TIME-VARYING NETWORKS ESTIMATION AND CHINESE WORDS SEGMENTATION

BY

XINXIN SHU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Statistics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Professor Annie Qu, Chair
Professor Douglas Simpson
Professor Jeff Douglas
Professor Xiaohui Chen

Abstract

This thesis contains two research areas including time-varying networks estimation and Chinese words segmentation. Chapter 1 introduces the background of the time-varying networks and the structure of Chinese language, followed by the motivations and goals for the research work.

In many biomedical and social science studies, it is important to identify and predict the dynamic changes of associations among network data over time. However, inadequate literature addresses the estimation of time-varying networks mainly because of extremely large volume of time-varying network data, leading to the computational difficulty.

In Chapter 2, we propose a varying-coefficient model to incorporate time-varying network data, and impose a piecewise-penalty function to capture local features of the network associations. The advantages of the proposed approach are that it is nonparametric and therefore flexible in modeling dynamic changes of association for network data problems, and capable of identifying the time regions when dynamic changes of associations occur. To achieve local sparsity of network estimation, we implement a group penalization strategy involving overlapping parameters among different groups. We also develop a fast algorithm, based on the smoothing proximal gradient method, which is computationally efficient and accurate. We illustrate the proposed method through simulation studies and children's attention deficit hyperactivity disorder fMRI data, and show that the proposed method and algorithm efficiently recover dynamic network changes over time.

The digital information has become an essential part of modern life, from scientific research, entertainment business, product marketing to national security protection. So developing fast automatic process of information extraction becomes extremely demanding. Chinese language is the second popular language among all internet users but is still severely under-studied, mainly due to the challenge of its ambiguity nature.

In Chapter 3, we propose a new method for word segmentation in Chinese language processing. The

Chinese language is the second most popular language among all internet users, but it is still not well-studied. Segmentation becomes crucial for Chinese language processing, since it is the first step to develop a fast automatic process of information extraction. One major challenge is that the Chinese language is highly context-dependent, and is very different from English. We propose a machine-learning model with computationally feasible loss functions which utilize linguistically-embedded features. The proposed method is investigated through the Peking university corpus Chinese documents. Our numerical study shows that the proposed method performs better than existing top competitive performers.

I dedicate this dissertation to my family.

You have always been, and will always be, my inspiration.

Acknowledgments

My first debt of gratitude must go to my advisor Dr. Annie Qu. She continuously provided her patience, enthusiasm, immense knowledge and advise for me to proceed through my Ph.D. study and research and complete my dissertation. It has been an honor to be her Ph.D. student. Without her guidance and constant feedback this PhD would not have been achievable.

I also need to thank my thesis and preliminary examination committee: Dr. Douglas Simpson, Dr. Jeffrey Douglas, Dr. Xiaohui Chen and Dr. Xiaofeng Shao, for their encouragement, insightful comments and suggestions. In addition, I would like to acknowledge my collaborators, Dr. Lan Xue, Dr. Xiaotong Shen and Dr. Junhui Wang for useful discussions as well as valuable and pleasant collaborative experiences.

Many thanks are given to my fellow students at the Department of Statistics. Particularly, I would like to thank Jingfei Zhang, Yeonwoo Rho, Peibei Shi, Xuan Bi, Chung Eun Lee, Jin Wang, Kevin He, Christopher Kinson for their support, encouragement and friendship. My sincere thanks also goes to Melissa Banks. She has always gone far beyond your job description to help us through the Ph.D. study. Her efforts and time are greatly appreciated.

Finally, I want to thank my parents for unending encouragement and support throughout my life , particularly during my years of study and research. I would also like to thank my wife Lisa. Her unwavering love, encouragement, and dedication carried me on through difficult times in the final year.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Time-varying networks

In social science, genomic, environmental and biomedical studies, it is scientifically important to identify and predict associations and interactions among genes, spatial locations, social structures, or biological or medical samples effectively. Network modeling (e.g., Kolaczyk, 2009) can effectively quantify the associations among variables. Our method is motivated by a children's attention deficit hyperactivity disorder study, where the data can be obtained from the ADHD-200 sample initiative website (http://www.nitrc.org/frs/?group id=383). The test samples contain fMRI data from different regions of interest of ADHD children's brains, which are repeatedly measured at many time points. We are interested in identifying associations and interactions among different regions of interest of the brain over time so we can better understand how ADHD patients' brains function.

Figure 1.1: Change of associations among different sites of a brain over three time-points

Figure 1.1 illustrates the dynamic changes of associations among several regions of interest of a brain over three time-points. The challenges of analyzing this type of data are that the measurements could be corrupted by noise introduced through various sources. We are interested in extracting the underlying

signals of associations through modeling responses of brain activities over time. This can be formulated as a time-varying network problem, where the regions of interest are variables or nodes in the network, and the associations among regions of interest represent edges connecting nodes of the network.

Recent development on network modeling includes high-dimensional graphical models by Meinshausen and Bühlmann (2006); Friedman et al. (2007); and Peng et al. (2009). The central idea of these approaches is to estimate the precision matrix or the inverse of the covariance matrix which provides a conditional correlation interpretation among variables in the graph, where zero partial correlation implies pairwise conditional independence. In addition, Shen et al. (2012) and Zhu et al. (2013) develop simultaneous grouping pursuit and feature selection for high-dimensional graphs. For multiple graphs, Guo et al. (2011) jointly estimate graphical models to capture the dependence among multiple graphs and their common structure, and Zhu et al. (2014) propose the maximum penalized likelihood approach to model structural changes over multiple graphs to incorporate dependency among interacting units.

Most of the existing literature targets the network data problem observed at one-time-point only. However, networks can be observed at multiple time-points where the dynamic change of associations is of scientific interest and requires quantification. For example, in gene expression data, functional magnetic resonance imaging (fMRI), and social network data, it is common that associations can change over time, and therefore it is important to model and estimate the dynamic changes of the network structure.

Modeling time-varying network data could be statistically and computationally challenging as the network structures over time could be quite complex, involve large-dimensional parameter estimation, and be computationally highly intensive with high-dimensional matrix operations. Existing approaches for time-course network data include linear mixed-effect modeling to incorporate temporal correlation (Shojaie and Michailidis, 2010), the kernel-reweighted logistic regression method for time-evolving network structure (Song et al., 2009; Kolar et al., 2010), and time-varying Markov random fields (Kolar and Xing, 2009). However, these approaches are mainly for the estimation of time-varying networks, and are not designed for model selection to capture the change of associations in local time regions.

In Chpater 2, we propose a local varying-coefficient model, aiming to estimate network associations and detect dynamic changes. We show that the proposed approach can correctly identify zero correlations and consistently estimate strengths of nonzero correlations in dynamic networks. One distinct feature of the proposed model is that we are able to detect local features of time-varying networks, and provide the

detection and estimation simultaneously. In addition, to handle large size network data, we proposed an algorithm which can significantly reduce the computational complexity.

1.2 Chinese words segmentation

Digital information has become an essential part of modern life, from media news, entertainment business, distance learning and communication, research on product marketing, to potential threat detection and national security protection. With the explosive amount of information gathered nowadays, manual information processing becomes far from sufficient and developing fast automatic process of information extraction becomes extremely crucial.

However, Chinese language processing is still an area which has been severely under-studied. This is likely due to specific challenges caused by the characteristics of Chinese language. Word segmentation is considered a crucial step towards Chinese language processing tasks, due to the unique characteristics of Chinese language structure. Chinese words generally are composed of multiple characters without any delimiter appearing between words. For example, the word 博客 "blog" consists of two characters, 博 "plentiful" and 客 "guest". If characters in a word are treated individually rather than together, it could lead to a completely different meaning. Good word segmenters could correctly transform text documents into a collection of linguistically meaningful words, and make it possible to extract information accurately from the documents. Therefore accurate segmentation is a prerequisite step for Chinese document processing. Without effective word segmentation of Chinese documents, it is more difficult to extract correct information given the ambiguous nature of Chinese words.

One major challenge in Chinese word segmentation is that the Chinese language is a highly context-dependent or strongly analytic language. The major differences between Chinese and English are listed as follows. Chinese morphemes corresponding to words have little inflection. English, on the other hand, equipped with inflections is more context-independent. There are a large amount of Chinese words which have more than one meaning under different contexts. For example, the original meaning of word 水分 means "water", but could also mean "inflated;" The word 算账 has double meanings of "balance budget" or "reckoning." Chinese has no tense on verbal inflections to distinguish past such as "-ed", current such as "-ing" and future activities, no number marking such as "-s" in English to distinguish singular versus plural, and no upper or lower case marking to indicate the beginning of a sentence. In addition, English morphemes

can have more than one syllable, while Chinese morphemes are typically monosyllabic and written as one character (Wong et al., 2010).

Another challenge is that the number of Chinese characters is much greater than the number of letters in English. The Kangxi dictionary during the Qing dynasty in the 17th century records around $47,035$ characters. Nowadays the number of characters has almost doubled to $87,019$, according to the Zhonghua Zihai dictionary (Zhonghua Book Company, 1994). Moreover, new Chinese characters are being created by internet users with the exponential speed of the internet in this information age.

In addition, the writing of Chinese characters is not unified because there are two versions of character writing. One is based on traditional characters and the other is simplified character writing. Simplified characters are officially used in the mainland of China, whereas traditional characters are maintained by Taiwan, Hong Kong and Macau. This leads to different coding systems for electronic Chinese documents and webpages. There are three main different coding systems, namely, GB, Big5, and Unicode. The GB encoding scheme is applied to simplified characters, while Big5 is for traditional characters. Unicode can be applied to both writing styles. One advantage of the Unicode system is that GB and Big5 can be converted into Unicode.

In Chapter 3, we propose a novel approach by incorporating linguistic rules into a statistical framework for Chinese segmentation. The proposed model has two advantages. First, new words which are not in the sample corpus are able to be identified through linguistic rules. Second, estimation complexity can be substantially minimized through linguistically-embedded constraints, and thus higher accuracy of segmentation can be attained. Most importantly, the procedure of optimization for the proposed model can be parallelly implemented through transforming nonconvex optimization problems to many subproblems of convex minimization, leading to the scalability for high-volume digital documents.

Chapter 2

Time-varying networks estimation and dynamic model selection

2.1 Introduction

A time-varying network model is a popular tool for identifying dynamic features and time-evolving associations in many sorts of networks such as social networks, gene networks and environmental networks. We propose a dynamic network model to capture the change of associations through a varying-coefficient model (Hastie and Tibshirani, 1993). One advantage of the proposed approach is that it is nonparametric and therefore flexible in modeling the changes of coefficients. Another advantage is that we are able to locate the time region when dynamic changes of associations occur. This is applicable in identifying the change of associations among different regions of interest over time as in the example of fMRI data for ADHD patients, which could be potentially useful for detecting the dynamic changes of brain functions.

In order to achieve local sparsity for the network data, we propose a piecewise penalized loss function incorporating the local features of the varying-coefficient models in the dynamic modeling. The piecewise penalization strategy involves overlapping spline-coefficient parameters among different penalty groups. However, the popular coordinate-wise descent algorithm cannot be applied in our optimization. We propose an alternative algorithm which is computationally efficient and accurate based on the proximal gradient method. The advantage of this approach is that it does not involve large-dimensional matrix inversion, and is capable of handling large-dimensional network data.

One computational challenge we face for time-varying network data is that the volume of this type of data is extremely large, as it includes observations for many nodes over many time points. For example, when the network size is about 100 and observed over 50 time points, the dimension of the matrix operation could reach 10^5 in iteration process. Existing methods for handling time-varying networks mainly target relatively small network sizes with limited time points. Therefore there is a great demand to develop computationally efficient and fast algorithms to solve the large-dimensional time-varying network problem.

The proposed group penalization strategy effectively ensures local sparsity; however, it brings additional computational cost in the optimization process, as it requires a high degree of memory storage and matrix operations for solving the dynamic network problem. In theory, it is also more challenging to establish local-feature model selection consistency than global-feature model selection consistency. We show that the proposed method identifies zero estimators in the non-signal time regions, and estimates the partial correlation functions uniform consistently in the signal regions.

This Chapter is organized as follows. Section 2.2 proposes the penalized polynomial spline method for time-varying network data. Section 2.3 provides the smoothing proximal gradient algorithm to capture the dynamic change of network data over time. Section 2.4 presents asymptotic theory of model selection local consistency. In Section 2.5, we compare the numerical performances of the proposed smoothing proximal gradient algorithm with other existing approaches. Section 2.6 illustrates the proposed method for the fMRI data of ADHD patients. The final section provides concluding remarks and a brief discussion.

2.2 Time-varying networks

A network can be defined as an undirected graph, where the p vertices can be represented as the p-dimensional random variables $(y_1, \ldots, y_p)'$, and the corresponding covariance matrix Σ is a $(p \times p)$-dimensional positive definite matrix. An edge connects variables y_i and y_j if and only if the correlation between y_i and y_j is nonzero. We define the precision matrix as $\Sigma^{-1} = (\sigma^{ij})_{p\times p}$ and $\rho^{ij} = -(\sigma^{ij}/\sqrt{\sigma^{ii}\sigma^{jj}})$ for $1 \leq i, j \leq p$, and ρ^{ij} is the partial correlation between y_i and y_j given other variables $y_{-(i,j)}$. We can model y_i through other y_j's based on $y_i = \sum_{j\neq i} \rho^{ij}\sqrt{(\sigma^{jj}/\sigma^{ii})}y_j + \varepsilon_i$, where $\text{var}(\varepsilon_i) = 1/\sigma^{ii}$ and ε_i is uncorrelated with $y_{-i} = \{y_j : 1 \leq j \neq i \leq p\}$.

The following joint loss function for a sample of size n is proposed by Peng et al. (2009) for network data observed at one time point:

$$L(\boldsymbol{\rho}, \boldsymbol{\sigma}, \mathbf{y}) \;=\; \frac{1}{2n}\sum_{k=1}^{n}\sum_{i=1}^{p} w_i \left(y_i^k - \sum_{j\neq i}^{p} \rho^{ij}\sqrt{\frac{\sigma^{jj}}{\sigma^{ii}}}y_j^k \right)^2, \tag{2.1}$$

where $\mathbf{y} = \{\mathbf{y}^k\}_{k=1}^n$ with $\mathbf{y}^k = (y_1^k, \ldots, y_p^k)'$ and y_i^k being the ith variable for the k-th subject, $\boldsymbol{\rho} = (\rho^{12}, \ldots, \rho^{(p-1)p})$, $\boldsymbol{\sigma} = \{\sigma^{ii}\}_{i=1}^p$, $\mathbf{w} = \{w_i\}_{i=1}^p$ are nonnegative weights and w_i typically can be chosen as $\text{var}^{-1}(\varepsilon_i) = \sigma^{ii}$. Note that $\rho^{ij} = \rho^{ji}$. So the total number of parameters in $\boldsymbol{\rho}$ is $(p-1)p/2$. The partial

correlation parameters $\boldsymbol{\rho}$ can be estimated by minimizing the loss function in (2.1).

In this chapter, we are interested in modeling the dynamic change in partial correlation where the partial correlation could be time-varying. Let $\mathbf{y}(t) = \big(y_1(t), \cdots, y_p(t)\big)'$ be a set of time-varying variables observed at time t, and $\{\mathbf{y}(t), t \in \mathbf{I}\}$ be the corresponding continuous stochastic process defined on a compact interval I. Without loss of generality, let $\mathbf{I} = [0, 1]$. Suppose the data consists of n subjects with measurements taken at m discrete time-points $0 \leq t_{k1} < \cdots < t_{km} \leq 1$ for each subject $k = 1, \ldots, n$, and its observation $\mathbf{y}^k(\mathbf{t}_k) = \big(\mathbf{y}^k(t_{k1}), \cdots, \mathbf{y}^k(t_{km})\big)'$ is a discrete realization of the continuous stochastic process $\{\mathbf{y}(t), t \in \mathbf{I}\}$. Here $\mathbf{y}^k(t_{ku}) = \big(y_1^k(t_{ku}), \cdots, y_p^k(t_{ku})\big)'$ for $u = 1, \ldots, m$, with $y_i^k(t_{ku})$ being the ith variable observed at the time t_{ku} for the kth subject. We propose the following joint loss function for time-varying networks,

$$L(\boldsymbol{\rho}, \boldsymbol{\sigma}, \mathbf{t}, \mathbf{y}) = \frac{1}{2nm} \sum_{k=1}^{n} \sum_{i=1}^{p} \sum_{u=1}^{m} w_{i_{ku}} \left(y_i^k(t_{ku}) - \sum_{j \neq i}^{p} \rho^{ij}(t_{ku}) \sqrt{\frac{\sigma^{jj}(t_{ku})}{\sigma^{ii}(t_{ku})}} y_j^k(t_{ku}) \right)^2, \tag{2.2}$$

where $\boldsymbol{\rho} = \big\{\rho^{12}(\mathbf{t}_k), \ldots, \rho^{(p-1)p}(\mathbf{t}_k)\big\}_{k=1}^{n}$, $\boldsymbol{\sigma} = \big\{\sigma^{ii}(\mathbf{t}_k)\big\}_{i=1,k=1}^{p,n}$ and $\mathbf{y} = \big\{\mathbf{y}^k(\mathbf{t}_k)\big\}_{k=1}^{n}$ with $\mathbf{t}_k = (t_{k1}, \ldots, t_{km})'$. Both the weights and the components in the concentration matrix can vary over time in the model (2.2). In addition, the functions $\boldsymbol{\rho}(t) = \{\rho^{12}(t), \cdots, \rho^{(p-1)p}(t)\}'$ are the time-varying coefficients, and can be approximated by spline functions. We apply the spline approximation here since it provides a good approximation of any smooth functions, even with a small number of knots.

In classical polynomial spline estimation, each time-varying partial coefficient function $\rho^{ij}(t)$ can be approximated by a spline function. Suppose$\{\nu_h\}_{h=1}^{N_n}$are N_n interior knots within the interval $[0, 1]$. Let Υ be a partition of the interval $[0, 1]$ with N_n knots, that is $\Upsilon_n = \{0 = \nu_0 < \nu_1 < \cdots < \nu_{N_n} < \nu_{N_n+1} = 1\}$. The polynomial splines of order $q + 1$ are functions with q-degree of polynomials on intervals $[\nu_{h-1}, \nu_h), h = 1, \ldots, N_n$ and $[\nu_{N_n}, \nu_{N_n+1}]$, and $q - 1$ continuous derivatives globally. We denote the space of such spline functions by $G_n = G(q, \Upsilon_n)$. Let $\nu_{-q} = \cdots = \nu_{-1} = 0$ and $\nu_{N_n+2} = \cdots = \nu_{N_n+q+1} = 1$ be auxiliary knots, and $(t)_+^q = t^q I(t \geq 0)$. For any function g on $[0, 1]$, the divided difference of g on a grid of m points

$0 \leq \nu_1^* \leq \nu_2^* \leq \cdots \leq \nu_m^* \leq 1$ is defined by

$$\begin{aligned}
[\nu_i^*]g &= g(\nu_i^*), \\
[\nu_1^*, \cdots, \nu_m^*]\, g &= \frac{[\nu_2^*, \cdots, \nu_m^*]g - [\nu_1^*, \cdots, \nu_{m-1}^*]g}{\nu_m^* - \nu_1^*} \quad \text{for } \nu_m^* \neq \nu_1^*, \\
[\nu_1^*, \cdots, \nu_m^*]\, g &= \frac{d^{m-1}}{dt^{m-1}} g\left(\nu_1^*\right)/(m-1)! \quad \text{for } \nu_1^* = \nu_m^* \in (0,1), \\
[\nu_1^*, \cdots, \nu_m^*]\, g &= \frac{d_+^{m-1}}{dt^{m-1}} g\left(\nu_1^*\right)/(m-1)! \quad \text{for } \nu_1^* = \nu_m^* = 0, \\
[\nu_1^*, \cdots, \nu_m^*]\, g &= \frac{d_-^{m-1}}{dt^{m-1}} g\left(\nu_1^*\right)/(m-1)! \quad \text{for } \nu_1^* = \nu_m^* = 1,
\end{aligned}$$

where d_-^h/dt^h and d_+^h/dt^h denote the hth left and right derivatives of a given function. Then the normalized B spline basis of G_n is defined as,

$$B_h(t) = (-1)^{q+1}\left(\nu_h - \nu_{h-q-1}\right)\left[\nu_{h-q-1}, \cdots, \nu_h\right](t - \nu_q)_+^q, \;\; h = 1, \ldots, J_n.$$

As a result, for any function $g \in G_n$, one has $g(\cdot) = \sum_{h=1}^{J_n} \beta_h B_h(\cdot)$ for some coefficient $\beta = \left(\beta_1, \ldots, \beta_{J_n}\right)^T$. The function $\rho^{ij}(t)$ for any $1 \leq i < j \leq p$ can be approximated by

$$\rho^{ij}(t) \approx g^{ij}(t) = \sum_{h=1}^{J_n} \beta_h^{ij} B_h(t) = (\boldsymbol{\beta}^{ij})' \mathbf{B}(t),$$

where $\boldsymbol{\beta}^{ij} = \left(\beta_1^{ij}, \ldots, \beta_{J_n}^{ij}\right)'$ is a set of coefficients, and $\mathbf{B}(t) = (B_1(t), \ldots, B_{J_n}(t))'$ are B-spline bases. In practice, different B-spline bases can be used to approximate different $g^{ij}(t)$. For simplicity, the same set of B-spline bases are used for different partial correlation functions in this chapter. The advantages of spline approximation for the time-varying coefficient model are that it is computationally fast and efficient. In the traditional polynomial spline estimation, one replaces ρ^{ij} with the spline g^{ij} in (2.2) and estimates the spline coefficients $\left\{\boldsymbol{\beta}^{ij}, 1 \leq i < j \leq p\right\}$ by minimizing (2.2). In this chapter, we are more interested in locally sparse estimators of the partial correlations that characterize dynamic changes of network associations over time.

The B-spline basis function has a desirable local property. For any interval constructed by two consecutive knots, denote as (ν_{h-1}, ν_h) for $1 \leq h \leq N_n + 1$. If $t \in (\nu_{h-1}, \nu_h)$, the spline function $g^{ij}(t)$ is only affected by basis functions $B_h, \ldots, B_{h+q}$. Therefore, the spline function $g^{ij}(t)$ is locally zero within the in-

terval (ν_{h-1}, ν_h) if and only if the spline coefficients $\boldsymbol{\gamma}_h^{ij} = (\beta_h^{ij}, \ldots, \beta_{(h+q)}^{ij})'$ are all zero. In addition, the whole region $[0, 1]$ can be divided into $N_n + 1$ intervals by the spline knots. We then propose the following piecewise penalized loss function to achieve sparsity for the network data,

$$L(\boldsymbol{\beta}, \boldsymbol{\sigma}, \mathbf{t}, \mathbf{y}) = \frac{1}{2nm} \sum_{k=1}^{n} \sum_{i=1}^{p} \sum_{u=1}^{m} w_{i_{ku}} \left(y_i^k(t_{ku}) - \sum_{j \neq i}^{p} \sum_{h=1}^{J_n} \beta_h^{ij} B_h(t_{ku}) \sqrt{\frac{\sigma^{jj}(t_{ku})}{\sigma^{ii}(t_{ku})}} y_j^k(t_{ku}) \right)^2 + \sum_{i<j}^{p} \sum_{h=1}^{N_n+1} P_\lambda(\|\boldsymbol{\gamma}_h^{ij}\|), \tag{2.3}$$

where $\boldsymbol{\beta} = (\beta_1^{1,2}, \ldots, \beta_{J_n}^{1,2}, \ldots, \beta_1^{p-1,p}, \ldots, \beta_{J_n}^{p-1,p})'$ is a $p(p-1)J_n/2$-dimensional parameter vector, $\boldsymbol{\sigma} = \left\{\sigma^{ii}(\mathbf{t})\right\}_{i=1}^{p}$ with $\mathbf{t} = (\mathbf{t}_1, \ldots, \mathbf{t}_n)'$, P_λ is a penalty function chosen from LASSO, adaptive LASSO or SCAD penalties, λ is a tuning parameter, $\|\cdot\|$ is the L_2-norm, and $\boldsymbol{\gamma}_h^{ij}$ can be shrunk towards zero if the magnitude of $\boldsymbol{\gamma}_h^{ij}$ is sufficiently small. The penalty term is different from a typical penalty term as we incorporate the local features of varying-coefficient models and ensure local sparsity of the dynamic modeling. Both $\boldsymbol{\beta}$ and $\boldsymbol{\sigma}$ are unknown parameters but $\boldsymbol{\beta}$ is the main parameter of our interest. To estimate $\boldsymbol{\beta}$ in the penalized loss (2.3), $\boldsymbol{\sigma}$ needs to be specified and a two-step iterative procedure will be proposed in the algorithm in the next section.

Let $\widetilde{\mathbf{y}}_{iu} = \sqrt{\frac{w_{iu}}{nm}} \mathbf{y}_i(t_u)$, $\mathbf{y}_i(t_u) = (y_i^1(t_u), \ldots, y_i^n(t_u))'$, $\widetilde{\mathbf{y}}_i = (\widetilde{\mathbf{y}}_{i1}', \ldots, \widetilde{\mathbf{y}}_{im}')'$, and $\mathcal{Y}_n = (\widetilde{\mathbf{y}}_1', \ldots, \widetilde{\mathbf{y}}_p')'$ be a nmp-dimensional vector. Let $\mathcal{X}_n = (\widetilde{\mathbf{x}}_{(1,2)}', \ldots, \widetilde{\mathbf{x}}_{(p-1,p)}')$ be a $(nmp) \times \{p(p-1)J_n/2\}$-dimensional matrix, with $\widetilde{\mathbf{x}}_{(i,j)} = (\mathbf{0}_1, \ldots, \mathbf{0}_{i-1}, \mathbf{z}_{(i,j)}^j, \mathbf{0}_{i+1}, \ldots, \mathbf{0}_{j-1}, \mathbf{z}_{(i,j)}^i, \ldots, \mathbf{0}_p)'$, where $\mathbf{0}_k = \{0\}_{J_n \times nm}$, and $\mathbf{z}_{(i,j)}^j = \left(\mathbf{z}_{(i,j),1}^j, \ldots, \mathbf{z}_{(i,j),m}^j\right)'$, with $\mathbf{z}_{(i,j),u}^j = \left(\mathbf{B}(t_{1u}) \sqrt{\frac{\widetilde{\sigma}^{jj}(t_{1u})}{\widetilde{\sigma}^{ii}(t_{1u})}} y_j^1(t_{1u}), \ldots, \mathbf{B}(t_{nu}) \sqrt{\frac{\widetilde{\sigma}^{jj}(t_{nu})}{\widetilde{\sigma}^{ii}(t_{nu})}} y_j^n(t_{nu})\right)$, for $u = 1, \ldots, m$, and $\widetilde{\sigma}^{ii}(t_u) = \sigma^{ii}(t_u)/w_{iu}$. Then the corresponding loss function (2.3) is equivalent to

$$L(\boldsymbol{\beta}, \boldsymbol{\sigma}, \mathcal{Y}_n) = \frac{1}{2} \|\mathcal{Y}_n - \mathcal{X}_n \boldsymbol{\beta}\|^2 + \sum_{i<j}^{p} \sum_{h=1}^{N_n+1} P_{\lambda_n}(\|\boldsymbol{\gamma}_h^{ij}\|). \tag{2.4}$$

Let $\hat{\boldsymbol{\beta}}$ be the minimizer of object function (2.3) or (2.4). Then the resulting estimator for the partial correlation function $\rho^{ij}(t)$ is defined as $\hat{\rho}^{ij}(t) = \hat{\boldsymbol{\beta}}^{ij} \mathbf{B}(t)$.

2.3 Algorithm

2.3.1 Proximal gradient approximation

In this section, we propose an algorithm to obtain an optimal solution for the objective function (2.4). Let the penalty function $P_\lambda(\|\boldsymbol{\gamma}_h^{ij}\|)$ in (2.4) follow the adaptive Lasso penalty (Tibshirani, 1996; Zou, 2006), that is, $P_\lambda(\|\boldsymbol{\gamma}_h^{ij}\|) = \lambda\tau_h^{ij}\|\boldsymbol{\gamma}_h^{ij}\|$, where $\tau_h^{ij} = 1/\|\tilde{\gamma}_h^{ij}\|^r$ with $r > 0$ and $\tilde{\gamma}_h^{ij}$ is a consistent estimator of γ_h^{ij}. So the penalty term can be considered as adaptive group LASSO with overlapping groups. When the groups overlap, if one group is shrunk to zero, all the coefficients in this group shrink to zero even though some coefficients in this group also belong to other nonzero-coefficient groups. The solution space and theoretical properties of the group LASSO with overlaps are discussed in Jenatton et al. (2009) and Obozinski et al. (2013), which indicate that traditional algorithms for LASSO cannot be directly applied to the penalized loss function in (2.3).

However, since the dual norm of the L_2-norm is still the L_2-norm, the L_2-norm $\boldsymbol{\gamma}_h^{ij}$ can be formulated as $\max_{\|\boldsymbol{\alpha}_h^{ij}\|\leq 1}(\boldsymbol{\alpha}_h^{ij})'\boldsymbol{\gamma}_h^{ij}$, where $\boldsymbol{\alpha}_h^{ij} \in R^{(p+1)}$ is an auxiliary vector associated with $\boldsymbol{\gamma}_h^{ij}$. A similar transformation and its properties have been discussed in Chen et al. (2012), Jacob et al. (2013) and Obozinski et al. (2013). Let $Q = \{\boldsymbol{\alpha} | \|\boldsymbol{\alpha}_h^{ij}\| \leq 1, 1 \leq i < j \leq p, h = 1, \cdots, N_n + 1\}$. We can rewrite the group adaptive LASSO penalty for the overlapping parameters in (2.3) as follows:

$$g_0(\boldsymbol{\beta}) = \lambda \sum_{i<j}^{p} \sum_{h=1}^{N_n+1} \tau_h^{ij} \|\boldsymbol{\gamma}_h^{ij}\| = \max_{\boldsymbol{\alpha}\in\mathbf{Q}} \sum_{i<j}^{p} \sum_{h=1}^{N_n+1} \lambda\tau_h^{ij} (\boldsymbol{\alpha}_h^{ij})'\boldsymbol{\gamma}_h^{ij} = \max_{\boldsymbol{\alpha}\in\mathbf{Q}} \boldsymbol{\alpha}' C\boldsymbol{\beta}, \tag{2.5}$$

where $C \in R^{[(q+1)(N_n+1)p(p-1)/2]\times[p(p-1)J_n/2]}$ is an indicator matrix with the element defined as

$$C_{(k,l)} = \begin{cases} \lambda\tau_h^{ij} & k = (r-1)(N_n+1)(q+1) + (h-1)(q+1) + v, l = (r-1)J_n + (h-1) + v \\ 0 & \text{otherwise} \end{cases},$$

where $r = (i-1)(p-i+2) + (j-i-1)$ and $v = 1, \cdots, (q+1)$. Note that C is a very sparse matrix with only one non-zero element in each row, and therefore only requires a relatively small amount of memory storage in the optimization procedure. Through the transformation, the group penalization terms no longer present overlapping parameters.

However, this introduces a new problem, that the penalty function $g_0(\boldsymbol{\beta})$ in (2.5) is a non-smooth function of $\boldsymbol{\beta}$. To circumvent this problem, we need to build a smooth function to approximate $g_0(\boldsymbol{\beta})$. Let $D = \max_{\boldsymbol{\alpha}\in\mathbf{Q}} \|\boldsymbol{\alpha}\|^2/2$ and

$$g_\mu(\boldsymbol{\beta}) = \max_{\boldsymbol{\alpha}\in\mathbf{Q}} \left(\boldsymbol{\alpha}' C\boldsymbol{\beta} - \frac{\mu}{2}\|\boldsymbol{\alpha}\|^2\right), \tag{2.6}$$

where μ is the tolerance parameter. Then $g_\mu(\boldsymbol{\beta})$ is a quadratic approximation for $g_0(\boldsymbol{\beta})$ with the maximum difference of μD. That is,

$$g_0(\boldsymbol{\beta}) - \mu D \leq g_\mu(\boldsymbol{\beta}) \leq g_0(\boldsymbol{\beta}).$$

In order to control the maximum difference, we choose the tolerance level $\epsilon = \mu D$, or equivalently $\mu = \epsilon/D$. Consequently, the loss function in (2.4) can be approximated by

$$\widetilde{L}(\mu, \boldsymbol{\beta}, \boldsymbol{\sigma}) = \frac{1}{2}\|\mathcal{Y}_n - \mathcal{X}_n\boldsymbol{\beta}\|^2 + g_\mu(\boldsymbol{\beta}).$$

To minimize the loss function $\widetilde{L}(\mu, \boldsymbol{\beta})$, we need to calculate the gradient of $\widetilde{L}(\mu, \boldsymbol{\beta})$. For any $\mu > 0$, $g_\mu(\boldsymbol{\beta})$ is convex and continuously differentiable and the corresponding gradient function $\nabla g_\mu(\boldsymbol{\beta})$ is $C'\boldsymbol{\alpha}^*$, where $\boldsymbol{\alpha}^*$ is the optimal solution in (2.6). Let $\boldsymbol{u}_h^{ij} = \lambda\tau_h^{ij}\boldsymbol{\gamma}_h^{ij}/\mu$ and the closed form of $\boldsymbol{\alpha}^*$ can be expressed as

$$(\boldsymbol{\alpha}_h^{ij})^* = \begin{cases} \frac{\boldsymbol{u}_h^{ij}}{\|\boldsymbol{u}_h^{ij}\|}, & \text{if } \left\|\boldsymbol{u}_h^{ij}\right\| > 1 \\ \boldsymbol{u}_h^{ij}, & \text{if } \left\|\boldsymbol{u}_h^{ij}\right\| \leq 1 \end{cases}. \tag{2.7}$$

Therefore the partial derivative $\nabla\widetilde{L}(\mu, \boldsymbol{\beta}, \boldsymbol{\sigma})$ with respect to $\boldsymbol{\beta}$ can be calculated as $\mathcal{X}_n' (\mathcal{X}_n\boldsymbol{\beta} - \mathcal{Y}_n) + C'\boldsymbol{\alpha}^*$. Moreover, $\nabla\widetilde{L}(\mu, \boldsymbol{\beta}, \boldsymbol{\sigma})$ is Lipschitz-continuous with the Lipschitz constant

$$M = \lambda_{\max}\left(\mathcal{X}_n'\mathcal{X}_n\right) + \frac{\|C\|^2}{\mu},$$

where $\lambda_{\max}$ is the largest eigenvalue of $(\mathcal{X}_n)'\mathcal{X}_n$ and $\|C\| = \max_{\|\boldsymbol{\alpha}\|\leq 1} \|C\boldsymbol{\alpha}\|$. The proximal operator can be defined as

$$Q_L(\boldsymbol{\beta}, \boldsymbol{\beta}', \boldsymbol{\sigma}) = \left\{\widetilde{L}(\mu, \boldsymbol{\beta}', \boldsymbol{\sigma}) + \nabla L(\mu, \boldsymbol{\beta}', \boldsymbol{\sigma})(\boldsymbol{\beta} - \boldsymbol{\beta}') + \frac{M}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|^2\right\},$$

and $\boldsymbol{\beta}$ can be updated at the $(l+1)$th iteration by applying the proximal gradient algorithm through

$$\begin{aligned}\boldsymbol{\beta}^{(l+1)} &= arg\min_{\boldsymbol{\beta}} Q_L(\boldsymbol{\beta}, \boldsymbol{\beta}^{(l)}, \boldsymbol{\sigma}) \\ &= arg\min_{\boldsymbol{\beta}} \left\{ \widetilde{L}(\mu, \boldsymbol{\beta}^{(l)}, \boldsymbol{\sigma}^{(l)}) + \nabla L(\mu, \boldsymbol{\beta}^{(l)}, \boldsymbol{\sigma}^{(l)})(\boldsymbol{\beta} - \boldsymbol{\beta}^{(l)}) + \frac{M}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}^{(l)}\|^2 \right\}. \quad (2.8)\end{aligned}$$

Convergence is guaranteed since the inequality $\widetilde{L}(\mu, \boldsymbol{\beta}^{(l+1)}, \boldsymbol{\sigma}^{(l)}) \leq Q_L(\boldsymbol{\beta}, \boldsymbol{\beta}^{(l)}, \boldsymbol{\sigma}^{(l)})$ holds for each iteration. It is not difficult to check the inequality holds, and the details are discussed in Chen et. al (2008). The above penalization strategy is able to achieve sparsity corresponding to the group parameters $\boldsymbol{\gamma}_h$; however, it does not guarantee the sparsity of each element in $\hat{\boldsymbol{\beta}}$ obtained from (2.8). Alternatively, we can set $\beta_h^{ij} = 0$ if the $\|\beta_h^{ij}\| < \epsilon^*$ for a small tolerance level ϵ^*. For $\boldsymbol{\sigma}$, if each subject is observed at the same time over m time points, i.e. $t_{ku} = t_u$ for any $k = 1, \ldots, n$ and $u = 1, \ldots, m$, then each component of $\boldsymbol{\sigma}^{(l+1)} = \left\{ \left((\sigma^{11})^{(l+1)}(t_u), \cdots, (\sigma^{pp})^{(l+1)}(t_u) \right) \right\}_{u=1}^{m}$ at the $l+1$-th iteration can be updated by

$$\frac{1}{(\sigma^{ii})^{(l+1)}(t_u)} = \frac{1}{n}\sum_{k=1}^{n}\left(y_i^k(t_u) - \sum_{j\neq i}^{p}\sum_{h=1}^{J_m}(\beta_h^{ij})^{(l)} B_h^{ij}(t_u)\sqrt{\frac{(\sigma^{jj})^{(l)}(t_u)}{(\sigma^{ii})^{(l)}(t_u)}} y_j^k(t_u) \right)^2, \quad (2.9)$$

and the weight component for the ith subject is $w_{iu}^{(l+1)} = (\sigma^{ii})^{(l+1)}$. If each subject is observed at the different m time points, one can get an update of $\left(\sigma^{ii}\right)^{(l+1)}(t)$ using a polynomial spline estimation method. To be more specific, let $\hat{\varepsilon}_i^2(t_{ku}) = \left(y_i^k(t_{ku}) - \sum_{j\neq i}^{p}\sum_{h=1}^{J_m}(\beta_h^{ij})^{(l)} B_h^{ij}(t_{ku})\sqrt{\frac{(\sigma^{jj})^{(l)}(t_{ku})}{(\sigma^{ii})^{(l)}(t_{ku})}} y_j^k(t_{ku}) \right)^2$. For each $i = 1, \ldots, p$, one can estimate $\sigma^{ii}(t)$ by a polynomial spline regression using $\left\{1/\hat{\varepsilon}_i^2(t_{ku})\right\}_{k=1,u=1}^{n,m}$ as the response variables, and the spline basis generated on time points $\{(t_{ku})\}_{k=1,u=1}^{n,m}$ as explanatory variables. We summarize the algorithm as follows.

Algorithm 1 Proximal gradient algorithm for estimating partial correlation networks

Input: Set desired tolerance levels ϵ and ϵ^*, obtain $\mu = \epsilon/D$ and matrix C, and calculate the step size M; initialize the parameters $\boldsymbol{\beta}, \boldsymbol{\sigma}$ as $\boldsymbol{\beta}^{(0)}$ and $\boldsymbol{\sigma}^{(0)}$.

Output: $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\sigma}}$.

1: Compute $\boldsymbol{\alpha}^*$ according to (2.7) and calculate $\nabla\widetilde{L}(\boldsymbol{\beta}^{(l)}, \mu) = \mathcal{X}_n'(\mathcal{X}_n\boldsymbol{\beta}^{(l)} - \mathcal{Y}_n) + C'\boldsymbol{\alpha}^*$;

2: Obtain $\boldsymbol{\beta}^{(l+1)}$ by minimizing (2.8), i.e., $\boldsymbol{\beta}^{(l+1)} = arg\min_{\boldsymbol{\beta}} Q_L(\boldsymbol{\beta}^{(l)}, \boldsymbol{\beta})$, and set the elements in $\boldsymbol{\beta}^{(l+1)}$ less than ϵ^* as zero;

3: Update $\boldsymbol{\sigma}^{(l+1)}$ and $\mathbf{w}^{(l+1)}$ by calculating (2.9);

4: Return to Step 1 if $\left\| Q_L(\boldsymbol{\beta}^{(l+1)}, \boldsymbol{\beta}^{(l)}, \boldsymbol{\sigma}^{(l+1)}) - Q_L(\boldsymbol{\beta}^{(l)}, \boldsymbol{\beta}^{(l-1)}, \boldsymbol{\sigma}^{(l)}) \right\| > \epsilon$.

The proximal gradient method has the following advantages: (a) we can construct a smoothing approximation to the objective function, which makes the convergence fast; (b) it does not require large matrix inversion and only involves sparse matrix operations. These could reduce algorithm complexity and improve computational speed significantly.

2.3.2 Tuning parameters selection

The choice of tuning parameters is critical as this determines the performance of the proposed method. Tuning parameter selection for the varying-coefficient model involves two parts. One is the selection of the sequence of knots for the polynomial spline, and the other is the selection of the tuning parameter in the penalty function. For convenience, we use equally-spaced knots and the number of interior knots is chosen to be the order of $N_n = n^{1/(2q+3)}$ where n is the sample size and q is the order of the polynomial spline. A similar method for knot selection can be found in Huang et al. (2004), Xue et al. (2010) and Xue and Qu (2012).

In the process of selecting the tuning parameters associated with the penalty function, we use the Bayesian Information Criteria (BIC) procedure, which can often be found in the model selection literature (e.g., Qu and Li, 2006; Wang, Li and Tsai, 2007). Specifically, given the tuning parameters λ_n, denote the estimator $\widehat{\boldsymbol{\beta}}^{\lambda_n}$, and calculate the estimators $\boldsymbol{\sigma}^{\lambda_n}$ and w^{λ_n} through the formula (2.9). Let κ_n be the total number of nonzero elements in $\widehat{\boldsymbol{\beta}}^{\lambda_n}$. Then the BIC function is given as

$$
\begin{aligned}
BIC(\lambda_n) \;=\; & \frac{1}{2}\sum_{k=1}^{n}\sum_{i=1}^{p}\sum_{u=1}^{m}\widehat{w}_{iu}^{(\lambda_n)}\left(y_i^k(t_{ku}) - \sum_{j\neq i}^{p}\sum_{h=1}^{J_n}(\widehat{\beta}_h^{ij})^{(\lambda_n)}B_h^{ij}(t_{ku})\sqrt{\frac{(\widehat{\sigma}^{jj})^{(\lambda_n)}(t_{ku})}{(\widehat{\sigma}^{ii})^{(\lambda_n)}(t_{ku})}}y_j^k(t_{ku})\right)^2 \\
& + \frac{\kappa_n \log(nm)}{nm},
\end{aligned}
$$

and the turning parameter $\widehat{\lambda}$ is selected by minimizing $BIC(\lambda)$.

2.4 Asymptotic theory

In this section, we investigate the asymptotic properties of the varying-coefficient estimator $\widehat{\boldsymbol{\rho}}(t)$ based on the polynomial spline approximation. Since one distinct feature of our approach is the estimation and

selection of local features in dynamic network modeling, we will focus on establishing the local-feature model selection consistency of $\widehat{\boldsymbol{\rho}}(t)$. That is, if true $\boldsymbol{\rho}(t)$ is 0 for any given region, the estimators of $\boldsymbol{\rho}(t)$ is 0 with probability approaching 1.

Before presenting the asymptotic properties of the proposed model, we first introduce the following regularity conditions that are required to establish the asymptotic properties.

C1: The weights $\{w_{it}\}_{i=1}^{p}$ are uniformly finite for $t \in I$. That is, there exist positive constants w_0 and w_∞ such that $0 < w_0 \leq \min_i\{w_{it}\} \leq \max_i\{w_{it}\} \leq w_\infty < \infty$ for any $t \in I$.

C2: For any $\eta > 0$, there exists a constant c such that for sufficient large n, $\max_{1\leq i\leq p} \sup_{t\in\mathbf{I}} |\widehat{\boldsymbol{\sigma}}^{ii}(t) - \boldsymbol{\sigma}^{ii}(t)| \leq c\sqrt{\frac{log(n)N_n}{n}}$ holds with probability $1 - O(n^{-\eta})$.

C3: The eigenvalues of the true covariance matrix $\Sigma(t)$ are assumed to be uniformly bounded for $t \in I$. That is, $0 < \inf_{t\in\mathbf{I}} \lambda_{min}(\Sigma(t)) \leq \sup_{t\in\mathbf{I}} \lambda_{max}(\Sigma(t)) < \infty$ where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of $\Sigma(t)$ respectively. We assume the time-varying random variables $Y_i(t)$ are uniformly bounded for each $i = 1, \ldots, p$. That is, there exists a positive constant $c > 0$, such that $\max_{i=1,\ldots,p} \sup_{t\in I} |Y_i(t)| \leq c$ almost surely.

C4: The observation times $\{t_{ku}\}_{k=1,u=1}^{n,m}$ are independent and follow a distribution $f_T(t)$ on I, and $f_T(t)$ is absolutely continuous and bounded away from zero and infinity.

C5: The partial correlation function $\boldsymbol{\rho}(\cdot)$ has κ continuous derivatives with $\kappa > 0$.

C6: For any $1 \leq i \neq j \leq p$, there exists a region $E^{ij} \subset I$ such that $\rho^{ij}(t) = 0$ if $t \in E^{ij}$ and $\rho^{ij}(t) \neq 0$ if $t \in (E^{ij})^c$. For simplicity, we further assume that $E^{ij} = [e_1^{ij}, e_2^{ij}]$ is a closed interval and $\left|\rho^{ij}(t)\right| > a\lambda_n$ if $t \in (0, e_1^{ij} - \lambda_n) \cup (e_1^{ij} + \lambda_n, 1)$.

C7: The set of knots denoted as $\Upsilon_n = \{0 = \nu_0 < \nu_1 < \cdots < \nu_{N_n} < \nu_{N_n+1} = 1\}$ is quasi-uniform, i.e., there exists $b > 0$ such that

$$\frac{\max(\nu_{h+1} - \nu_h, h = 0, \cdots, N_n)}{\min(\nu_{h+1} - \nu_h, h = 0, \cdots, N_n)} \leq b.$$

C8: The number of interior knots N_n and tuning parameters λ_n satisfy

$$\lambda_n \to 0, N_n^{(p+2)}\lambda_n \to \infty, \sqrt{N_n/nm}/\lambda_n \to 0.$$

Condition C1 indicates that the weights are bounded away from 0 and infinity. Condition C2 assumes that there exists a consistent estimator for $\sigma^{ii}(t)$, for each $i = 1, \ldots, p$. Similar conditions of C1 and C2 can also be found in Peng et al. (2009), which can be met by estimating $\widehat{\boldsymbol{\sigma}}(t)$ using the residuals of least-square fitting as discussed in the algorithm. Conditions C3, C4, C5 and C7 are standard conditions in the polynomial spline framework, and are required to ensure consistency for spline estimation of the varying coefficient model. Similar conditions can be found in Huang et al. (2002), Xue and Qu (2012), and Xue et al. (2013). Condition C6 is used to separate time regions between zero correlation and nonzero correlation, and thus leads to consistency of the partial correlation estimators.

Theorem 2.4.1. *Under conditions (C1)-(C8), for any* $1 \leq i < j \leq p$*,*

$$\sup_{0<t<1} \left|\widehat{\rho}^{ij}(t) - \rho^{ij}(t)\right| = O_p\left(\sqrt{\frac{N_n^3}{nm} + N_n^{-(q+1)}}\right).$$

Furthermore, for any interior point t *in* W^{ij}*, the probability* $P(\widehat{\rho}^{ij}(t) = 0) \to 1$.

Theorem 2.4.1 shows that, with probability approaching to one, the estimator by minimizing model (2.3) can correctly identify zero estimators in the non-signal time regions, and consistently estimate the partial correlation functions uniformly in the time regions with signal. Therefore, the proposed method can correctly produce a locally sparse network and efficiently model the dynamic change of the network data for sufficiently large data. The proof of the Theorem is provided in the Appendix.

2.5 Simulation

In this section, we conduct simulation studies to illustrate the performance of the proposed method based on the proximal gradient method (SPG) described in Section 3. We first compare the performance of SPG using different degrees of polynomial spline. Then the proposed approach with the best order of B-spline approximation is selected to compare with other existing approaches such as SPACE (Peng et al., 2009), the kernel-based method (Kolar et al., 2010) and the alternating direction method of multipliers (ADMM). Note

that these existing approaches do not apply directly in our dynamic partial correlation networks, therefore we provide the adaptive versions of the existing approaches for our setting.

We generate dynamic networks assuming that the network structures have disjointed blocks. Networks with disjointed blocks are quite common in many applications where networks are only connected within blocks, but are not associated with each other between blocks. See examples from Girvan and Newman (2002) and Valencia et al. (2009) on brain and biological functions, gene expressions, social, sports and computer network associations. In the following simulations, the number of disjointed blocks is 3. To generate the concentration matrix at time t, we first create an initial matrix $(A_t)_{p\times p}$ with three blocks as

$$\begin{pmatrix} A_t^1 & & \\ & A_t^2 & \\ & & A_t^3 \end{pmatrix},$$

where the diagonal entries for each block $A_t^k, k = 1, 2, 3$ are all set to be one, and the off-diagonal entries of A^k are set to be $f_k(t)U$ with U following the Bernoulli distribution of the probability ω being 1. Here the functions $f_k(t), k = 1, 2, 3$ are defined as follows:

$$f_1(t) = \begin{cases} 5(t-0.5)^2 - 0.125, & \text{if} \quad 1 \le t \le 0.342 \\ 0, & \text{if} \quad 0.342 < t \le 0.658 \\ -5(t-0.5)^2 + 0.125, & \text{if} \quad 0.658 < t \le 1 \end{cases}, \quad f_2(t) = \begin{cases} -3t + 0.9, & \text{if} \quad 0 \le t \le 0.3 \\ 0, & \text{if} \quad 0.3 < t \le 0.7 \\ 3t - 2.1, & \text{if} \quad 0.7 < t \le 1 \end{cases},$$

$$f_3(t) = \begin{cases} -22.5(t-0.5)^2 + 0.9, & \text{if} \quad 0.3 \le t \le 0.7 \\ 0, & \text{if} \quad \text{o.w.} \end{cases}.$$

The plots of $f_k(t)$ are provided in Figure 2.1. The parameter ω can be used to control the number of nonzero entries of each block in the concentration matrix, where the networks are sparse if ω is small. Here we consider moderate strength of associations among nodes in the network, and the probability $\omega = 0.8$ is selected in our simulations. In addition, we also follow a similar strategy as in Peng et al. (2009) to make sure that the simulated covariance matrix is positive definite.

We first compare the performances of local signal selection using the linear, quadratic and cubic spline approximations in the simulation studies. Various network sizes of $p = 18, 54$ and 108, and time length

$T = 50$ are considered here. The sample size is chosen as $n = 200$.

Table 2.1 provides the comparisons of model selection performance of the smoothing proximal gradient method (SPG) in detecting the true time-varying signals under different orders of spline approximations. Here correct-fitting (C), over-fitting (O) and under-fitting (U) are calculated as the percentages of time-points out of T equally-spaced time-points at interval $[0, 1]$ where both true signal and non-signal points are identified correctly; true non-signal points are misclassified as signal ones; and true signal points are not selected, respectively. Table 2.1 indicates that the SPG with linear spline tends to select correct edges with the highest frequency, compared to the quadratic and cubic splines. When the network size increases from 18 to 108, the percentage of selecting correct associations decreases about 9.8% in the linear spline approach. When the network size is 108, the percentage of selecting correct edges based on the SPG is about 83.0% for the linear spline approach. This simulation indicates that the SPG with linear spline has the best performance in detecting the local changes of network associations, compared to the quadratic and cubic splines.

We further compare the performance of the proposed model with SPACE, the kernel-based method (KEN) and the ADMM approaches. The SPACE method which is developed for networks at one time point, does not consider the correlation among networks at different time points. The kernel-based method could not be applied to estimate the partial correlations. To the best of our knowledge, the ADMM cannot be applied directly on dealing with overlapping groups for the penalized loss function. Therefore, we develop KEN for handling partial correlations and the ADMM for overlapping group LASSO. Both methods are described below.

The description of KEN is provided as follows. At each time point t_u, we use a kernel smoothing approach to estimate the partial correlation $\boldsymbol{\rho}(t_u)$. The estimator $\widehat{\boldsymbol{\rho}}(t_u)$ is obtained by minimizing the following objective function:

$$L(\boldsymbol{\sigma}(t_u), \boldsymbol{\rho}(t_u), \mathbf{y}) = \frac{1}{2} \sum_{k=1}^{n} \sum_{i=1}^{p} \sum_{u=1}^{m} w_u \left(y_i^k(t_u) - \sum_{j \neq i}^{p} \rho^{ij}(t_u) \sqrt{\frac{\sigma^{jj}(t_u)}{\sigma^{ii}(t_u)}} y_j^k(t_u) \right)^2 + \lambda \|\boldsymbol{\rho}(t_u)\|,$$

where the weights w_u are defined as $w_u = \frac{K_h(t-t_u)}{\sum_{u'=1}^{m} K_h(t_{u'}-t_u)}$ with nonnegative and symmetric kernel function $K_h(\cdot) = K(\cdot/h)$. The bandwidth h for the kernel function can be controlled by users. Kolar et al. (2010) proposed a coordinate-wise descent method. We follow the same algorithm in the simulation. The algorithm

is summarized in Algorithm 2.

Algorithm 2 The algorithm for the kernel based model

Input: Set initial values $\boldsymbol{\rho}(t_u)^{(0)}$
Output: $\widehat{\boldsymbol{\rho}}(t_u)$.
1: Update each element $\rho^{ij(l+1)}(t_u)$ in $\boldsymbol{\rho}(t_u)$ by calculating the solution to the following optimization $\min_{\rho^{ij}(t_u)} L(\boldsymbol{\sigma}^{(l)}, \rho^{12(l+1)}(t_u), \cdots, \rho^{(i,j-1)(l+1)}, \rho^{ij}(t_u), \rho^{(i,j+1)(l)}(t_u), \cdots, \rho^{(p-1,p)(l)}(t_u), \mathbf{y})$
2: Update $\boldsymbol{\sigma}^{(l+1)}(t_u) = \frac{1}{1/n\|\mathbf{y}(t_u)-\widehat{\mathbf{y}}(t_u)\|^2}$
3: Return to Step 1 until convergence.

For the ADMM algorithm, we apply the same strategy in the SPG algorithm to approximate $g_0(\boldsymbol{\beta})$ by $g_\mu(\boldsymbol{\beta})$ as defined in (2.6). To get the solution for the model with overlapping group adaptive LASSO, the problem can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{\beta},\boldsymbol{\beta}^*} \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\boldsymbol{\beta}\|^2 + g_\mu(\boldsymbol{\beta}^*), \\ &s.t. \quad \boldsymbol{\beta} = \boldsymbol{\beta}^*. \end{aligned} \tag{2.10}$$

Then the scaled augmented Lagrangian (Boyd et al. 2010) for this problem is given by

$$L_\rho = \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\boldsymbol{\beta}\|^2 + g_\mu(\boldsymbol{\beta}^*) + \frac{\kappa}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}^* + \boldsymbol{\eta}\|_2^2, \tag{2.11}$$

where $\boldsymbol{\eta}$ are dual variables and κ is a scalar and can be preset. Therefore, the ADMM algorithm under (2.11) leads to three iteration steps for $\boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\eta}$. The details are provided at the $(l+1)$-th step,

$$\begin{aligned} \boldsymbol{\beta}^{(l+1)} &= arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\boldsymbol{\beta}\|^2 + \frac{\kappa}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}^{*(l)} + \boldsymbol{\eta}^{(l)}\|_2^2, \\ \boldsymbol{\beta}^{*(l+1)} &= arg\min_{\boldsymbol{\beta}} g_\mu(\boldsymbol{\beta}^*) + \frac{\kappa}{2}\|\boldsymbol{\beta}^{(l+1)} - \boldsymbol{\beta}^* + \boldsymbol{\eta}^{(l)}\|_2^2, \\ \boldsymbol{\eta}^{(l+1)} &= \boldsymbol{\eta}^{(l)} + \left(\boldsymbol{\beta}^{(l+1)} - \boldsymbol{\beta}^{*(l+1)}\right). \end{aligned} \tag{2.12}$$

The first minimization problem in (2.12) is easy to solve since the objective function is quadratic. The function $g_\mu(\boldsymbol{\beta}^*)$ in the second minimization is a smoothing function and thus can be approximated by the Taylor expansion at $\boldsymbol{\beta}^{*(l)}$, i.e. $g_\mu(\boldsymbol{\beta}^*) \approx g_\mu(\boldsymbol{\beta}^{*(l)}) + 1/2\nabla g_\mu(\boldsymbol{\beta}^{*(l)})(\boldsymbol{\beta}^* - \boldsymbol{\beta}^{*(l)})$. Thus $\nabla g_\mu(\boldsymbol{\beta}^*) \approx \nabla g_\mu(\boldsymbol{\beta}^{*(l)})/2 = C'\boldsymbol{\alpha}^{*(l)}/2$, where $\boldsymbol{\alpha}^{*(l)}$ can be calculated by (2.7) corresponding to $\boldsymbol{\beta}^{*(l)}$. So the solution $\boldsymbol{\beta}^{*(l+1)} = \boldsymbol{\beta}^{(l+1)} + \boldsymbol{\eta}^{(l)} - \lambda C'\boldsymbol{\alpha}^{*(l)}/(2\kappa)$. The algorithm is summarized in Algorithm 3.

Algorithm 3 Alternating direction method of multipliers for estimating partial correlation networks

Input: Set desired tolerance levels ϵ, ϵ^* and scalar κ, obtain $\mu = \epsilon/D$ and matrix C; Initialize the parameters $\boldsymbol{\beta}, \boldsymbol{\sigma}$ as $\boldsymbol{\beta}^{(0)}$ and $\boldsymbol{\sigma}^{(0)}$.

Output: $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\sigma}}$.

1: Compute $\boldsymbol{\alpha}^{*(l)}$ according to (2.7);

2: Obtain $\boldsymbol{\beta}^{(l+1)}, \boldsymbol{\beta}^{*(l+1)}, \boldsymbol{\eta}^{(l+1)}$ according to (2.12), and set the elements in $\boldsymbol{\beta}^{(l+1)}$ less than ϵ^* as zero;

3: Update $\boldsymbol{\sigma}^{(l+1)}$ and $\mathbf{w}^{(l+1)}$ by calculating (2.9).

4: Return to Step 1 if $\left\|\boldsymbol{\beta}^{(l+1)} - \boldsymbol{\beta}^{*(l+1)}\right\| > \epsilon$.

Both KEN and ADMM require to calculate the inverse of matrix. When the network size is large, the dimension of the matrix could be extremely large, then the computation of the matrix inversion becomes infeasible.

We compare the performance of these four methods under the network sizes of 18, 54 and 108 with sample size $n = 200$ and time length $T = 50$ based on 100 simulations. Since Table 2.1 indicates that SPG with the linear spline outperforms the quadratic and cubic splines, we use the linear spline for the SPG in the following comparison.

Table 2.2 provides the model selection performance of the SPG, ADMM, SPACE and KEN under various network sizes. The SPG and ADMM have similar performance and are the best in the sense of selecting the true model with the highest frequency when the network size is 18 or 54. When the network size increases to 108, the rate of selecting the correct model for SPACE goes down to 51.2%. This is probably due to the overfitting problem of SPACE. In comparison, the SPG still has a correct-fitting rate of 83.0%. However, neither ADMM nor KEN is feasible due to the problem of high-dimensional matrix inversion for the ADMM approach and a highly intensive computing procedure for the kernel method. We tried the SparseM package in R, the Eigen package and SparseLib++ in C++ which are designed for high-dimensional matrix operations, but none of these are able to solve these problems.

Table 2.2 also provides the average computing time per simulation run for each method. We run simulations on a cluster server running a Linux system equipped with 2.67GHz CPU and 48GB memory. The computing time increases significantly as the dimension of matrix operations increases exponentially from 10^2 to 10^5 when the network size increases from 18 to 108. SPACE is the fastest among all the four methods since it does not utilize neighboring information of the time-points observed from the same subject. KEN is the slowest of all since it requires updating neighborhood information for each nonparametric coefficient estimation at each iteration. The computing time for the SPG algorithm ranges from 27.46 seconds to 1.04

hours per run when the network size increases from 18 to 108. We are not able to record the time for KEN and ADMM when $p = 108$ due to infeasibility issues for these two approaches. In summary, SPG is the best among all four methods if we consider the computational feasibility and correct-fitting performance.

We also compare the number of edges correctly identified by these four methods with a moving tuning parameter. Figure 2.2 shows that the BIC reaches the minimum if the tuning parameter is selected as $\lambda = 0.145$ when the network size is 18, the sample size is 200 and the number of time-points is 50. In addition, Figure 2.3 indicates that both SPG and ADMM have the highest ratio of correctly identified edges over total detected edges for almost any given tuning parameter. For example, when the number of total detected edges equals the number of true edges (1876), the SPG and ADMM are able to identify 1444 and 1441 correct edges, respectively, whereas KEN detects 1345 correct edges, and SPACE detects only 1243 correct edges.

2.6 Application

In this section, we analyze the data obtained from the attention deficit hyperactivity disorder (ADHD) study. ADHD is a mental disorder found in children and adolescents, and common symptoms include being easily distracted, impulsiveness, and restlessness. To better understand how ADHD patients' brains function and react to different stimulants, we focus on identifying associations and interactions among different regions of interest (ROI) of the brain. One distinct feature of ADHD patients is to have high variability of brain function over time; therefore, it is scientifically important to identify the dynamic changes of association among different regions of interest of the brain to locate the ADHD pathology.

The ADHD-200 samples contain fMRI data which are repeatedly measured over time. We choose 78 patients from the ADHD-200 test samples obtained from the Oregon Health and Science University. The fMRI data are processed by an automated anatomical labeling software package and digital atlas for the human brain, and are collected from 116 regions of interest of the brain over 74 time-points, with a few seconds between each two time-points. We apply only the SPG and SPACE methods to this data, since the ADMM and KEN approaches are not able to handle the network size of 116. The number of connections among ROIs at each time point is shown in Table 2.3. Note that SPACE identifies more than 2000 connections at most of the time-points, in contrast to the SPG method which identifies at most 78 connections at each time point. The over-identifying problem of SPACE makes it difficult to select any

useful connections. In the following, we provide data analysis and graphical illustration based on the SPG only.

Figure 2.4 illustrates the associations and connections of 116 regions of interest formulated as a network at time points $t = 1, 10, 20, 50, 60$ and 74. Note that each region of interest in the brain is represented as nodes or vertices with either green or pink color, and the associations among nodes are connected with blue lines. The color pink of a node represents five or more associations with other regions of interest, and the color green of a node indicates less than five associations with other regions of interest.

We are able to identify the dynamic changes of associations among the 116 regions of interest over time. Specifically, the ADHD patients experience three distinct periods of brain activities during the test. The number of connections at each time point is shown in Table 2.3. At the beginning of the test, the ADHD patients' brains are active. However, when the test proceeds further, the ADHD patients' brains are mostly in a resting state, since there are only a few connections among the 116 regions of interest, with most of the regions of interest containing less than 36 connections. This is possibly due to the fact that patients are less disturbed in the middle of the experiment, since there is actually no stimulus imposed on their brains. In the later stage of the test when $t > 57$, patients' brains again have more connections among regions of interest, as patients might anticipate something happening by the end of the experiment. These phenomena are also indicated in Figure 2.4, showing that there are more associations among regions of interest between $t = 1$ and $t = 10$, and $t = 60$ and $t = 74$, but fewer brain activities between $t = 20$ and $t = 50$.

Table 2.4 confirms our findings and indicates that there are few associations between $t = 20$ and $t = 55$, with only 2 vertices having three or more connections during this period. However, between time points $t = 1$ and 19, there are 15 vertices containing three or more connections among regions of interest, and between $t = 56$ and 74, there are 14 vertices having three or more connections. The corresponding names of those ROIs with three or more connections and their gray levels are provided in Table 2.5 (gray level is defined as the volume of gray matter in a ROI, and gray matter distinguished from white matter consists of cell bodies, neuropil, glial cells and capillaries). These findings could be helpful in studying ADHD patients' brain function over time, even without any stimulation.

Figure 2.5(a) illustrates the locations of certain ROIs in the brain using an automated anatomical labeling (AAL) software package. Here different ROIs are marked as different colors. Note that most of the ROIs have counterparts located on the opposite side of the brain, and are marked as the same color. For example,

the cyan blue color is used for both Temporal_Mid_L and Temporal_Mid_R in Figure 2.5(a). However, these counterpart ROIs are not necessarily associated with each other. Figure 2.5(a) shows 50 out of the 116 ROIs, and Figure 2.5(b) provides a partial network of the ROIs to illustrate the associations based on the selected 15 ROIs. The partial network is quite sparse. For better visualization of the associated network, Figure 2.5(c) also provides the associated names of the 15 selected ROIs.

In addition, we also provide an animated video in the file "ADHD.mp4" to illustrate the dynamic changes for 116 regions of interest of the brain over 74 time points. The colors of the nodes in the video ranges from red to purple, blue and green, which reflects the level of connections with other ROI over the entire time period. The red nodes are the most active ROIs with the number of connections ranging from 30 to 36; the purple nodes have a number of connections from 18 to 29; while the blue and green nodes have moderate to few associations with other ROIs of the brain, ranging between 8 to 17 and 0 to 7, respectively.

2.7 Discussion

The time-varying network model is powerful for identifying time-evolving associations for brain and biological functions, gene networks, social networks and environmental networks over time. In this chapter, we develop a local varying-coefficient model to effectively quantify and detect dynamic changes in network associations and interactions. One distinct feature of the proposed approach is that we are able to incorporate local features of a nonparametric function, and provide local-signal detection and estimation simultaneously for time-varying network data.

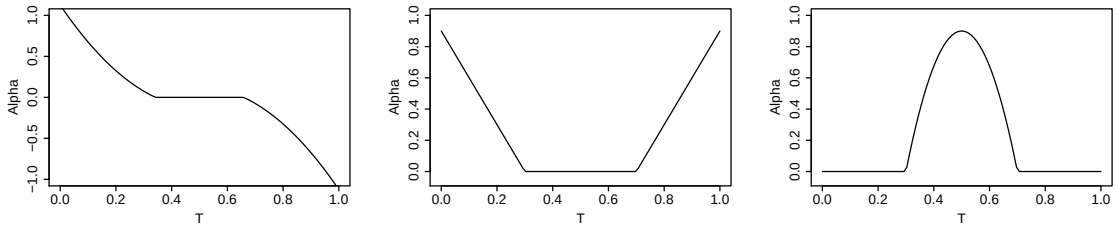
We propose a piecewise penalized loss function such that the coefficients associated with the varying-coefficient model at the local region are shrunk to zero if the magnitude of the grouped coefficients is sufficiently small. This has significant advantages over the traditional varying-coefficient model selection approach without incorporating local features, especially for time-varying network data, since the network associations could be quite volatile over time, and local-region estimation and signal detection are of more scientific interest than global-feature selection. Our simulation studies and data application to the ADHD study indicate that the proposed method is quite effective at capturing the local features of the time-varying network data.

However, it is quite computationally challenging to develop highly computationally intensive algorithms in order to achieve local-sparsity properties in estimation and local-signal detection. The group penalization

strategy involves overlapping parameters among different groups, which makes the optimization process extremely challenging when the network size is large. To overcome these difficulties, we develop a smoothing proximal gradient method, which does not require inverting the large-dimensional matrix. The developed algorithm has significant computational advantages in increasing computational speed and efficiency. Most importantly, the proposed smoothing proximal gradient algorithm is able to analyze a relatively large size of network data within a reasonable time frame. We also compare the ADMM and kernel-based algorithms which require inverting a large-dimensional matrix, and therefore cannot feasibly estimate large size network data.

Theoretically, we show that the proposed method achieves model selection consistency in local regions, and provides a uniform rate of convergence for local-signal coefficient estimators. Scientifically, it is important to detect dynamic changes in networks, as identifying the associations of biological functionalities over time can help us to better understand the mechanisms of network change.

2.8 Figures and Tables



(a) $f_1(t)$ (b) $f_2(t)$ (c) $f_3(t)$

Figure 2.1: The function $f(t)$ at time interval $t \in [0, 1]$

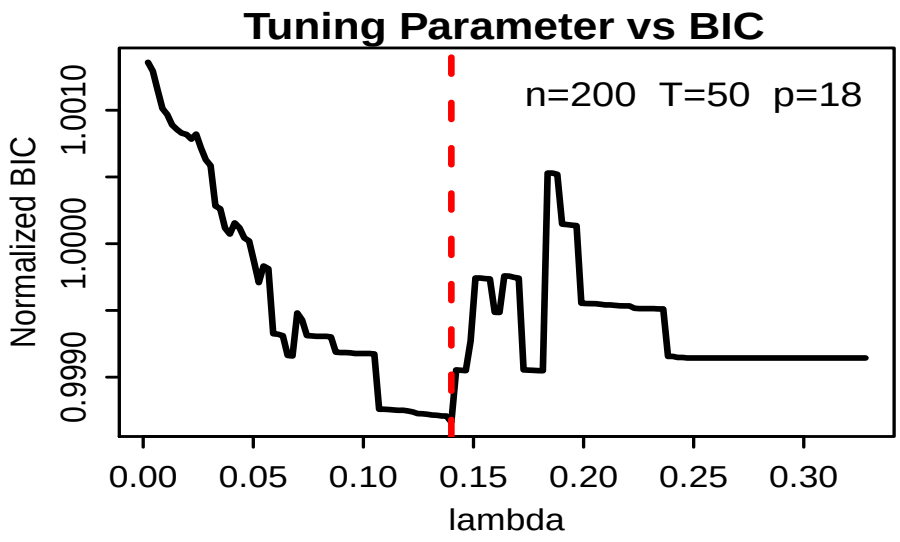


Figure 2.2: The plot of moving tuning parameter versus the BIC for the SPG algorithm when $n = 200$, $T = 50$ and $p = 18$.

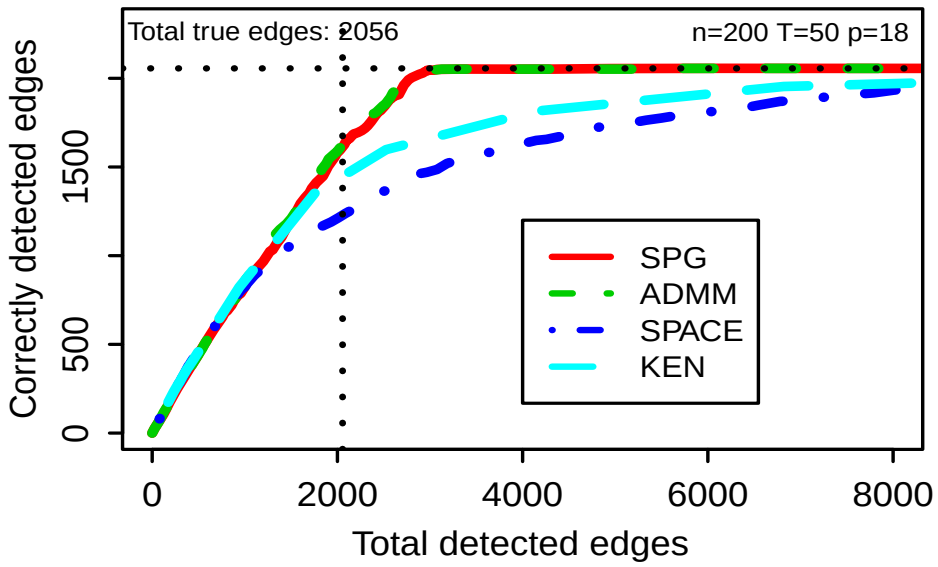


Figure 2.3: Correctly detected edges versus the total detected edges using the four methods.

Table 2.1: Model selection performance of the smoothing proximal gradient method (SPG) for three-block disjointed networks with the number of time-points $T = 50$ and sample size 200 based on 100 simulation runs.

	Network size	C	O	U	Time
Linear	p=18	0.920	0.056	0.024	27.46
	p=54	0.859	0.071	0.070	467.71
	p=108	0.830	0.023	0.147	3726.48
Quadratic	p=18	0.887	0.063	0.050	41.87
	p=54	0.838	0.073	0.089	670.89
	p=108	0.799	0.088	0.113	7510.36
Cubic	p=18	0.860	0.091	0.049	60.13
	p=54	0.791	0.099	0.110	1192.30
	p=108	0.764	0.113	0.123	14102.38

Table 2.2: Model selection performance of SPG, ADMM, SPACE and KEN for three-block disjointed networks with the number of time-points $T = 50$ and sample size 200 based on 100 simulation runs.

Network size	Methods	C	O	U	Time per run (seconds)
p=18	SPG	0.920	0.056	0.024	27.46
	ADMM	0.920	0.055	0.025	10.53
	SPACE	0.907	0.011	0.082	1.33
	KEN	0.909	0.065	0.026	109.35
p=54	SPG	0.859	0.071	0.070	467.71
	ADMM	0.860	0.068	0.072	286.87
	SPACE	0.691	0.220	0.089	36.39
	KEN	0.786	0.123	0.091	14328.74
p=108	SPG	0.830	0.023	0.147	3726.48
	ADMM	NA	NA	NA	NA
	SPACE	0.512	0.418	0.070	349.98
	KEN	NA	NA	NA	NA

Table 2.3: Number of associations identified by SPG and SPACE from time-points 1 to 74.

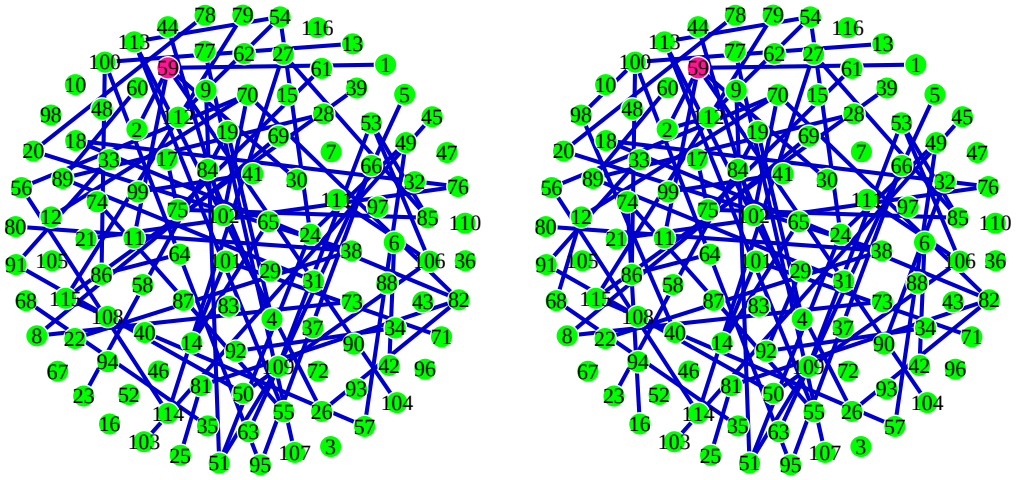
Method	Number of associations from 1 to74
SPG	70 77 77 77 76 77 77 76 77 77 77 77 77 77 78 77 77 77 77
	35 36 35 35 35 35 35 35 35 35 35 35 35 35 36 35 35 35
	34 34 34 34 34 34 34 34 34 34 34 34 34 35 34 34 34 34
	76 76 76 75 76 76 76 76 76 77 76 76 76 76 76 76 76 66
SPACE	3024 3102 3257 2059 2691 2839 3278 2962 3111 3080 2926 2946 2833 3079 3171
	3156 3067 2932 3129 2955 2934 3025 1998 3076 3130 3278 3230 2786 3176 2828
	2979 2981 3057 3045 2695 3070 2665 3120 3090 2916 3054 2982 2670 3038 2836
	2969 3006 3154 2756 3056 3179 3024 2975 2974 3067 3273 1956 3157 2707 3132
	3115 2948 2799 2967 3028 3059 2969 3165 3089 3039 3109 2950 3103 2779

Table 2.4: ROIs with 5 or more associations identified by SPG from time-points 1 to 74.

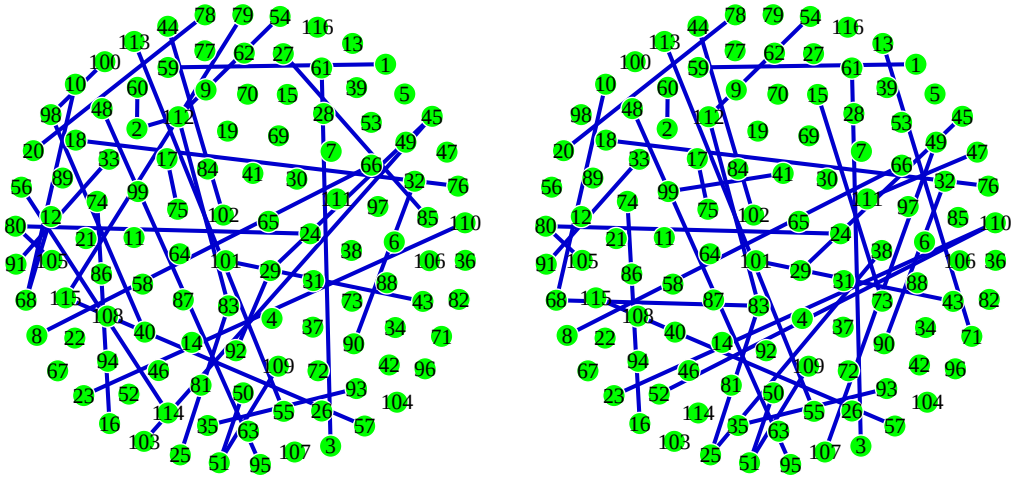
Time(t)	ROIs with 3 or more associations	Total
1-19	24 38 51 53 54 59 70 75 82 85 89 100 106 113 115	15
20-55	83 112	2
58-74	5 25 32 52 63 71 76 81 82 90 95 100 110 116	14

Table 2.5: Name list of ROIs with 3 or more associations identified by SPG

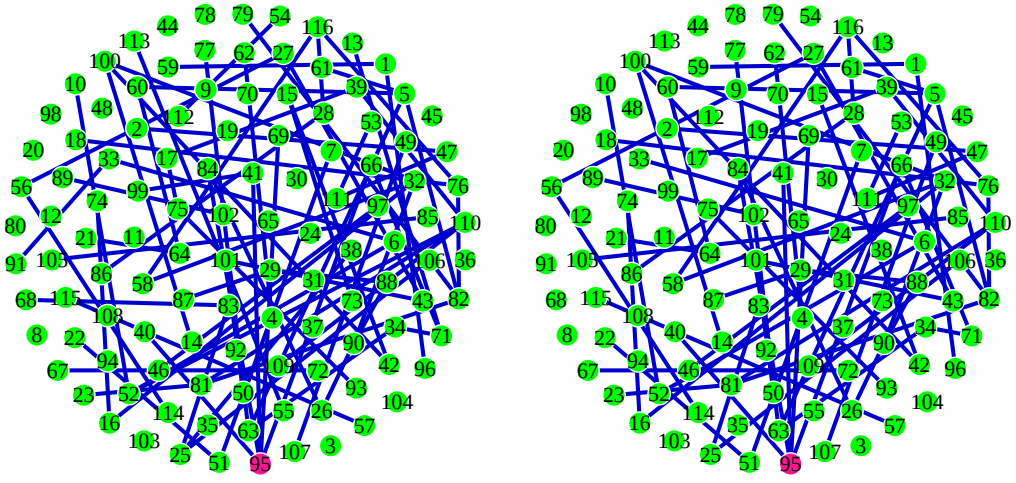
Number	Name	Gray level
5	Frontal_Sup_Orb_L	2111
24	Frontal_Sup_Medial_R	2602
25	Frontal_Mid_Orb_L	2611
32	Cingulum_Ant_R	4002
38	Hippocampus_R	4102
51	Occipital_Mid_L	5201
52	Occipital_Mid_R	5202
54	Occipital_Inf_R	5302
59	Parietal_Sup_L	6101
63	SupraMarginal_L	6211
70	Paracentral_Lobule_R	6402
71	Caudate_L	7001
75	Pallidum_L	7021
76	Pallidum_R	7022
82	Temporal_Sup_R	8112
83	Temporal_Pole_Sup_L	8121
85	Temporal_Mid_L	8201
89	Temporal_Inf_L	8301
90	Temporal_Inf_R	8302
95	Cerebelum_3_L	9021
100	Cerebelum_6_R	9042
106	Cerebelum_9_R	9072
110	Vermis_3	9110
112	Vermis_6	9130
113	Vermis_7	9140
115	Vermis_9	9160
116	Vermis_10	9170



(a) $t = 1$ (b) $t = 10$

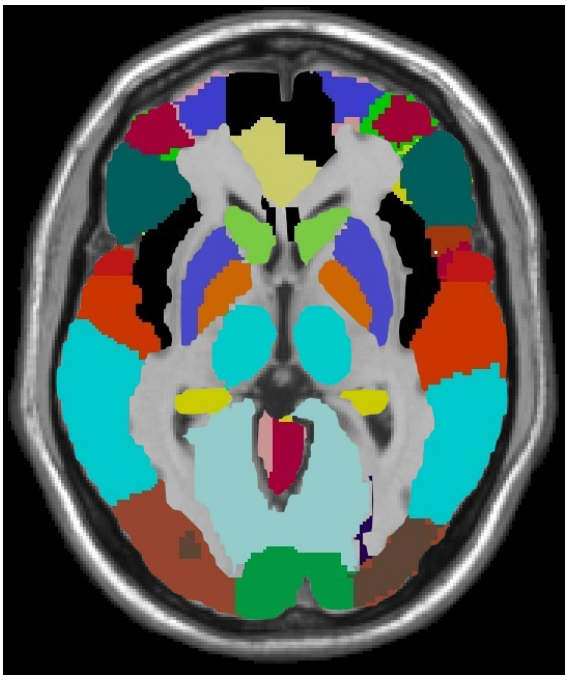


(c) $t = 20$ (d) $t = 50$

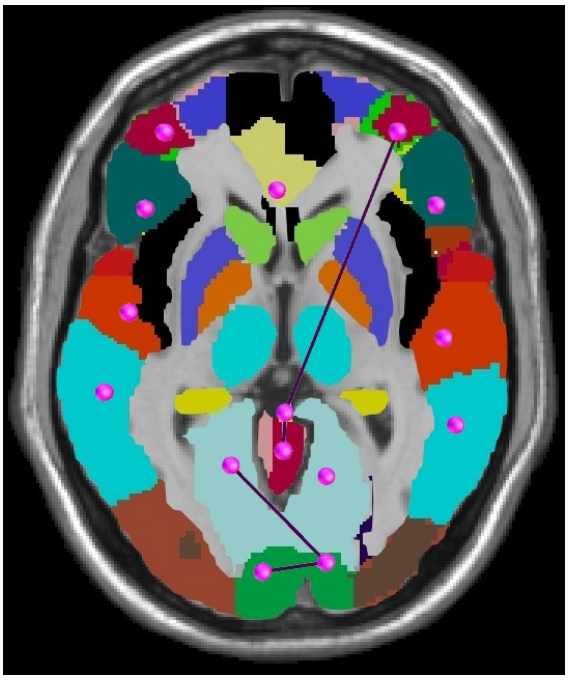


(e) $t = 60$ (f) $t = 74$

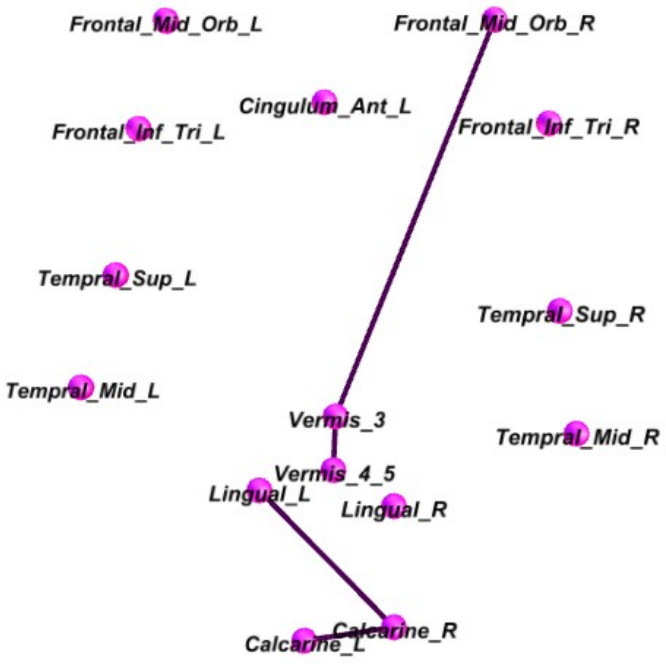
Figure 2.4: Estimation of brain networks of ADHD-200 data at time-points $t = 1, 10, 20, 50, 60$ and 74.



(a) AAL ROIs' locations in brain



(b) The partial network of ROIs



(c) ROIs' names in the partial network

Figure 2.5: Illustration of AAL ROIs in the brain and its networks

Chapter 3

Words segmentation in Chinese language processing

3.1 Introduction

In this chapter, we focus on problems arising from Chinese natural language processing, and in particular we address problems of word segmentation. This is an important area and quite timely, since the Chinese language has become the second most popular language among all internet users. In 2000, there were about 22.5 million Chinese internet users. However, after rapid growth in the last decade, there were over 530 million internet users in 2012 writing text documents in Chinese, consisting of 22.4% of all internet users, compared to 29.4% in English. The online sales and entertainment businesses also promote the popularity of Chinese in the digital world. For example, Amazon.cn data (Zhang et al., 2009) consists of 5×10^5 Chinese reviews on various products.

Studies on Chinese word segmentation are still quite limited. Existing methods for Chinese segmentation are essentially based on either characters, words or their hybrids (Sun, 2010; Gao et al., 2005). Teahan et al. (2000) proposed a word-based method by applying forward or backward maximum matching strategies. Their method requires an existing corpus as a reference to identify exact character sequences and then segment character by character sequentially, through processing documents in either a forward or backward direction. This method is also developed by Chen et al. (1999) and Nie et al. (1994). One obvious drawback of this approach is that the segmentation heavily relies on the coverage of the given corpus, and thus is not designed for identifying new words which are not in the corpus.

The main idea of character-based methods is that the segmentation is considered as a sequence of labeling problems (Xue, 2003). That is, the location of characters in a word is labeled through statistical modeling such as conditional Gaussian random fields (CRF; Lafferty et al., 2001; Chang et al., 2008) or structured support vector machine (SVM^{struct}; Tsochantaridis et al., 2005) based on hinge loss. Xue and Shen (2003) proposed a maximum entropy approach which combines both character-based and word-based methods.

Specifically, their idea is to integrate forward/backward maximum matching with statistical or machine-learning models to improve segmentation performance. Sun and Xu (2011) proposed a unified approach for a learning segmentation model from both training and test datasets to improve segmentation accuracy. These approaches, however, suffer a major drawback, in that they do not utilize available linguistic information which can enhance the segmentation (Gao et al., 2005). In addition, some current segmenters treat word segmentation and new word identification as two separate processes (Chen, 2003; Wu and Jiang, 2000), which may lead to inconsistent results in segmentation. Other methods of segmentation are embedded into other processing procedures such as translation, to serve a specific purpose, for example, Chinese-English translation (Xu et al., 2008; Zhang et al., 2008). Those methods unified with other processing approaches have not been proposed for general use.

Most existing character-based and word-based methods have a drawback in that they are incapable of identifying new words not appearing in training documents. Although in some situations character-based methods tend to outperform word-based methods in terms of segmentation accuracy (Wang et al., 2010), the enormous variety of different permutations of Chinese characters makes the computation of segmentation intractable. To overcome these problems, we propose a model which incorporates linguistic rules to detect new words and semantic associations in context, in the meantime utilizing machine-learning technology to reduce computational complexity.

This Chapter is organized as follows. Section 2 proposes the linguistically-embedded learning model. Section 3 designs a computational strategy to meet computational challenges in solving large-scale optimization for the proposed model. In section 4, the proposed method is illustrated by applying to the Peking university corpus in the SIGHAN Bakeoff. The final section provides concluding remarks and a brief discussion.

3.2 Words segmentation

Segmentation in Chinese language processing is a crucial step because there is no boundary delimiter among consecutive Chinese words. Moreover, most Chinese characters can appear in different position within different words. Table 3.1 shows an example of the Chinese character 发 "happen" may occur in three different positions. This problem makes it impossible to simply determine word boundaries just through detecting certain types of Chinese characters, even though the number of characters is finite. As long

as a character can occur in different positions within different words, itself cannot be used to determine word boundaries since its position could vary. In fact, a character appearing in multiple positions leads to ambiguities of various kinds. For instance, a segmenter could segment the sentence 网球拍卖完了 as 网球拍/卖完/了 "Tennis racquets are sold out," while can also segment the sentence as 网球/拍卖完/了 "Tennis ball(s) is/are auctioned." The segmenter therefore would face a dilemma since different segmentation may lead to different meanings. The ambiguity of the Chinese language makes extremely challenging since mechanical methods such as tabulating frequencies of key words in context can not be applied to collecting text information in Chinese language processing.

In this section, we introduce the proposed model by incorporating linguistic rules into the model for Chinese segmentation. Specifically, two key components will be integrated with a new class of surrogate segmentation loss functions in the model: (1) linguistically meaningful features through higher-order n-gram templates; (2) linguistically-embedded constraints. In summary, we construct possible linguistical features using different order gram templates to build the candidate set of features, and select significant features from the candidate set based on the modeling to utilize the existing linguistic rules.

The proposed model has two major advantages. One is that the proposed model is able to identify new words which are not in the sample corpus. The other is that estimation complexity incorporating linguistic rules can be substantially minimized and higher segmentation accuracy can be achieved. The model is described as follows.

3.2.1 Linguistically-embedded learning framework

In this section, we first introduce character-based framework, and then illustrate how to incorporate character-based framework into the proposed model. Let T be the number of characters in one sentence, and the corresponding sentence is denoted as $\mathbf{c} = c_1 \dots c_T$ and the set of its segmentation locators as $\mathbf{s} = s_1 \dots s_T$, where each character c_t corresponds to a segmentation locator s_t, and $s_t \in \mathcal{S}$. Meng et al. (2010) suggests that a simple 4-tag set $\mathcal{S} = \{B, M, E, S\}$ is sufficient for unique determination and segmentation, where B, M, E, and S denote the beginning, middle and the end of a word, and a single-character word, respectively. For instance, consider the sentence $\mathbf{c} =$ 我们将创造美好的新世纪 "we will create a bright future" with $T = 11$ characters, where $c_1 =$ 我, $c_2 =$ 们, ..., $c_{11} =$ 纪, and $\mathbf{s} = BESBEBESBME$. So the linguistically meaningful segmentation is: 我们 将 创造 美好 的 新世纪. Segmentation accura-

cy and computation efficiency are two important and desirable properties we intend to achieve. The 4-tag segmentation rule is effective to achieve both of these properties.

To identify the segmentation locators for each Chinese sentence, we construct the segmentation model based on training data $(\mathbf{c}_i, \mathbf{s}_i)_{i=1}^n$, mapping from $f : \mathcal{C}^T \to \mathcal{S}^T$, where $\mathbf{c}_i$ and $\mathbf{s}_i$ are the character and locator vectors in the i-th sentence and n is the number of sentences. For instance, $f(\{$ 我, 们$\}) = \{B, E\}$. The model is built on minimizing segmentation error calculated by $E\{I(\mathbf{S} \neq f(\mathbf{C}))\}$, where I is an indicator, or equivalently, $\hat{f} = \arg\min_f \sum_{i=1}^n L(f, \mathbf{c}_i, \mathbf{s}_i)$, where $L(f, \mathbf{c}, \mathbf{s})$ is a surrogate loss function for $I(\mathbf{S} \neq f(\mathbf{C}))$. For example, $L(f, \mathbf{c}, \mathbf{s})$ can be $0/1$ loss, i.e. $L(f, \mathbf{c}, \mathbf{s}) = 0$ if $\mathbf{s} = f(\mathbf{c})$, and $L(f, \mathbf{c}, \mathbf{s}) = 0$ otherwise. However, f is an ultra-high dimensional function when the size of a Chinese document is large. To reduce the dimensionality, we construct a weighted function for each sentence $f(\mathbf{c}_i, \mathbf{s}_i) = \sum_{k=1}^K \sum_{t=1}^T \lambda_k f_k(\mathbf{s}, \mathbf{c}, t)$, where $f_k(\mathbf{s}, \mathbf{c}, t)$ is a linguistically meaningful feature, K is the number of features and $\Lambda = (\lambda_1, \cdots, \lambda_K)^T$ are the weights which describe the relative importance of features in $f(\mathbf{c}, \mathbf{s})$. The construction of $f_k(\mathbf{s}, \mathbf{c}, t)$ is introduced in the next section. To obtain weights Λ, we minimize the following cost function

$$\underset{\Lambda}{\operatorname{argmin}} \sum_{i=1}^n L\Big(\sum_{t=1}^T \sum_{k=1}^K \lambda_k f_k(\mathbf{s}_i, \mathbf{c}_i, t) \Big) + \eta \sum_{k=1}^K J(\lambda_k) \text{ subject to } \lambda_k \geq \lambda_j \text{ for all } (k, j) \in \mathcal{I}, \tag{3.1}$$

where the index set $\mathcal{I}$ contains all pairs of features with importance ordering based on the linguistic rules, $J(\Lambda)$ is a regularizer such as LASSO (Tibshirani, 1996), SCAD (Fan and Li, 2001) and truncated L_1 penalty (Shen et al., 2012) to ensure the model sparsity and η is a tuning parameter. We obtain $\widehat{f}(\mathbf{c}, \mathbf{s})$ with selected important features through minimizing (3.1), and the sequence $\mathbf{c}$ can be segmented by $\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \widehat{f}(\mathbf{c}, \mathbf{s})$. That is to optimize the segmentation by maximizing the value of the linear combination of those important features in a document. How to build linguistically-embedded features $f_k(\mathbf{s}, \mathbf{c}, t)$'s is illustrated in the following section.

3.2.2 Construction of linguistically-embedded features

We incorporate linguistic language rules into features $f_k(\mathbf{s}, \mathbf{c}, t)$'s through effective word categorization for Chinese words. This categorization method is first introduced in Gao et al. (2005) with five categories: lexical words (LW), morphologically derived words (MDW), factoids (FT), named entities (NE) and new

words (NW). The taxonomy in Chinese words is summarized in Table 3.2.

We first define N-gram templates and apply the templates to construct binary features. The unigram (i.e. 1-gram) templates contain $I(s(0) = s_t, c(-1) = c_{t-1})$, $I(s(0) = s_t, c(0) = c_t)$ and $I(s(0) = s_t, c(+1) = c_{t+1})$, and bigram (i.e. 2-gram) templates include $I(s(0) = s_t, c(-1) = c_{t-1}, c(0) = c_t)$ and $I(s(0) = s_t, c(0) = c_t, c(+1) = c_{t+1})$, where $c(-1)$, $c(0)$, $c(+1)$ and $s(0)$ denote the the previous, current and next characters, and the tag for the current character, respectively. The unigram templates contain single character's information on the previous, current or next characters given the current segmentation locator, while the bigram templates include two consecutive characters' information through combining the previous or next character, with the current character. To illustrate this, for the previous example, $\mathbf{c} =$ 我们将创造美好的新世纪, $\mathbf{s} = \{BESBEBESBME\}$, each of the first and the last character has 2 unigram and 1 bigram features, and each of any middle character has 3 unigram and 2 bigram features, then this generates 31 unigram and 20 bigram features in a total. The higher order gram templates can be defined in a similar way. However, higher the order gram templates are used, the more complex the model will be. Fortunately, mastering around 3000 characters is sufficient to understand 99% of Chinese documents (Wong et al., 2000). Moreover, the proportions of words with one, two, three and four or more characters are 5%, 75%, 14% and 6% respectively. Therefore, unigram, bigram, trigram and quadgram templates are sufficient to capture most of Chinese words information. For lexicon words, as shown in the above example, we can apply unigram templates and bigram templates to construct their binary features. Take 我们 "we" in the above sentence as an example, the feature functions are

$f_1 = I(s(0) = B, c(0) =$我$)$, $f_2 = I(s(0) = B, c(+1) =$们$)$, $f_3 = I(s(0) = B, c(0) =$我$, c(+1) =$们$)$, $f_4 = I(s(0) = E, c(-1) =$我$)$, $f_5 = I(s(0) = E, c(0) =$们$)$, $f_6 = I(s(0) = E, c(+1) =$将$)$, $f_7 = I(s(0) = E, c(-1) =$我$, c(0) =$们$)$, $f_8 = I(s(0) = E, c(0) =$们$, c(+1) =$将$)$.

For morphologically derived words, factoids and named entities, we use trigram and quadgram templates such that the five main morphological rules are incorporated. The five rules include affixation (e.g., 老师们"teachers" is teacher + plural), reduplication (e.g., 马马虎虎"careless" reduplicates and emphasizes 马虎), splitting (e.g., 吃了饭 "already ate" splits a lexical word 吃饭 "eat" by a particle "了"), merging (e.g., 上下文 "context" merges 上文 "above text" and 下文"following text"), and head particle (e.g., 拿出来"take out" is the head 拿"take" + the particles 出来"out"). For instance, the head particle rule yields trigram template $I(s(0) = B, s(+1) = M, s(+2) = E, c(0) = e_0, (c(+1), c(+2)) = (e_1, e_2))$,

where e_0 is a head character such as 拿 "take" and 放 "put," and (e_1, e_2) takes value in a set of selected particles including 出来 "out," 进去 "in," 下去 "down"; the reduplication rule leads to quadgram templates $I(s(0) = B, s(+1) = M, s(+2) = M, s(+3) = E, c(0) = c(+1), c(+2) = c(+3))$.

The factoid words mainly consist of numeric and foreign characters, such as a number 一千零二十四 "1024" or a foreign organization "FBI." Given the set of numeric and foreign characters $\mathcal{F}$, the factoid words lead to trigram templates $I(s(0) = B, c(-1) \notin \mathcal{F}, (c(0), c(+1)) \in \mathcal{F})$, $I(s(0) = M, (c(-1), c(0), c(+1)) \in \mathcal{F})$, and so on. Named entities include frequently-used Chinese names for persons, locations and organizations. A person's name requires extensive enumeration to identify since it does not follow any language rules. In contrary, names for locations and organization can be identified by using built-in feature templates. For example, an organization template $I(s(-2) = B, s(-1) = M, s(0) = E, c(0) \in \mathcal{L})$, where $\mathcal{L}$ is a collection of keywords such as 部、局、委员会 "ministry, bureau and committee."

For new words, it is more challenging in Chinese segmentation. There is not much literature in the identification of new words, though it has substantial impact on the performance of word segmentation. Therefore, good strategies are necessary to detect new words utilizing the linguistic rules and current features. For example, enumeration can be used to detect new factoid words and named entities, as discussed above. In addition, features constructed for the lexicon and morphologically derived words can also be employed to detect new words. Specifically, certain characters are always located at the beginning or at the end of a Chinese word, so new words containing those characters can be easily detected by using the unigram template. For instance, 反 "anti-" typically appears at the beginning of a Chinese word, so the unigram template $I(s(0) = B, c(0) = 反)$ can be used to detect new words such as 反对 "disagree" and 反抗 "resist." In addition, if a new word satisfies the splitting rule as discussed above, trigram templates can be utilized such as $I(s(-1) = B, s(0) = M, s(+1) = E, c(0) = 了)$ for detecting new words like 吐了槽 "already complained". The most important strategy is to incorporate contextual information. For example, if a detected word 价格 "price" is observed many times in a document, it is very likely that certain new words are also associated with price in the same document, such as 上涨 "rise." Then these information can be built into features such as $I(s(0) = B, s(+1) = E, (c(0), c(1)) = 上涨)$ to detect more new words.

3.2.3 Linguistically-embedded constraints and computational feasible losses

In this section, we illustrate how linguistically-embedded constraints can be integrated into (3.1), leading to the importance of features being ranked and thus the effective size of the parameter space can be reduced. As discussed above, some Chinese characters appear much more frequently at the beginning of a word. For instance, 读 "read," has a chance of 74% to occur in the beginning position (Li et al., 2004). So simple constraints can be formulated to obtain the relative order of weight λ_k's. In this example, we can put the weight λ_k for $I(s(0) = B, c(0) = 读)$ larger than weights associated with $I(s(0) = M, c(0) = 读)$, $I(s(0) = E, c(0) = 读)$ and $I(s(0) = S, c(0) = 读)$. In addition, the existing linguistic rules should always be considered and incorporated into the constraints. More language and linguistic rules can be found in Gao et al. (2005). For example, the merging rule implies that λ_k for 上下文 "context" with $I(s(0) = B, s(+1) = M, s(+2) = E, (c(0), c(+1), c(+2)) = 上下文)$ should be largest compared to other weights for it with other trigram features.

To further facilitate the computation, a surrogate loss $L(f, \mathbf{c}_i, \mathbf{s}_i) = L(\Lambda^T F_{\mathbf{c}_i,\mathbf{s}_i})$ is needed, where $\Lambda^T F_{\mathbf{c}_i,\mathbf{s}_i}$ can be regarded as the generalized function margin in multiclass classification (Vapnik, 1998). We propose to use the hinge loss $L(u) = (1-u)_+$ for the segmentation formulation, which is often used in large margin classification such as the support vector machine (SVM; Cortes and Vapnik, 1995). The hinge loss works well as a loss function for large margin classification, since the more the margin is violated, the larger the loss is. As shown in Figure 3.1, the hinge loss is convex, though not differentiable everywhere, and can be easily solved by many efficient optimization algorithms.

Specifically, the model in (3.1) with the hinge loss can be formulated as

$$\begin{aligned} \text{argmin}_{\Lambda,\xi} \quad & \sum_{i=1}^{n} \xi_i + \eta \sum_{k=1}^{K} J(\lambda_k) \\ \text{s.t.} \quad & 1 - \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(\mathbf{s}_i, \mathbf{c}_i, t) \le \xi_i, \;\; \xi_i \ge 0; \; i = 1, \ldots, n, \\ & \lambda_k \ge \lambda_j \text{ for all } (k, j) \in \mathcal{I}, \end{aligned} \tag{3.2}$$

where ξ_i is a slack variable for the hinge loss of each sentence.

There are two main advantages using the hinge loss for the proposed model. First, the model (3.2) only has $2n + 1 + |\mathcal{I}|$ constraints, where $|\mathcal{I}|$ is the number of elements in $\mathcal{I}$. These constraints are much easier

to manage than the exponential order of operations required by the methods of conditional random fields (CRF) and structured support vector machine (SVM^{struct}). The SVM^{struct} solves classification problems involving multiple dependent output variables or structured outputs, which is applicable for complex outputs such as natural language parsing. CRF is conditional random fields, which combines conditional models with the global normalization of random field models. However, both methods bring exponential number of constraints. Secondly and most importantly, optimization process of (3.2) can be efficiently implemented through parallel computing and thus make segmentation scalable. The details of the parallelization algorithm for scalable implementation is provided in Section 3.

Other surrogate loss functions are also possible. Shen et al. (2003) proposed a ψ-loss function $L(u) = \psi(u) = \min(1, (1-u)_+)$ as shown in Figure 3.2. The ψ-loss could attain the optimal rate of convergence under certain assumptions and outperform the hinge loss in general. Intuitively, the advantage of the ψ-loss lies in the fact that it is much closer to the 0-1 loss $I(u > 0)$ in defining the segmentation error, especially when u is negative. As a consequence, the ψ-loss is much less affected by an outlying mis-classified sentence with negative functional margin $\Lambda^T F_{\mathbf{c}_i,\mathbf{s}_i}$. More detailed discussion can be found in Shen et al. (2003).

The model in (3.1) with the ψ-loss can be formulated as

$$\begin{aligned} \text{argmin}_{\Lambda,\xi} \quad & \sum_{i=1}^{n} \psi\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(\mathbf{s}_i, \mathbf{c}_i, t)\right) + \eta \sum_{k=1}^{K} J(\lambda_k) \\ \text{s.t.} \quad & \lambda_k \geq \lambda_j \text{ for all } (k,j) \in \mathcal{I}. \end{aligned} \tag{3.3}$$

To solve the optimization problem in (3.3), a non-convex optimization algorithm, called difference convex algorithm (DCA; An and Tao, 1997) needs to be employed due to the non-convex ψ-loss function. The key idea is to decompose the non-convex ψ-loss as a difference of two convex loss functions, $\psi(u) = \psi_1(u) - \psi_2(u)$, and then approximate $\psi_2(u)$ with iteratively updated tangent hyperplane. The algorithm is guaranteed to converge super-linearly, and thus its computing complexity is not much different from the model (3.2) when n is not too large. However, the training set for a typical segmentation task is often very large with $n = 10^6$ or higher. Therefore, in this article, we proceed with the hinge loss in the real application.

3.2.4 Relation with sentiment analysis

Words segmentation is a prerequisite step for sentiment analysis. A major purpose of sentiment analysis is to determine and summarize text writers' attitudes and opinions towards a specific event, or to classify the polarity of any given text document, e.g., whether the expressed opinion is positive, negative or neutral. Automating sentiment analysis has many potential applications in the areas of cyber-security, online opinion surveys and e-commons from social media and network.

For illustration, Let Y be an ordinal response with a finite set of ordered categories $\{1, \ldots, J\}$. Here we use three categories of opinions "negative," "neutral" and "positive" ($J = 3$). A more refined ordinal category system such as the 5-star in tripadvisor.com or 10-score ranking used in priceline.com can also be incorporated into the framework. In general, we employ a set of decision functions $f_j(\mathbf{x});\ j = 1, \ldots, J-1$ to predict the outcome of Y, where $f_j(\mathbf{x}) \leq 0$ corresponds to $Y \leq j$. So the decision rule is defined as $\phi(\mathbf{x}) = \text{argmin}_j\{j : f_j(\mathbf{x}) \leq 0\}$ and $\phi(\mathbf{x}) = J$ when all $f_j(\mathbf{x}) > 0$. This decision rule is simple and has one good property, that is the decision rule $\phi(\mathbf{x}) = Y$ given the outcome Y. In other word, the decision rule has no difference from the outcome when the outcome is known. Based on this decision rule, when j for $f_j(\mathbf{x}) \leq 0$ is smaller, the decision rule $\phi(\mathbf{x})$ is smaller and the outcome Y is smaller. This decision rule is straightforward. For example, suppose a customer on tripadvisor.com uses different words such as "worst", "bad" and "good" in his/her reviews to describe three different hotels, he/she is expected to rate the hotels from lower to higher in the 5-star system. If we code those words into decision functions and could predict the outcomes (ratings) according to the decision rule.

The accuracy of ϕ can be assessed through a loss function based on the weighted classification errors defined as

$$Err(\phi) = E|\phi(\mathbf{X}) - Y| = \sum_{j=1}^{J-1} E\big(I(f_j(\mathbf{X})\,\text{sign}(Y - j) \leq 0)\big), \tag{3.4}$$

where the weights in (3.4) are proportional to $|\phi(\mathbf{X}) - Y|$ so the order preference can be incorporated. Note that the classification error is expected to be large only when $f_j(\mathbf{X}) \leq 0$ and $Y > j$ or $f_j(\mathbf{X}) > 0$ and $Y \leq j$, or equivalent to say that the decision rule $\phi(\mathbf{X})$ is not monotonic increasing function of the outcome Y.

To construct a cost function associated with the loss defined in ((3.4)), we first define the generalized functional margin $\mathbf{u} = (u_1, \ldots, u_{J-1})$ as $u_j(\mathbf{f}, \mathbf{x}, y) = f_j(\mathbf{x})\,\text{sign}(y - j)$ for each j, where $\text{sign}(+) = 1$

and $\text{sign}(0) = \text{sign}(-) = -1$. To differentiate the above loss function $L(u)$, we define the following as the cost function:

$$\sum_{i=1}^{n} L(\mathbf{u}(\mathbf{f}, \mathbf{x}_i, y_i)) + \sum_{j=1}^{J-1} P_\lambda(f_j), \tag{3.5}$$

where $L(\cdot, \cdot)$ is a surrogate large margin loss, $(\mathbf{x}_i, y_i)_{i=1}^n$ is a training sample, and P_λ is a regularizer to penalize features which do not contribute on margin separator. The surrogate large margin loss $L(\mathbf{u})$ can have various forms. For example, $L(\mathbf{u}) = \sum_j l(u_j)$ is an averaged form and $L(\mathbf{u}) = \max_j l(u_j)$ is a minimax form. Here $l(u)$ can be the hinge loss $l(u) = (1-u)_+$ (Cortes and Vapnik, 1995), or the ψ-loss $l(u) = \psi(u)$ (Shen et al., 2003). In addition, another form of $l(u)$ is $\log\big(1 + \exp(-u)\big)$ (Zhu and Hastie, 2005), which leads to the cumulative logit model with $\text{logit}\Big\{\text{Prob}(Y \le j|\mathbf{x})\Big\} = -f_j(\mathbf{x})$.

It is necessary for sentimental analysis to construct language-specific covariates $\mathbf{x}$. For example, $\mathbf{x}$ could be the counts of keywords associated with opinions, the number of appearances of common phrases under a given subject such as electronic products' feedback, hotel reviews, or political opinions. These covariates could be a mix of binary, counts or numerical ordinal rankings from individual reviews. For Chinese documents, the features incorporated in segmentation are very important as effective segmentation improves the accuracy of sentiment analysis substantially. For illustration, to handle ambiguous Chinese words such as 好 "good" discussed in section 3.2.2, it would be beneficial to utilize the constructed features in Section 3.2.2 as the covariates in (3.5), where $f_j(\mathbf{x}_i)$ is modeled as $\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_{jk}f_k(\mathbf{c}_i, \mathbf{s}_i, t)$, and $\mathbf{c}_i$ and $\mathbf{s}_i$ are the character and tag sequences for the i-th textual review and $f_k(\mathbf{c}_i, \mathbf{s}_i, t)$ are the constructed features described in Section 2.2. With the informative features built in (3.5), segmentation of Chinese words can be obtained by minimizing the sentiment analysis error as opposed to the segmentation error. Consequently, it is capable of detecting ambiguous words such as 好 in 很好 "very good" and 好糟糕 "really terrible," even if the training corpus may not contain opinion terms such as 糟糕 "terrible." This hybrid framework differentiates Chinese sentiment analysis from English sentiment analysis significantly where segmentation is not required.

3.3 Algorithm and computation

An efficient optimization strategy as well as parallelization in computing will improve computational efficiency significantly. A computational strategy based on the idea of "decomposition and combination" is

designed to meet computational challenges in solving large-scale optimization for the proposed model. The procedure is summarized in Algorithm 4

Algorithm 4 Chinese words segmentation procedure based on penalized hinge loss

Input: Training samples $\{w_i\}_{i=1}^n$ and test documents $\mathcal{C}$.

1: Build features $f_k(\mathbf{s}_i, \mathbf{c}_i, t)$ with N-gram templates for every $k = 1, \cdots, K, i = 1, \cdots, n$ and $t = 1, \cdots, T$.

2: Initialize λ_k for $k = 1, \cdots, K$ and build constraints $\lambda_k \leq \lambda_j$ for all $(k, j) \in \mathcal{I}$ through linguistic rules.

3: Implement the ad hoc cutting-plane algorithm to get estimates $\hat{\lambda}_k$ for $k = 1, \cdots, K$ by minimizing (3.2).

4: Predict segmentation locators $\mathbf{s}$ by maximizing $\hat{f}(\mathbf{c}, \mathbf{s}) \equiv \sum_{k=1}^K \sum_{t=1}^T \hat{\lambda}_k f_k(\mathbf{c}, \mathbf{s}, t)$ for any $\mathbf{c} \in \mathcal{C}$.

Output: segmented words $\{w_i\}_{i=1}^{n^{te}}$ for $\mathcal{C}$.

Specifically, we use the truncated L_1-penalty (Shen et al., 2012) for $J(\Lambda)$, i.e. $J(\cdot) = \eta \min(\|\cdot\|/\nu, 1)$ with η and ν being tuning parameters. The truncated L_1-penalty has three advantages. It performs adaptive selection of linguistic features with weights Λ and corrects the Lasso bias through tuning ν. It is also capable of handling small weights of linguistic features through tuning ν and thus helps the improvement of accuracy in segmentation. In addition, it has piecewise linear property which gains computational advantages.

The optimization is carried out using an ad hoc cutting-plane algorithm. The idea of the cutting-plane method is to refine iteratively feasible sets by means of linear inequalities. Let $\mathcal{M}$ be the set of constraints in model (3.2) and $\mathcal{W} \subset \mathcal{M}$ be the current working set of constraints. In each iteration, the algorithm computes the solution over the current working set $\mathcal{W}$, finds the most violated constraint in $\mathcal{M} \backslash \mathcal{W}$, and adds it to the working set. The algorithm stops until all violations are smaller than the tolerance ϵ. The ad hoc cutting-plane algorithm is illustrated in Algorithm 5. The computational efficiency of cutting-plane algorithm for

Algorithm 5 The cutting-plane algorithm for model (3.2)

Input: η, ϵ, initial $\mathcal{W} = \emptyset$

Output: $\hat{\Lambda}$

1: Compute $\hat{\Lambda}^{(m)} = \text{argmin}_{\Lambda, \xi} \sum_{i=1}^n \xi_i + \eta \sum_{k=1}^K J(\lambda_k)$ s.t. $\mathcal{W}$;

2: Obtain the constraint $l^{(m)} \in \mathcal{M}$ which has the largest violation in $\mathcal{M}$ given $\hat{\Lambda}^{(m)}$;

3: Set $\mathcal{W} = \mathcal{W} \cup s_m$;

4: Return to step 1 until no violation is larger than ϵ.

hinge loss has been extensively investigated by empirical studies. It is much faster than conventional training methods derived from decomposition methods (Joachims et al., 2009). Note that model (3.2) contains a large amount of features which make the computation challenging. For example, assuming that $\mathcal{C}$ contains the 1000 most common Chinese characters and the character locator set $\mathcal{S}$ has 4 tags $\{B, M, E, S\}$, the unigram

and bigram templates involve $3 \times |\mathcal{S}| \times |\mathcal{C}| + 2 \times |\mathcal{S}| \times |\mathcal{C}| \times |\mathcal{C}|$ or roughly 8×10^6 features. To further accelerate the computation, parallel strategy can be used in Step 2.

To handle large size of documents or texts, MapReduce computation can be utilized to break large problems into many small subproblems in a recursive and parallel manner. In particular, we decompose our cost functions and regularizers over many observations by transforming complicated nonconvex optimization problems to many subproblems of convex minimization. This alleviates high storage costs and increases computational speed through parallel computing. To achieve this purpose, there are two tools available, OpenMP, the multi-platform shared-memory parallel programming platform (http://www.openmp.org), and Mahout, a library for scalable machine learning and data mining. Those strategies allow us to analyze data containing several billions (10^9) of observations on a single machine with reasonable computational speed.

Other penalty functions are also optional if they could bring advantages for model (3.2). For example, if the regularizer L_2 norm penalty is used, model (3.2) becomes optimization problems of convex functions. In fact, the involved optimization problems can be solved by sequential quadratic programming (QP) or linear programming (LP), respectively. Parallelized LP problem has been extensively studied (Dongarra et al., 2002), and QP can also be parallelized through a parallel gradient projection-based decomposition method (Zanni et al., 2006). The key idea is to split the original problem into a sequence of smaller QP subproblems and parallelize the most demanding task of the gradient projection within each QP subproblem. With the parallelized QP and LP implementations, we are able to process Chinese text datasets of size $O(10^7)$ under a common PC. Therefore, the personalized model for (2) becomes ideal for parallelization and mapReduce, and is able to handle large-scale applications.

3.4 Application to Peking university corpus

In this section, we analyze the corpora obtained from SIGHAN Bakeoff (http://www.sighan.org), which are popular corpora in Chinese language processing competition. There are four datesets released in SIGHAN's International Chinese Word Segmentation Bakeoff, and they are Academia Sinica (AS), City University of Hong Kong (HK), Peking University (PK) and Microsoft Research corpora (MSR). Each corpus is coded by Unicode and consists of training and test sets. The number of words in each corpus is shown in Table 3.3. In the Bakeoff corpora, Out-of-vocabulary (OOV) words are defined as words in the test set not occurring in the training set. The contents in the corpora are carefully selected and the domains in the corpora are

widely distributed, including politics, economics, culture, law, science and technology, sport, military and literature. Therefore, the corpora are representative enough to investigate performances of Chinese word segmentation methods.

PK is used in our experiments to investigate the proposed model. We notice in Table 3.3 that PK is well balanced in terms of OOV percentage and sizes of training and test sets, compared to the other three corpora. In the PK corpus, the training set has $161,212$ sentences, i.e. $n = 161,212$, consisted of around 1.1 million words while the test set has $14,922$ sentences with 17 thousand words. Approximately 6.9% words in the test set are OOV, in which about 30% OOV are new words of main-specific or time-sensitive, such as 三通 "three links" and 非典 "SARS". In addition, more than 85% of all new words fall into types of either 2-character new word or 2-character word followed with a character. Some fragments of the PK training and test sets are shown in Figure 3.3 and 3.4.

We provide a study to test our proposed method in the two settings of model (3.2). The simplest setting $method^{12}$ only contains the corresponding unigram and bigram templates, while $method^{1234}$ has extra trigram and quadgram templates. Those two settings are compared with the two top performers in the second international Chinese word segmentation bakeoff. Performance of Chinese word segmenters is generally reported in terms of three performance metric criteria: precision (P), recall (R) and evenly-weighted F-measure (F). The precision is the fraction of correct segmented words, the recall is the fraction of correct words which are segmented, and the evenly-weighted F-measure is the harmonic mean of the precision and recall (F=2PR/(P+R)).

The results are shown in Table 3.4. The proposed method overall has higher recall than the two top performers. The simplest setting $method^{12}$ attains to 92.9% precision, 96.5% recall and 94.7% in F-measure, delivering competitive performance against the two top performers. Note that $method^{12}$ has relatively low precision and high recall compared to the two top performers. This is because unigram and bigram templates only contain the information of the consecutive two characters, reluctant to segment words with three or more characters. When trigram and quadgram templates are used in the proposed method, significant improvement in precision are achieved through higher-order templates, and $method^{1234}$ attains to 95.1%, which has almost no difference from the two top performers. Moreover, $method^{1234}$ delivers significantly higher recall with 96.9%, leading to highest F-measure with 96.0%. Therefore $method^{1234}$ outperforms the other three methods without considering computing time.

The computing time of these four methods is not investigated since the information of the two top performers is not available on the SIGHAN website. However, The computing time of $method^{1234}$ increases significantly, compared with $method^{12}$, as more and higher order n-gram templates are embedded into the model. In our experience, $method^{12}$ runs for about 8 days for the PK corpus while $method^{1234}$ takes more than two weeks in a common personal computer. The order of n-gram templates consequently should be selected according to the specific situation if time is a sensitive factor. In summary, the proposed method outperforms the current top performers for the PK corpus when sufficient n-gram templates are incorporated. But the higher order n-gram templates lead to significant computing time increase. For this reason, the improvement of the algorithm and parallel strategies are necessary to reduce the computational challenges.

3.5 Discussion

We propose a learning framework with linguistically-embedded features for Chinese word segmentation. The proposed model is a character-based method by constructing the feature functions mapping from characters to segmentation locators in words. The key idea is to build feature functions through N gram templates, not only containing the information of the character itself but incorporating information of its former and later characters. To make the model more scalable, the hinge loss that is often used in classification, is employed to reduce the number of constraints as well as to simplify their forms.

Most existing approaches only incorporate very limited linguistic information, ignoring well established linguistic rules. One main advantage of the proposed model is that linguistic rules can be incorporated into the model through adjusting the weights of each linguistic features and thus dramatically improve the segmentation accuracy.

Computational tractability is the crucial requirement of segmentation because of a vast amount of data containing increasing volumes of text information over time. The most important property of the proposed model is that optimization process can be efficiently implemented through transforming complicated non-convex optimization problems to many subproblems of convex minimization and computing these convex subproblems parallelly, and thus becomes scalable for high-volume datasets.

Accuracy and computational complexity are always trade-off. Higher order gram templates lead to higher accuracy but could result in significant increase of computation cost. For different corpora, the order of grams should be carefully selected for different categories of words. In addition, it is still a challenge

for OOV segmentation, but better strategies and more linguistic rules can be incorporated into the proposed model.

3.6 Figures and Tables

Table 3.1: A character can appear in different positions within different words

position	example
beginning	发生 “to happen”
middle	始发站 “starting station”
end	头发 “hair”

Table 3.2: Taxonomy in Chinese words

category	subcategory	examples
LW	lexical word	学生，照片，约会
MDW	affixation	老师们
	reduplication	马马虎虎
	splitting	吃了饭
	merging	上下文
	head+particle	拿出来
FT	date & time	5月3日，六月五日，12点半，三点二十分
	number & fraction	一千零二十四，4897，60%，百分之一，1/6
	email & website	johnson@email.com, www.google.com
NE	person name	张三，约翰
	location name	北京，上海
	organization name	长城，大都会博物馆
NW	new word	吐槽，非典

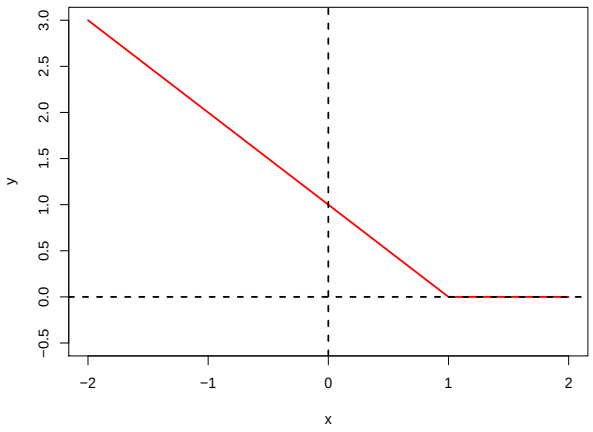


Figure 3.1: A plot of hinge loss function

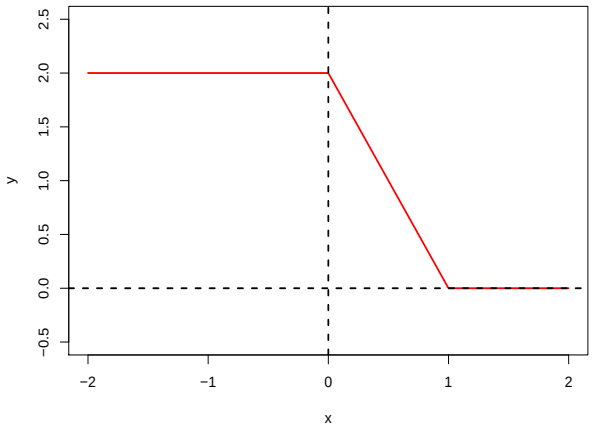


Figure 3.2: A plot of psi loss function

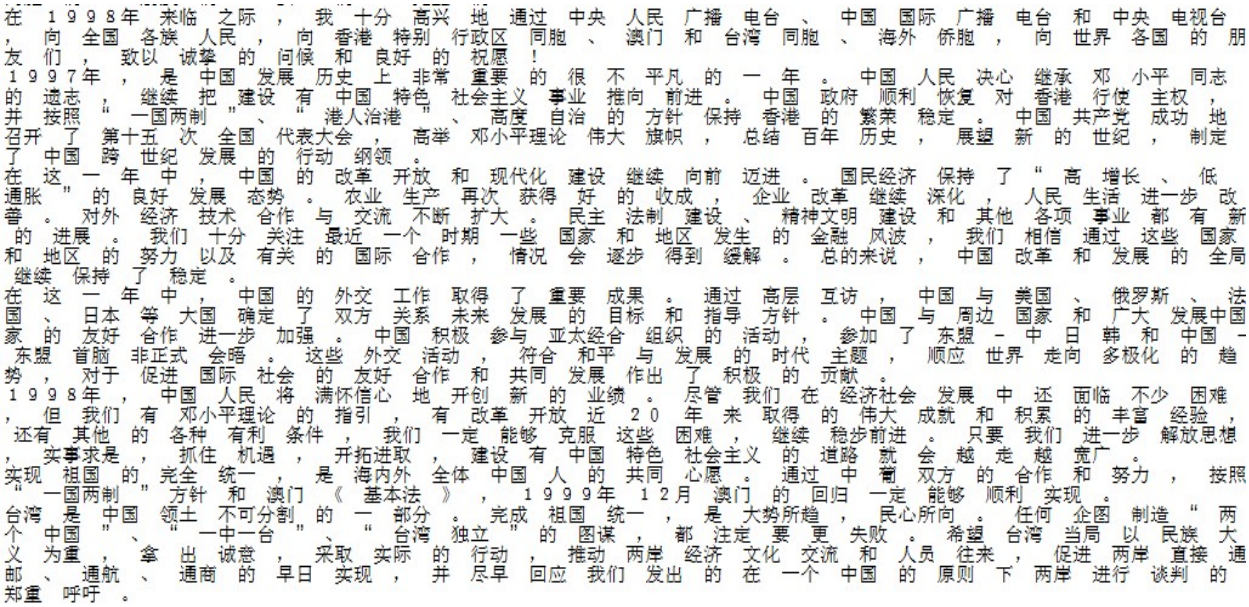


Figure 3.3: Fragments of the PK training set.

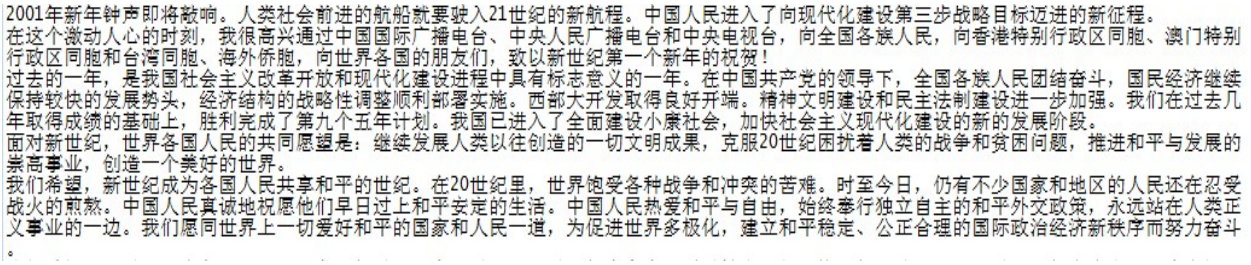


Figure 3.4: Fragments of the PK test set.

Table 3.3: SIGHAN's corpora.

corpora	# training words (in thousands)	# test words (in thousands)	OOV(%)
AS	5800	12	0.021
HK	240	35	0.071
PK	1100	17	0.069
MSR	20,000	226	0.002

Table 3.4: Comparison of two top performers and the proposed method

method	precision	recall	F-measure
$method^{12}$	0.929	0.965	0.947
$method^{1234}$	0.951	0.969	0.960
1st in bakeoff	0.952	0.951	0.952
2nd in bakeoff	0.955	0.939	0.947

References

An, H. and Tao, P. (1997), Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms. *Journal of Global Optimization*, **11**, 253-285.

Agarwal, D., Zhang, L. and Mazumder, R. (2011). Modeling item-item similarities for personalized recommendations on Yahoo! front page. *Ann. Appl. Stat.* **5**, 1839-1875.

Basu, S., Pan, W., Shen, X., Oetting, B. (2011). Multi-locus association testing with penalized regression. *Genetic Epidemiology* **35**, 755-765.

Bautin, M., Vijayarenu, L. and Skiena, S. (2008). International sentiment analysis for news and blogs. Proceedings of the International Conference on Weblogs and Social Media.

Bo, P., Lillian, L. and Shivakumar, V. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Empirical Methods on Natural Language Processing,* 79-86.

Chang, P., Galley, M., Manning, C., (2008). Optimizing Chinese Word Segmentation for Machine Translation Performance. *Proceedings of the Third Workshop on Statistical Machine Translation*, 224-232.

Chen, A. (2003). Chinese word segmentation using minimal linguistic knowledge. *SIGHAN*, 148-151.

Chen, S. and Goodman, J., (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language* **13**, 359-394.

Chen, X., Lin, Q., Kim, S., Carbonell, J. G. and Xing, E. P. (2012). Smoothing proximal gradient method for general structured sparse regression. *Ann. Applied Statist.* **6**, 719-752.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Mach. Learn.* **20**, 273-297.

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

Devore, R. A. and Lorentz, G. G. (1993). *Constructive Approximation*. Springer-Verlag, Berlin Heidelberg.

Dongarra, J., Foster, I., Fox, G., Gropp, W., Kennedy, K., Torczon, L. and White, A. (2002). The Sourcebook of Parallel Computing. *Morgan Kaufmann Publishers*, San Francisco.

Fan, J. and Li, R. (2001). Variable Selection via nonconcave penalized likelihood and its oracle properties. *Journal of American Statistical Association* **96**, 1348-1360.

Friedman, J. H., Hastie, T. and Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**, 432-441.

Gao, J., Li, M., Wu, A., and Huang, C. (2005). Chinese word segmentation and named entity recognition: a pragmatic approach. *Computational Linguistics Journal* **31**, 531-574.

Girvan, M. and Newman, M.E.J. (2002). Community structure in social and biological networks. *Proc. Natl Acad. Sci.* **99**, 7821-7826.

Guo, J., Levina, L., Michailidis, G. and Zhu, J. (2011). Joint estimation of multiple graphical models. *Biometrika* **98**, 1-15.

Hastie, T. and Tibshirani, R. (1993). Varying-coefficient models. *J. Roy. Statist. Soc. Ser. B* **55**, 757-796.

Huang, J. Z. (1998). Projection estimation in multiple regression with application to functional ANOVA models. *Ann. Statist.* **26**, 242-272.

Huang, J. Z., Wu, C. O. and Zhou, L. (2002). Varying-coefficent models and basis function approximations for the analysis of repeated measurements. *Biometrika* **89**, 111-128.

Huang, J. Z., Wu, C. O. and Zhou, L. (2004). Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statist. Sinica* **14**, 763-788.

Jacob, L., Obozinski, G. and Vert, G. (2013). Group lasso with overlap and graph lasso. Manuscript.

Jenatton, R., Audibert, J.-Y. and Bach, F. (2009). Structured variable selection with sparsity inducing norms. Manuscript.

Joachims, T., Finley, T., and Yu, C. (2009). Cutting-plane training of structural SVMs. *Machine Learning* **27**, 27-59.

Kolaczyk, E.D. (2009). *Statistical analysis of network data: Methods and models.* New York, Springer.

Kolar, M., Parikh, A. and Xing, E. P. (2010). On sparse nonparametric conditional covariance selection. *The 27th International Conference on Machine Learning*.

Kolar, M. and Xing, E. P. (2009). Sparsistent estimation of time-varying discrete Markov random fields. Manuscript.

Lafferty, J., McCallum, A. and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th Inter. Conf. on Machine Learning*, Morgan Kaufmann, pp. 282-289.

Li, H., Huang, C., Gao, J. and Fan, X. (2004). The use of SVM for Chinese new word identification. In Proc. 1st Inter. Joint Conf. on Natural Language, Springer, pp. 497-504.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the LASSO. *Ann. Statist.* **34**, 1436-1462.

Meng, W., Liu, L. and Chen, A. (2010). A comparative study on Chinese word segmentation using statistical models. In *IEEE International Conference on Software Engineering and Service Sciences*, pp. 482-486.

Obozinski, G., Jacob, L. and Vert, G. (2013). Group lasso with overlaps: the latent group lasso approach. Manuscript.

Pang, B., Lee, L. and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 79 - 86. East Stroudsburg, PA: Association for Computational Linguistics.

Peng, J., Wang, P., Zhou, N. and Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *J. Amer. Statist. Assoc.* **104**, 735-746.

Qu, A. and Li, R. (2006). Quadratic inference functions for varying coefficient models with longitudinal data. *Biometrics* **62**, 379-391.

Shen, X., Huang, H. and Pan, W. (2012). Simultaneous supervised clustering and feature selection over a graph. *Biometrika* **99**, 899-914.

Shen, X., Pan, W., and Zhu, Y. (2012). Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association* **107**, 223-232.

Shen, X., Tseng, G., Zhang, X., and Wong, W. (2003). On ψ-learning. *Journal of the American Statistical Association*, **98**, 724-734.

Shojaie, A. and Michailidis, G. (2010). Network enrichment analysis in complex experiments. *Stat. Appl. Genet. Mol. Biol.*, **9** Art. 22.

Song, L., Kolar, M. and Xing, E. P. (2009). KELLER: Estimating time-evolving interactions between genes. *Bioinformatics* **25**, 128-136.

Sun, W. (2010). Word-based and character-based word segmentation models: comparison and combination. *Proceedings of the 23rd International Conference on Computational Linguistics*, 1211-1219.

Sun, W., and Xu., J. (2011). Enhancing Chinese word segmentation using unlabeled data. *Empirical Methods in Natural Language Processing*.

Teahan, W., Wen, Y. and Witten, I. (2000). A compression-based algorithm for Chinese word segmentation. *Computational Linguistics* **26**, 375-393.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *J. R. Stat. Soc. Ser. B* **58**, 267-288.

Tsochantaridis, I., Joachims, T., Hofmann, T. and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* **6**, 1453-1484.

Turney, P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proceedings of the Association for Computational Linguistics,* 417- 424. East Stroudsburg, PA: Association for Computational Linguistics.

Valencia, M., Pastor, M. A., Fernadez-Seara, M. A., Artieda, J., Martinerie, J., and Chavez, M. (2009). Complex modular structure of large-scale brain networks. *Chaos* **19**, 023119.

Vapnik, V. (1998). Statistical learning theory, Chichester, UK, Wiley.

Wang, H., Li, R. and Tsai, C.-L. (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika* **94**, 553-568.

Wang, K., Zong, C. and Su, K.Y. (2010). A character-based joint model for Chinese word segmentation. *Proceedings of the 23rd International Conference on Computational Linguistics,* 1173-1181.

Wiebe, J., Wilson, T., Bruce, R., Bell, M. and Martin, M. (2004). Learning subjective language. *Computer Linguistics* **30**, 277-308.

Wu, A. and Jiang, Z. (2000). Statistically-enhanced new word identification in a rule-based Chinese system. *Proceeding of the 2rd ACL Chinese Processing Workshop*, 41-66.

Xu, J., Gao, J., Toutanova, K., and Ney, H., (2008). Bayesian semi-supervised Chinese word segmentation for statistical machine translation. *'08 Proceedings of the 22nd International Conference on Computational Linguistics* **1**, 1017-1024.

Xue, L. and Qu, A. (2012). Variable selection in high-dimensional varying-coefficient models with global optimality. *J. Mach. Learn. Res.* **13**, 1973-1998.

Xue, L., Qu, A. and Zhou, J. (2010). Consistent model selection for marginal generalized additive model for correlated data. *J. Amer. Statist. Assoc.* **105**, 1518-1530.

Xue, L., Shu, X., Shi, P., Wu, C. and Qu, A. (2013). Local feature selection in varying coefficient models. Manuscript.

Xue, L. and Yang, L. (2006). Additive coefficient modeling via polynomial spline. *Statist. Sinica* **16**, 1423-1446.

Xue, N. (2003). Chinese word segmentation as Character Tagging . *Computational Linguistics and Chinese Language Processing* **8**, 29-48.

Xue, N. and Shen, L. (2003). Chinese word segmentation as LMR tagging. *Proceedings of 2nd SIGHAN Workshop on Chinese Language Processing*, 176-179.

Zanni, L., Serafini, T. and Zanghirati, G. (2006). Parallel software for training large scale support vector machines on multiprocessor systems. *Journal of Machine Learning Research* **7**, 1467-1492.

Zhang, R., Yasuda, K., and Sumita, E. (2008). Chinese word segmentation and statistical machine translation. ACM *Transactions on Speech and Language Processing* **5**, 4:1-4:19.

Zhonghua Zihai dictionary. Zhonghua Book Company, 1994.

Zhu, Y., Shen, X., and Pan, W. (2013). Simultaneous grouping pursuit and feature selection in regression over an undirected graph. *J. Amer. Statist. Assoc.* **108**, 713-725.

Zhu, Y., Shen, X., and Pan, W. (2014). Structural pursuit over multiple undirected graphs. Manuscript.

Zou, H. (2006). The adaptive LASSO and its oracle properties. *J. Amer. Statist. Assoc.* **101**, 1418-1429.

Appendix

3.7 Time-varying networks estimation and dynamic model selection

In the following, let C_κ be the space of κ-times continuously differentiable functions on $[0,1]$, and G_n be the spline approximation space of order p and knot sequence Υ_n. Let $\|\cdot\|_2$ be the usual vector or function L_2 norm, unless otherwise defined. We denote any positive constants by the same letters c, C without distinction in each case. In the following, we assume that functions $\left\{\sigma^{ii}(t)\right\}_{j=1}^{p}$ are known. But the proof follows similarly for any $\left\{\hat{\sigma}^{ii}(t)\right\}_{j=1}^{p}$ that satisfy condition (C2).

Let M be the model space as a collection of vectors of functions each with $p\left(p-1\right)/2$ elements,

$$M=\left\{\rho\left(t\right)=\left(\rho^{ij}\left(t\right),1\leq i<j\leq p\right),\rho^{ij}\left(t\right)\in C_{\kappa}\right\},$$

and let the approximation space be defined similarly as,

$$M_{n}=\left\{g\left(t\right)=\left(g^{ij}\left(t\right),1\leq i<j\leq p\right),g^{ij}\left(t\right)\in G_{n}\right\}.$$

For any $\rho\in M$, we define the theoretical and empirical norms on M respectively as

$$\|\rho\|^{2}=E\left[\sum_{i=1}^{p}\left(\sum_{j\neq i}\rho^{ij}(T)\sqrt{\frac{\sigma^{jj}(T)}{\sigma^{ii}(T)}}Y_{j}(T)\right)^{2}\right],$$

and

$$\|\rho\|_{n}^{2}=\frac{1}{nm}\sum_{k=1}^{n}\sum_{u=1}^{m}\sum_{i=1}^{p}\left(\sum_{j\neq i}^{p}\rho^{ij}(t_{u})\sqrt{\frac{\sigma^{jj}(t_{u})}{\sigma^{ii}(t_{u})}}y_{j}^{k}(t_{u})\right)^{2}.$$

Lemma 3.7.1. *Under conditions C1 and C2, there exist constants $C > c > 0$, such that*

$$C \sum_{1\le i<j\le p} \left\|\rho^{ij}(T)\right\|_2^2 \ge \|\rho\|^2 \ge c \sum_{1\le i<j\le p} \left\|\rho^{ij}(T)\right\|_2^2,$$

where $\left\|\rho^{ij}(T)\right\|_2^2 = E\left(\rho^{ij}(T)\right)^2$.

Proof: Let $\rho^i(T) = \left(\rho^{ij}(T), j \ne i\right)$, and $\widetilde{\mathbf{Y}}_i(T) = \left(\sqrt{\frac{\sigma^{jj}(t)}{\sigma^{ii}(t)}} Y_j(T), j \ne i\right)$. Then by condition (C2), there exist constants $c > 0$,such that

$$\begin{aligned}\|\rho\|^2 &= E\left[\sum_{i=1}^p \left(\rho^i(T)\right)^T \widetilde{\mathbf{Y}}_i^T(T)\, \widetilde{\mathbf{Y}}_i(T)\, \rho^i(T)\right] \ge cE\left[\sum_{i=1}^p \left(\rho^i(T)\right)^T \rho^i(T)\right] \\ &= cE\left[\sum_{i=1}^p \sum_{j\ne i} \left(\rho^{ij}(T)\right)^2\right] = 2c \sum_{1\le i<j\le p} E\left(\rho^{ij}(T)\right)^2 = 2c \sum_{1\le i<j\le p} \left\|\rho^{ij}(T)\right\|_2^2.\end{aligned}$$

The other side of the inequality follows similarly from condition (C2). □

Lemma 3.7.2. *Under conditions C1,C2, C5 and C8, there exist constants $c, C > 0$ such that, except on an event whose probability goes to zero, as $n \to \infty$, for any vector β_n of length $p(p-1)J_n/2$, one has,*

$$c\|\rho\|^2 \le \|\rho\|_n^2 \le C\|\rho\|^2.$$

Proof: The proof follows similarly from Lemma 4 of Huang (1998), and Lemma A.4 of Xue and Yang (2006). □

Lemma 3.7.3. *Given Conditions C1, C2, C5, C6 and C8, there exist constants $C > c > 0$ such that, except on an event whose probability goes to zero, as $n \to \infty$, for any vector β_n of length $p(p-1)J_n/2$, one has,*

$$\frac{c}{N_n}\|\beta_n\|_2^2 \le \frac{1}{nm}\beta_n^T \mathcal{X}_n \beta_n \le \frac{C}{N_n}\|\beta_n\|_2^2.$$

Proof. Write $\beta_n = \left(\beta^{ij}, 1 \le i < j \le p\right)^T$, and $\beta^{ij} = \left(\beta_h^{ij}, h = 1, \ldots J_n\right)^T$. Let $g^{ij} = \sum_{h=1}^{J_n} \beta_h^{ij} B_h \in G_n$, and $g = \left(g^{ij}(t), 1 \le i < j \le p\right) \in M_n$. Then$\|g\|_n^2 = \frac{1}{nm}\beta_n^T X_n \beta_n$. By Lemmas 3.7.1 and 3.7.2, one has

$$c \sum_{1\le i<j\le p} \left\|g^{ij}\right\|_2^2 \le \frac{1}{nm}\beta_n^T \mathcal{X}_n \beta_n \le C \sum_{1\le i<j\le p} \left\|g^{ij}\right\|_2^2,$$

in which $\left\|g^{ij}\right\|_2^2 = E\left(\sum_{h=1}^{J_n}\beta_h^{ij}B_h(T)\right)^2$. Furthermore, Theorem 5.4.2 of Devore and Lorentz (1993) entails that there exists a constant $c>0$, such that

$$E\left(\sum_{h=1}^{J_n}\beta_h^{ij}B_h(T)\right)^2 \geq c\sum_{h=1}^{J_n}\left(\beta_h^{ij}\right)^2 E\left(B_h^2(T)\right) \geq \frac{c}{N_n}\sum_{h=1}^{J_n}\left(\beta_h^{ij}\right)^2.$$

Therefore, $\frac{1}{nm}\beta_n^T X_n\beta_n \geq \frac{c}{N_n}\left\|\beta_n\right\|_2^2$ for some $c>0$. The other side of the inequality follows similarliy from the triangular inequality that $E\left(\sum_{h=1}^{J_n}\beta_h^{ij}B_h(T)\right)^2 \leq 2\sum_{h=1}^{J_n}\left(\beta_h^{ij}\right)EB_h^2(T) \leq \frac{2}{N_n}\sum_{h=1}^{J_n}\left(\beta_h^{ij}\right)^2$. □

To establish our theoretical results, we first introduce an oracle estimator, which estimates each $\rho^{ij}(t)$ under the assumption that the null regions of each $\rho^{ij}(t)$ is known. It is constructed only for the proof of the asymptotic results, and is not useful for analyzing real data. One notes that, for each end point of the null region $E^{ij} = [e_1^{ij}, e_2^{ij}]$ in condition (C7), there exist knots $\nu_{l_1^{ij}}$ and $\nu_{l_2^{ij}}$ in the knot sequence $\Upsilon = \{0=\nu_0<\nu_1<\cdots<\nu_{N_n}<\nu_{N_n+1}=1\}$ such that $e_1^{ij}-\lambda_n \in [\nu_{l_1^{ij}}, \nu_{l_1^{ij}+1})$ and $e_2^{ij}+\lambda_n \in [\nu_{l_2^{ij}}, \nu_{l_2^{ij}+1})$. Let $J_{ij} = \left\{1,\ldots,\nu_{l_1^{ij}}-1, \nu_{l_2^{ij}}+p+1,\ldots,J_n\right\}$. An oracle estimator $\widetilde{\beta}^{(o)}$ is constructed by taking all coefficients $\widetilde{\beta}_k^{ij}=0$, for $k=\nu_{l_1^{ij}},\ldots,\nu_{l_2^{ij}}+p$ and estimating the other coefficients by minimizing the sum of the squares

$$\frac{1}{2nm}\sum_{i=1}^{p}\sum_{k=1}^{n}\sum_{u=1}^{m} w_{iu}\left(y_i^k(t_{ku}) - \sum_{j\neq i}^{p}\sum_{h\in\mathcal{J}_{ij}}\beta_h^{ij}B_h(t_{ku})\sqrt{\frac{\sigma^{jj}(t_{ku})}{\sigma^{ii}(t_{ku})}}y_j^k(t_{ku})\right)^2.$$

Denote the resulting oracle estimator of the partial coefficient functions by $\widetilde{\rho}^{ij}(t), 1\leq i<j\leq p$. Then $\widetilde{\rho}^{ij}(t)=0$, for any $t\in\left[\nu_{l_1^{ij}}, \nu_{l_2^{ij}}\right)$. Let $G_{ij}^{(o)}$ be a subspace of G_n with spline functions of the form $g^{(0)}=\sum_{k\in J_{ij}}\beta_k B_k$. Let $M_n^{(o)} = \left\{g(t) = \left(g^{ij}(t), 1\leq i<j\leq p\right), g^{ij}(t)\in G_{ij}^{(o)}\right\}$ be the oracle spline space. Then the oracle estimator defined above can be viewed as the least square estimator on $M_n^{(o)}$.

Lemma 3.7.4. *Under conditions (C5)-(C7), there exists a spline function $g_{ij}^{(o)}\in G_{ij}^{(o)}$, such that* $\sup_{0<t<1}\left|\rho^{ij}-g_{ij}^{(o)}\right| = O\left(N_n^{-(p+1)}+\lambda_n\right)$.

Proof: The approximation theory in Deboor (2001) entails that there exists a spline function $g_{ij}\in\mathcal{G}_n^{(p)}$ such that $\sup_{0<t<1}\left|\rho^{ij}(t)-g_{ij}(t)\right| = O\left(N_n^{-(p+1)}\right)$, where $g_{ij}=\sum_{h=1}^{N_n+p+1}\beta_h^{ij}B_h$ for a set of

coefficients $\left\{\beta_h^{ij}\right\}_{h=1}^{N_n+p+1}$. Now let $g_{ij}^* = \sum_{h\in J_{ij}} \beta_h^{ij} B_h$. Then $g_{ij}^* \in G_{ij}^{(o)}$, and

$$\begin{aligned}
\sup_{0<t<1} \left|\rho^{ij} - g_{ij}^*\right| &\leq \sup_{0<t<1} \left|\rho^{ij} - g_{ij}\right| + \sup_{0<t<1} \left|g_{ij} - g_{ij}^*\right| \\
&\leq 2 \sup_{0<t<1} \left|\rho^{ij} - g_{ij}\right| + \sup_{t\in[w_{l1}-\lambda_n, w_{l1})} \left|\rho^{ij}\right| + \sup_{t\in[w_{l2}, w_{l2}+\lambda_n)} \left|\rho^{ij}\right| \\
&= O\left(N_n^{-(p+1)} + \lambda_n\right). \quad \square
\end{aligned}$$

Lemma 3.7.5. *Given Conditions C1, C2, C5, C6 and C8, we have, for $1 \leq i < j \leq p$, the oracle estimators satisfy*

$$\begin{aligned}
\left\|\widetilde{\rho}^{ij} - \rho^{ij}\right\|_2 &= O_p\left(\sqrt{\frac{N_n}{nm}} + N_n^{-(q+1)}\right), \\
\sup_{t\in\mathbf{I}} \left|\widetilde{\rho}^{ij}(t) - \rho^{ij}(t)\right| &= O_p\left(\frac{N_n^{3/2}}{\sqrt{nm}} + N_n^{-(q+1)}\right). \qquad (3.6)
\end{aligned}$$

Lemma 3.7.6. *Suppose conditions (C1)-(C6) hold. Let $Z_h^{ijk}(t) = B_h(t)\sqrt{\frac{\sigma^{jj}(t)}{\sigma^{ii}(t)}} y_j^k(t)$, and for any $1 \leq i, j \leq p$, let*

$$\widetilde{\mathbf{c}}_h^{ij}(\beta) = -\frac{1}{nm} \sum_{k=1}^{n} \sum_{u=1}^{m} Z_h^{ijk}(t_{ku}) \left(y_i^k(t_{ku}) - \sum_{j'\neq i}^{p} \sum_{h\in\mathcal{J}_{ij}} \beta_h^{ij'} Z_h^{ij'k}(t_{ku}) \right),$$

and $\mathbf{c}_h^{ij}(\beta) = \widetilde{\mathbf{c}}_h^{ij}(\beta) + \widetilde{\mathbf{c}}_h^{ji}(\beta)$ for any $1 \leq i < j \leq p$. Then for any η_n that satisfies $\frac{1}{\eta_n}\sqrt{\frac{\log(N_n)}{N_n nm}} \to 0$, and $\left(N_n^{-(p+2)} + \lambda_n/N_n\right)/\eta_n \to 0$, one has

$$P\left(\max_{1\leq i<j\leq p, h\in J_{ij}} \left\|\mathbf{c}_h^{ij}\left(\widehat{\beta}^{(0)}\right)\right\|_2 \geq \eta_n\right) \to 0.$$

Proof: By Lemma 3.7.5, there exists a constant $c > 0$ and spline functions $s^{ij} \in G_{ij}^{(o)}$, such that

$$\max_{1\leq i<j\leq p} \sup_{0<t<1} \left|\rho^{ij}(t) - s^{ij}(t)\right| \leq c N_n^{-(p+1)}. \qquad (3.7)$$

Let $e_h^{ij} = -\frac{1}{nm}\sum_{k=1}^{n}\sum_{u=1}^{m} Z_h^{ijk}(t_u)\left(y_i^k(t_u) - \sum_{j'\neq i}^{p}\rho^{ij'}(t_u)\sqrt{\frac{\sigma^{j'j'}(t_u)}{\sigma^{ii}(t_u)}}y_{j'}^k(t_u)\right)$,

$$\delta_h^{ij} = \frac{1}{nm}\sum_{k=1}^{n}\sum_{u=1}^{m} Z_h^{ijk}(t_u)\left(\sum_{j'\neq i}^{p}\left[s^{ij'}(t_u) - \rho^{ij'}(t_u)\right]\sqrt{\frac{\sigma^{j'j'}(t_u)}{\sigma^{ii}(t_u)}}y_{j'}^k(t_u)\right),$$

and $\varepsilon_h^{ij}(\beta) = -\frac{1}{nm}\sum_{k=1}^{n}\sum_{u=1}^{m} Z_h^{ijk}(t_u)\left\{\sum_{j'\neq i}^{p}\left[s^{ij'}(t_u) - \sum_{h=1}^{J_n}\beta_h^{ij}B_h^{ij}(t_u)\right]\sqrt{\frac{\sigma^{j'j'}(t_u)}{\sigma^{ii}(t_u)}}y_{j'}^k(t_u)\right\}$. Then one has

$$\widetilde{\mathbf{c}}_h^{ij}(\beta) = e_h^{ij} + \delta_h^{ij} + \varepsilon_h^{ij}(\beta).$$

Following similar arguments of Lemma 7 in Xue and Qu (2012), there exists a $c > 0$ such that

$$E\left(\max_{1\leq i<j\leq p, h\in J_{ij}}\left|e_h^{ij}\right|\right) \leq c\sqrt{\log(N_n)/(N_n nm)}.$$

Therefore, by Markov's inequality, one has

$$P\left(\max_{1\leq i<j\leq p, h\in J_{ij}}\left|e_h^{ij}\right| > \frac{\eta_n}{2}\right) \leq \frac{2c}{\eta_n}\sqrt{\frac{\log(N_n)}{N_n nm}} \to 0, \tag{3.8}$$

as $n\to\infty$, by condition (C8). On the other hand,

$$\begin{aligned}
\max_{1\leq i<j\leq p, h\in J_l}\left|\delta_h^{ij}\right| &\leq \max_{1\leq i<j\leq p, k\in J_{ij}}\frac{1}{nm}\sum_{i=1}^{p}\sum_{k=1}^{n}\sum_{u=1}^{m} Z_h^{ijk}(t_u)\left(\sum_{l\neq i}^{p}\left[s^{il}(t_u) - \rho^{il}(t_u)\right]\sqrt{\frac{\sigma^{ll}(t_u)}{\sigma^{ii}(t_u)}}y_l^k(t_u)\right)\\
&\leq cN_n^{-(p+1)}\frac{1}{nm}\max_{1\leq i<j\leq p, k\in J_{ij}}\sum_{i=1}^{p}\sum_{k=1}^{n}\sum_{u=1}^{m}\left|Z_h^{ijk}(t_u)\right|\sum_{j'\neq i}^{p}\sqrt{\frac{\sigma^{ll}(t_u)}{\sigma^{ii}(t_u)}}\left|y_l^k(t_u)\right|\\
&\leq c\left(N_n^{-(p+1)} + \lambda_n\right)/N_n.
\end{aligned} \tag{3.9}$$

Finally, by the definition of $\widehat{\beta}^{(0)}$, one has $\varepsilon_h^{ij}\left(\widehat{\beta}^{(0)}\right) = 0$, for $h\in J_l$. Then Lemma 3.7.6 follows from (3.8) and (3.9). □

3.7.1 Proof of Theorem 2.4.1

For notation simplicity, we assume $w_{iu} = 1$ in (2.3). The proof for the general form of w_{iu} follows similarly under condition (C1). Let $c(\beta) = \frac{1}{2nm}\sum_{i=1}^{p}\sum_{k=1}^{n}\sum_{u=1}^{m}\left(y_i^k(t_u) - \sum_{j\neq i}^{p}\sum_{h=1}^{J_n}\beta_h^{ij}B_h^{ij}(t_u)\sqrt{\frac{\sigma^{jj}(t_u)}{\sigma^{ii}(t_u)}}y_j^k(t_u)\right)^2$,

$c_h^{ij}(\beta) = \partial c(\beta)/\partial \beta_h^{ij}$, and $\overline{c}_h^{ij}\left(\widehat{\beta}\right) = \left(c_h^{ij}(\beta) \vdots, \ldots, \vdots c_{h+p}^{ij}(\beta)\right)$. By the KKT condition, $\widehat{\beta}$ is the solution of the minimization problem (2.4) if and only if

$$\begin{aligned} c_h^{ij}\left(\widehat{\beta}\right) + \sum_{s=\max(h-p,1)}^{\min(h,N_n+1)} \frac{\lambda_n w_s^{ij}}{\left\|\widehat{\gamma}_s^{ij}\right\|}\widehat{\beta}_h^{ij} &= 0, \quad if \quad \widehat{\beta}_k^{ij} \neq 0, \\ \left\|\overline{c}_k^{ij}\left(\widehat{\beta}\right)\right\|_2 &\leq \lambda_n w_s^{ij}, \quad if \quad \widehat{\beta}_k^{ij} = 0. \end{aligned} \tag{3.10}$$

Let $\beta^{ij} = \left(\beta_1^{ij}, \ldots, \beta_{J_n}^{ij}\right)^T$, and $\beta_{\mathcal{J}_{ij}}^{ij} = \left(\beta_k^{ij}, k \in \mathcal{J}_{ij}\right)^T$. Define $\beta_{\mathcal{J}_{ij}^C}^{ij}$ similarly. Let $\widehat{\beta} = \left(\widehat{\beta}^{ij}, 1 \leq i < j \leq p\right)$ such that for each $\widehat{\beta}^{ij}$ with $\widehat{\beta}_{\mathcal{J}_{ij}^C}^{ij} = 0$ and $\widehat{\beta}_{\mathcal{J}_{ij}}^{ij}$ solving

$$c_k^{ij}(\beta) + \sum_{s=\max(k-p,1)}^{\min(k,N_n+1)} \frac{\lambda_n w_s^{ij}}{\left\|\widehat{\gamma}_s^{ij}\right\|}\beta_k^{ij} = 0, \tag{3.11}$$

for $k \in J_{ij}$. Then Lemma 3.7.6 and condition (C9) entail that

$$P\left(\max_{1\leq i<j\leq p, k\in J_l} \left\|\overline{\mathbf{c}}_k^{ij}\left(\widehat{\beta}\right)\right\|_2 \geq \lambda_n w_k^{ij}\right) \to 0.$$

Therefore, $\widehat{\beta}$ satisfies the KKT condition, and is the solution of the adaptive Lasso objective function. Now let $\widehat{\rho}^{ij} = \left(\widehat{\beta}^{ij}\right)^T B$ be the corresponding estimator of the partial correlation functions. It is clear from the definition of $\widehat{\beta}^{ij}$ that $\widehat{\rho}^{ij}(t) = 0$ for $t \in E^{ij} = [e_1^{ij}, e_2^{ij}]$. We now show that $\sup_{0\leq t\leq 1}\left|\widehat{\rho}^{ij}(t) - \rho^{ij}(t)\right| = O_p\left(\frac{N_n^{3/2}}{\sqrt{n}} + N_n^{-(p+1)}\right)$. Note that for each $1 \leq i < j \leq p$ and $k \in J_l$, $\widetilde{\beta}_k^{ij} = \widehat{\beta}_k^{ij} = 0$, and for any $k \in J_{ij}$, write

$$\widehat{\beta}_k^{ij} = \widetilde{\beta}_k^{ij} + \boldsymbol{\delta}_k^{ij},$$

for some δ_k^{ij}. Let $\delta_n = \left(\boldsymbol{\delta}_k^{ij}, 1 \leq i < j \leq p, k = 1, \ldots, J_n\right)$. Then by (3.11), one has for $k \in J_{ij}$,

$$c_k^{ij}\left(\widetilde{\beta} + \boldsymbol{\delta}_n\right) + \sum_{s=\max(k-p,1)}^{\min(k,N_n+1)} \frac{\lambda_n w_s^{ij}}{\left\|\widehat{\gamma}_s^{ij}\right\|}\left(\widetilde{\beta}_k^{ij} + \boldsymbol{\delta}_k^{ij}\right) = 0.$$

Or in matrix form, one has

$$\mathbf{C}_{J_{ij}}^{ij}\left(\widetilde{\beta} + \boldsymbol{\delta}_n\right) + \mathbf{W}_{J_{ij}}^{ij}\left(\widetilde{\beta}_{\mathcal{J}_{ij}}^{ij} + \boldsymbol{\delta}_{\mathcal{J}_{ij}}^{ij}\right) = 0,$$

where $C_{J_{ij}}^{ij} = \left(c_k^{ij}, k \in J_{ij}\right)^T$, and $W_{J_{ij}}^{ij} = diag\left\{\left(\mathbf{w}_k^{ij}, k \in J_{ij}\right)\right\}$ with $w_k^{ij} = \sum_{s=\max(k-p,1)}^{\min(k,N_n+1)} \lambda_n w_s^{ij} / \left\|\widehat{\boldsymbol{\gamma}}_s^{ij}\right\|$. Note that, by the definition of $\widetilde{\beta}$, $C_{J_{ij}}^{ij}\widetilde{\beta} = 0$. Let $C_n = \left(\mathbf{C}_{J_{ij}}^{ij}, 1 \le i < j \le p\right)$, $W_n = \left(\mathbf{W}_{J_{ij}}^{ij}, 1 \le i < j \le p\right)$, and $\widetilde{\beta}^{(A)} = \left(\widetilde{\beta}_{\mathcal{J}_{ij}}^{ij}, 1 \le i < j \le p\right)$. Then one has

$$\boldsymbol{\delta}_n = \left(\mathbf{C}_n + \mathbf{W}_n\right)^{-1} \mathbf{W}_n \widetilde{\beta}^{(A)},$$

and

$$\widehat{\rho}^{ij}(t) - \widetilde{\rho}^{ij}(t) = \left(\mathbf{B}(t)\right)^T \delta_n = \left(\mathbf{B}(t)\right)^T \left(\mathbf{C}_n + \mathbf{W}_n\right)^{-1} \mathbf{W}_n \widetilde{\beta}^{(A)},$$

in which the eigenvalues of C_n are of order $1/N_n$ by Lemma 3.7.3. Note that the diagonal matrix W_n with its elements $w_{lj}^* = \sum_{s=\max(j-p,1)}^{\min(j,N_n+1)} \lambda_n w_{nls} / \left\|\widehat{\boldsymbol{\gamma}}_{ls}^{AL}\right\| = \lambda_n O_p(1)$. Therefore

$$\begin{aligned}
\sup_{0<t<1} \left|\widehat{\rho}^{ij}(t) - \widetilde{\rho}^{ij}(t)\right| &\le N_n\lambda_n \sup_{0<t<1} \left|\mathbf{B}_{J_{ij}^C}^T(t)\, \widetilde{\beta}_{\mathcal{J}_{ij}}^{ij}\right| = N_n\lambda_n \sup_{0<t<1} \left|\widetilde{\rho}^{ij}(t)\right| \\
&= O_p(N_n\lambda_n) = o\left(\frac{N_n^{3/2}}{\sqrt{nm}}\right).
\end{aligned}$$

Then Theorem 2.4.1 follows from the triangular inequality and condition (C8). □