists of a set of n vertices V and a real-valued $n \times n$ matrix $A = \{a_{ij}\}$ where

$$a_{ij} = \begin{cases} \neq 0, & \text{if there is an edge between vertex } i \text{ and } j, \\ 0, & \text{otherwise.} \end{cases}$$

Here, a_{ij} specifies the relationship between vertex i and vertex j. The matrix A is often called adjacency matrix because its elements indicate whether a pair of vertices are connected, or

in other words, adjacent to each other. For unweighted networks, a_{ij} is either 0 or 1. If we want to keep a record of how intensively two vertices are connected, for example, how long it is between two cities in a highway network, the nonzero element of matrix A can be greater than 1, which corresponds to a weighted network. An example of an undirected, unweighted network is $V = \{1, 2, 3\}$ and

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

The plot of this simple network is as follows

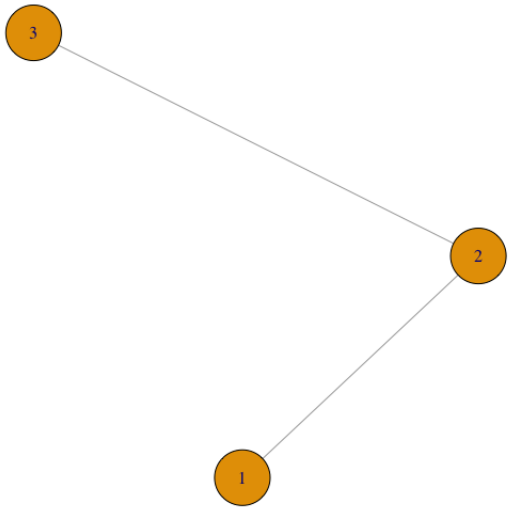

Figure 1.1: A simple network example.

In this example, the network has three vertices $\{1, 2, 3\}$ and two edges. The vertex 1 is linked to vertex 2 and vertex 3 is linked to vertex 2.

1.2.2 Properties of a Network

There are several statistics that can be used to summarize properties of a network, including size, degree, density, as well as various centrality measures. We overview these statistics in this section.

The *network size* is defined as the number of vertices of a network G. We define the *neighborhood* of the vertex i, denoted by N_i, as the set of vertices that are linked to vertex i, i.e., $N_i \triangleq \{j : a_{ij} = 1\}$. We further define the *degree* of vertex i, d_i as the cardinality of N_I, i.e., $d_i = \#\{j : a_{ij} = 1\} = \#N_i$. The *density* of a network G measures the fraction edges in a network; it is defined as

$$D_G = \frac{\#\text{ of edges}}{\#\text{ of possible edges}}.$$

Centrality measures how important a vertex i is in a network. It is widely used in many applications, such as detecting the most influential individual in a social network, finding the information spreader of internet, and locating the key infrastructure in an urban network. There are various ways to quantify centrality, and each of them is used to capture different aspects of "importance" of a vertex. In summary, there are four commonly used centrality measures, listed as follows.

1) **Degree Centrality**. The degree centrality of a vertex i, is defined to be its normalized degree, i.e.,

$$D_i = \frac{1}{n-1} d_i$$

where d_i is the degree of vertex i and n is the total number of vertices in the network.

2) **Closeness Centrality**. The closeness centrality (Bavelas, 1950) measures the importance of a vertex i by using its average distances to other vertices. The shorter the average distances to others, the more central is the vertex i. Mathematically, the close-

ness centrality of vertex i, C_i is defined as the reciprocal of the average distance, i.e.,

$$C_i = \frac{n-1}{\sum_{j \in G, j \neq i} d_{ij}}.$$

3) **Betweenness Centrality**. The betweenness centrality (Freeman, 1977) claims that a vertex is more central if it is found in the shortest paths between two other vertices. Denote $\sigma_{kj}(i)$ the number of shortest paths between vertex k and vertex j where vertex i is in the shortest path, and denote σ_{kj} the total number of the shortest paths between vertex k and vertex j. The betweenness centrality of vertex i is defined to be

$$B_i = \sum_{k \neq j, i \notin \{k,j\}} \frac{\sigma_{kj}(i)}{\sigma_{kj}} \cdot \frac{1}{\frac{(n-1)(n-2)}{2}},$$

where $\frac{(n-1)(n-2)}{2}$ is a normalizing factor that counts the number of vertices pairs that do not involve vertex i.

4) **Eigenvector Centrality**. The eigenvector centrality (Newman, 2008) measures vertex i's importance in two aspects:

   (i) How many connections it has, which is similar to the degree centrality.

   (ii) How important are the vertices that are connected to vertex i.

A vertex i is important if it is linked to many other important vertices. To calculate the eigenvector centrality, let $x = \{x_i\}_{i=1,\ldots,n}$ be the eigenvector of the adjacency matrix $A = (a_{ij})$, so that $Ax = \lambda x$, where λ is the largest eigenvalue of the adjacency matrix A. The eigenvector centrality of vertex i is the ith element of x.

1.2.3 Community Detection

A classical random graph, introduced by Erdos and Rényi (1960), assumes that for each pair of vertices, the probability of having an edge is the same across the whole network. Therefore,

the edges are distributed homogeneously across all vertices. Most networks encountered in the real world, however, demonstrate large degree of internal inhomogeneity.

Community patterns, i.e., groups of vertices that are frequently connected within the group but loosely connected outside the group, are often observed. Community detection (also called graph or network clustering) aims to find out vertex groups with such clustering patterns within a network. For example, professional social network websites cluster users with similar backgrounds or interests. The communities found in the professional social network websites offer great convenience for building an efficient job recommendation system.

Community detection is also an indispensable tool to locate vertices of special importance. For example, one may cluster the vertices by finding the hidden communities and the boundaries of the communities, and then identify the vertices that have many connections with others in the community. Identifying the "key player" may help social researchers understand the information flow in the community and design an effective intervention strategy. Those who connects in between communities could play essential roles in holding different communities together and spreading information across the network.

Finding communities within an arbitrary network can be challenging. Two fundamental problems need to be addressed before proposing a community detection algorithm. The first is to come up with an appropriate quantitative definition of the community. Depending on the specific problems of study, no specific definition is universally accepted. The second problem is how to partition the network in order to group its vertices into communities. In most situations, the number and size of communities remain unknown. Although we could detect communities by enumerating all possible partitions and pick the one that best suits the community definition, the number of possible partitions grows exponentially with the size of the network, making enumeration computationally intractable. Therefore, instead of the brute-force approach, many sophisticated algorithms have been developed. These algorithms can be roughly assigned into three categories. The first category consists of traditional algorithm-based methods such as hierarchical clustering and spectral clustering. The second category consists methods that are based on some pre-defined criteria. Their goal is to find the network

partitions by optimizing these criteria. A representative of such criteria is the Newman-Girvan Modularity, which leads to the discovery of a large family of algorithms, including the Louvain algorithm, the greedy algorithm, and the Newman's leading eigenvector algorithm. The third category consists of model-based methods. These methods usually introduce latent variables to assign vertices into different communities, and then fit a probabilistic model to estimate the latent variables. A well-known example is the Stochastic Block Model (SBM).

1.3 Outline of the Dissertation

The rest of this dissertation is organized as follows. In Chapter 2, we propose a novel Bayesian method for region selection in the framework of functional data regression. The selection of regions is achieved through encouraging sparse estimation of the regression coefficient. Nonzero regions of the estimated coefficient function correspond to the selected regions. Besides working on the FLM framework, we also consider the functional probit model with binary responses and functional predictors. In Chapter 3, we focus on constructing diversified portfolios based on the co-movement network of stock returns. During a training period, we construct the stock co-movement network using stock returns in the U.S. stock market. Based on the co-movement network, we detect communities and construct a portfolio by selecting stocks from different communities. We finally calculate the cumulative returns of the constructed portfolio over a testing time period and compare them with that of the market index (the S&P 500 index). This modeling strategy is applied to data from two time periods, one during 2003-2005 (a period during which no financial crisis presents), and the other during 2007-2009 (a period during which a financial crisis presents). During each time period, training and testing procedures are performed sequentially.

Chapter 2 Bayesian Region Selection in Functional Data Regression

2.1 Introduction

The ever-growing modern technology enables automatic collection of high-dimensional data in functional form, with basic observational units being curves or surfaces measured over fine grids. Examples include signals, images and longitudinal trajectories. These data promote the development of *functional data analysis*. Although functional data carry excessive amount of information, due to their heterogeneous nature, some regions may play a more important role in decision-making than other regions. Here we list several examples:

1) In cancer imaging, one aims to detect abnormal lesions in tissue that may reflect cancerous change. This can be done through identifying special regions in images such as PET or MRI.

2) Recent studies show that controlling progesterone levels may prevent preterm birth. An important question to ask is: at which stage of pregnancy the progesterone level is most associated with preterm birth? This question can be answered by detecting regions on a longitudinal trajectory.

3) In mass spectrometry data, all meaningful information is contained in the peaks of the spectral curves. The essential goal is to detect peaks that are differentially expressed across samples. Peak detection on spectral data is a special type of region selection problem.

As indicated by these applications, the "regions of interest" has great significance in guiding decision-making. It is therefore desirable to identify local regions on functional data that are relevant to the problem of interest. Although various ad hoc methods are available on feature extraction in machine learning literature, these methods cannot take advantage of

the additional information implied by properties of the underlying functions, thereby usually leading to suboptimal results.

In functional data analysis literature, while various methods have been proposed for selection among basis coefficients of a single functional predictor (Lee and Park, 2012), among multiple functional predictors (Zhu and Cox, 2009; Zhu et al., 2010; Fan et al., 2015; Gertheiss et al., 2013), or among additive components in the functional additive model (Zhu et al., 2014), there is a limited literature on selection of local features within functional object. Notable works that consider local feature selection include James et al. (2009), Zhou et al. (2013), Koltchinskii and Minsker (2013), and Lin et al. (2015).

James et al. (2009) use a p-dimensional basis to approximate the regression coefficient function $\beta(t)$ in a functional linear model framework and apply a lasso-type penalty to a discretized approximation of the dth derivative $\beta^{(d)}(t)$. Sparsity in the zeroth derivative generates zero regions while sparsity in higher derivatives ensures smoother fit.

Zhou et al. (2013) also consider functional linear model and try to encourage zero-values of coefficient function at sub-regions. They propose a two-stage estimator by combining a Dantzig selector and a group SCAD penalty. In stage one, the Dantzig selector is used to provide a rough initial estimate of the regression coefficient function based on B-spline representation. In stage two, they use a grouped SCAD method and a boundary grid-search algorithm to refine the zero regions and to achieve estimation of regression coefficient.

Recently, Lin et al. (2015) proposed a functional generalization of the SCAD penalty and named it "functional SCAD" (fSCAD in short). They combined fSCAD with smoothing splines to develop a one-stage procedure called SLoS (Smooth and Locally Sparse). This procedure penalizes $\beta(t)$ via SCAD penalty and smooths $\beta(t)$ jointly so it can identify the zero sub-regions of $\beta(t)$ and produce a smooth estimate of $\beta(t)$ in the non-zero sub-regions simultaneously.

The existing methods all rely on penalized regression to achieve region selection. They have several limitations: (i) the uncertainty of the regression coefficient is often hard to characterize due to theoretical difficulties; (ii) a single basis is often limited in characterizing

the heterogeneous local features of functional data; (iii) existing methods focus primarily on regressing a continuous scalar variable on a functional predictor, and may not be applicable to more complicated situations such as the binary response case. These limitations may be addressed through Bayesian approaches. However, to the best of our knowledge, no Bayesian method is available for the functional data region selection problem.

In this chapter, we propose a general Bayesian regression framework for region selection and estimation. The proposed approach adopts compactly supported basis to capture local features, takes advantage of the Ising prior to encourage spatially structured shrinkage for the regression coefficient function. The selection of regions is achieved through encouraging sparse estimation of the regression coefficient. Nonzero regions of the estimated coefficient function correspond to the selected regions. In particular, we adopt compactly supported and potentially over- complete basis to capture local features of the regression coefficient function, and assume spike-slab priors to coefficients of the bases functions. To encourage continuous shrinkage of nearby regions, we adopt an Ising hyper-prior to take into account the neighboring structure of the bases functions, represented by an undirected graph. Posterior sampling is performed through Markov chain Monte Carlo algorithms.

The outline for the rest of this chapter is as follows. In Section 2.2, we describe how to connect the functional regression model with compactly supported basis representation and structured variable selection to accomplish the region selection task. To be specific, we start by representing the functional linear regression model with compactly supported basis in Section 2.2.1, introduce a variable selection prior and an Ising hyper-prior for structured shrinkage in Section 2.2.2. The posterior inference will be demonstrated in Section 2.2.3, followed by an extension to the binary response case in Section 2.2.4. Simulation results are presented in Section 2.3 and real data applications to Tecator and Sonar data are provided in Section 2.4. Finally, we discuss related issues in Section 2.5 and Section 2.6.

2.2 The Bayesian Region Selection Model

2.2.1 Functional Linear Regression and Basis Representation

Suppose we obtain functional observations from n objects. For $i = 1, \ldots, n$, let $X_i(t)$ be the function observed from the i-th object which takes value in $L^2(\mathcal{T})$ with $\mathcal{T}$ being a compact domain. We treat the observations $\{X_i(t), i = 1, \ldots, n\}$ as predictors. We assume that the response y_i forms a linear model with $X_i(t)$ (Ramsay and Dalzell, 1991) (Hastie and Mallows, 1993):

$$y_i = b_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt + \epsilon_i. \tag{2.1}$$

Here b_0 is the intercept. For all i, we assume that ϵ_i are i.i.d with distribution $N(0, \sigma^2)$ and that $\beta(t) \in L^2(\mathcal{T})$. We further assume that $\{X_i(t)\}$ are collected on a dense grid $\boldsymbol{t} = (t_1, \ldots, t_m)$ in $\mathcal{T}$, and $\beta(t)$ is exactly zero on a continuous set $\mathcal{T}_0$, where $\mathcal{T}_0 \subset \mathcal{T}$ and $\mathcal{T}_0 \neq \varnothing$. Thus, only on non-zero regions of $\mathcal{T}$ there exists association between y_i and $X_i(t)$. We propose multiple sets of B-spline basis $\boldsymbol{\Phi} = \left(\{\phi_j^l(t)\}_{j=1}^{q_l+k_l}, l = 1, \ldots, L\right)$. For the l-th set $\{\phi_j^l(t)\}_{j=1}^{q_l+k_l}$, q_l is the order of the B-spline functions and $k_l - 1$ is the number of interior knots. We can expand $\beta(t)$ by using these sets of basis:

$$\beta(t) = \sum_{l=1}^{L} \sum_{j=1}^{q_l+k_l} b_j^l \phi_j^l(t), \tag{2.2}$$

where b_j^l is the coefficient associated with the jth basis component of the l-th basis set. We notice that $\left(\{\phi_j^l(t)\}_{j=1}^{q_l+k_l}, l = 1, \ldots, L\right)$ forms a potentially over-complete basis so it is natural to assume that the coefficient set $\left(\{b_j^l\}_{j=1}^{q_l+k_l}, l = 1, \ldots, L\right)$ is sparse. We plug (2.2) into (2.1) and get

$$\begin{aligned} y_i &= b_0 + \int_{\mathcal{T}} X_i(t)[\sum_{l=1}^{L} \sum_{j=1}^{q_l+k_l} b_j^l \phi_j^l(t)]dt + \epsilon_i \\ &= b_0 + \sum_{l=1}^{L} \sum_{j=1}^{q_l+k_l} b_j^l \int_{\mathcal{T}} X_i(t)\phi_j^l(t)dt + \epsilon_i. \end{aligned} \tag{2.3}$$

Denote $\int_{\mathcal{T}} X_i(t)\phi_j^l(t)dt$ as c_{ij}^l, then (2.3) can be represented by

$$y_i = b_0 + \sum_{l=1}^{L}\sum_{j=1}^{q_l+k_l} b_j^l c_{ij}^l + \epsilon_i. \tag{2.4}$$

The notation of equation (2.4) can be further simplified by concatenating coefficients b_0 and $\left(\left\{b_j^l\right\}_{j=1}^{q_l+k_l}, l = 1,\ldots,L\right)$ to make one vector $\boldsymbol{b}$. So (2.4) can be written as

$$y_i = \boldsymbol{c}_i^T\boldsymbol{b} + \epsilon_i, \tag{2.5}$$

where $\boldsymbol{c}_i = \left(1, c_{i,1}^1,\ldots,c_{i,q_1+k_1}^1,\ldots,c_{i,1}^L,\ldots,c_{i,q_L+k_L}^L\right)^T$ and $\boldsymbol{b} = \left(b_0, b_1^1,\ldots,b_{q_1+k_1}^1,\ldots,b_1^L,\ldots,b_{q_L+k_L}^L\right)^T$. We can write (2.5) compactly in a matrix form

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{b} + \boldsymbol{\epsilon}. \tag{2.6}$$

where $\left\{\boldsymbol{c}_i^T, i = 1,\ldots,n\right\}$ constitutes the rows of matrix $\boldsymbol{C}$, $\boldsymbol{y} = (y_1,\ldots,y_n)^T$ and $\boldsymbol{\epsilon} = (\epsilon_1,\ldots,\epsilon_n)^T$.

If $X_i(t)$ is observed as $W_i(t)$ with measurement error $\delta_i(t)$, i.e.,

$$W_i(t) = X_i(t) + \delta_i(t), \quad t \in \mathcal{T},$$

then model (2.1) becomes

$$\begin{aligned} y_i &= b_0 + \int_{\mathcal{T}} W_i(t)\beta(t)dt + \epsilon_i \\ &= b_0 + \int_{\mathcal{T}} (X_i(t) + \delta_i(t))\beta(t)dt + \epsilon_i \\ &= b_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt + \int_{\mathcal{T}} \delta_i(t)\beta(t)dt + \epsilon_i \\ &= b_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt + \tilde{\delta}_i + \epsilon_i, \end{aligned} \tag{2.7}$$

where $\tilde{\delta}_i = \int_{\mathcal{T}} \delta_i(t)\beta(t)dt$. Similar to (2.5), (2.7) can be written in a vectorized form: $y_i =$

$\boldsymbol{c}_i^T\boldsymbol{b} + \tilde{\epsilon}_i$ with $\tilde{\epsilon}_i = \epsilon_i + \tilde{\delta}_i$. Assume $\tilde{\delta}_i \sim N(0, \sigma_1^2)$ so we have $\tilde{\epsilon}_i \sim N(0, \sigma^2 + \sigma_1^2)$. Similar to (2.6), (2.7) can be expressed compactly in a matrix form:

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{b} + \tilde{\boldsymbol{\epsilon}},$$

where $\tilde{\boldsymbol{\epsilon}} = (\tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_n)^T$. In the later derivation, we will adopt the formula in (2.1). Bear in mind that the case with random error can be transformed into the same framework.

2.2.2 Prior Setups

In order to encourage zero sub-regions in $\beta(t)$, we need to encourage sparsity of $\boldsymbol{b}$. We do this by using the spike-and-slab prior which introduces a latent variable $\gamma_j \in \{0, 1\}, j = 1, \ldots, p$. Here, $p = \sum_{l=1}^{L}(q_l + k_l)$. The conditional prior of b_j given γ_j is

$$b_j \mid \gamma_j, \sigma^2 \sim (1 - \gamma_j)I_0 + \gamma_j N(\mathbf{0}, \sigma^2 v^2), \quad j = 1, \ldots, p,$$

$$b_0 \sim N(0, v_0^2),$$

where v^2 is a hyper-parameter which controls the scale of each b_j's variance. The variance of b_0, v_0^2 is set to be a large value. In matrix form, we have

$$\boldsymbol{b} \mid \boldsymbol{\gamma}, \sigma^2 \sim N\big(\mathbf{0}, \sigma^2\Sigma_\gamma\big) \tag{2.8}$$

with $\Sigma_\gamma = \mathrm{diag}\big\{v_0^2, \big(\gamma_j v^2 + (1 - \gamma_j)I_0\big)_{j=1}^{p}\big\}$.

Since the neighborhood structure of the basis functions in $\boldsymbol{\Phi}$ reflects the correlation pattern between components of $\boldsymbol{b}$, we need to estimate $\boldsymbol{b}$ in (2.6) while taking into account both the sparsity and the neighborhood structure. In traditional spike-and-slab variable selection framework, the prior for $\boldsymbol{\gamma}$ is assumed to be i.i.d Bernoulli distribution. Here, borrowing the idea of Li and Zhang (2010), we assume $\boldsymbol{\gamma}$ has an Ising prior so that we can perform a structured variable selection. Let $\boldsymbol{a} = \big(a_1, \ldots, a_p\big)^T$ be a real vector and $D = \big\{d_{jk}\big\}_{p\times p}$ be a

symmetric real-valued matrix with $d_{jj} = 0, j = 1, \ldots, p$. The prior of $\boldsymbol{\gamma}$ has the density

$$p(\boldsymbol{\gamma}) \propto \exp\{\boldsymbol{a}^T\boldsymbol{\gamma} + \boldsymbol{\gamma}^T D\boldsymbol{\gamma}\}.$$

Here, $\boldsymbol{a}$ controls the sparsity of $\boldsymbol{\gamma}$ and $\{d_{jk}\}_{p\times p}$ represents the prior belief on the strength of coupling between the pairs of neighbors (j, k). In our study, we assume that $d_{jk} > 0$, which corresponds to positive coupling strength. However, d_{jk} can sometimes be negative, which is often called antiferromagnetic (Azcoiti et al., 2017). When d_{jk} is positive, neighboring components of $\boldsymbol{\gamma}$ tend to have similar values; When d_{jk} is negative, neighboring components of $\boldsymbol{\gamma}$ tend to have opposite values. We define two B-spline basis functions to be neighbors if they have over-lapped non-zero regions. For non-neighbor pairs (j', k'), we set $d_{j'k'} = 0$. Notice that when all $D \equiv \mathbf{0}$, the Ising prior above is reduced to the independent Bernoulli prior. Without prior knowledge of $\boldsymbol{\gamma}$, we assume that all entries of $\boldsymbol{a}$ equal to a constant a. We also assume all non-zero elements of D to be d. So the Ising prior can be reduced to

$$p(\boldsymbol{\gamma}) \propto \exp\{a\mathbf{1}^T\boldsymbol{\gamma} + d\boldsymbol{\gamma}^T G\boldsymbol{\gamma}\}, \tag{2.9}$$

where $G = \{g_{jk}\}$ and $g_{jk} = 1$ if (j, k) is a pair of neighbors and 0 otherwise. We assume an inverse Gamma prior for σ^2, i.e.,

$$\sigma^2 \sim IG(\nu/2, \nu\lambda/2). \tag{2.10}$$

When $\nu = 0$, the inverse Gamma prior reduces to a flat prior for σ^2, $p(\sigma^2) \propto \frac{1}{\sigma^2}$, which is adopted in this research.

2.2.3 Posterior Inference

When $\{\gamma_j, j = 1, \ldots, p\}$ is given, we have information about which elements of $\boldsymbol{b}$ are zero. We use $\boldsymbol{b}_\gamma$ to denote the nonzero components of $\boldsymbol{b}$ and $\boldsymbol{C}_\gamma$ to denote the submatrix of $\boldsymbol{C}$ whose columns are the columns corresponding to the non-zero entries of $\boldsymbol{b}$. The conditional

distribution of $\boldsymbol{y}$ given $\boldsymbol{\gamma}, \boldsymbol{b}_\gamma$ and σ^2 is

$$\boldsymbol{y} \mid \boldsymbol{b}_\gamma, \boldsymbol{\gamma}, \sigma^2 \sim N\big(\boldsymbol{C}_\gamma \boldsymbol{b}_\gamma, I_n \sigma^2\big). \tag{2.11}$$

With the conditional distribution in (2.11) and the priors of $\boldsymbol{b}$, σ^2, and $\boldsymbol{\gamma}$ in (2.8), (2.10) and (2.9), we get the joint conditional posterior distribution of $\boldsymbol{b}$, σ^2 and $\boldsymbol{\gamma}$ given $\boldsymbol{y}$ by

$$p(\boldsymbol{b}, \sigma^2, \boldsymbol{\gamma} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{b}_\gamma, \boldsymbol{\gamma}, \sigma^2) p(\boldsymbol{b}_\gamma \mid \boldsymbol{\gamma}, \sigma^2) p(\boldsymbol{\gamma}) p(\sigma^2). \tag{2.12}$$

Based on (2.12), the conditional distribution of $\boldsymbol{b}$ given σ^2, $\boldsymbol{\gamma}$ and $\boldsymbol{y}$ is

$$\boldsymbol{b} \mid \sigma^2, \boldsymbol{\gamma}, \boldsymbol{y} \sim N\big(K^{-1} M, K^{-1}\big), \tag{2.13}$$

where $K = \big(\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1}\big)\sigma^{-2}$ and $M = \boldsymbol{C}_\gamma^T \sigma^{-2} \boldsymbol{y}$. By integrating out $\boldsymbol{b}$ from (2.12), we get the conditional distribution of σ^2 given $\boldsymbol{\gamma}$ and $\boldsymbol{y}$:

$$\sigma^2 \mid \boldsymbol{\gamma}, \boldsymbol{y} \sim \text{Inv-Gamma}(\tilde{a}, \tilde{b}), \tag{2.14}$$

where $\tilde{a} = \frac{n}{2}$ and $\tilde{b} = \frac{1}{2} \boldsymbol{y}^T L_\gamma \boldsymbol{y}$ with $L_\gamma = I - \boldsymbol{C}_\gamma \big(\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1}\big)^{-1} \boldsymbol{C}_\gamma^T$. In order to sample from $p(\boldsymbol{\gamma} \mid \boldsymbol{y})$, we would like to derive the conditional posterior odds of γ_j:

$$O_j = \frac{P\big(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{y}\big)}{P\big(\gamma_j = 0 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{y}\big)}, \tag{2.15}$$

where $\boldsymbol{\gamma}_{(-j)} = \big\{\gamma_l, l \neq j\big\}$. Here, O_j can be factored by

$$O_j = \frac{P\big(\boldsymbol{y} \mid \gamma_j = 1, \boldsymbol{\gamma}_{(-j)}\big)}{P\big(\boldsymbol{y} \mid \gamma_j = 0, \boldsymbol{\gamma}_{(-j)}\big)} \cdot \frac{P\big(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}\big)}{P\big(\gamma_j = 0 \mid \boldsymbol{\gamma}_{-j}\big)}. \tag{2.16}$$

To find (2.16) we need to obtain the probability density function (pdf) of $p(\boldsymbol{y} \mid \boldsymbol{\gamma})$ and the prior odds

$$\frac{P\big(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}\big)}{P\big(\gamma_j = 0 \mid \boldsymbol{\gamma}_{(-j)}\big)}.$$

The marginal likelihood $p(\boldsymbol{y} \mid \boldsymbol{\gamma})$ can be derived by integrating out $\boldsymbol{b}$ and σ^2 from

$$p(\boldsymbol{y} \mid \boldsymbol{b}_\gamma, \boldsymbol{\gamma}, \sigma^2) p(\boldsymbol{b}_\gamma \mid \boldsymbol{\gamma}, \sigma^2) p(\sigma^2),$$

which gives

$$p\big(\boldsymbol{y} \mid \boldsymbol{\gamma}\big) = \frac{\Gamma\big(\frac{n}{2}\big)\pi^{-\frac{n}{2}} \left|\Sigma_\gamma^{-\frac{1}{2}}\right| \left|C_\gamma^T C_\gamma + \Sigma_\gamma^{-1}\right|^{-\frac{1}{2}}}{\big(\boldsymbol{y}^T L_\gamma \boldsymbol{y}\big)^{\frac{n}{2}}}. \tag{2.17}$$

Equation (2.17) can be used to obtain the bayes factor

$$\frac{P\big(\boldsymbol{y} \mid \gamma_j = 1, \boldsymbol{\gamma}_{(-j)}\big)}{P\big(\boldsymbol{y} \mid \gamma_j = 0, \boldsymbol{\gamma}_{(-j)}\big)}. \tag{2.18}$$

The prior odds can be easily derived from the Ising prior

$$\frac{P\big(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}\big)}{P\big(\gamma_j = 0 \mid \boldsymbol{\gamma}_{(-j)}\big)} = \frac{P\big(\gamma_j = 1, \boldsymbol{\gamma}_{(-j)}\big)}{P\big(\gamma_j = 0, \boldsymbol{\gamma}_{(-j)}\big)} = \exp\big(a + 2d \cdot \mathbf{1}^T \boldsymbol{\gamma}_{(-j)}\big), j = 1, \ldots, p. \tag{2.19}$$

With the conditional posterior odds, $j = 1, \ldots, p$, γ_j can be sampled from Bernoulli $\left(\frac{O_j}{1+O_j}\right)$ for $j = 1, \ldots, p$.

MCMC algorithm for the continuous response case (gibbs) Based on the conditional distributions derived above, we propose an MCMC algorithm for posterior sampling. The steps of the MCMC algorithm are described as follows.

Step 0 Set initial values for $\boldsymbol{\gamma}$, $\boldsymbol{b}$ and σ^2 as $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{b}^{(0)}$ and $\sigma^{2(0)}$.

Step 1 For $j = 1, \ldots, p$, conditioning on $\boldsymbol{y}$, sample $\gamma_j \sim \text{Bernoulli}\big(\frac{O_j}{1+O_j}\big)$.

Step 2 Update σ^2 based on $P(\sigma^2 \mid \boldsymbol{\gamma}, \boldsymbol{y})$.

step 3 Update $\boldsymbol{b}$ based on $P(\boldsymbol{b} \mid \sigma^2, \boldsymbol{\gamma}, \boldsymbol{y})$.

Repeat step 1-3 until the maximum number of iterations is achieved.

Before running the MCMC algorithm, we need to set initial values for $\boldsymbol{\gamma}$, $\boldsymbol{b}$ and σ^2. Denote their initial values by $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{b}^{(0)}$ and $\sigma^{2(0)}$ respectively. To encourage sparsity, we fit a Lasso Regresion on $\boldsymbol{y}$ and C to get $\boldsymbol{b}^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$, and estimate $\sigma^{2(0)}$ by using

$$\frac{1}{n}\big(\boldsymbol{y} - C\boldsymbol{b}^{(0)}\big)^T\big(\boldsymbol{y} - C\boldsymbol{b}^{(0)}\big).$$

2.2.4 Extension to the Probit Model Case

In the model proposed in Section 2.2.1, if the response y_i takes values in $\{0, 1\}$, we introduce a univariate latent variable z_i which links the response y_i to the predictor (Zhu et al., 2007, 2010). In particular,

$$y_i = \begin{cases} 1, & \text{if } z_i < 0, \\ 0, & \text{if } z_i \geq 0, \end{cases} \tag{2.20}$$

$$z_i = \boldsymbol{c}_i^T\boldsymbol{b} + \epsilon_i, \quad \epsilon_i \sim N(0, 1), \quad i = 1, \ldots, n. \tag{2.21}$$

In (2.21), the structures of $\boldsymbol{c}_i$ and $\boldsymbol{b}$ are exactly the same as those in the linear model case. Notice that there is no σ^2 in the distribution of ϵ_i. For $j = 1, \ldots, p$, the conditional prior of b_j given γ_j is

$$b_j \mid \gamma_j \sim (1 - \gamma_j)I_0 + \gamma_j N(0, v^2), \quad j = 1, \ldots, p,$$

$$b_0 \sim N\big(0, v_0^2\big).$$

If we write the expressions above in a matrix form, we have

$$\boldsymbol{b} \mid \boldsymbol{\gamma} \sim N\big(\boldsymbol{0}, \Sigma_\gamma\big). \tag{2.22}$$

with $\Sigma_\gamma = \text{diag}\big\{v_0^2, \big(\gamma_j v^2 + (1 - \gamma_j)I_0\big)_{j=1}^p\big\}$. Same as before, we assume that $\boldsymbol{\gamma}$ has an Ising prior, i.e.,

$$p(\boldsymbol{\gamma}) \propto \exp\big\{a\boldsymbol{1}^T\boldsymbol{\gamma} + d\boldsymbol{\gamma}^T G\boldsymbol{\gamma}\big\}, \tag{2.23}$$

where a, d, G are explained in Section 2.2.2. From (2.20) and (2.21), it is easy to see that the conditional distribution of z_i given y_i and $\boldsymbol{b}$ is a truncated normal distribution, i.e.,

$$z_i \mid y_i, \boldsymbol{b} \sim \mathrm{TN}\big(\boldsymbol{c}_i^T \boldsymbol{b}, 1\big)\big\{I\big(z_i < 0\big)I\big(y_i = 1\big) + I\big(z_i \geq 0\big)I\big(y_i = 0\big)\big\}. \tag{2.24}$$

Denote $\boldsymbol{z} = (z_1, \ldots, z_n)^T$. With the conditional distribution (2.24) and the prior distributions of (2.22) and (2.23), we get the joint posterior distribution of $\boldsymbol{b}$, $\boldsymbol{\gamma}$ and $\boldsymbol{z}$ given $\boldsymbol{y}$ by

$$p\big(\boldsymbol{b}, \boldsymbol{\gamma}, \boldsymbol{z} \mid \boldsymbol{y}\big) \propto p(\boldsymbol{z} \mid \boldsymbol{b}_\gamma, \boldsymbol{\gamma}, \boldsymbol{y}) p\big(\boldsymbol{b}_\gamma \mid \boldsymbol{\gamma}\big) p\big(\boldsymbol{\gamma}\big). \tag{2.25}$$

After integrating out $\boldsymbol{b}$ in (2.25), the marginal conditional posterior distribution of $\boldsymbol{\gamma}$ given $\boldsymbol{z}$ and $\boldsymbol{y}$ is

$$p\big(\boldsymbol{\gamma} \mid \boldsymbol{z}, \boldsymbol{y}\big) \propto |2\pi\Sigma_\gamma|^{-\frac{1}{2}} \left|2\pi \widetilde{K}_\gamma^{-1}\right| \exp\big\{\frac{1}{2}\widetilde{M}_\gamma^T \widetilde{K}_\gamma \widetilde{M}_\gamma\big\} p\big(\boldsymbol{\gamma}\big), \tag{2.26}$$

where $\widetilde{K}_\gamma = \big(C_\gamma^T C_\gamma + \Sigma_\gamma^{-1}\big)$ and $\widetilde{M}_\gamma = C_\gamma^T \boldsymbol{z}$. Similar as in the linear model case, we use the conditional posterior odds $\widetilde{O}_j$ to sample γ_j. In particular, $\widetilde{O}_j$ takes the form

$$\widetilde{O}_j = \frac{P\big(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{z}, \boldsymbol{y}\big)}{P\big(\gamma_j = 0 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{z}, \boldsymbol{y}\big)}. \tag{2.27}$$

Here, $\widetilde{O}_j$ can be easily derived based on (2.26). Details for the formula are provided in Appendix B. From (2.25), we also obtain the conditional distribution of $\boldsymbol{b}$ given $\boldsymbol{z}$, $\boldsymbol{\gamma}$ and $\boldsymbol{y}$. In particular,

$$\boldsymbol{b} \mid \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{y} \sim N\big(\widetilde{K}_\gamma^{-1} \widetilde{M}_\gamma, \widetilde{K}_\gamma^{-1}\big). \tag{2.28}$$

MCMC algorithm for the binary response case (gibbs) Based on the proposed probit model and posterior inference, we now illustrate the MCMC algorithm for the posterior sampling:

Step 0 Set initial values for $\boldsymbol{\gamma}$ and $\boldsymbol{b}$ as $\boldsymbol{\gamma}^0$ and $\boldsymbol{b}^0$.

Step 1 For $i = 1, \ldots, n$, conditioning on y_i, $\boldsymbol{b}$ and $\boldsymbol{\gamma}$, sample z_i from:

$$z_i \mid y_i, \boldsymbol{b}, \boldsymbol{\gamma} \sim \mathrm{TN}\big(\boldsymbol{c}_i^T \boldsymbol{b}, 1\big)\big\{I\big(z_i < 0\big)I\big(y_i = 1\big) + I\big(z_i \geq 0\big)I\big(y_i = 0\big)\big\}.$$

Step 2 For $j = 1, \ldots, p$, conditioning on $\boldsymbol{z}$, $\boldsymbol{y}$, sample γ_j from Bernoulli$\big(\frac{\widetilde{O}_j}{1+\widetilde{O}_j}\big)$.

Step 3 Given $\boldsymbol{z}$, $\boldsymbol{\gamma}$ and $\boldsymbol{y}$, sample $\boldsymbol{b}$ from

$$\boldsymbol{b} \sim p\big(\boldsymbol{b} \mid \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{y}\big).$$

Repeat step 1-3 until maximum iteration number is achieved.

To get an initial value of $\boldsymbol{b}$ and $\boldsymbol{\gamma}$ we can fit a Lasso-glm Regression with $\boldsymbol{y}$ and C to get the initial value for $\boldsymbol{\gamma}$ and $\boldsymbol{b}$, $\boldsymbol{\gamma}^0$ and $\boldsymbol{b}^0$.

2.3 Simulation Study

We designed two simulation studies to demonstrate the performance of our proposed approach, one for the continuous response case and the other for the binary response case. Our goal is to compare the proposed Bayesian Functional Region Selection (BRS) methods with existing methods. For the continuous response case, we will compare our methods with four other methods, two of which do not encourage sparse estimation—FLM through B-spline basis representation (FLMBBR) (Ramsay and Silverman, 1997) and FLM through Functional Principal Components Analysis (FLMFPC) (Yao et al., 2005); the other two methods encourage sparse estimation of $\beta(t)$—the Functional Linear Regression That's Interpretable(FLiRTI) method of James et al. (2009) and the Smooth and Locally Sparse(SLoS) method of Lin et al. (2015). For the binary response case, since no existing approach is available for the sparse estimation of $\beta(t)$, we only compared our method with two existing methods that are based on functional generalized linear models (FGLM), including the basis representation approach using B-splines (FGLMBR) by Febrero-Bande et al. (2012) and the Functional Principle Component Analysis

(FGLMFPC) based approach by Müller and Stadtmüller (2005). Note that since all existing methods we compared with are Frequentist methods and do not have uncertainty result, we have performed 1000 bootstraps and used the bootstrap distribution of the estimated $\beta(t)$ (and σ^2) to characterize the uncertainty of these estimates.

To produce functional predictors, we generated $n = 300$ smooth trajectories by sampling from a Gaussian process with mean $\mu(t) = \sqrt{16t} + 0.6\sin(16t) - 1.5$ and covariance kernel $\kappa(s,t) = (\alpha|s-t|)^{\nu} K_{\nu}(\alpha|s-t|)$, where K_{ν} is the modified Bessel function of the second kind. Here, we set $\nu = 0.5$ and $\alpha = 1.5$. These trajectories are sampled on an equally spaced dense grid of $T = [0,1]$, with the number of grid points $m = 400$. The true regression coefficient function was set to be $\beta(t) = \cos(4\pi t)\, I_{[0,\,0.375]}(t) + \cos(16\pi t)\, I_{[0.71875,\,0.78125]}(t)$, which consists of two non-zero regions at $[0, 0.375]$ and $[0.71875, 0.78125]$ as well as two zero regions. The function $\beta(t)$ has lower frequency on the first non-zero region and higher frequency on the second non-zero region. In the continuous response case, $\{Y_i\}$ are generated from $N\big(\int_T X_i(t)\beta(t)dt,\, \sigma^2\big)$ with $\sigma = 0.05$. Here, the integral was evaluated using numerical integration via trapezoidal rule. In the binary response case, we rescale the $\beta(t)$ used in the continuous response case by 20 to improve signal-to-noise ratio. We denote the rescaled true $\beta(t)$ $\beta^*(t)$. Binary responses were simulated by first sampling $Z_i \sim N\big(\int_T X_i(t)\beta^*(t)dt, 1\big)$ and then setting $Y_i = I_{\{Z_i<0\}}$.

2.3.1 Results for the Continuous Response Case

For the continuous response data, the proposed BRS method introduced in Sections 2.2.1–2.2.3 was applied. In order to capture local features at both lower and higher frequencies, we let $\boldsymbol{\Phi}$ to contain B-spline basis functions at two different scales (i.e., $L = 2$), one with $(q_1 = 2, k_1 = 20)$, another with $(q_2 = 3, k_2 = 40)$. This gives 65 basis functions. To avoid numerical collinearity, we let the knots locations of the two B-spline sets to be uniformly distributed on the interval T. The neighborhood structure of all basis functions is summarized using an adjacency matrix $\boldsymbol{G} = \big(g_{jk}\big)$, i.e., if the jth and the kth basis functions have overlapped supports, set $g_{jk} = 1$, otherwise set $g_{jk} = 0$. We initialized $\boldsymbol{b}$, $\boldsymbol{\gamma}$ and σ^2 using the Lasso

regression (Tibshirani, 1994) and applied a Gibb sampler to obtain posterior samples. We adopted the simplified two-parameter formulation of the Ising hyper-prior and set the hyper-parameters to be $a = -9$, $d = 1.5$ and $v = 10$. We performed 6000 MCMC iterations and used 5000 iterations as the burn-in period. Posterior samples of $\beta(t)$ was constructed using basis expansion: $\beta^{(h)}(t) = \sum_{l=1}^{2} \sum_{j=1}^{q_l+k_l} b_j^{l(h)} \phi_j^l(t)$ for $h = 1, \ldots, H$. Here h is the index for posterior samples and H is the total number of posterior samples obtained after burn-in.

The posterior mean or bootstrap mean of $\beta(t)$, denoted by $\widehat{\beta}(t)$, and its 95% pointwise credible band (PCB95) and 95% simultaneous credible band (SB95) are calculated and plotted in Figure 2.1. The 95% pointwise credible band was obtained by taking the 0.025 and 0.975 quantiles of the posterior samples pointwisely; and the 95% simultaneous credible band was calculated by finding a constant M so that 95% of the posterior samples of $\beta(t)$ fall in the interval $\widehat{\beta}(t) \pm M s_j(t)$, where $s_j(t)$ denotes the standard deviation at t (Ruppert et al., 2003).

In Figure 2.1, we compared the $\beta(t)$ estimate of BRS with that obtained by FLiRTI, SLoS, FLMBBR and FLMFPC. From Figure 2.1, we observed that the proposed BRS method successfully identified the two non-zero regions. In contrast, FLMBBR, FLMFPC, and SLoS obtained non-zero $\beta(t)$ on all regions; FLiRTI identified zero regions but the values of $\beta(t)$ on the high frequency region were under-estimated.

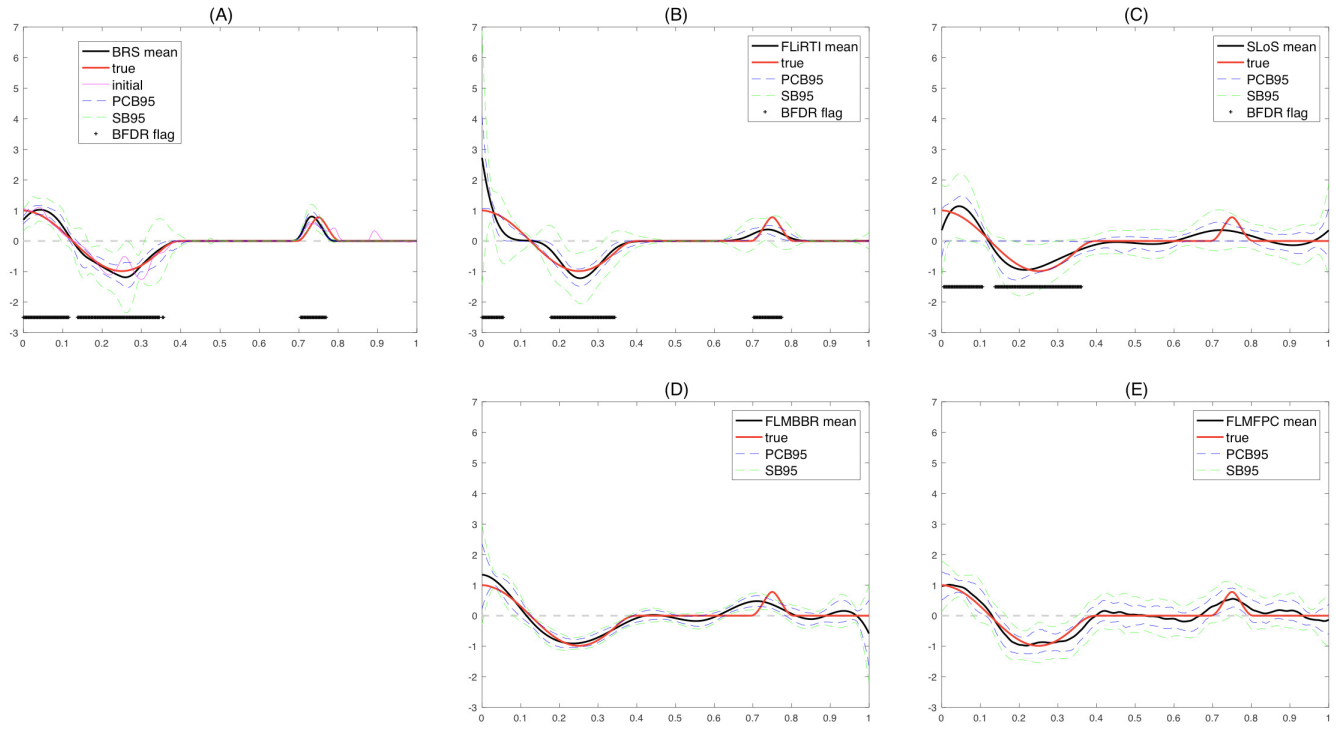

Figure 2.1: Estimates of $\beta(t)$ using BRS, compared with FLiRTI, SLoS, FLMBBR and FLMFPC. The blue dashed curves illustrates the 95% pointwise credible band and the green dashed curves illustrates the 95% simultaneous credible band. In BRS, FLiRTI and SLoS, the zero region is flagged by "+" sign at the bottom of the plot. (A) BRS, (B) FLiRTI, (C) SLoS, (D) FLMBBR, (E) FLMFPC.

In addition to the two figures above, we measured the estimation performance of $\beta(t)$ using five summary statistics, including (1) Integrated Width (IWidth) of the pointwise credible band or bootstrap confidence band calculated by IWidth $= \int_T \{U(t) - L(t)\}\, dt$, where $U(t)$, $L(t)$ are the upper and lower bounds of the 95% credible bands; (2) Integrated Squared Error (ISE) that characterizes the variability of the posterior mean about the truth, calculated by ISE $= \int_T \left\{\widehat{\beta}(t) - \beta(t)\right\}^2 dt$; (3) Integrated Posterior Variability (IPVar) that characterizes the variability of the posterior (bootstrap) samples about the posterior mean (bootstrap mean), calculated by IPVar $= H^{-1} \sum_{h=1}^{H} \int_T \left\{\beta^{(h)}(t) - \widehat{\beta}(t)\right\}^2 dt$; (4) Coverage Probability for the 95% Pointwise Credible Band (CPrPCB95), obtained by counting the proportion of $\beta(t)$ falling in the 95% pointwise credible band; and (5) Coverage Probability for the 95% Simultaneous Credible Band (CPrSB95), which counts the proportion of $\beta(t)$ falling in the 95% simultaneous credible band. In addition, we also calculated two summary statistics to assess the estimation

of σ^2. They are Squared Error (SE) defined by $\text{SE} = \left(\widehat{\sigma} - \sigma\right)^2$ and the Posterior Variance (PVar) defined by $\text{PVar} = H^{-1}\sum_{h=1}^{H}\left(\sigma^{(h)} - \widehat{\sigma}\right)^2$, where $\widehat{\sigma}$ denotes the posterior mean of σ.

Table 2.1: Simulation results: a comparison of BRS with existing methods (FLMBBR, FLMFPC, FLiRTI, and SLoS) for the estimation of $\beta(t)$ and σ for both continuous response and binary response cases. Note that for the binary response case, the statistics for the σ parameter are omitted because there is no σ in the model.

	Method	$\beta(t)$					σ	
		IWidth	ISE	IPVar	CPrPCB95	CPrSB95	SE	PVar
Case 1	BRS	0.195	0.013	0.008	0.805	0.913	2.01×10^{-6}	5.04×10^{-6}
	FLMBBR	0.393	0.030	0.014	0.763	0.878	1.40×10^{-5}	3.36×10^{-6}
	FLMFPC	0.631	0.019	0.026	1.000	1.000	6.72×10^{-5}	6.69×10^{-5}
	FLiRTI	0.238	0.050	0.009	0.743	1.000	1.01×10^{-5}	2.68×10^{-5}
	SLoS	0.647	0.030	0.024	0.960	1.000	2.91×10^{-6}	5.21×10^{-5}
Case 2	BRS	7.358	17.242	11.78	0.898	0.953	–	–
	FGLMBR	31.35	263.56	80.84	0.648	0.945	–	–
	FGLMFPC	2.479	75.693	0.511	0.575	0.593	–	–

Results of summary statistics are listed in the top panel of Table 2.1. From Table 2.1, we observe that BRS produced more accurate estimation of $\beta(t)$ than all existing methods, as indicated by the smallest IWidth, ISE, and IPVar. For the coverage probabilities, FPCA, FLiRTI, and SLoS have wider confidence bands, which induce higher coverage probabilities for either CPrPCB95 or CPrSB95, but their also have larger estimation errors as indicated by higher ISE values. For the estimation of σ, BRS achieves the lowest SE and PVar among all methods.

2.3.2 Results for the Binary Response Case

For the binary response case, we initialized $\mathbf{b}$ and $\boldsymbol{\gamma}$ by applying the generalized linear model with Lasso regularization (Friedman et al., 2010), and applied a Gibbs sampler following the conditional distributions in Section 2.2.4. We set the hyper-parameters to be $a = -9$, $d = 1.5$

and $v = 1$. A total of 6000 MCMC iterations are performed with a burn-in period of 5000 iterations. In the bottom panel of Table 2.1, we list the summary statistics for the estimation of $\beta(t)$. Also, the posterior mean or bootstrap mean of $\beta(t)$, denoted by $\widehat{\beta}(t)$, and its 95% pointwise credible band (PCB95) and 95% simultaneous credible band (SB95) are calculated and plotted in Figure 2.2. Looking at these result, we notice that $\beta(t)$ is much harder to be estimated compared to the continuous response case. This is not a surprise because binary responses carry much less information than continuous responses. Nevertheless, BRS still gives the lowest ISE among all three methods. Furthermore, the point-wise band and the simultaneous band of BFRS achieve approximately 0.95 coverage probabilities, which is higher than both the FGLMBR and FGLMFPC methods.

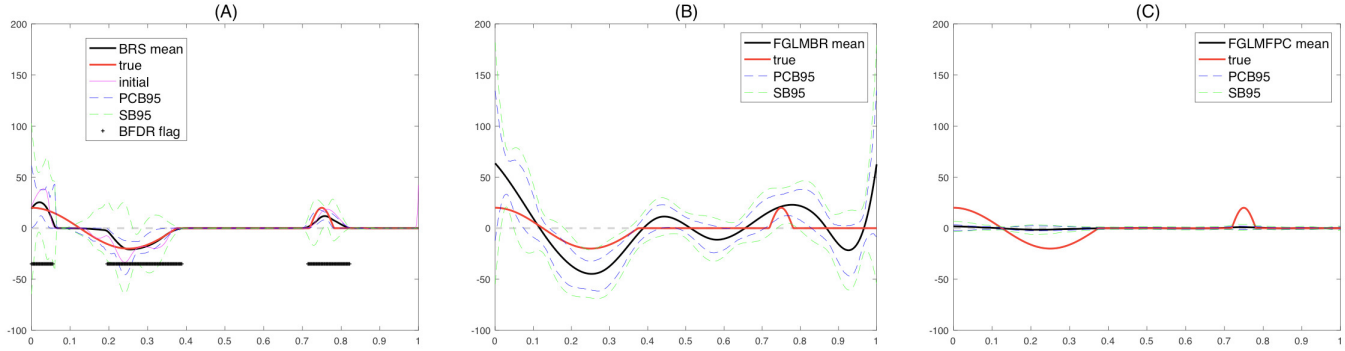

Figure 2.2: Estimates of $\beta(t)$ using BRS, compared with FGLMBR and FGLMFPC. The blue dashed curves illustrates the 95% pointwise credible band and the green dashed curves illustrates the 95% simultaneous credible band. (A) BRS, (B) FGLMBR, (C) FGLMFPC.

2.4 Real Data Application

2.4.1 Application to the Tecator Data

To demonstrate the performance of the proposed method, we apply the proposed BRS method to the Tecator data by regressing the protein content of a meat sample on the near infrared absorbance spectral measurement. The dataset consists of measurements from 240 meat sample collected by the company Tecator. It is publicly available on the Statlib website (`http://lib.stat.cmu.edu/datasets/`). The near infrared spectral measurements were obtained using a spectrometer, and the spectral curves were recorded at wavelengths ranging from 850 nm to 1050 nm. Each meat sample consists of 100 channel spectrum (100 grid points) for absorbance, as well as three potential responses (water, fat and protein). Figure-2.3 demonstrate a plot of the spectral curves. We choose protein as the response. We have two major interests for this analysis. First, we want to see whether zero regions of the coefficient function can be detected by using BRS. Second, we want to evaluate the prediction performance of BRS. To make the calculation more stable, we did a transformation from the original time domain $[850, 1050]$ to $[0, 1]$ by letting $t' = \frac{t-850}{200}$. Like in the simulation study, we introduce two B-spline sets $\left\{\phi_j^1(t')\right\}_{j=1}^{q_1+k_1}$ and $\left\{\phi_j^2(t')\right\}_{j=1}^{q_2+k_2}$ with $\big(q_1 = 2, k_1 = 10\big)$ and $\big(q_2 = 3, k_2 = 30\big)$. The knots of the two basis sets are uniformly distributed along the domain $[0, 1]$. After determining the adjacency matrix G, we set a = -9, d = 1.5 and v = 25. A total of 6000 MCMC iterations and a 5000 burn-in period is adopted in the computation. In Figure 2.4, we plotted the estimated $\beta(t)$ using the proposed BRS approach and compared with estimates obtained using four other approaches (FLiRTI, SLoS, FLMBBR, and FLMFPC).

From Figures 2.4 we can see that all five estimated coefficient functions show somewhat similar patterns. All of them achieve the peak around 0.15 and reach the valley around 0.4. For the three region selection methods (BRS, FLiRTI, and SLoS), the plot of BRS shows that the region $[0.5, 1]$ is exactly zero, while FLiRTI shows the region $[0.5, 1]$ is roughly zero and SLoS only suggests a small zero region around $[0.8, 0.9]$. For the two non-region selection

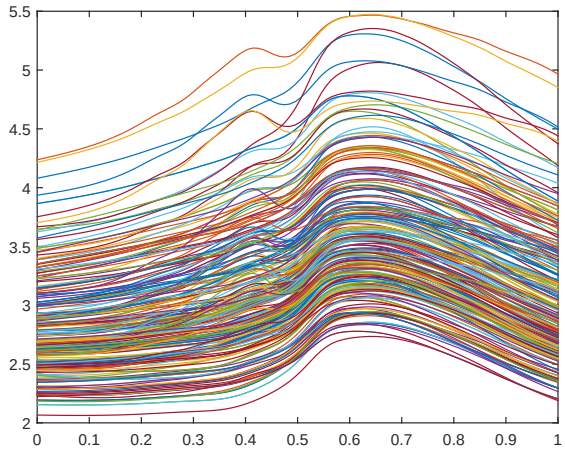

Figure 2.3: The near infrared absorbance spectrum.

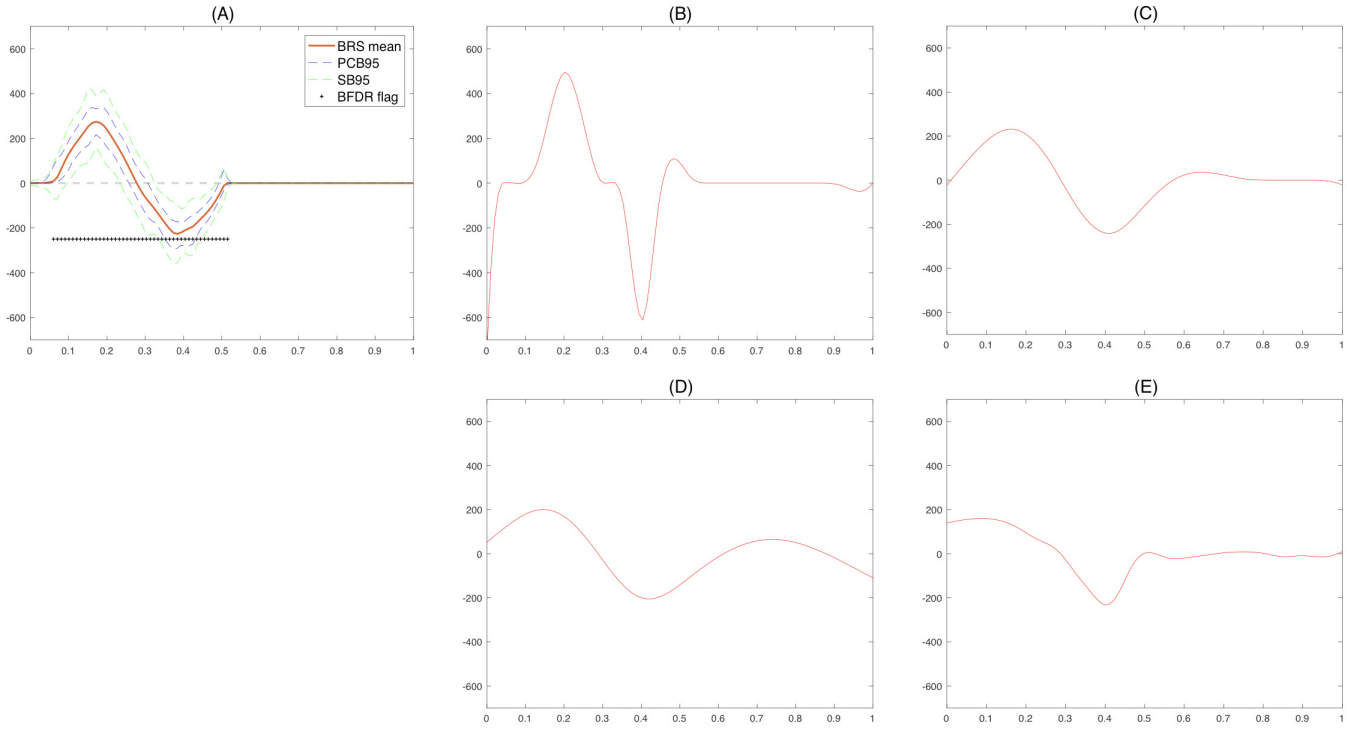

Figure 2.4: Estimates of $\beta(t)$ using BRS, compared with FLiRTI, SLoS, FLMBBR and FLMFPC for the Tecator data. (A) BRS, (B) FLiRTI, (C) SLoS, (D) FLMBBR, (E) FLMFPC. The non-zero regions detected by the BRS method is denoted by the "+" signs on the bottom region of (A).

methods (FLMBBR and FLMFPC), the plot of FLMFPC demonstrates that region $[0.5, 1]$ is close to zero. In contrast, FLMBBR cannot detect any zero regions. Furthermore, the magnitude of the coefficient function generated by using FLiRTI is larger than that produced by all other four methods. In addition to estimate $\beta(t)$, we also compared the prediction performance of BRS with the four other methods. In particular, we applied 10-fold cross-validation to evaluate the performance of predicting the protein content and calculated the averaged Root Mean Squared Error (RMSE). The result is shown in Table 2.2. From Table 2.2, we see that FLiRTI produces the smallest RMSE for prediction. We can see that the three region selection methods (BRS, FLiRTI, SLoS) achieve lower average RMSE compared to the two non-region-selection methods (FLMBBR, FLMFPC). The average RMSE of BRS is between that of FLiRTI and SLoS, indicating a reasonably good prediction performance among the region selection methods.

Table 2.2: The average RMSE for prediction by using 10-fold CV for the five methods.

BRS	FLiRTI	SLoS	FLMBBR	FLMFPC
1.6409	1.4998	1.6655	1.7774	2.0099

2.4.2 Application to the Sonar Data

Sonar (Sound Navigation and Ranging) is a technique that uses sound propagation to detect, localize, and identify objects for purposes such as navigation. In a study that investigates echoes in natural environment, a sonar device was used to collect echoes from three types of terrain substrates: grass, sand, and a simulated tropical rainforest floor. Here we use a subset of the collected data which contain measurements from grass and rainforest. The data consist of 1160 echo envelopes, of which 580 are from grass and 580 are from rainforest. Response is 1 if echo is from the grass and 0 if from the rainforest. The time domain $t \in T$ ranges from 4.96 ms to 20.39 ms and the total number of grid points is 644. We want to see whether BRS can detect zero regions of the coefficient function for a classification problem and whether it has the ability to perform binary classification with a reasonably high accuracy. We apply three methods (BRS, FGLMBR, FGLMFPC) and plot the estimated coefficient function $\beta(t)$. A 10-fold cross-validation is used and the averaged AUC (Area Under the Curve) is adopted to evaluate the classification performance of the three methods. To make the calculation more stable, we transformed the time domain $t \in [4.96, 20.3920]$ to $[0, 1]$ by letting $t' = \frac{t-4.96}{15.4320}$. We introduce two B-spline sets $\left\{\phi_j^1(t')\right\}_{j=1}^{q_1+k_1}$ and $\left\{\phi_j^2(t')\right\}_{j=1}^{q_2+k_2}$ with $\big(q_1 = 2, k_1 = 20\big)$ and $\big(q_2 = 3, k_2 = 30\big)$. The knots of the two basis sets are uniformly distributed on the domain $[0, 1]$. After determining the adjacency matrix G, we set a = -9, d = 1.5 and v = 20. A total of 6000 MCMC iterations with a 5000 burn-in period is adopted in the computation. We show the plots of the estimated coefficient functions in Figures 2.5.

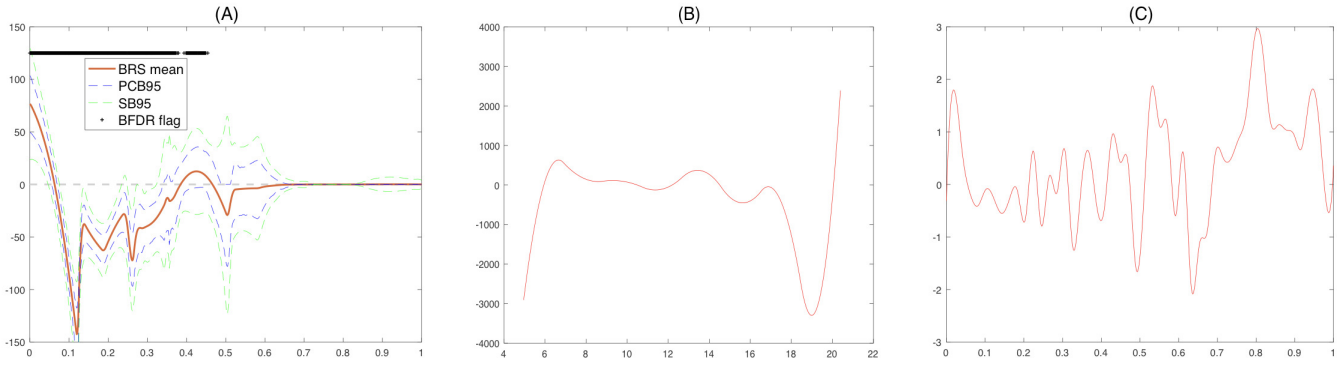

Figure 2.5: Estimates of $\beta(t)$ using BRS, compared with FGLMBR and FGLMFPC for the Sonar data. (A) BRS, (B) FGLMBR, (C) FGLMFPC. The non-zero regions detected by the BRS method is denoted by the "+" signs on the top region of (A).

From Figures 2.5, we can see that the three estimated coefficient functions show totally different patterns. The shapes and magnitude for the estimated coefficient functions are different. FGLMFPC gives the smallest magnitude while FGLMBR gives the largest. Only BRS is able to detect zero regions. The averaged AUC for 10-fold cross validation by using each of the three methods is listed in Table 2.3. Table 2.3 shows that our method BRS outperforms FGLMFPC and is comparable with FGLMBR. This analysis demonstrates that our method can detect zero regions and enjoys reasonable prediction performance.

Table 2.3: The AUC of prediction by using the 10-fold CV.

Method	BRS	FGLMBR	FGLMFPC
Average AUC	0.9936	0.9962	0.9877

2.5 Hyperparameter Setups

Setting hyper-parameters plays an important role in Bayesian inference. For Ising prior, special attention needs to be paid to the phase transition issue. The phase transition is a phenomenon which occurs when a small change in a parameter leads to a large-scale, qualitative change in the state of a system. It is well known that Ising models with dimension greater than one exhibit this phenomenon at or close to the phase transition boundary for some choices of

the hyper-parameters (Li and Zhang, 2010). The phase transition can lead to several serious consequences such as a dramatic change in the proportion of $\gamma_i = 1$ while performing Bayesian variable selection for high-dimensional $\boldsymbol{\gamma}$ and non-convergence of the MCMC algorithm.

Under the setting of the Ising prior (2.9), two hyper-parameters need to be pre-determined, the sparsity parameter a ($a \leq 0$) and the smoothness parameter d ($d \geq 0$). As we mentioned before, a controls the sparsity of $\boldsymbol{\gamma}$ and d quantifies the coupling strength of the neighbor pairs (γ_j, γ_k) where $k \in N(j)$ and $N(j)$ is the neighbor of j. In this section, we performed a numerical analysis, following McEvoy et al. (2013), to study the properties of Ising prior. This analysis provides a guidance on setting the parameters for the Ising prior. Our goal is to chose a and d to take into account prior knowledge about $\boldsymbol{\gamma}$ and to avoid phase transition issues.

First, we carry out an investigation on how d affects the conditional probability $P(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)})$ in the Ising prior, which takes the form

$$P(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}) = \frac{1}{1 + \exp(-a - 2d \sum_{k \in N(j)} I(\gamma_k = 1))}. \tag{2.29}$$

When $a = 0$, (2.29) becomes

$$P(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}, a = 0) = \frac{1}{1 + \exp(-2d \sum_{k \in N(j)} I(\gamma_k = 1))}. \tag{2.30}$$

Here, (2.30) is greater than or equal to (2.29). We can further maximize (2.30) by replacing $\sum_{k \in N(j)} I(\gamma_k = 1))$ by $|N(j)|$, where $|N(j)|$ denotes the total number of neighbors of γ_j. Therefore, maximum conditional probability can be written as

$$\frac{1}{1 + \exp(-2d|N(j)|)}.$$

We plot the maximum conditional probability for different values of d in Figure 2.6. From Figure 2.6, we notice that when $d = 0$, $\text{Max}P(\gamma_j \mid \boldsymbol{\gamma}_{(-j)})$ is a constant, 0.5, which corresponds to the situation when no coupling strength exists among $\boldsymbol{\gamma}$. In this case, the Ising prior reduces

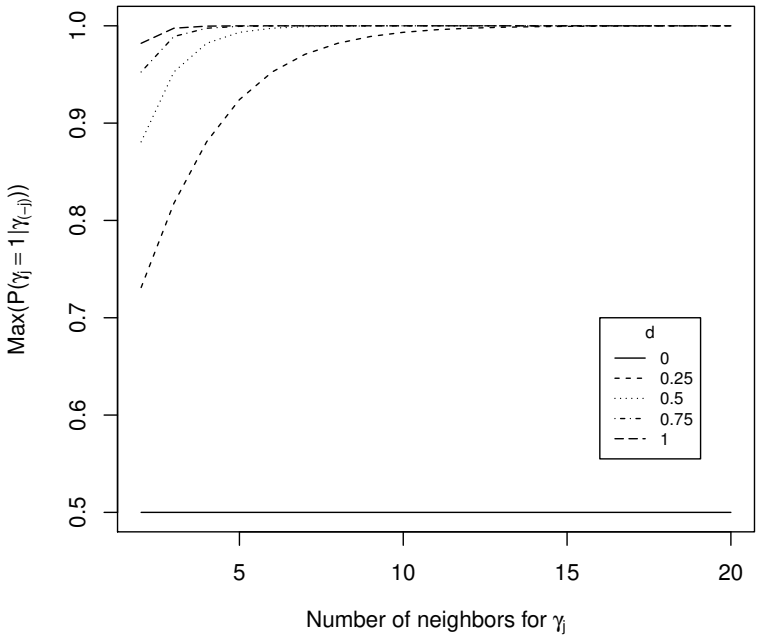

Figure 2.6: Maximal conditional prior probability for $\gamma_j = 1$.

to Bernoulli(0.5). As d increases, the value of $\text{Max}P(\gamma_j \mid \boldsymbol{\gamma}_{(-j)})$ increases faster towards 1, indicating that the smoothness parameter d plays an import role in controlling the coupling strengths between components of $\boldsymbol{\gamma}$.

In addition to the maximum conditional probability, we also performed a gibbs-sampling scheme to empirically evaluate how the sparsity parameter a and the smoothness parameter d jointly influence the expected number of 1's in $\boldsymbol{\gamma}$. The neighboring structure of $\boldsymbol{\gamma}$ is given by the adjacency matrix G from Section 2.3.1. We adopt the following gibbs-sampling scheme

Step 0 : Set initial values for $\boldsymbol{\gamma}$.

Step 1 : For $j = 1, \ldots, p$, calculate $h_j = \exp(-a - 2d \sum_{k \in N(j)} I(\gamma_k = 1))$, and sample $\gamma_j \sim$ Bernoulli$\big(\frac{1}{1+h_j}\big)$.

We ran the above algorithm to generate 1500 samples of $\boldsymbol{\gamma}$ for different combinations of the Ising parameters (a, d). In Figure 2.7, we plot how the average number of 1's in $\boldsymbol{\gamma}$ changes for different $(a, d)'s$. For a fixed d, if we reduce the absolute value of a, the average number of

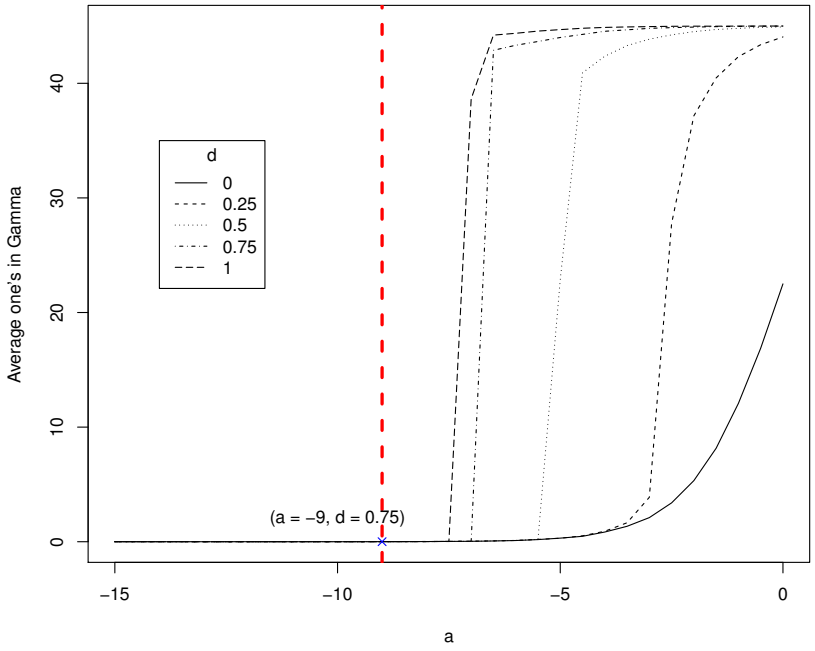

Figure 2.7: Average number of 1's in $\boldsymbol{\gamma}$ for different a and d.

1's will increase drastically at certain value, exhibiting a phase transition. Generally, a and d need to be chosen carefully to prevent phase transition phenomenon. The parameter a used in our simulation is set to be -9, and d is set to be 0.75, marked by red dashed line in Figure 2.7. These parameters are located in a non-phase-transition region. The expected number of 1's in $\boldsymbol{\gamma}$ is 0.004 based on 1500 gibbs samples after a 10,000 burn-in period.

2.6 Discussion

To detect regions of functional data that are relevant to the variables of interest, we have proposed the first Bayesian method for functional data region selection. The proposed method achieves region selection via sparse estimation of the regression coefficient function in a functional regression framework. We have investigated both continuous response and binary response cases. In comparison with existing methods which are all from Frequentist perspective, our approach provides full Bayesian inference that naturally characterizes uncertainty of the model parameters, allows over-complete basis which brings extra flexibility of capturing heterogeneous local features of functional data, and facilitates convenient computation via MCMC algorithms.

In our simulation and real data analysis, we have focused on relatively smooth regression coefficient functions, so that the total number of basis functions used is relatively small ($<$100). When the number of basis functions is large (e.g., $>$1000), computational scalability needs to be carefully investigated. Figure 2.8 shows an empirical analysis on the computational scalability, in which we calculated run-time of the core Gibbs sampler (coded in C++) for 1,000 MCMC iterations under different number of bases. Let p denote the number of bases, and assume that the number of nonzero components of $\beta(t)$ increases with p at a rate $p^{1/r}$, $r > 1$, then the run-time of our current Gibbs sampler is on the scale of $O(p^{(3+r)/r})$.

Rue et al. (2009) argues that since $\boldsymbol{b}$ and the latent variable $\boldsymbol{z}$ are highly correlated, there might be a mixing issue in the MCMC procedure for the Probit model. While the theoretical convergence of a Gibbs sampler is guaranteed by the ergodic theorem (Tierney, 1994), the practical mixing and convergence often need to be monitored and tested. In practice, the convergence of MCMC can be checked using various diagnostic tools such as the trace plots, the autocorrelation plots, and testing approaches such as Geweke's Z-statistics (Geweke et al., 1991) or the Gelman-Rubin diagnostics (Gelman et al., 1992). An overview of the diagnostic tools can be found in Brooks and Gelman (1998). In our simulation and real data analysis, we have used relatively small number of basis functions ($<$100), and the convergence has been

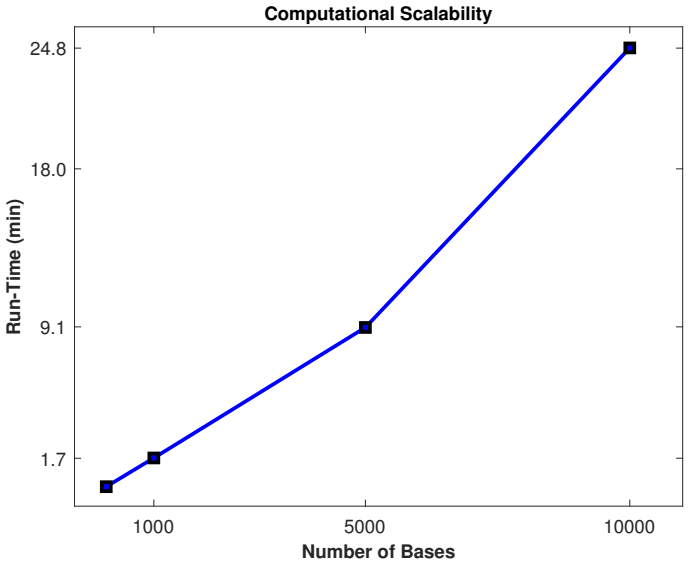

Figure 2.8: Computational scalability.

checked by visualizing the trace plots of the MCMC samples. In case that more basis functions are involved, more automatic diagnostic testing need to be adopted to guarantee convergence.

We have been focused on one-dimensional domains $T \in \mathbb{R}$. Extensions to higher-dimensional domains is straightforward. In high dimensional domains, finite element basis or wavelet basis may be more convenient to use than multi-scale B-splines. We have adopted MCMC algorithms via a Gibbs sampler. Fast stochastic search algorithms can be used via improved MCMC such as the shot-gun algorithm (Hans et al., 2007). Furthermore, improved mixing can be achieved by assuming truncated sparsity prior (Yang et al., 2016).

Chapter 3 Constructing Diversified Portfolios by Detecting Communities in Stock Co-movement Network

3.1 Introduction

In financial investment, risk refers to any uncertainty about the investment that may negatively affect the investor's financial welfare. The uncertainty in the market and the potential bias in human decision making can result in a high risk lurking in the invested assets, which might lead to a massive loss of the original investment.

In financial investment, investors are looking for opportunities with high returns in general. However, these opportunities are usually coupled with higher level of risk because of the trade-off between risk and return. Therefore, controlling the risk is indispensable in the search of high-return opportunities.

Various theoretical work and approaches for portfolio management have been proposed. Markowitz (1952) founded the modern portfolio theory by linking the risk and return of portfolio. Markowitz's theory provides an approach for assets allocation under given risk and return by solving an optimization problem. However, the two major drawbacks of his approach are as follows: (1) Theoretically, it only gives suggestions on asset allocation rather than stock picking; (2) To solve the assets allocation problem, optimizing algorithms are often used, which requires inverting correlation matrices of stock returns. The computation becomes intractable when the population of stocks is large. In order to alleviate the second drawback, some strategies were developed by Amenc and Le Sourd (2005). However, their assumption and complicated computation procedures limited their usage in real world applications.

Comparing with practitioners who make subjective investment decisions, computer scientists and statisticians have been using Machine Learning and Data Mining to come up with a data-driven investment strategy. According to modern portfolio theory, one of the most

effective ways to lower the risk and increase the reward is diversifying assets. The goal of diversification is to select stocks in a careful way to make low risk portfolio. The logic behind this strategy is simple—if someone invests in a portfolio whose stocks do not share similar moving trends, he will less likely lose a lot for the return of the portfolio. Through clustering the stocks with some pre-defined similarity measure, one identifies assets with different behaviors. The cluster information helps the investor construct portfolios by picking assets that are different from each other, which is essential in reducing the overall risk. In the finance literature, there exists a large number of research works in stock clustering. For example, Da Costa Jr et al. (2005) applied the hierarchical clustering algorithm to select stocks of major companies in North and South America; Nanda et al. (2010) deployed K-means, fuzzy C-means, and self-organizing map algorithms to cluster stocks and build portfolios in the Indian stock market.

Another unsupervised learning approach is to utilize the network analysis. A network can be constructed based on the co-movement behavior of the stock. Here, co-movement means that return changes over time, the other one changes in the same direction. For example, two stocks may have rather different return levels, yet similar co-movement patterns (Boginski et al., 2006). Stocks with similar co-movement patterns tend to have similar risk levels. Network provides a concise summary of the co-movements among stocks. Using a network, stocks can be represented by vertices, and co-movements between them can be represented by edge pairs. We therefore consider using network to model the co-movement behaviors of stocks, and community detection algorithms can be used to partition the network into different communities/groups (Fortunato, 2010). Similar to the clustering approach, stocks can be picked from different communities, thus achieve the goal of diversification.

In this study, we want to know how different community detection algorithms can help portfolio selection. To provide a strategy of diversification, we propose a pipeline that integrates social network analysis with network community detection algorithms so that we can easily pick stocks with diversified risks based on the historical data of stock returns. Specifically, stock data over a certain transaction period are used to build an unweighted network.

The network is then partitioned into several co-movement groups by using community detection algorithms. Based on these groups, stocks are selected and added to the portfolio to achieve diversity.

We will evaluate our investment strategy using the stock return data from two time periods 2003-2005 and 2007-2009, and apply a wide range of community detection algorithms for community detection and portfolio construction. Our results show that the resulting portfolios demonstrate clear advantages over the benchmark, i.e., the market index such as S&P 500 which represents the return of the total market, for both time periods.

Network-based portfolio construction has also been studied by Koochakzadeh et al. (2011), in which the authors construct a weighted network based on the co-movement of the stock return and partitioned the network by using the Louvain algorithm. Compared with Koochakzadeh et al. (2011)'s work, our analysis is more general in several aspects. First, our analysis investigates how different community detection algorithms perform in portfolio selection and return. Second, unlike Koochakzadeh et al. (2011), our approach provides an automatic pipeline for network construction, portfolio selection, and performance evaluation.

3.2 Network Construction

We measure the performance of a stock using its return. Stock return can be calculated by the relative difference of stock prices over a certain period of time. For example, the return of stock i at time t is calculated by

$$R_{it} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}}, \tag{3.1}$$

where $P_{i,t}$ is the price of stock i at time t. The strength of co-movement between a pair of stocks (e.g., i and stock j) is quantified by their Pearson-Correlation, calculated by

$$c_{ij} = \frac{\sum_{t=1}^{T}(R_{it} - \overline{R}_i)(R_{jt} - \overline{R}_j)}{\sqrt{\sum_{t=1}^{T}(R_{it} - \overline{R}_i)^2}\sqrt{\sum_{t=1}^{T}(R_{jt} - \overline{R}_j)^2}}, \tag{3.2}$$

where $\overline{R}_i = \frac{1}{T}\sum_{t=1}^{T} R_{it}$.

We follow Boginski et al. (2006) to model the stock co-movements by using a network. The simplest representation of a network is an undirected and unweighted graph, in which any pair of vertices are either connected or not connected by an edge. A graph can be denoted by $G(V, A)$, which consists of a vertex set $V = \{v_1, \ldots, v_n\}$ and an adjacency matrix A that models the edge structures. Here, A is a binary matrix of size $n \times n$, denoted by $A = \{a_{ij}\}_{i,j=1,\ldots,n}$, where

$$a_{ij} = \begin{cases} 1, & \text{an edge exists between vertex } i \text{ and vertex } j, \\ 0, & \text{otherwise.} \end{cases}$$

Here, a_{ij} indicates certain relationship, for example, the co-movement relationship between vertex i and vertex j.

In a stock co-movement network, each stock is represented by a vertex, and an edge is added between a pair of vertices if the corresponding stocks co-move. Similar as in Boginski et al. (2006), we determine the co-movement by setting a threshold θ to c_{ij} in Equation (3.2). To be specific, if c_{ij} is larger or equal to θ, we claim that stock i and stock j have a similar co-movement pattern, i.e., the time series of the returns of one stock resemble that of the other. If that is the case, the corresponding vertices are connected by an edge. This θ serves as a parameter which decides the structure of the co-movement network.

3.3 Community Detection

In real world, many networks have a large degree of internal inhomogeneities, as represented by sub-group structures. Specifically, a subset of the vertices may be densely connected with each other but loosely connected to vertices that are not in the subset. Such special pattern is named as a community structure and the subsets of vertices are called communities. Vertices from the same community are expected to share common properties or similar behaviors.

By applying community detection algorithms on the stock co-movement network, we ex-

pect to find communities which contain stocks with similar movement patterns over time. Since different algorithms may lead to different communities and portfolio selections. It is of interest to investigate how different algorithms perform in finding a well-performed portfolio. In this study, we will consider six different state-of-art community detection algorithms. These algorithms can be categorized into four families, including the *modularity optimization* approach (including the Louvain algorithm, the Fast Greedy algorithm, and the Leading Eigenvector algorithm), the *flow optimization* approach (including the InfoMap algorithm), the *spectral clustering* approach, and the *stochastic block model*. In the following, I will briefly overview the main ideas of these network community detection approaches.

- **The Modularity Optimization Approach** Modularity (Newman, 2006b) has been widely used to measure the strength of division of a network into communities. The modularity of a network is a scalar ranging from -1 to 1. It is defined as the difference between the fraction of edges within groups and the expected fraction if the edges are randomly assigned to any pair of vertices. When the network has weighted edges, the modularity can be represented by

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(g_i, g_j).$$

  Here, A_{ij} stands for the edge weight between vertex i and vertex j, $k_i = \sum_j A_{ij}$ is the total weights of all edges connected to vertex i, g_i is the community to which vertex i is assigned, the δ-function is 1 if $g_i = g_j$ and 0 otherwise, and $m = \frac{1}{2} \sum_{ij} A_{ij}$, which is the sum of all edge weights in the graph. In an unweighted network, the weight of each edge is treated as 1.

  One popular algorithm that is based on modularity optimizaton is the *Louvain* algorithm. Louvain (Blondel et al., 2008) is a community detection algorithm that aims to maximize the modularity. It consists of two iterative stages. At the first stage, we start by assuming that each vertex in the network is an individual community. For each vertex i, we calculate the change of modularity if remove vertex i from its own community

and merge it into its neighbor, vertex j's community. The change of modularity can be calculated by

$$\Delta Q = \left[\frac{\Sigma_{in} + 2k_{i,in}}{2m} - \left(\frac{\Sigma_{in} + 2k_{i,in}}{2m}\right)^2\right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right],$$

where Σ_{in} is the sum of the weights of the community to which vertex i is moving to, Σ_{tot} is the sum of the weights of all edges that are connected to vertices in the community to which vertex i is moving to. Here, k_i is the weighted degree of i, $k_{i,in}$ is the sum of the weights of all edges that link i and other vertices in the community to which vertex i is moving to. m is the sum of the weights of all edges in the network. Once ΔQ is calculated for all communities to which vertex i is connected to, i is placed into the community with the greatest modularity increase. If no increase happens for all communities, i stays in its original community. This process is performed repeatedly and sequentially to all vertices until no modularity increase occurs. Once the modularity reaches its local maxima, the first stage ends. At the second stage, communities from the first stage are treated as “big vertices". All edges between vertices within a community are represented by a self-loop. Edges that connect vertices within a community to those in another community are represented by weighted edges between the two big vertices. Once the new network is constructed, we perform the first stage again to merge the new vertices. The iteration stops when no change of modularity happens and a maximum of modularity is achieved.

Besides the Louvain algorithm, another popular modularity based algorithm is the *Greedy* algorithm, proposed by Newman (2004). The Greedy algorithm focuses on iteratively merging two communities if the modularity increases. It involves the following steps. We start by treating each vertex as a single community and inspecting each pair of communities that are connected by at least one edge. We then compute the change of modularity ΔQ if combine them and merge the community pair with the largest ΔQ. We repeat there steps until all vertices are merged into a single big community. At each

step, we record the Q value. We select the partition with the maximal Q value. This algorithm has a faster implementation developed by Clauset et al. (2004), which gives results identical to the original algorithm.

The third algorithm that is based on modularity optimization is the *Leading Eigenvector* algorithm proposed by Newman (2006a). The goal is to maximize the modularity based on spectral decomposition. To explain the idea, we consider the case of two communities. Let $\boldsymbol{s}$ to be an indicator vector for community assignment, with $s_i = 1$ if the vertex i is located in the first community and $s_i = -1$ if in the second community. The modularity function can then be written as

$$Q = \frac{1}{4m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] [s_i s_j + 1] = \frac{1}{4m} \sum_{i,j} D_{ij} s_i s_j = \frac{1}{4m} \boldsymbol{s}^T D \boldsymbol{s}.$$

Here, we let $D_{ij} = A_{ij} - \frac{k_i k_j}{2m}$ and denote $D = \{D_{ij}\}$ and $\boldsymbol{s} = (s_1, \ldots, s_n)$. The vector $\boldsymbol{s}$ can be written as a linear combination of the normalized eigenvectors $\boldsymbol{u}_i$ associated with the matrix D such that $\boldsymbol{s} = \sum_{i=1}^{n} a_i \boldsymbol{u}_i$ and $a_i = \boldsymbol{u}_i^T \boldsymbol{s}$. Using this along with the fact that β_i is the eigenvalue of D corresponding to the eigenvector $\boldsymbol{u}_i$, we get

$$Q = \frac{1}{4m} \sum_{i=1}^{n} (\boldsymbol{u}_i^T \boldsymbol{s})^2 \beta_i.$$

The idea of the Leading Eigenvector method is to look for the largest positive eigenvalue of D, denoted as β_1, then group the vertices according to signs of the elements in $\boldsymbol{u}_1$.

The extension of the algorithm to more than two communities relies on improving modularity by dividing a community g with n_g vertices into two communities. The change of modularity can be calculated by

$$\Delta Q = \frac{1}{4m} \sum_{i,j \in g} \left[D_{ij} - \delta_{ij} \sum_{k \in g} D_{ik} \right] \delta_i \delta_j = \frac{1}{4m} \boldsymbol{s}^T D^{(g)} \boldsymbol{s},$$

where $D^{(g)}$ is an $n_g \times n_g$ matrix with elements $D^{(g)}_{ij} = D_{ij} - \delta_{ij} \sum_{k \in g} D_{ik}$. The algorithm

stops when there are no more positive eigenvalues.

- **The Flow Optimization Approach** We now introduce the general idea of the flow optimization approach. The *InfoMap* algorithm is one algorithm for solving the flow optimization problem. Consider a network with m potential communities, we use a set of labels to summarize the trajectory of an object travelling in the network randomly. This set is called a code. A good code should utilize the fact that the randomly walking object is more likely to stay in a community of the network for a long time than traveling between communities. Rosvall et al. (2009) pointed out that the procedure of finding such code leads to an optimal partition of the network, and this can be achieved by minimizing the following map equation

$$L(M) = H(Q)\sum_{i=1}^{m} q_i + \sum_{i=1}^{m} p^i H(P^i), \tag{3.3}$$

where $L(M)$ is a quality function whose value varies for different partitions M of the network. The goal is to find the minimal L which corresponds to the best partition of the network. In Equation (3.3) q_i is the probability for the object to leave community i, p^i is the proportion of the movements of the object within community i, $H(Q)$ represents the entropy of the code to describe all of the communities and Q is the code set, $H(P^i)$ stands for the entropy of movements made by the object inside community i where P^i describes all movements that happen within community i. To be specific, $H(Q) = -\sum_{i=1}^{m} \frac{q_i}{\sum_{j=1}^{m} q_j} \log\left(\frac{q_i}{\sum_{j=1}^{m} q_j}\right)$, $p^i = \sum_{\beta \in i} p_\beta + q_i$, and $H(P^i) = -\frac{q_i}{q_i + \sum_{\beta \in i} p_\beta} \log\left(\frac{q_i}{q_i + \sum_{\beta \in i} p_\beta}\right) - \sum_{\alpha \in i} \frac{p_\alpha}{q_i + \sum_{\beta \in i} p_\beta} \log\left(\frac{p_\alpha}{q_i + \sum_{\beta \in i} p_\beta}\right)$. Here, p_β is the probability for the object to reach vertex β. A popular way to solve the flow optimization problem is to follow the similar procedure as in the Louvain algorithm. The only difference is that in the Louvain algorithm, the change ΔQ is to increase Q whereas in flow optimization, the change ΔL is to decrease L.

- **The Spectral Clustering** The general spectral clustering technique (Von Luxburg,

2007) makes use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimension reduction before clustering. In network analysis, the similarity matrix corresponds to the adjacency matrix $A = \{A_{ij}\}_{i,j=1,\ldots,n}$. The steps for detecting communities of a network by using the Spectral Clustering method is as follows. Denote the degree of vertex i as $d_i = \sum_{j=1}^{n} A_{ij}$, the degree matrix D as $D = \text{diag}\{d_1, \ldots, d_n\}$, and the Graph Laplacian L as $L = D - A$. We start by finding the K first eigenvectors of L. Let U be the matrix formed by the first K eigenvectors, i.e. $U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K]$. We construct a matrix Y by normalizing each row of matrix U so that each row has norm one. $Y_{ij} = \frac{U_{ij}}{\left[\sum_{j=1}^{n} U_{ij}^2\right]^{\frac{1}{2}}}$. We then treat each row of Y as a point and classify the rows into K classes via the K-means algorithm. We assign vertex i to cluster j if and only if row i of matrix $\{Y_{ij}\}$ was assigned to cluster j. A key step of spectral clustering is to pre-determine the number of communities K. One practical method is called the eigen-gap heuristic. The idea is to choose K such that all eigenvalues $\lambda_1, \ldots, \lambda_K$ are very small, but λ_{K+1} is relatively large. For example, Figure 3.1 plots 10 smallest eigenvalues of L, which shows a gap at 4. In this case, $K = 4$ is an appropriate choice for the number of communities.

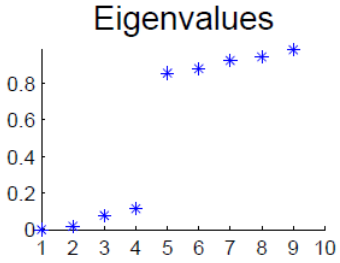

Figure 3.1: The 10 smallest eigenvalues.

- **The Stochastic Block Model (SBM)** Statistical inference offers model-based approaches to tackle the community detection problem. These approaches rely on fitting a statistical model to identify community structures. An example is the stochastic block model (SBM). The stochastic block model (SBM) (Abbe, 2017) is a probabilistic model used to detect communities of a network. Assume that the number of communities K

is fixed, and the communities are indexed by k, $k = 1, \ldots, K$. Define a binary latent random vector $\boldsymbol{Z}_i = [z_{i1}, \ldots, z_{iK}]$, $i = 1, \ldots, n$. We assign the i-th vertex into community g if $z_{ig} = 1$ and $z_{ik} = 0$ for all $k \neq g$. Here, $\boldsymbol{Z}_i$ is assumed to follow a multinomial distribution. $\boldsymbol{Z}_i \sim \text{Multi}(1, \boldsymbol{\alpha})$ with $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]^T$ and $\sum_{k=1}^{K} \alpha_k = 1$. Conditional on the community assignments of each vertex, the edge between vertex i and vertex j can be modeled by $X_{ij} \mid Z_{ik} = 1, Z_{jl} = 1 \sim Ber(\pi_{kl}), \quad k, l = 1, \ldots, K$ where $X_{ij} = A_{ij}$ and A is the adjacency matrix. Denote the posterior probability of point i being from community k as τ_{ik}, called variational parameter. After introducing a set of variational parameters $\boldsymbol{\tau} = \{\tau_{ik}\}_{i=1,\ldots,n;k=1,\ldots,K}$, we can obtain the estimate of $\boldsymbol{\tau}$, $\boldsymbol{\pi}$ and $\boldsymbol{\alpha}$ by using the variational inference:

$$\hat{\tau}_{ik} \propto \hat{\alpha}_k \prod_{i \neq j}^{n} \prod_{l=1}^{K} \left[\hat{\pi}_{kl}^{x_{ij}} (1 - \hat{\pi}_{kl})^{1 - x_{ij}} \right]^{\hat{\tau}_{jl}},$$

$$\hat{\pi}_{kl} = \frac{\sum_{i \neq j}^{n} \hat{\tau}_{ik} \hat{\tau}_{jl} x_{ij}}{\sum_{i \neq j}^{n} \hat{\tau}_{ik} \hat{\tau}_{jl}},$$

$$\hat{\alpha}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{\tau}_{ik}.$$

The community assignment for the i-th vertex can be estimated by

$$\hat{Z}_{ik} = \begin{cases} 1, & k = \text{Argmax}_{k'} \hat{\tau}_{ik'}, \\ 0, & \text{otherwise.} \end{cases}.$$

To determine the number of communities K we use the Integrated Classification Likelihood (ICL) criterion proposed by Biernacki et al. (2000). For a model with K^* communities, denoted by M_{K^*}, the ICL can be calculated by

$$\text{ICL}(M_{K^*}) = \log f(\boldsymbol{x}, \hat{\boldsymbol{z}} \mid M_{K^*}, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\alpha}}) - \frac{1}{2} \frac{K^*(K^* + 1)}{2} \log \left[\frac{n(n-1)}{2} \right] - \frac{K - 1}{2} \log(n).$$

The larger the ICL, the better the M_{K^*} is.

3.4 Stock Selection Using Importance Measures

If a collection of stocks have similar co-movement pattern, it is high likely that they will be classified into the same community. These communities define a partition of the stock market. When creating a portfolio, a basic idea is to select the stocks that behave distinctly in their movement in order to reduce risk. A reasonable idea is to choose a single stock that is the representative of its community and ensure that each stock selected in the final portfolio are different from the others. Thus, a selection criterion is required. For example, the eigenvector centrality of a network (Newman, 2008) provides a measure of the importance of a vertex in social network, and can serve the purpose of selecting a "representative" from a community. The calculation of this centrality measure depends not only on the number of its connection but also on the quality of the connection (i.e. the importance of the vertices which are connected to the vertex of interest). For each community, we assign each vertex a score calculated by the eigenvector centrality score. The vertex with the highest eigenvector centrality score is added into the portfolio.

3.5 A Workflow for Portfolio Construction and Evaluation

Figure 3.2 presents a workflow we adopted to construct a portfolio and evaluate its performance. We first choose a large group of stocks over a training time period and calculate the pairwise correlation of the returns based on Equation (3.2). We then dichotomize the pairwise correlations using the pre-selected threshold value θ to construct the adjacency matrix of an undirected, unweighted network. Based on the network, we apply a network community detection algorithm to group stocks with similar co-movement patterns. For stocks in the same group, we pick the one with the largest eigenvector centrality score and add it to the portfolio. Finally, we assign weights to each stock in the portfolio and evaluate the portfolio's return in a testing period.

Figure 3.2: The proposed workflow to construct portfolios.

3.6 Results of the CRSP Data Analysis

We applied our analytical workflow to the stock return data obtained from the Center for Research in Security Prices (CRSP) database. The CRSP database is maintained by the University of Chicago. It provides raw historical stock market data for academic research. The raw historical stock market data contain the following variables:

1) Daily prices for a total of 29,865 stocks in 11,137 days, saved in a large matrix (11,137 rows $\times$ 29,865 columns). It represents the daily prices for 29,865 stocks over 43 years (1972-2015).

2) The weekly market value of each stock's firm over 43 years.

3) Three weekly indices over 43 years. The first is the value-weighted return index, the second is the equal-weighted return index, and the third is the S&P composite index.

To obtain the two datasets used in this study, several preprocessing steps have been taken on the raw dataset. Details are described as follows:

I. We select two subsets of two distinct time periods. The first time period is from 2003 to 2006 and the second one is from 2007 to 2010. Notice that the 2008 financial crisis occurred in the second time period.

II. Inside each of the two subsets, stocks are chosen based the market value of their corresponding firms. A total of 1000 stocks with the largest firm market values are selected.

III. For the 1000 selected stocks, we calculate the weekly and monthly returns based on their daily prices.

With these preprocessing, the two datasets used in the study are a total of 1000 stocks' weekly and monthly returns for the period of 2003-2006 and 2007-2010. In Figure 3.3, we show the boxplots of the Pearson correlations (A) and two different co-movement patterns (B)-(C). In Figure (A), we observe that the median pairwise correlation of the weekly return for each of the four years (2003-2006) are about 0.25, and the correlations of year 2005 and 2006 are smaller than that of year 2003 and 2004. Figure (B) illustrates an example on a pair of stocks with evident co-movement pattern. It shows that, during the year of 2003, the stock id 70519 fluctuated in a way close to that of stock 59408. Figure (C) describes a situation when two stocks behave differently, which demonstrates that stock 17778 and stock 18163 do not share the same moving pattern in weekly return in the year of 2003.

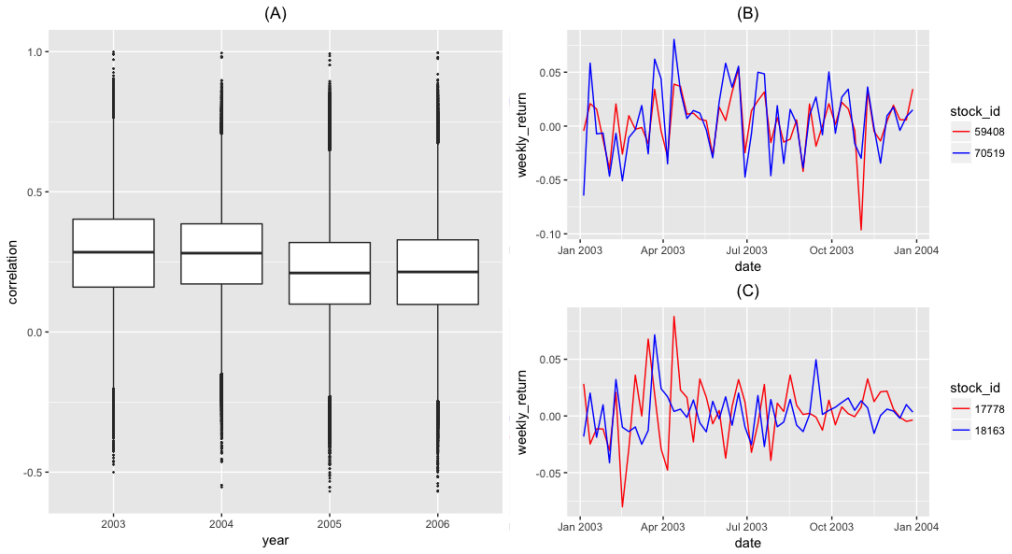

Figure 3.3: The correlation pattern of weekly return for the period 2003-2006. (A) Boxplots of the pairwise correlations of weekly return for each year in 2003-2006. (B) An example of co-movement. (C) An example of non-comovement.

Before presenting the result, we briefly summarize the data analysis procedure. Recall that the goal of this analysis is to evaluate the influence of a variety of community detection algorithms in portfolio construction and return. We analyzed the two datasets following the analytical workflow described before. Take the first dataset (period 2003-2006) as an example,

the procedure is illustrated as follows:

Step 1: Use the weekly return data of year 2003 and the threshold value $\theta = 0.3$ to construct a network.

Step 2: Apply the six aforementioned community detection algorithms (the Louvain method, the Greedy algorithm, the Leading Eigenvector method, the InfoMap method, the Spectral Clustering method, and the Stochastic Block Model) to partition the network obtained in Step 1 into different communities.

Step 3: For each of the community detected, select the stock from the community with the largest eigenvector centrality and add this stock to the portfolio set. Equal weight is assigned to each stock in the portfolio.

Step 4: Assess the performance of the six portfolios by calculating its monthly return of year 2004.

Step 5: Repeat Steps 1-4 for $\theta = 0.5$, and $\theta = 0.7$.

In order to investigate how this investment strategy performs in the following two years, i.e., the data of 2004 and 2005 are used to construct portfolios and the performance is evaluated for 2005 and 2006 respectively. In the following sections, we will demonstrate the results of our analysis for the two time periods.

3.6.1 Results for Period I (Year 2003-2006)

For the weekly return data of each year (2003, 2004, and 2005), we constructed a network using the weekly return data and a pre-specified threshold value θ. We then applied the six aforementioned community detection algorithms on the network to construct portfolios. The performance of the constructed portfolios was evaluated for the following year.

Take the weekly return data of year 2003 as an example. We consider the case $\theta = 0.5$. Two statistics, the modularity (MOD) and the number of communities (n_c), were used to summarize the community detection result. The Area under the curve (AUC) is adopted to measure the performance of the portfolio for the following year (2004). Specifically, the AUC is given as follows: suppose that $p(t)$ is the cumulative return of a portfolio or a market index at a certain time t, the area under the curve (AUC) for $p(t)$ is: AUC $= \int_t p(t)dt$. If a portfolio has constantly higher cumulative return than that of a market index at all time points, its AUC will be larger than that of the market index.

Table 3.1 summarizes the community detection results and portfolio performance. From Table 3.1, we can see that for each of the three years, the portfolios constructed by setting the threshold value $\theta = 0.5$ perform better in terms of AUC, compared with that of $\theta = 0.3$ and $\theta = 0.7$. The modularity measure does not seem to be related to how the portfolio performs. For example, when $\theta = 0.7$ and the portfolios are assessed for year 2006, although the spectral clustering gives 0 modularity, its portfolio achieve 14.0659 in AUC. In contrast, while the Louvain method leads to the highest modularity, it does not produce a portfolio with a larger AUC. Additionally, we observe that larger θ value ($\theta = 0.7$) generates sparser network, on which the Louvain Method (LV), the Fast-Greedy Method (FG), and the Leading Eigenvector Method (LE) produce a large number of communities. An exception is the Stochastic Block Model (SBM) method, which results in large number of communities for small theta values ($\theta = 0.3$). Another exception is the Spectral Clustering (SC) method, which detects the largest total number of communities at $\theta = 0.5$. Overall, all six portfolios are perform better than S&P 500 in terms of the AUC values for each year and each θ setup. In Figure 3.4, we plot

the cumulative returns of six portfolios resulted from the six community detection algorithms for 2004-2006 by letting $\theta = 0.5$. Figure 3.4 shows that with the exceptions of SBM during January 2005 and May 2005, all community detection methods result in cumulative portfolio returns superior than that of the S&P 500 index. The SBM algorithm results in portfolios with lower returns than all other five algorithms during 2004-2005, but higher returns than these algorithms during 2006.

Table 3.1: Summary statistics for period I (2004-2006): modularity of the network communities (MOD), area under the curve of cumulative return (AUC), the total number of communities for a network (n_c), the Louvain Method (LV), the Fast-Greedy Method (FG), the Leading Eigenvector Method (LE), the InfoMap Method (IM), the Spectral Clustering (SC), the Stochastic Block Model (SBM), the S&P 500 Index (SP500)

		2004			2005			2006		
Threshold	Index	MOD	AUC	n_c	MOD	AUC	n_c	MOD	AUC	n_c
	LV	0.0738	13.1121	5	0.1075	12.3012	3	0.1505	11.4390	3
	FG	0.0660	13.0465	4	0.0920	12.7438	3	0.1147	14.4935	4
	LE	0.0677	13.1938	4	0.0973	12.7251	3	0.1425	11.4390	3
$\theta = 0.3$	IM	0	14.2391	2	0	10.6871	1	0	12.8415	1
	SC	0	14.2391	2	0	10.6871	1	0	12.8415	1
	SBM	0.0046	12.6976	33	0.0117	12.1188	33	0.0199	13.5091	38
	SP500	–	12.1929	–	–	11.9439	–	–	12.6050	–
	LV	0.1874	14.8680	43	0.3023	13.6041	22	0.4050	13.2305	57
	FG	0.1599	14.6886	46	0.2762	14.0569	21	0.3820	13.2660	64
	LE	0.1603	14.6450	41	0.2678	13.7903	18	0.3805	13.2974	52
$\theta = 0.5$	IM	0.0709	14.3465	57	0.0882	13.6399	27	0.3832	13.2444	102
	SC	0.0002	14.6748	39	0	14.1223	15	0.0006	13.2992	48
	SBM	0.0546	12.8000	24	0.1032	12.9215	27	0.2065	13.9671	17
	SP500	–	12.1929	–	–	11.9439	–	–	12.6050	–
	LV	0.5933	13.4105	592	0.7515	12.5780	623	0.4374	13.1328	755
	FG	0.5405	13.3838	595	0.7503	12.5642	623	0.3664	13.1158	757
	LE	0.5547	13.4042	592	0.7412	12.5622	623	0.4368	13.1058	757
$\theta = 0.7$	IM	0.5803	13.3730	610	0.7324	12.5762	645	0.4252	13.1300	770
	SC	0	12.2090	4	0	11.4473	1	0	14.0659	2
	SBM	0.4020	12.3096	10	0.5161	14.5242	6	0.2901	13.1507	6
	SP500	–	12.1929	–	–	11.9439	–	–	12.6050	–

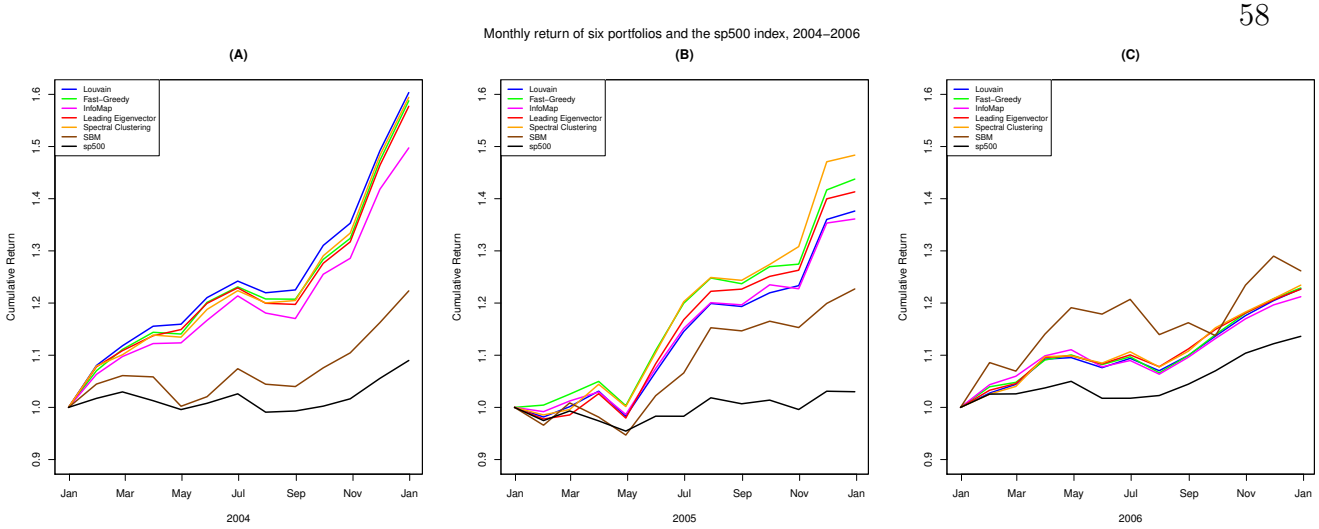

Figure 3.4: Plot for the portfolios' performances in year 2004-2006, $\theta = 0.5$. (A) The 2004 monthly return of six portfolios constructed from the weekly return of 2003 when $\theta = 0.5$, compared with the monthly return of S&P 500 index, (B) The 2005 monthly return of six portfolios constructed from the weekly return of 2004 when $\theta = 0.5$, compared with the monthly return of S&P 500 index, (C) The 2006 monthly return of six portfolios constructed from the weekly return of 2005 when $\theta = 0.5$, compared with the monthly return of S&P 500 index,

3.6.2 Results for Period II (Year 2007-2010)

Table 3.2 shows the summary statistics for the community detection and portfolio performance for the data during 2008-2010, a period when the financial crisis of 2007-2008 happens. From Table 3.2, we notice that all portfolios and the S&P 500 index result in much lower cumulative returns than the 2004-2006 period in terms of AUCs. For year 2009 and 2010, the portfolios associated with $\theta = 0.5$ and $\theta = 0.7$ have better performance than those of $\theta = 0.3$. When $\theta = 0.5$, the portfolios created by all six community detection algorithms outperform the S&P 500 index in terms of AUC from 2008 to 2010.

Similar to what has been observed from Table 3.1, the modularity measure does not appear to be relevant to how the portfolios perform. Larger θ values still lead to sparser networks, which leads to an immense amount of communities detected by LV, FG, and LE. For SBM and SC, the relationship between the number of communities and the threshold value θ is not clear.

We plot the performance of the portfolios constructed by using the six community detection methods in Figure 3.5. From Figure 3.5, we see that, all cumulative monthly returns show a downward trend. This can be explained by the stock market drop occurred over 2007-2008. The situation starts getting improved in 2009 when all six portfolios and the S&P 500 index begin moving upwards since March, 2009. In 2010 the cumulative monthly return of the six portfolios still enjoys advantage over the S&P 500 index. All six portfolios perform similar during 2010.

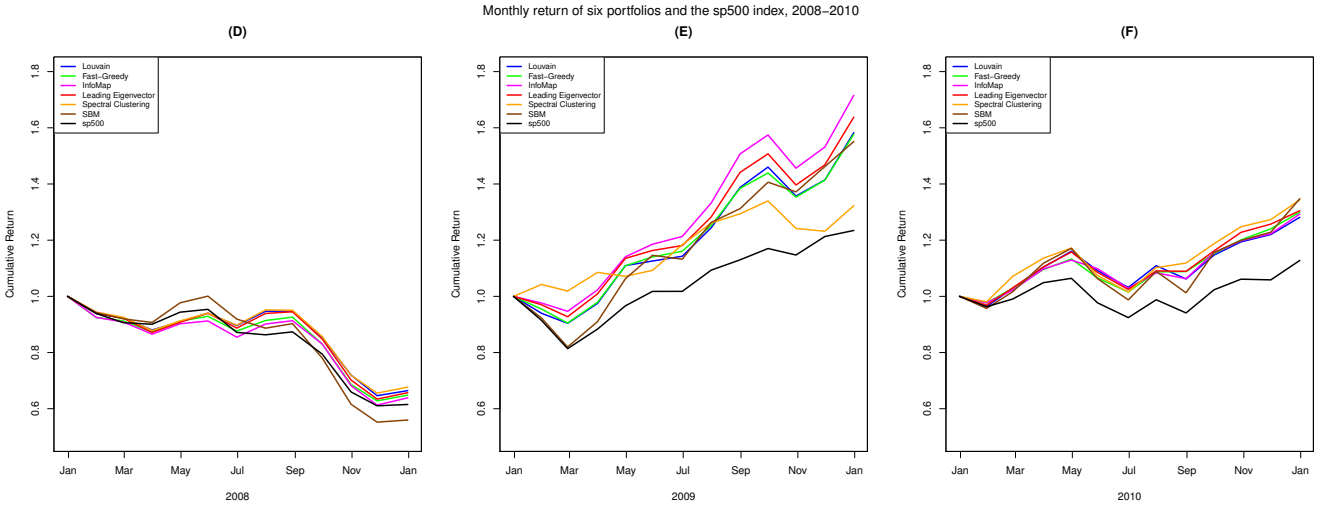

Figure 3.5: Plot for the portfolios' performances in year 2008-2010, $\theta = 0.5$. (D) The 2008 monthly return of six portfolios constructed from the weekly return of 2007 when $\theta = 0.5$, compared with the monthly return of S&P 500 index, (E) The 2009 monthly return of six portfolios constructed from the weekly return of 2008 when $\theta = 0.5$, compared with the monthly return of S&P 500 index, (F) The 2010 monthly return of six portfolios constructed from the weekly return of 2009 when $\theta = 0.5$, compared with the monthly return of S&P 500 index,

3.6.3 Conclusion

The results demonstrated in Sections 3.6.1–3.6.2 show several patterns. First, the threshold value θ has a large impact on the performances of the portfolios. The sparsity of the network changes with the value of θ. Higher θ usually results in sparser network. For these two data sets, letting $\theta = 0.5$ seems to result in better portfolio returns. When $\theta = 0.5$, the first four algorithms (LV, FG, LE and IM) are likely to generate portfolios performing similarly. With

Table 3.2: Summary statistics for period II (2008-2010): Modularity of the network communities (MOD), The Area under the curve of monthly return (AUC), the total number of communities for a network (n_c), the Louvain Method (LV), the Fast-Greedy Method (FG), the Leading Eigenvector Method (LE), the InfoMap Method (IM), Spectral Clustering (SC), the Stochastic Block Model (SBM), the S&P 500 Index (SP500)

		2008			2009			2010		
Threshold	Index	MOD	AUC	n_c	MOD	AUC	n_c	MOD	AUC	n_c
	LV	0.0703	9.8824	3	0.0402	12.4817	3	0.0229	12.4344	4
	FG	0.6516	12.5782	3	0.0399	11.8808	2	0.0229	12.1191	3
	LE	0.0650	10.2761	2	0.0393	11.2350	2	0.0225	11.7007	2
$\theta = 0.3$	IM	0	11.0194	1	0	11.2104	1	0	11.5062	1
	SC	0	11.0428	2	0	11.2104	1	0	11.5062	1
	SBM	0.0019	10.5540	39	-0.0024	14.8462	38	-0.0054	12.8866	34
	SP500	–	10.1237	–	–	12.4841	–	–	12.1038	–
	LV	0.2088	10.4315	20	0.1176	14.3514	8	0.0716	13.2524	11
	FG	0.1670	10.2358	22	0.1078	14.3795	9	0.0556	13.2345	9
	LE	0.1768	10.3658	19	0.1082	14.8001	7	0.0683	13.3503	9
$\theta = 0.5$	IM	0.0002	10.1282	21	0	15.2444	6	0.0001	13.2227	11
	SC	0	10.4699	17	0	14.0235	4	0	13.5499	5
	SBM	0.0421	10.1837	26	0.0073	14.0847	50	0.0005	13.1717	41
	SP500	–	10.1237	–	–	12.4841	–	–	12.1038	–
	LV	0.6412	10.4535	528	0.3116	15.3904	136	0.2120	12.2362	181
	FG	0.6373	10.4689	529	0.2899	15.7839	137	0.1678	12.1842	184
	LE	0.6141	10.4655	526	0.2706	15.5647	136	0.1994	12.1909	183
$\theta = 0.7$	IM	0.6139	10.4558	556	0.2033	15.4733	157	0.0400	12.2185	199
	SC	0.0229	10.3109	11	0	14.3433	4	0.0002	12.9826	6
	SBM	0.4339	9.8026	8	0.0993	14.1196	28	0.0494	12.7590	24
	SP500	–	10.1237	–	–	12.4841	–	–	12.1038	–

$\theta = 0.5$, the portfolio derived by using the SBM performs quite differently from the other five algorithms during 2004-2006, and the portfolio resulting from SC performs similarly compared with the first four (LV, FG, LE and IM) except for 2009.

Moreover, the portfolios generated by most community detection algorithms usually show advantages over the S&P 500 if the threshold value θ is chosen appropriately. Apart from year 2008, the portfolios created by all six community detection algorithms always outperform the S&P 500 market index. During the 2008 financial crisis period, the cumulative returns of the six portfolios are still comparable with, if not better than the S&P 500 index.

For the SC and SBM method, the number of communities K needs to be decided in order to do further partition on the network, whereas for the cases of LV, FG, LE and IM, K is not needed. When applying SC, K is chosen by observing if there exist gaps for the eigenvalues. This process can be somewhat subjective. The K in SBM has been chose by maximizing ICL. It usually takes a long searching time for SMB to detect a K to achieve the largest ICL, and the portfolio suggested by SBM can have worse return than that proposed by the other five approaches.

One might wonder why we assign equal weight when allocating asset to a portfolio. To my knowledge, assigning equal weights is a crucial strategy to achieve well-performed portfolio. Imposing different weights on each stock in the portfolio set can be viewed as buying more shares in a particular stock than others, which is equivalent to buying additional stocks which are acting exactly the same. Therefore, the effect of risk diversification maybe hindered. In section 3.7, we will perform a numerical study to demonstrate the effect of equal versus unequal weights.

3.7 A Study of Some Related Issues

In this section, we perform several extra data analysis tasks to study some related issues, including the influence of different model choices, the interpretation of stock communities, and parameter tuning. While doing the analysis, we consider the influence of an individual

factor and keep other factors and the community detection algorithm fixed. These analyses provide insights into addressing the following questions: 1. which correlation measure we shall use in the construction of co-movement network; 2. whether we shall choose equal weights vs. unequal/optimal weights in the portfolio construction; 3. whether there is potential association between the resulting stock communities and which industry the stocks belong to; and 4. how to tune the correlation threshold θ in the construction of co-movement network.

3.7.1 Correlation Measures

At the first step of the analytical workflow, one needs to calculate the pairwise correlation for each pair of the stock returns in order to construct the co-movement network. We have focused on using the Pearson Correlation (3.2) to quantify the co-movement between two stock returns. In addition to Pearson Correlation, there are some other correlation options such as the Copula correlation and the Kendal's correlation. Here, we will provide an overview of these two correlation measures and a demonstration of the performance of the resulting portfolios if using these correlations.

The *Kendall's correlation* coefficient measures the association between two vectors (X, Y) by using their relative ranks. For two pairs of (X, Y), (x_i, y_i) and (x_j, y_j), when $\text{sign}(x_i - x_j)\text{sign}(y_i - y_j) > 0$, concordance happens. Otherwise, when $\text{sign}(x_i - x_j)\text{sign}(y_i - y_j) < 0$, discordance occurs. The formula of the Kendall's correlation $Cor_K$is

$$Cor_K = \frac{\text{total concordant pairs} - \text{total discordant pairs}}{n(n-1)/2} = \frac{2}{n(n-1)} \sum_{i<j}^{n} \text{sign}(x_i - x_j)\text{sign}(y_i - y_j).$$

The *copula correlation* (Ding and Li, 2013) can measure all deterministic relationships between two continuous variable, not just linear relationship that Pearson's correlation measures. The Sklar's theorem (Sklar, 1959) declares that for two random variables X and Y, the jointed CDF of X and Y, $F_{X,Y}(x, y)$, can be expressed as a function of their marginal CDFs, $F_X(x)$ and $F_Y(y)$. The function is called copula and denoted by C, satisfying $F_{X,Y}(x, y) =$

$C[F_X(x), F_Y(y)]$. The copula correlation Cor_c is defined as

$$Cor_c = \frac{1}{2} \int \int_{[0,1]\times[0,1]} |c(u,v) - 1| du dv,$$

where $c(u,v) = \frac{\partial^2}{\partial u \partial v} C(u,v)$.

A comparison of different correlation measures To see how the choice of correlation measures influences the performance of the generated portfolio, we construct three portfolios based on the 2003 weekly return data and the 2007 weekly return data (the data in the financial crisis period) by using the three correlation measures and letting $\theta = 0.5$. Similar to what we've done before, the performance of the portfolios is assessed by using the cumulative monthly return data in the following year. Figure 3.6 shows the cumulative monthly returns of the three portfolios and the S&P 500 index in 2004. We notice that in this situation, the portfolios based on Kendall's correlation and the Copula correlation behave similarly but no better than that based on the Pearson correlation. Figure 3.7 shows the cumulative monthly returns of the three portfolios and the S&P 500 index in 2008. We observe that during the financial crisis period (2008), no evident difference exists for all three portfolios' performance in terms of the cumulative monthly returns. To sum up, there is no clear sign of improvement in portfolio returns by considering the Kendall's correlation and the Copula correlation as compared with the Pearson correlation.

3.7.2 Weight Assignments

In our analysis, to construct a portfolio, equal weights have been assigned to each stock in the portfolio. In general, the weights can be any positive numbers as long as they sum up to 1. Here, we briefly discuss some other weighting approaches and compare their performance with the equal weighting (EW) strategy.

Mean-Variance Portfolio Optimization (MVPO) The Mean-Variance portfolio optimization procedure (Markowitz, 1952), has been widely used in modern portfolio analysis. It

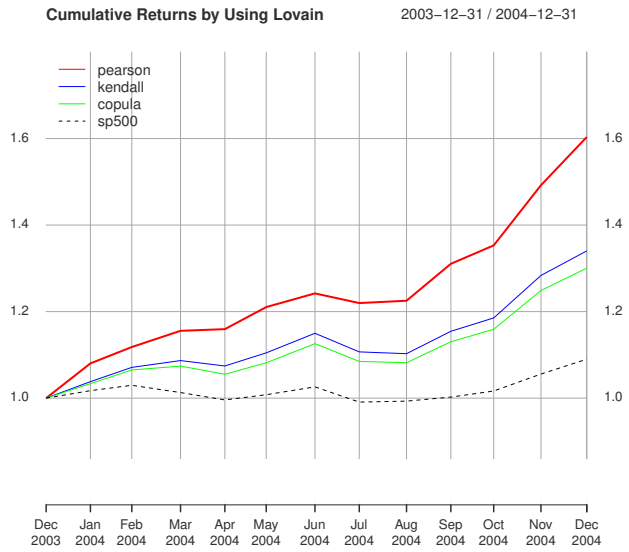

Figure 3.6: The performance of the constructed portfolios in 2004 by using three correlation measures with $\theta = 0.5$. The community detection algorithm used is the Louvain algorithm. Results are compared with the performance of the S&P 500 index.

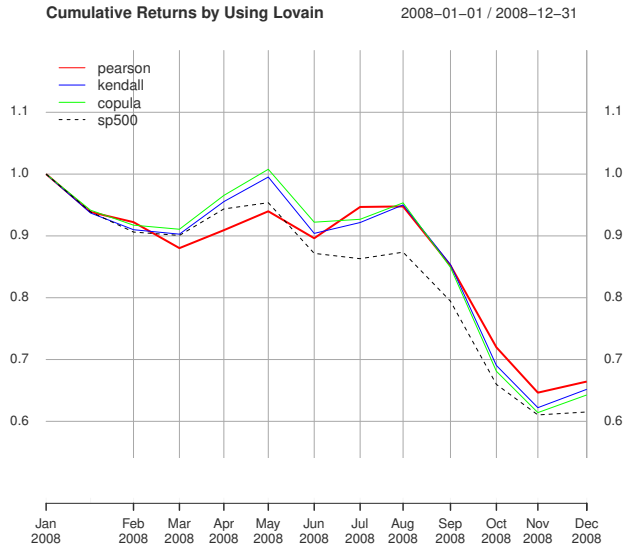

Figure 3.7: The performance of the constructed portfolios in 2008 by using three correlation measures with $\theta = 0.5$. The community detection algorithm used is the Louvain algorithm. Results are compared with the performance of the S&P 500 index.

provides a standard way to find weights for a portfolio. Suppose that we have k assets, and the return during a specific period can be represented as a k-dimensional random vector $\boldsymbol{R}$, and $\boldsymbol{R} = (R_1, \ldots, R_k)^T$. Denote the mean and the covariance matrix of $\boldsymbol{R}$ as $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively. We assign each stock a weight w_i, $i = 1, \ldots, k$, so that $\sum_{i=1}^{k} w_i = 1$. These weights can be written as a k-dimensional vector $\boldsymbol{w} = (w_1, \ldots, w_k)^T$. In matrix form, the portfolio return $\sum_{i=1}^{k} w_i R_i$ can be written as $\boldsymbol{w}^T\boldsymbol{R}$. Its mean is $\mathrm{E}[\boldsymbol{w}^T\boldsymbol{R}] = \boldsymbol{w}^T\boldsymbol{\mu}$ and its variance is $\mathrm{Cov}(\boldsymbol{w}^T\boldsymbol{R}) = \boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}$. Under this setup, the optimal weights are found by maximizing the following objective function

$$Q(\boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{\mu} - \lambda \cdot \boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w}.$$

where $\lambda \geq 0$ is the Arrow-Pratt risk aversion index (Pratt, 2013), which measures the trade-off between risk and return.

Optimization of Portfolio Weights influenced by Market Index (OPWMI) Azizah et al. (2017) come up with a portfolio weighting strategy that extends the Mean-Variance procedure by regressing the return of each stock on a standard market index return. In particular, denote the return for stock i at a certain time period by R_i, we model R_i by a simple linear regression and regress R_i on the market index R_m at the same time period, i.e.,

$$R_i = \alpha_i + \beta_i R_m + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma_i^2)$. The least-squared estimators for α_i, β_i and σ_i^2 are denoted by $\hat{\alpha}_i$, $\hat{\beta}_i$ and $\hat{\sigma}_i^2$, respectively. The estimated mean and variance of R_m is denoted by $\overline{R}_m$ and $\hat{\sigma}_m^2$, respectively. Same as in the Mean-Variance procedure, let w_i be the weight assigned to R_i, the portfolio return, R_p for a certain evaluation period can be calculated by

$$R_p = \sum_{i=1}^{k} w_i R_i = \boldsymbol{w}^T\boldsymbol{R}.$$

The expectation of R_p is $E[R_p] = \boldsymbol{w}^T E[\boldsymbol{\alpha} + \boldsymbol{\beta} R_m]$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T$ and $\boldsymbol{\beta} =$

$(\beta_1, \ldots, \beta_n)^T$. This quantity can be estimated by $\boldsymbol{w}^T(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}}\overline{R}_m)$. The covariance of R_p, $Cov(R_p)$ is

$$Cov(R_p) = \boldsymbol{w}^T Cov(\boldsymbol{\alpha} + \boldsymbol{\beta}R_m + \boldsymbol{\epsilon})\boldsymbol{w} = \boldsymbol{w}^T(\boldsymbol{\beta}\boldsymbol{\beta}^T\sigma_m^2 + \boldsymbol{\Sigma})\boldsymbol{w}$$

where $\boldsymbol{\Sigma} = diag(\sigma_1^2, \ldots, \sigma_k^2)$. The estimator for the covariance matrix is $\boldsymbol{w}^T(\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^T\hat{\sigma}_m^2 + \hat{\boldsymbol{\Sigma}})\boldsymbol{w}$. With the above results, the weight seeking problem can be converted into an optimization problem:

$$\boldsymbol{w} = \arg_{\boldsymbol{w}}\max(\boldsymbol{w}^T(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}}\overline{R}_m) - \lambda \cdot \boldsymbol{w}^T(\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^T\hat{\sigma}_m^2 + \hat{\boldsymbol{\Sigma}})\boldsymbol{w}),$$

subject to $\boldsymbol{w}^T\mathbf{1} = 1$ and $0 < w_i < 1$ for all $i = 1, \ldots, k$. Here the parameter λ is a tuning parameter that controls the trade-off between return and risk. To determine λ, Grinold and Kahn (2000) have suggested a formula,

$$\lambda = \frac{\mu_m}{2\sigma_m^2}, \tag{3.4}$$

where μ_m and σ_m^2 are the mean and variance of the market index returns. Details of the derivation can be found in Appendix C.

To exam the effect of this weighting strategy on the resulting portfolio, we assigned the EW weights and OPWMI weights separately to the stocks selected based on the weekly return of 2003, 2004, and 2005 by using the Pearson correlation with $\theta = 0.5$ and the Louvain method. The performance of the portfolios is evaluated by using the following years' monthly returns and compared with the monthly return of the S&P 500 index.

Figure 3.8 shows the cumulative monthly returns of the OPWMI-based portfolio, EW-based portfolio and the S&P 500 index over 2004, 2005, and 2006, respectively. From Figure 3.8 we see that the performance of OPWMI-based portfolio is superior to that of EW during 2005 and 2006. In 2004, however, the EW-based portfolio does a better job.

Figure 3.9 shows the cumulative monthly returns of the OPWMI-based portfolio, the EW-based portfolio, together with that of the S&P 500 index for the period between 2008 and

2010, respectively. We observe that the OPWMI-based portfolio outperforms the EW-based one over the 2008 financial crisis period. In 2010, the EW-based portfolio starts to outperform the OPWMI-based portfolio since April. It is worth noting that the OPWMI method fails to find the optimal weights based on the 2008 stock return data due to numerical issues induced by the irregular behavior of the stock returns during the financial crisis period. In contrast, the EW strategy does not involves solving an optimization problem hence is free of such a problem.

Overall, we can see that the performance of OPWMI and EW varies at different years. There is no clear evidence that which one is better than the other. However, both of them tend to outperform the market index. Hence, both can be sensible options when constructing a portfolio.

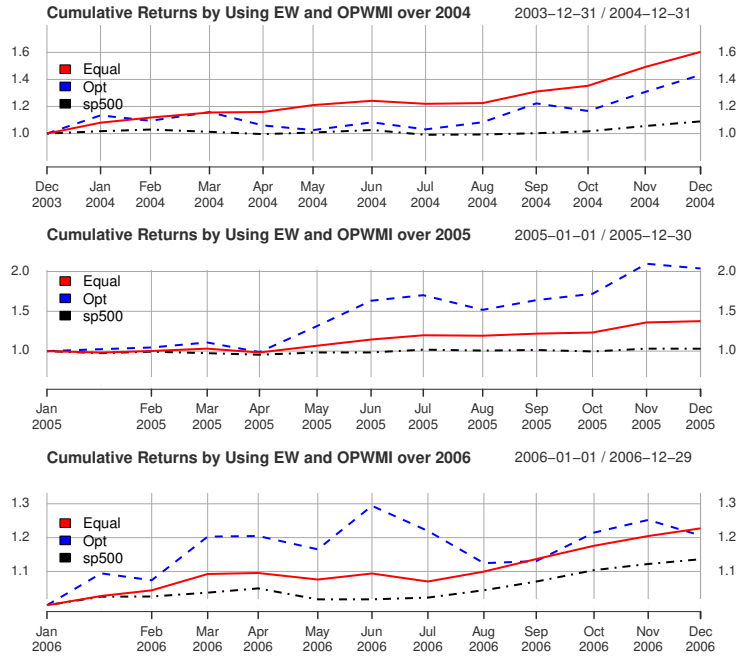

Figure 3.8: The performance of the OPWMI-based portfolio and the EW-based portfolios in 2004-2006 by the Pearson correlation with $\theta = 0.5$ and the Louvain algorithm, compared with the performance of the S&P 500 index.

The Value Weight portfolio The market-value weight (VW) is another alternative weighting approach in portfolio construction. For stock i, denote the market capitalization of its company as E_i. The VW approach relies on setting the weight for stock i to be $\frac{E_i}{\sum_{i=1}^{k} E_i}$. Pae

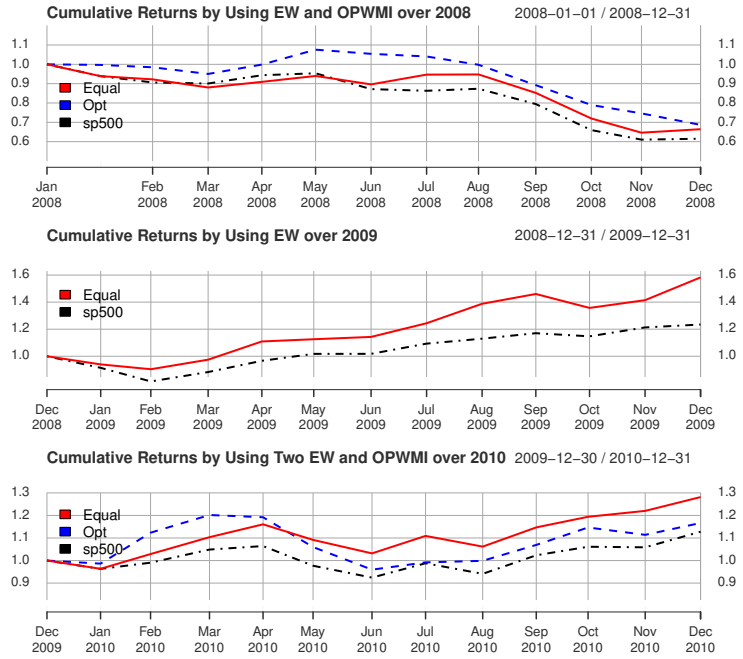

Figure 3.9: The performance of the OPWMI-based portfolio and the EW-based portfolios in 2008-2010 by using the Pearson correlation with $\theta = 0.5$ and the Louvain algorithm, along with the performance of the S&P 500 index.

and Sabbaghi (2015) argue that a EW-based portfolio usually results in higher return than a VW-based portfolio. A sensible explanation is that, by using value-weight, the performance of a portfolio tends to be driven by the dominating stocks (i.e., big companies with large market values), which diminishes the effect of risk diversification. For this reason, we do not recommend to use value-weighted approach.

3.7.3 Association Between Community Assignment and the Industrial Sector

One may be interested in knowing whether there is any association between stock communities and their industry segmentation. We investigated this issue by comparing the community detection outputs with the gsector index of the stocks. The gsector is the indicator that categorizes the stock market into major industrial sectors, in accordance with the Global Industry Classification Standard (GICS).

For two stocks sharing the same gsector number, it is reasonable to conjecture that they have similar movement so that they are likely to be found in the same community by the

community detection algorithms. To verify this conjecture, we compared the community detection outputs of 2003's stock return data with the stocks' gsectors. We listed a two-way table that shows the distribution of stocks into communities (using Louvain with $\theta = 0.5$) in rows and gsector in columns. Due to the large amount of missing-values in gsector indices, we did not see a clear association between gsector and the community assignment.

3.7.4 Tuning of the Thresholding Parameter θ

Table 3.3: Performance of portfolio made by using the 2004-2006 and 2008-2010 weekly return data, with the Louvain method and three different threshold values of θ.

		2003	2004	2005	2007	2008	2009
$\theta = 0.3$	Network density	0.4657	0.4530	0.2894	0.5663	0.8132	0.8235
	Relative portfolio size	0.5 %	0.3 %	0.3 %	0.3 %	0.3 %	0.4 %
	AUC	13.1121	12.3012	11.4390	9.8824	12.4817	12.4344
$\theta = 0.5$	Network density	0.1017	0.0722	0.0350	0.1255	0.4133	0.4582
	Relative portfolio size	4.3 %	2.2 %	5.7 %	2 %	0.8 %	1.1 %
	AUC	**14.8680**	**13.6041**	**13.2305**	10.4315	14.3514	**13.2524**
$\theta = 0.7$	Network density	0.0041	0.0023	0.0022	0.0042	0.0559	0.0723
	Relative portfolio size	59.2 %	62.3 %	75.5 %	52.8 %	13.6 %	18.1 %
	AUC	13.4105	12.5780	13.1328	**10.4535**	**15.3904**	12.2362

As mentioned in the previous sections, the thresholding parameter θ plays an important role in portfolio construction as it controls the network density and affects the community detection results. While there is no gold standard on the choice of θ. It is possible to provide some guidelines by analyzing the properties of the networks and the performance of the resulting portfolios. To further investigate the effect of θ on the network/community properties and portfolio performance, we listed three statistics in Table 3.3. These statistics are based on applying the Louvain algorithm on networks constructed under different θ's for 2003-2005 and 2007-2009. The three statistics are (1) network density, the density of the network based on Pearson correlation and the θ threshold; (2) relative portfolio size, the

number of stocks in the portfolio divided by the total number of stocks; and (3) AUC, the area under the cumulative portfolio return curve for the following year. The highest AUC is highlighted by bold font.

From Table 3.3, we observe that a network that produces portfolios with good performance satisfy at least one of the following two conditions: (1) the network density is appropriately low, ranging from 0.01 to about 0.1, and (2) the relative size of the portfolio varies between 1% to 10 %. For the weekly return data of 2003, 2004, and 2005, the best performing portfolios satisfy both conditions. For the weekly return data of 2008 and 2009, the best performing portfolios satisfy condition (1) and condition (2), respectively. For the weekly return data of 2007, the best performing portfolio in 2008 does not meet either of the condition (1) or (2). The reason might be the abnormal behavior of the whole stock market during the financial crisis period. In practice, we can try different θ values to obtain a network that satisfies at least one of the two conditions. This provide a practical guideline for determining θ.

Chapter 4 Conclusion

In this thesis, I investigated two types of high-dimensional data—functional data which have infinite dimensionality and network data which have special topological structrue.

For the functional data analysis, I have focused on combing functional regression with basis expansion using compactly supported basis and Bayesian variable selection to develop a novel Bayesian regression method for region selection and estimation. To encourage continuous shrinkage of nearby regions, we adopt an Ising hyper-prior to take into account the neighboring structure. Our method shows the advantage of detecting zero and non-zero regions compared to traditional functional regression. The results are comparable with existing frequentist approaches on detecting important regions.

For the network data analysis, I adopt networks to model the stock co-movements and come up with a portfolio selection strategy. In particular, we first apply network community detection algorithms to effectively diversify our assets, then choose the most "important" stock from each cluster to construct the portfolio. This network-based investment strategy provides portfolios with constantly higher returns than the market index. In our data analysis, we also provided a comparison of several popular network community detection algorithms and evaluated their performance over two different financial periods.

Appendix A

Derivation for (2.13), (2.14), and (2.18)

From the likelihood function in (2.11) and priors in (2.8) and (2.10), we have

$$\begin{aligned} p(\boldsymbol{b}, \sigma^2 \mid \boldsymbol{\gamma}, \boldsymbol{y}) &\propto p(\boldsymbol{y} \mid \boldsymbol{b}_\gamma, \boldsymbol{\gamma}, \sigma^2) p(\boldsymbol{b}_\gamma \mid \boldsymbol{\gamma}, \sigma^2) p(\sigma^2) \\ &= \left|2\pi\sigma^2 I\right|^{-\frac{1}{2}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{C}_\gamma \boldsymbol{b}_\gamma)^T (\boldsymbol{y} - \boldsymbol{C}_\gamma \boldsymbol{b}_\gamma)\Big\} \\ &\cdot \left|2\pi\sigma^2 \Sigma_\gamma\right|^{-\frac{1}{2}} \exp\Big\{ -\frac{1}{2}\boldsymbol{b}_\gamma^T \Sigma_\gamma^{-1} \boldsymbol{b}\Big\} p(\sigma^2). \end{aligned} \tag{4.1}$$

Based on (4.2), the posterior distribution of $\boldsymbol{b}$ can be written as

$$\begin{aligned} p(\boldsymbol{b} \mid \sigma^2, \boldsymbol{\gamma}, \boldsymbol{y}) &\propto p(\boldsymbol{b}, \sigma^2 \mid \boldsymbol{\gamma}, \boldsymbol{y}) \\ &\propto (2\pi\sigma^2)^{-\frac{n}{2}} \left|2\pi\sigma^2 \Sigma_\gamma\right|^{-\frac{1}{2}} \left|2\pi K^{-1}\right|^{-\frac{1}{2}} \left|2\pi K^{-1}\right|^{\frac{1}{2}} \\ &\cdot \exp\Big\{ -\frac{1}{2}\left[\boldsymbol{b}_\gamma^T (\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1})\sigma^{-2} \boldsymbol{b}_\gamma - 2\boldsymbol{b}_\gamma^T \boldsymbol{C}_\gamma^T \sigma^{-2} \boldsymbol{y} + M^T K^{-1} M\right]\Big\} \\ &\cdot \exp\Big\{ -\frac{1}{2} M^T K^{-1} M\Big\} \exp\Big\{ -\frac{1}{2}\boldsymbol{y}^T \boldsymbol{y}\Big\} p(\sigma^2). \end{aligned} \tag{4.2}$$

Therefore, posterior distribution of $\boldsymbol{b}$ is

$$\boldsymbol{b} \mid \sigma^2, \boldsymbol{\gamma}, \boldsymbol{y} \sim N(K^{-1}M, K^{-1}),$$

where $K = (\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1})\sigma^{-2}$ and $M = \boldsymbol{C}_\gamma^T \sigma^{-2} \boldsymbol{y}$. After integrating out $\boldsymbol{b}$ in (4.2), the distribution of $\sigma^2 \mid \boldsymbol{\gamma}, \boldsymbol{y}$ can be derived, i.e.,

$$\begin{aligned} p(\sigma^2 \mid \boldsymbol{\gamma}, \boldsymbol{y}) &\propto \int_{\boldsymbol{b}} p(\boldsymbol{b}, \sigma^2 \mid \boldsymbol{\gamma}, \boldsymbol{y}) d\boldsymbol{b} \\ &\propto (2\pi)^{-\frac{n}{2}} (\sigma^2)^{-\frac{n}{2}} \left|2\pi\sigma^2 \Sigma_\gamma\right|^{-\frac{1}{2}} \left|2\pi\sigma^2 (C_\gamma^T C_\gamma + \Sigma_\gamma^{-1})^{-1}\right|^{\frac{1}{2}} \\ &\cdot \exp\Big\{ -\frac{1}{2\sigma^2} [\boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{C}_\gamma (\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1})^{-1} \boldsymbol{C}_\gamma^T \boldsymbol{y}] \Big\} \sigma^{-2} \\ &\propto (\sigma^2)^{-(\frac{n}{2}+1)} \exp\Big\{ -\frac{1}{2\sigma^2} [\boldsymbol{y}^T (I - \boldsymbol{C}_\gamma (\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1})^{-1} \boldsymbol{C}^T) \boldsymbol{y}] \Big\} \\ &\propto (\sigma^2)^{-(\frac{n}{2}+1)} \exp\Big\{ -\frac{1}{\sigma^2} (\frac{1}{2} \boldsymbol{y}^T L_\gamma \boldsymbol{y}) \Big\}, \end{aligned} \tag{4.3}$$

where L_γ is defined as $I - \boldsymbol{C}_\gamma (\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1})^{-1} \boldsymbol{C}^T$. Therefore, we have

$$\sigma^2 \mid \boldsymbol{\gamma}, \boldsymbol{y} \sim \text{Inv-Gamma}\big(\frac{n}{2}, \frac{1}{2} \boldsymbol{y}^T L_\gamma \boldsymbol{y}\big).$$

To find (2.18), we need find $p(\boldsymbol{y} \mid \boldsymbol{\gamma})$, which takes the form

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{\gamma}) &= \int_{\sigma^2} \int_{\boldsymbol{b}} p(\boldsymbol{y} \mid \boldsymbol{b}_\gamma, \boldsymbol{\gamma}, \sigma^2) p(\boldsymbol{b}_\gamma \mid \boldsymbol{\gamma}, \sigma^2) p(\sigma^2) d\boldsymbol{b} d\sigma^2 \\ &= \frac{\Gamma(\frac{n}{2}) (2\pi)^{-\frac{n}{2}} \left|\Sigma_\gamma^{-\frac{1}{2}}\right| \left|\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1}\right|^{-\frac{1}{2}}}{(\frac{1}{2} \boldsymbol{y}^T L_\gamma \boldsymbol{y})^{\frac{n}{2}}}. \end{aligned} \tag{4.4}$$

Let

$$\tilde{\boldsymbol{\gamma}} = \big\{\gamma_j = 1, \boldsymbol{\gamma}_{(-j)}\big\}, \quad \boldsymbol{\gamma}^* = \big\{\gamma_j = 0, \boldsymbol{\gamma}_{(-j)}\big\}.$$

We have

$$\log\big(P\big(\boldsymbol{y} \mid \tilde{\boldsymbol{\gamma}}\big)\big) = \log\Gamma\big(\frac{n}{2}\big) - \frac{n}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_{\tilde{\boldsymbol{\gamma}}}| - \frac{1}{2}\log\left|C_{\tilde{\boldsymbol{\gamma}}}^T C_{\tilde{\boldsymbol{\gamma}}} + \Sigma_{\tilde{\boldsymbol{\gamma}}}^{-1}\right| - \frac{n}{2}\log\big(\frac{1}{2} \boldsymbol{y}^T L_{\tilde{\boldsymbol{\gamma}}} \boldsymbol{y}\big)$$

and

$$\log\big(P\big(\boldsymbol{y} \mid \boldsymbol{\gamma}^*\big)\big) = \log\Gamma\big(\frac{n}{2}\big) - \frac{n}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_{\tilde{\boldsymbol{\gamma}}}| - \frac{1}{2}\log\left|C_{\boldsymbol{\gamma}^*}^T C_{\boldsymbol{\gamma}^*} + \Sigma_{\boldsymbol{\gamma}^*}^{-1}\right| - \frac{n}{2}\log\big(\frac{1}{2} \boldsymbol{y}^T L_{\boldsymbol{\gamma}^*} \boldsymbol{y}\big).$$

Since

$$\log|\Sigma_{\tilde{\boldsymbol{\gamma}}}| = 2\log v + \log|\Sigma_{\boldsymbol{\gamma}^*}|,$$

we have

$$\begin{aligned}
\log\left[\frac{P\big(\boldsymbol{y} \mid \gamma_j = 1, \boldsymbol{\gamma}_{(-j)}\big)}{P\big(\boldsymbol{y} \mid \gamma_j = 0, \boldsymbol{\gamma}_{(-j)}\big)}\right] &= \log[P(\boldsymbol{y} \mid \tilde{\boldsymbol{\gamma}})] - log[P(\boldsymbol{y} \mid \boldsymbol{\gamma}^*)] \\
&= -\log(v) + \frac{n}{2}\log\left(\frac{1}{2}\boldsymbol{y}^T L_{\gamma^*}\boldsymbol{y}\right) - \frac{n}{2}\log\left(\frac{1}{2}\boldsymbol{y}^T L_{\tilde{\gamma}}\boldsymbol{y}\right) \\
&\quad + \frac{1}{2}\log\left|C_{\boldsymbol{\gamma}^*}^T C_{\boldsymbol{\gamma}^*} + \Sigma_{\boldsymbol{\gamma}^*}^{-1}\right| - \frac{1}{2}\log\left|C_{\tilde{\boldsymbol{\gamma}}}^T C_{\tilde{\boldsymbol{\gamma}}} + \Sigma_{\tilde{\boldsymbol{\gamma}}}^{-1}\right|.
\end{aligned} \tag{4.5}$$

Therefore, (2.18) can be obtained by taking exponential on the result of (4.5).

Appendix B

Derivation for (2.26), (2.27), and (2.28)

From the likelihood function in (2.24) and priors in (2.22) and (2.23), we have the joint posterior distribution

$$\begin{aligned} p(\boldsymbol{b}, \boldsymbol{\gamma}, \boldsymbol{z} \mid \boldsymbol{y}) &\propto p(\boldsymbol{y} \mid \boldsymbol{b}_\gamma, \boldsymbol{\gamma}, \sigma^2) p(\boldsymbol{b}_\gamma \mid \boldsymbol{\gamma}, \sigma^2) p(\sigma^2) \\ &= |2\pi I_n|^{-\frac{1}{2}} \exp\Big\{ -\frac{1}{2} (\boldsymbol{z} - \boldsymbol{C}_\gamma \boldsymbol{b}_\gamma)^T (\boldsymbol{z} - \boldsymbol{C}_\gamma \boldsymbol{b}_\gamma) \Big\} \\ &\quad \cdot \left| 2\pi\sigma^2 \Sigma_\gamma \right|^{-\frac{1}{2}} \exp\Big\{ -\frac{1}{2} \boldsymbol{b}_\gamma^T \Sigma_\gamma^{-1} \boldsymbol{b} \Big\} p(\boldsymbol{b}_\gamma). \end{aligned} \tag{4.6}$$

Based on (4.6) we can derive the posterior distribution of $\boldsymbol{b}$ as

$$\begin{aligned} p(\boldsymbol{b} \mid \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{y}) &\propto p(\boldsymbol{b}, \boldsymbol{\gamma}, \boldsymbol{z} \mid \boldsymbol{y}) \\ &\propto \exp\Big\{ -\frac{1}{2} (\boldsymbol{z} - \boldsymbol{C}_\gamma \boldsymbol{b}_\gamma)^T (\boldsymbol{z} - \boldsymbol{C}_\gamma \boldsymbol{b}_\gamma) \Big\} \exp\Big\{ -\frac{1}{2} \boldsymbol{b}_\gamma^T \Sigma_\gamma^{-1} \boldsymbol{b} \Big\}. \end{aligned} \tag{4.7}$$

Therefore, the posterior distribution of $\boldsymbol{b}$ is

$$\boldsymbol{b} \mid \boldsymbol{z}, \boldsymbol{\gamma}, \boldsymbol{y} \sim N(\widetilde{K}_\gamma^{-1} \widetilde{M}_\gamma, \widetilde{K}_\gamma^{-1}),$$

where $\widetilde{K}_\gamma = (\boldsymbol{C}_\gamma^T \boldsymbol{C}_\gamma + \Sigma_\gamma^{-1})$ and $\widetilde{M}_\gamma = \boldsymbol{C}_\gamma^T \boldsymbol{z}$. After integrating out $\boldsymbol{b}$ in (4.6), the distribution of $\boldsymbol{\gamma} \mid \boldsymbol{z}, \boldsymbol{y}$ can be obtained as

$$
\begin{aligned}
p(\boldsymbol{\gamma} \mid \boldsymbol{z}, \boldsymbol{y}) &= p(\boldsymbol{\gamma}) \int_{\boldsymbol{b}} p(\boldsymbol{b}, \boldsymbol{\gamma}, \boldsymbol{z} \mid \boldsymbol{y}) d\boldsymbol{b} \\
&\propto p(\boldsymbol{\gamma}) \left|2\pi I_n\right|^{-\frac{1}{2}} \left|2\pi \Sigma_\gamma\right|^{-\frac{1}{2}} \\
&\cdot \int_{\boldsymbol{b}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{z} - \boldsymbol{C}_\gamma \boldsymbol{b}_\gamma)^T (\boldsymbol{z} - \boldsymbol{C}_\gamma \boldsymbol{b}_\gamma) - \frac{1}{2} \boldsymbol{b}_\gamma^T \Sigma_\gamma^{-1} \boldsymbol{b} \Big\} d\boldsymbol{b} \\
&\propto p(\boldsymbol{\gamma}) \left|2\pi \Sigma_\gamma\right|^{-\frac{1}{2}} \int_{\boldsymbol{b}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{b}_\gamma^T \widetilde{K}_\gamma \boldsymbol{b}_\gamma - 2\boldsymbol{b}_\gamma^T \widetilde{M}_\gamma + \widetilde{M}_\gamma^T \widetilde{K}_\gamma^{-1} \widetilde{M}_\gamma) \Big\} d\boldsymbol{b} \\
&\propto p(\boldsymbol{\gamma}) \left|2\pi \Sigma_\gamma\right|^{-\frac{1}{2}} \left|2\pi \widetilde{K}_\gamma^{-1}\right|^{\frac{1}{2}} \exp\Big\{ \frac{1}{2} \widetilde{M}_\gamma^T \widetilde{K}_\gamma^{-1} \widetilde{M}_\gamma \Big\} \exp\Big\{ -\frac{1}{2} \boldsymbol{z}^T \boldsymbol{z} \Big\} \\
&\propto p(\boldsymbol{\gamma}) \left|2\pi \Sigma_\gamma\right|^{-\frac{1}{2}} \left|2\pi \widetilde{K}_\gamma\right|^{-\frac{1}{2}} \exp\Big\{ \frac{1}{2} \widetilde{M}_\gamma^T \widetilde{K}_\gamma^{-1} \widetilde{M}_\gamma \Big\}.
\end{aligned}
\tag{4.8}
$$

To find (2.27), let

$$
\tilde{\boldsymbol{\gamma}} = \big\{ \gamma_j = 1, \boldsymbol{\gamma}_{(-j)} \big\}, \quad \boldsymbol{\gamma}^* = \big\{ \gamma_j = 0, \boldsymbol{\gamma}_{(-j)} \big\}.
$$

The conditional odds is

$$
\begin{aligned}
\widetilde{O}_j &= \frac{p(\gamma_j = 1 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{z}, \boldsymbol{y})}{p(\gamma_j = 0 \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{z}, \boldsymbol{y})} = \frac{p(\gamma_j = 1, \boldsymbol{\gamma}_{(-j)} \mid \boldsymbol{z}, \boldsymbol{y})}{p(\gamma_j = 0, \boldsymbol{\gamma}_{(-j)} \mid \boldsymbol{z}, \boldsymbol{y})} = \frac{p(\tilde{\boldsymbol{\gamma}} \mid \boldsymbol{z}, \boldsymbol{y})}{p(\boldsymbol{\gamma}^* \mid \boldsymbol{z}, \boldsymbol{y})} \\
&= \frac{\left|2\pi \Sigma_{\tilde{\gamma}}\right|^{-\frac{1}{2}}}{\left|2\pi \Sigma_{\gamma^*}\right|^{-\frac{1}{2}}} \cdot \frac{\left|2\pi \widetilde{K}_{\tilde{\gamma}}^{-1}\right|^{\frac{1}{2}}}{\left|2\pi \widetilde{K}_{\gamma^*}^{-1}\right|^{\frac{1}{2}}} \cdot \frac{p(\tilde{\boldsymbol{\gamma}})}{p(\boldsymbol{\gamma}^*)} \cdot \exp\Big\{ \frac{1}{2} \widetilde{M}_{\tilde{\gamma}}^T \widetilde{K}_{\tilde{\gamma}}^{-1} \widetilde{M}_{\tilde{\gamma}} - \frac{1}{2} \widetilde{M}_{\gamma^*}^T \widetilde{K}_{\gamma^*}^{-1} \widetilde{M}_{\gamma^*} \Big\} \\
&= \exp\left[\log \frac{p(\gamma_j \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{z}, \boldsymbol{y})}{p(\gamma_j \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{z}, \boldsymbol{y})} \right],
\end{aligned}
$$

where

$$
\begin{aligned}
\log \frac{p(\gamma_j \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{z}, \boldsymbol{y})}{p(\gamma_j \mid \boldsymbol{\gamma}_{(-j)}, \boldsymbol{z}, \boldsymbol{y})} &= -\log v + \frac{1}{2} \log \left|\widetilde{K}_{\tilde{\gamma}}^{-1}\right| - \frac{1}{2} \log \left|\widetilde{K}_{\gamma^*}^{-1}\right| \\
&\quad + \frac{1}{2} \widetilde{M}_{\tilde{\gamma}}^T \widetilde{K}_{\tilde{\gamma}}^{-1} \widetilde{M}_{\tilde{\gamma}} - \frac{1}{2} \widetilde{M}_{\gamma^*}^T \widetilde{K}_{\gamma^*}^{-1} \widetilde{M}_{\gamma^*} + \log \frac{p(\gamma_j \mid \boldsymbol{\gamma}_{(-j)})}{p(\gamma_j \mid \boldsymbol{\gamma}_{(-j)})}.
\end{aligned}
\tag{4.9}
$$

The last term of (4.9) is the logarithm of the prior odds, which can be obtained from (2.19).

Appendix C

Derivation for (3.4)

We now derive the choice of λ in (3.4) following Grinold and Kahn (2000). Consider combing the benchmark portfolio M with a risk-free portfolio F. The expected value of the return of M is $p_m\mu_m$ and the risk of M is $p_m^2\sigma_m^2$. Here, p_m is the proportion of the asset that is allocated to M. Similar to the mean-variance optimization procedure, the objective function for M is

$$Q_m(\lambda) = p_m\mu_m - \lambda p_m^2\sigma_m^2.$$

Setting the first order derivative of $Q_m(\lambda)$ with respect to p_m as 0, we have

$$\mu_m - 2\lambda p_m\sigma_m^2 = 0,$$

which leads to $\lambda = \frac{\mu_m}{2p_m\sigma_m^2}$. When $p_m = 1$, which means that all assets have been allocated to M, we find $\lambda = \frac{\mu_m}{2\sigma_m^2}$.

References

Abbe, E. (2017), "Community detection and stochastic block models: recent developments," *arXiv preprint arXiv:1703.10146*.

Amenc, N. and Le Sourd, V. (2005), *Portfolio theory and performance analysis*, John Wiley & Sons.

Azcoiti, V., Di Carlo, G., Follana, E., and Royo-Amondarain, E. (2017), "Antiferromagnetic Ising model in an imaginary magnetic field," *Physical Review E*, 96, 032114.

Azizah, E., Rusyaman, E., and Supian, S. (2017), "Optimization of investment portfolio weight of stocks affected by market index," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 166, p. 012008.

Bavelas, A. (1950), "Communication patterns in task-oriented groups," *The Journal of the Acoustical Society of America*, 22, 725–730.

Biernacki, C., Celeux, G., and Govaert, G. (2000), "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE transactions on pattern analysis and machine intelligence*, 22, 719–725.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008), "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, 2008, P10008.

Boginski, V., Butenko, S., and Pardalos, P. M. (2006), "Mining market data: a network approach," *Computers & Operations Research*, 33, 3171–3184.

Brooks, S. P. and Gelman, A. (1998), "General methods for monitoring convergence of iterative simulations," *Journal of Computational and Graphical Statistics*, 7, 434–455.

Cardot, H., Ferraty, F., and Sarda, P. (1999), "Functional linear model," *Statistics & Probability Letters*, 45, 11–22.

Cardot, H., Ferraty, F., and Sarda, P. (2003), "Spline estimators for the functional linear model," *Statistica Sinica*, 571–591.

Clauset, A., Newman, M. E., and Moore, C. (2004), "Finding community structure in very large networks," *Physical review E*, 70, 066111.

Da Costa Jr, N., Cunha, J., and Da Silva, S. (2005), "Stock selection based on cluster analysis," *Economics Bulletin*, 13.

Dauxois, J., Pousse, A., and Romain, Y. (1982), "Asymptotic theory for the principal component analysis of a vector random function: some applications to statistical inference," *Journal of multivariate analysis*, 12, 136–154.

De Boor, C., De Boor, C., Mathématicien, E.-U., De Boor, C., and De Boor, C. (1978), *A practical guide to splines*, vol. 27, Springer-Verlag New York.

Ding, A. A. and Li, Y. (2013), "Copula correlation: An equitable dependence measure and extension of pearson's correlation," *arXiv preprint arXiv:1312.7214*.

Eilers, P. H. and Marx, B. D. (1996), "Flexible smoothing with B-splines and penalties," *Statistical science*, 89–102.

Erdos, P. and Rényi, A. (1960), "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, 5, 17–60.

Fan, Y., James, G. M., Radchenko, P., et al. (2015), "Functional additive regression," *The Annals of Statistics*, 43, 2296–2325.

Febrero-Bande, M., de la Fuente, M. O., et al. (2012), "Statistical computing in functional data analysis: The R package fda. usc," *Journal of Statistical Software*, 51, 1–28.

Fortunato, S. (2010), "Community detection in graphs," *Physics reports*, 486, 75–174.

Freeman, L. C. (1977), "A set of measures of centrality based on betweenness," *Sociometry*, 35–41.

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

Gelman, A., Rubin, D. B., et al. (1992), "Inference from iterative simulation using multiple sequences," *Statistical Science*, 7, 457–472.

George, E. I. and McCulloch, R. E. (1993), "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, 88, 881–889.

Gertheiss, J., Maity, A., and Staicu, A.-M. (2013), "Variable selection in generalized functional linear models," *Stat*, 2, 86–101.

Geweke, J. et al. (1991), *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*, vol. 196, Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN, USA.

Grinold, R. C. and Kahn, R. N. (2000), "Active portfolio management," .

Hans, C., Dobra, A., and West, M. (2007), "Shotgun Stochastic Search for Large p Regression," *Journal of the American Statistical Association*, 102, 507–516.

Hastie, T. and Mallows, C. (1993), "[A Statistical View of Some Chemometrics Regression Tools]: Discussion," *Technometrics*, 35, 140–143.

James, G. M. (2002), "Generalized linear models with functional predictors," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 411–432.

James, G. M., Wang, J., and Zhu, J. (2009), "Functional linear regression that's interpretable," *The Annals of Statistics*, 2083–2108.

Koltchinskii, V. and Minsker, S. (2013), "L_1-Penalization in Functional Linear Regression with Subgaussian Design," *arXiv preprint arXiv:1307.8137*.

Koochakzadeh, N., Keshavarz, F., Sarraf, A., Rahmani, A., Kianmehr, K., Rifaie, M., Alhajj, R., and Rokne, J. (2011), "Stock investment decision making: A social network approach," in *Emerging Intelligent Technologies in Industry*, Springer, pp. 47–57.

Lee, E. R. and Park, B. U. (2012), "Sparse estimation in functional linear regression," *Journal of Multivariate Analysis*, 105, 1–17.

Li, F. and Zhang, N. R. (2010), "Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics," *Journal of the American statistical association*, 105, 1202–1214.

Lin, Z., Cao, J., Wang, L., and Wang, H. (2015), "A Smooth and Locally Sparse Estimator for Functional Linear Regression via Functional SCAD Penalty," *arXiv preprint arXiv:1510.08547*.

Markowitz, H. (1952), "Portfolio selection," *The journal of finance*, 7, 77–91.

Marx, B. D. and Eilers, P. H. (1999), "Generalized linear regression on sampled signals and curves: a P-spline approach," *Technometrics*, 41, 1–13.

McEvoy, B. W., Nandy, R. R., and Tiwari, R. C. (2013), "Bayesian approach for clinical trial safety data using an Ising prior," *Biometrics*, 69, 661–672.

Müller, H.-G. and Stadtmüller, U. (2005), "Generalized functional linear models," *Ann. Statist.*, 33, 774–805.

Nanda, S., Mahanty, B., and Tiwari, M. (2010), "Clustering Indian stock market data for portfolio management," *Expert Systems with Applications*, 37, 8793–8798.

Newman, M. E. (2004), "Fast algorithm for detecting community structure in networks," *Physical review E*, 69, 066133.

Newman, M. E. (2006a), "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, 74, 036104.

Newman, M. E. (2006b), "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, 103, 8577–8582.

Newman, M. E. (2008), "The mathematics of networks," *The new palgrave encyclopedia of economics*, 2, 1–12.

Pae, Y. and Sabbaghi, N. (2015), "Equally weighted portfolios vs value weighted portfolios: Reasons for differing betas," *Journal of Financial Stability*, 18, 203–207.

Pratt, J. W. (2013), "Risk aversion in the small and in the large," in *HANDBOOK OF THE FUNDAMENTALS OF FINANCIAL DECISION MAKING: Part I*, World Scientific, pp. 317–331.

Ramsay, J. and Silverman, B. (1997), *Functional Data Analysis*, Springer.

Ramsay, J. O. and Dalzell, C. (1991), "Some tools for functional data analysis," *Journal of the Royal Statistical Society. Series B (Methodological)*, 539–572.

Rosvall, M., Axelsson, D., and Bergstrom, C. T. (2009), "The map equation," *The European Physical Journal-Special Topics*, 178, 13–23.

Rue, H., Martino, S., and Chopin, N. (2009), "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations," *Journal of the royal statistical society: Series b (statistical methodology)*, 71, 319–392.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, Cambridge Series in Statistical and Probabilistic Mathematics, UK: Cambridge University Press.

Sklar, M. (1959), "Fonctions de repartition an dimensions et leurs marges," *Publ. inst. statist. univ. Paris*, 8, 229–231.

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

Tierney, L. (1994), "Markov chains for exploring posterior distributions," *the Annals of Statistics*, 1701–1728.

Von Luxburg, U. (2007), "A tutorial on spectral clustering," *Statistics and computing*, 17, 395–416.

Yang, Y., Wainwright, M. J., and Jordan, M. I. (2016), "On the computational complexity of high-dimensional Bayesian variable selection," *Ann. Statist.*, 44, 2497–2532.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005), "Functional data analysis for sparse longitudinal data," *Journal of the American Statistical Association*, 100, 577–590.

Yuan, M. and Cai, T. T. (2010), "A reproducing kernel Hilbert space approach to functional linear regression," *Ann. Statist.*, 38, 3412–3444.

Zhou, J., Wang, N.-Y., and Wang, N. (2013), "Functional linear model with zero-value coefficient function at sub-regions," *Statistica Sinica*, 23, 25.

Zhu, H. and Cox, D. D. (2009), "A functional generalized linear model with curve selection in cervical pre-cancer diagnosis using fluorescence spectroscopy," *Lecture Notes-Monograph Series*, 173–189.

Zhu, H., Vannucci, M., and Cox, D. D. (2007), "Functional Data Classification in Cervical Pre-cancer Diagnosis: A Bayesian Variable Selection Model." Joint Statistical Meetings Proceedings 2007.

Zhu, H., Vannucci, M., and Cox, D. D. (2010), "A Bayesian hierarchical model for classification with selection of functional predictors," *Biometrics*, 66, 463–473.

Zhu, H., Yao, F., and Zhang, H. H. (2014), "Structured functional additive regression in reproducing kernel Hilbert spaces," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76, 581–603.