Summer 2016

# Structural exploration and inference of the network

Ruihua Cheng
*New Jersey Institute of Technology*

# Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

# ABSTRACT

## STRUCTURAL EXPLORATION AND INFERENCE OF THE NETWORK

**by**
**Ruihua Cheng**

This dissertation consists of two parts. In the first part, a learning-based method for classification of online reviews that achieves better classification accuracy is extended. Automatic sentiment classification is becoming a popular and effective way to help online users or companies to process and make sense of customer reviews. The method combines two recent developments. First, valence shifters and individual opinion words are combined as bigrams to use in an ordinal margin classifier. Second, relational information between unigrams expressed in the form of a graph is used to constrain the parameters of the classifier. By combining these two components, it is possible to extract more of the unstructured information present in the data than previous methods, like support vector machine, random forest, hence gaining the potential of better performance. Indeed, the results show a higher classification accuracy on empirical real data with ground truth as well as on simulated data.

The second part deals with graphical models. Gaussian graphical models are useful to explore conditional dependence relationships between random variables through estimation of the inverse covariance matrix of a multivariate normal distribution. An estimator for such models appropriate for multiple graphs analysis in two groups is developed. Under this setting, inferring networks separately ignores the common structure, while inferring networks identically would mask the disparity. A generalized method which estimates multiple partial correlation matrices through linear regressions is proposed. The method pursues the sparsity for each matrix, similarities for matrices within each group, and the disparities for matrices between groups. This is achieved by a $\ell_1$ penalty and a $\ell_2$ penalty for the pursuit of sparseness

and clustering, and a metric that learns the true heterogeneity through optimization procedure. Theoretically, the asymptotic consistency for both constrained $\ell_0$ method and the proposed method to reconstruct the structures is shown. Its superior performance is illustrated via a number of simulated networks. An application to polychromatic flow cytometry data sets for network inference under different sets of conditions is also included.

# STRUCTURAL EXPLORATION AND INFERENCE OF THE NETWORK

by
Ruihua Cheng

A dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
Rutgers, The State University of New Jersey – Newark
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mathematical Sciences

Department of Mathematical Sciences
Department of Mathematics and Computer Science, Rutgers-Newark

August 2016

**APPROVAL PAGE**

**STRUCTURAL EXPLORATION AND INFERENCE OF THE NETWORK**

**Ruihua Cheng**

---

Dr. Ji Meng Loh, Dissertation Advisor — Date
Associate Professor of Mathematics, NJIT

---

Dr. Zhi Wei, Dissertation Co-Advisor — Date
Associate Professor of Computer Science, NJIT

---

Dr. Antai Wang, Committee Member — Date
Associate Professor of Mathematics, NJIT

---

Dr. Yixin Fang, Committee Member — Date
Assistant Professor of Population Health, NYU

---

Dr. Yi Chen, Committee Member — Date
Associate Professor of School of Management, NJIT

## BIOGRAPHICAL SKETCH

**Author:** Ruihua Cheng

**Degree:** Doctor of Philosophy

**Date:** August 2016

### Undergraduate and Graduate Education:

- Doctor of Philosophy in Mathematical Sciences,
  New Jersey Institute of Technology, Newark, NJ, USA, 2016
- Bachelor of Science in Information and Computing Science,
  Henan University of Economics and Law, Zhengzhou, Henan, CHN, 2012

**Major:** Probability and Applied Statistics

### Presentations and Publications:

Jie Zhang, Kuang Du, Ruihua Cheng, Zhi Wei, Chenguang Qin, Huaxin You, and Sha Hu, "Reliable Gender Prediction Based on Users' Video Viewing Behavior," *ICDM*, Submitted for Initial Review.

Ruihua Cheng, Learning-Based Method with Valence Shifters for Sentiment Analysis, *From Industrial Statistics to Data Science Conference*, University of Michigan, 2015.

Ruihua Cheng, Learning-Based Method with Valence Shifters for Sentiment Analysis, *Dana Knox Student Research Showcase*, New Jersey Institute of Technology, 2016.

*I dedicate this research to my family, who nursed me with affections, love and encouragement. Without their inspiration, guidance and dedication, I would not be able to get such success and honor.*

Ruihua Cheng

## ACKNOWLEDGMENT

My deepest gratitude is to my advisor, Dr. Ji Meng Loh and my coadvisor, Dr.Zhi Wei. I have been amazingly fortunate to have them give me the freedom and encouragement to explore research ideas while providing excellent guidance. Their persistent support and patience helped me overcome many difficult situations throughout my research. Without their continuous help this dissertation would not have been possible.

I would like to extend my gratitude to the other members of my dissertation committee, Dr. Antai Wang, Dr. Yixin Fang and Dr. Yi Chen for their support throughout my PhD.

I would also like to thank Dr. Jonathan Luke, and other faculty and staff members of the Department of Mathematical Sciences for their support during the last four years.

## TABLE OF CONTENTS

## TABLE OF CONTENTS
## (Continued)

**Chapter** **Page**

## LIST OF TABLES

## LIST OF FIGURES

**Figure** **Page**

## LIST OF FIGURES
## (Continued)

**Figure** **Page**

# CHAPTER 1

# INTRODUCTION

Text mining has become an important research area to automatically detect information contained in large numbers of text documents. In business, sentiment analysis, i.e., analyzing the positivity or negativity of text, has been applied to analyze users' comments, feedback or critiques towards a company's products or services. Due to ever-increasing amounts of such textual data, it is necessary to develop novel methods for improving the predictive accuracy of reviews.

There are two main approaches for detecting sentiment automatically. One approach uses lexicon-based methods to calculate a semantic orientation score of a document [61, 58, 29, 55, 56]. A second approach uses machine learning methods such as Support Vector Machine (SVM) classifiers on textual data using individual words as features or predictors [43, 48, 6, 3]. A recent work, [67], combined both approaches, using the lexicon-based method on the training data and a learning classifier on new data.

However, as [65] noted, at the individual word level, these methods do not distinguish differences in sentiment between words that have the same polarity, e.g., "good" and "great". At the level of phrases, these methods also fail to account for the modifying effects on sentiment of neighboring words, such as "very", "absolutely" or "extremely", which we call *intensifiers*. Our proposed method for sentiment classification aims to avoid the above two shortcomings.

To address the first shortcoming, we develop a statistic for polarity calculation that can distinguish and measure the difference in sentiment strength between opinion words or phrases. Furthermore, the relative sentiment between words or phrases are characterized by capturing the direction of their sentiment polarity. In our work, a

graph is constructed to capture and express all the relative sentiment strengths of opinion words or phrases.

To address the second shortcoming, motivated by the work of [44, 62, 33, 51], we take into account the effects on sentiment of neighboring words by including negators, valence intensifiers, and valence diminishers in our method. These three types of words can change the degree of the expressed sentiment. For example, a negation word like *not* reverses the sentiment of a opinion word. A valence intensifier like *deeply* in the phrase *deeply suspicious*, increases the intensity of the word *suspicious*. On the other hand, the valence diminisher *rather* in the phrase *rather efficient*, makes the statement less positive.

We combine the above two proposed extensions into a learning-based model for classification of sentiment. The main idea is to include valence shifters (intensifiers or diminishers) in the features and then training a learning-based model while simultaneously integrating the relative sentiments of features into the training procedure. The latter is achieved by using the graph of relative sentiments to provide constraints on the coefficients. By incorporating these two extensions, the new procedure may offer improvement in predictive accuracy.

Inference for graphical model has attracted a lot of attention in recent years, due to its advantage in gaining insights into patterns of association among observed variables. Many problems from such fields as biology, computer vision, and medicine [26, 31, 64], which often generate very high-dimensional data sets with moderate sample sizes, can be solved by the estimation of the partial correlation matrix, also known as the concentration or precision matrix. The goal is to discover the conditional independence in graphical models from a set of independent and identically distributed observations.

There are many methods in both statistics and computer science that have been devoted to the study of graphical models. Classical approaches are the greedy forward

or backward stepwise selection methods [13]. The forward selection method starts by adding the most significant edge into the empty set, and continues adding edges until a suitable stopping criterion based on an individual partial correlation test is satisfied. However, this procedure is computationally infeasible for high-dimensional data. Furthermore, this method does not correctly account for multiple testing [12]. Drton and Perlman [15] proposed a simultaneous testing procedure to control the overall error rate for the inclusion of incorrect edge. Nevertheless, it is too conservative due to its applicability only on low-dimensional data sets with a large number of observations. Other methods include Bayesian network modeling [21, 50] which can effectively infer networks by extracting meaningful insights from data sets, and Gaussian graphical modeling [49] which can determine which elements of the inverse covariance matrix are zero by a thresholding and false discovery approach.

Recent methods take the potential sparsity into account in the estimation step of the precision matrix. Since the network is sparse, the assumption that most variable pairs are conditionally independent under normality is reasonable for many real life problems. It has been shown in the literature that most genetic networks contains many genes with few interactions, and therefore is intrinsically sparse [57, 32, 24] . Representative examples in this category include Dobra et al. [14], who presented a novel Bayesian framework for building Gaussian graphical models by converting the dependency networks into compositional networks using the Cholesky decomposition. Moreover, Bickel and Levina [4] proposed to regularize covariance by banding the Cholesky factor and showed consistency of banded estimators in the operator norm under mild conditions. Huang et al. [30] proposed adding an $\ell_1$ penalty in the modified Cholesky decomposition step of the concentration matrix. The implicit regularizing assumption underlying those approaches is that variables which far apart have weak partial correlations [46]. But the Cholesky decomposition naturally requires ordering restriction of the variables, which makes the procedure

computationally intensive and even infeasible in that it has to determine the order of variables [38]. There are also a lot of literature that focus on the general case when the ordering of variables unavailable. A penalized maximum likelihood framework with an $\ell_1$ penalty imposed on the partial correlation estimation have been employed by [66, 2, 11, 20, 46], who adapt different interior-point optimization methods for computing the estimator. Li and Gui [38] considered a threshold gradient descent regularization procedure, in which the sparsity is accounted for by defining a loss function- the negative of the log likelihood function. Those approaches can be applied to situations where the number of samples is small relative to the number of dimensions.

The aforementioned literature mainly focused on estimating a single Gaussian graphical model. Nevertheless, it is more realistic in many applications to have multiple undirected graphs in a single group that observations correspond to distinct categories. There are some prior studies on estimating multiple graphical models for a single group. Guo et al. [26] proposed imposing an $\ell_1$ penalty on the common factors, which encourages the sparsity and similarity of edges across all individuals in the group and a second $\ell_1$ penalty on the category-specific features to allow edges to be specifically set to zero. Danaher et al. [10] introduced the fused penalty and group penalty simultaneously to encourage shared structure across all individuals. Zhu et al. [68] ever proposed the regularized maximum likelihood estimation method with nonconvex penalty for the pursuit of sparseness across all individuals and clustering among individuals in the group. Yajima et al. [63] used a Bayesian approach with stochastic simulation to jointly estimated the strength of association for the baseline group and differential group.

In this article, extending the prior studies on single group network inference, we consider the problem of two groups. As Friston [22] noted, such a situation often arises in brain region connectivity patterns estimation. The patterns vary

between different subjects both in healthy group and diseased group. In this case, one might want to estimate multiple graphical models for the healthy group and multiple graphical models for the diseased group. One would expect the graphical models within each group to be similar to each other, while that in different groups are allowed to vary from similar to unique since involved subjects not only share many common demographic or other covariate features, but also have considerable disparities arising from the fact that brain region connectivity patterns are often dysregulated in patient. In such situations, inferring the networks separately for each individual ignores the substantial commonality among the true graphical models. Conversely, failing to consider either the disparities between individuals within specific group or the heterogeneity among two groups in the graphical models may lead to inconsistent results. In order to make better use of the data, we need a principled method that jointly estimating two groups not only encourage common structure within the specific group but also allows for group-wise differences or certain similarity. In fact, the differences between the graphical models may be of scientific interest. To the best of our knowledge, this problem has not been properly addressed before.

Meinshausen and Bühlmann [41] ever proposed neighboorhood selection for each node separately by fiting LASSO model and showed that this is an approximation to the exact problem [66, 2, 9]. Based this method for single network inference, we propose extending it to the problem of estimating two groups of individual networks. Unlike the separation of each LASSO regression, we merge all adaptive LASSO linear regressions into a single learning model to simultaneously perform neighborhood selection for all nodes and all individuals.

Our model goes beyond inferring each network separately, and is especially useful when the sample size is relatively small. Based on prior knowledge of two groups, we add $\ell_2$ regularization into group-specific features. Borrowing information across networks within each specific group can encourage common structures and

reduce the variance of the estimates. Differing from the previous method for single group network inference, we simultaneously add a weight parameter denoted $a$ to constrain the difference between the two groups. We propose using two different ways to learn the value of $a$. First, consider $a$ to be part of the $\ell_2$ penalty and learn its value through cross validation method. This parametric-model-free approach is more commonly used in various statistics and machine learning literature. In this way, the proposed method is considered to be the regularized method as described in Sub-Section 3.1.2. We present the corresponding optimization procedure in Section 3.2 and theoretical properties in Section 3.3. The second way to tune this weight parameter is using a data-driven based adaptive method as described in Section 3.4, where $a$ represents the ratio of the expected network distance within the same group to that between two groups. It is a critical method to explicitly detect the underlying heterogeneity between two groups. By doing this, we gain the ability to learn insights on how strongly the true graph structures for the two groups are related. Our model is more flexible compared with previous methods. We use adaptive LASSO, not only for yielding a sparse solution, but also to improve the learned features by taking into account the prior knowledge through shared adaptive weights to regularize all individuals. Moreover, we can learn the importance score of selected features for differentiating the individuals which belong to different groups through a feature selection indicator. Theoretically, we provide the grouping property and derive finite-sample error bounds under certain conditions by the global minimizers of $\ell_0$-constrained method defined in (3.10) and its computationally surrogate, our proposed two methods defined in (3.2) in Section 3.3 and Section 3.4. The $\ell_0$ method is most general and ideal, but is computationally intensive. Hence, we propose the $\ell_1$ or $\ell_2$-penalized method, which is much more computationally effective to find solutions. Both methods can consistently reconstruct the sparsity, group-specific commonality and group-wise heterogeneity. Empirically, we demonstrate the effectiveness and

stability of our methods especially the adaptive method compared against competing methods, especially when the heterogeneity between two groups is large and present an application of the proposed adaptive method to signaling network inference.

The dissertation is organized as follows. Chapter 2 contains the first proposed project on sentiment analysis. Section 2.1 introduces related work on proposed valence shifters, semantic relationship and the prevailing learning-based classifier. We describe the methodology in detail in Section 2.2. The asymptotic consistency of proposed method is shown in Section 2.3. A simulation study evaluating the performance of the proposed method and comparing it with competing methods is reported in Section 2.4. The proposed method is illustrated using three real datasets in Section 2.5: hotel reviews from TripAdvisor.com, movies reviews from Internet Movie Database (IMDb) archive and restaurant reviews from OpenTable.com. Section 2.6 contains a discussion. Chapter 3 provides details for the estimation of Gaussian graphical models. Section 3.1 first introduces the joint sparse regression penalized method. Section 3.2 presents the optimization procedure for the proposed method. Section 3.3 illustrates the group effect of proposed model and shows the asymptotic consistency for the $\ell_0$-constrained method and our proposed method. Section 3.4 describes the proposed data-driven based adaptive method with its theoretical results. Section 3.5 includes simulation results. Section 3.6 demonstrates an application to signaling network inference. We briefly discuss the methods and results in section 3.7. Finally, the appendix contains proofs.

# CHAPTER 2

# PROPOSED LEARNING-BASED METHOD WITH VALENCE SHIFTERS FOR SENTIMENT ANALYSIS

## 2.1 Background and Related Work

In this section, we provide some background and review related work on sentiment analysis.

### 2.1.1 Contextual Valence Shift

The valence of a word used by an individual in say, a review, is defined as the degree of positivity or negativity of that word in conveying how that individual feels toward the subject of the review. Valence shifters are additional words that can modify the degree to which the original word is positive or negative. Table 2.1 below lists some examples of English words which can be used to intensify or diminish valence.

**Table 2.1** Examples of Words with Valence

| PART OF SPEECH | *Intensified Valence* | *Diminished Valence* |
|---|---|---|
| Adverbs | definitely, very, extremely | somewhat, barely, less |
| Adjectives | bright, authentic | worthless, weak, rough |
| Verbs | ensure, improve, assure | fail, discourage |
| Nouns | benefit, favor | disaster, bankruptcy |

In a sentence like *This hotel looked very good*, the phrase *very good* combines an adverb *very* with a polar adjective *good*. Here, the word *very* intensifies the valence so that *very good* should be considered more positive than *good* even through *very* on

its own, does not indicate any sentiment. Combining *very* with a negative adjective, like *expensive*, i.e., *very expensive*, on the other hand, should be characterized as more negative than *expensive*. To give another example, the sentence *This hotel is somewhat small*, the term *somewhat* diminishes valence, making this statement less negative. Hence, such valence shifters can play an important role in assessing sentiment in reviews. Section 2.2 contains a more detailed discussion.

### 2.1.2 Determining Semantic Word Similarity

Since the phrase *very good* has a higher polarity than *good*, and the two words *good* and *great* also have different positive polarities, we introduce two main methods to measure semantic word similarity, Latent Semantic Analysis (LSA) and Pointwise Mutual Information (PMI).

**Latent Semantic Analysis (LSA)** [37, 16] introduced LSA as a mathematical learning method to infer the contextual similarity of words for a large corpus of text. LSA word similarity relies on the distributional hypothesis that words occurring in the same contexts tend to have similar meanings. Table 2.2 shows a summary of the algorithm for LSA. Given a large corpus of text, LSA first creates a term-document matrix to capture the occurrences of terms in the documents (Step 1). Then local and global weighting functions are defined (Steps 2 and 3, respectively) and combined (Step 4) to reflect each word's importance. Singular value decomposition is applied to the reweighted matrix to obtain a new matrix with smaller dimensionality (Steps 5 and 6), where each word is represented by a vector in this new matrix. The similarity between two words is defined as cosine angle between their corresponding vectors.

**Pointwise Mutual Information (PMI)** A second method measuring semantic word similarity is PMI. [58] estimated word similarities by calculating PMI scores based on AltaVista's NEAR operator. For any two words, $A$ and $B$, the NEAR

**Table 2.2** A Description of LSA

1. Compute the matrix $C$ of word-by-document occurrences: $C[i, j]$, which represents how many times word $i$ occurs in document $j$.
2. Compute $LC$ from $C$ such $LC[i, j] = \log(1 + C[i, j])$.
3. Compute the entropy $H[i]$ of word $i$ as

$$H[i] = \Sigma_i - C[i, j]\log C[i, j].$$

4. Normalize the entries in $LC$: $N[i, j] = LC[i, j]/H[i]$.
5. Use singular value decomposition on $LC$ to obtain a matrix $Q$ of dimensionality $k$.
6. A word $i$ is represented as the vector $Q[i]$ and the similarity between words $i$ and $j$ is $\cos(Q[i], Q[j])$.

operator finds the number of documents (as determined by an AltaVista search) that contains these two words separated by at most a few words apart. The PMI between two words $A$ and $B$ captures how likely it is to find $B$ in a text given that the text contains $A$. It is a co-occurrence metric, in that it normalizes the probability of co-occurrence of two words with their individual probabilities of occurrence. The PMI between $A$ and $B$ can be calculated as:

$$PMI(A, B) = \log \frac{p(A, B)}{p(A)p(B)}.$$

The similarity between words $A$ and $B$ is then taken to be their PMI score.

**Comparison between LSA and PMI** [45] conducted a comparison of PMI with LSA and found that LSA outperforms PMI on a wide variety of evaluation tasks. Later, [27] showed that a combination of LSA and WordNet [18] performed well at measuring Semantic Textual Similarity for the Stanford Webbase corpus. WordNet is a large electronic database of English, where nouns, verbs, adjectives and adverbs are organized into synonyms, each expressing a distinct concept. The Stanford Webbase corpus is a dataset containing a collection of English paragraphs with more than three billion words obtained from the Stanford WebBase project in Feb 2007. It is one of the largest collections of textual data with balanced text and contains 100 million web pages from more than 50,000 websites. [45] also showed the Stanford Webbase Corpus is a good representation of everyday use of the English language.

The method proposed by [27] for measuring word similarity first uses LSA to obtain basic word similarity scores. Then it adjusts the scores by incorporating WordNet's information about synonyms so that the final similarity score between words $x$ and $y$ given by

$$sim(x, y) = sim_{LSA}(x, y) + 0.5e^{-\alpha D(x,y)}, \tag{2.1}$$

where $D(x, y) \in \{0, 1, 2, 3\}$ is the minimal path distance between $x$ and $y$. The path distance is based on WordNet information about synonyms illustrated by [39] in Figure 2.1, which is a fragment of the semantic hierarchy of WordNet. The closer $x$ and $y$ are related, the smaller the value of $D(x, y)$. For example, the shortest path between boy and girl is boy-male-person-female-girl, so the minimum length of path is 4. [39] suggests setting $\alpha$ to be 0.2, based on their experimental results. We will use this method to construct our graph of relational information between words or phrases.

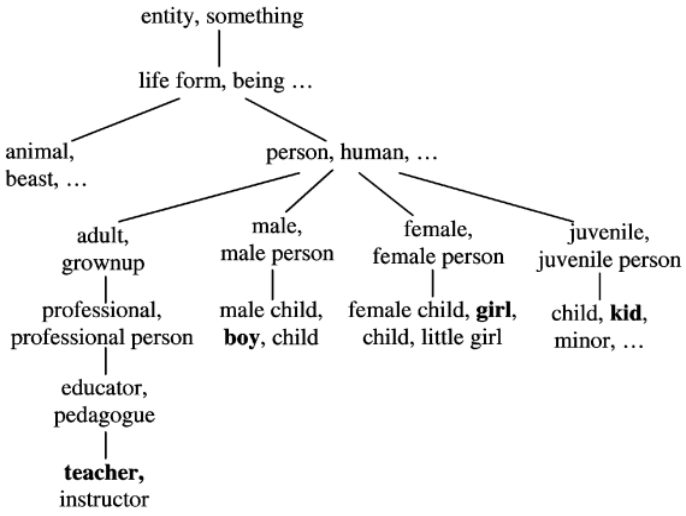



**Figure 2.1:** Hierarchical semantic knowledge base. "$\cdots$" indicates that some words in the class were omitted to save space.

### 2.1.3 Random Forests (RF)

The Random Forests (RF) method was introduced by [5] for feature (variable) selection by ranking the importance of variables and improved predictions for decision tree models. RF feature selection is a combination of variable subset selection and bootstrapping with variable ranking. The main idea is to generate a vast number of decision trees, which are used to determine the most popular variables based on performance. Similarly, to classify a new object from an input vector, the input vector is passed down each tree in the forest. Each tree yields a classification, referred to

as a "vote" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

Each tree is grown as follows: First, if the number of cases in the training set is $N$, sample $N$ cases at random with replacement from the original data. The sample will be the training set for growing the tree. Second, if there are $M$ input variables, $m$ variables with $m \ll M$, are randomly selected at each node and used to determine the best split at that node. The value of $m$ is held constant throughout the procedure. Each tree is grown to the largest extent possible, without any pruning.

### 2.1.4 Machine Learning Methods: SVM and LMOC

**Support Vector Machine (SVM)** Support Vector Machine [8] is a discriminative classifier by finding the best hyperplane that separates all data points of one class from those of another class. Most learning techniques do not perform well on datasets where the number of features is large compared to the sample size. SVMs are believed to be an exception [23]. To understand our methodology, familiarity with linear SVMs is required, and a brief introduction follows.

**Two-class SVM** Consider the simplest case: two class-classification, that is given training data $\{X_1, ..., X_n\}$ that are vectors in some feature space $X \subseteq \mathbb{R}^d$, we use hyperplanes to separate the data according to their labels $\{y_1, .., y_n\}$ where $y_i$ falls within two classes, $y_i \in \{-1, 1\}$. The idea is to find hyperplanes that separate the training data by as wide a margin as possible, see Figure 2.2.

All vectors lying on one side of the hyperplane as labeled as -1, and all vectors lying on the other side are labeled as 1, hence the decision function is

$$f(x) = sign(\boldsymbol{\beta}^T x + \beta_0). \tag{2.2}$$

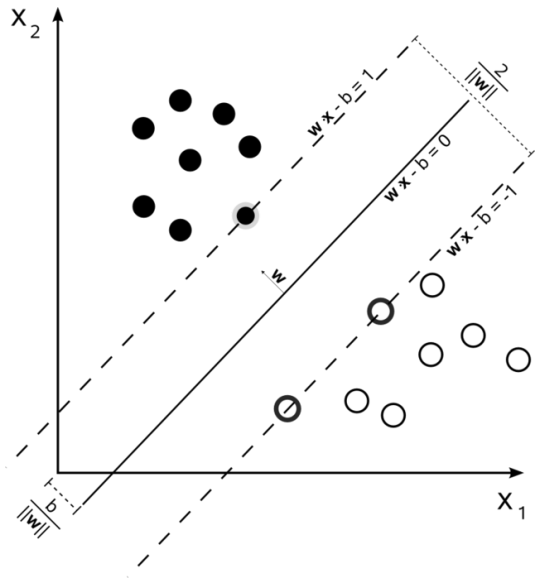



**Figure 2.2:** A simple linear support vector machine.

The training instances that lie closest to the hyperplane are called support vectors. The SVM optimization criterion is

$$\min_{\boldsymbol{\beta}} \frac{||\boldsymbol{\beta}||^2}{2} \text{ subject to } \; y_i(\boldsymbol{\beta}^T x_i + \beta_0) \geq 1, \text{for all } i. \tag{2.3}$$

In practice, one allows for error terms in case there is no hyperplane:

$$\min_{\boldsymbol{\beta},\beta_0,\xi_i} \left( \frac{||\boldsymbol{\beta}||^2}{2} + C \sum_i \xi_i \right) \text{subject to } y_i(\boldsymbol{\beta}^T x_i + \beta_0) \geq 1 - \xi_i, \text{for all } i, \tag{2.4}$$

where $C$ is a regularization parameter that balances between margin size and training error. The quantities $\xi_i$, called slack variables, measure the degree of misclassification.

One can solve this with Lagrange multipliers so that $\beta$ is given by $\beta = \sum_i \alpha_i y_i x_i$. The vectors $x_i$ for which $\alpha_i$ are non-zero are the support vectors. Optimization criterion thus becomes

$$\max_{\alpha_i} L_d = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j \text{ subject to } \sum_i \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C. \tag{2.5}$$

The SVM finds the $\alpha_i$ that correspond to the maximal margin hyperplane.

In general, the dot product can be replaced by a kernel matrix $K(i,j) = \phi(x_i) \cdot \phi(x_j)$ or

a positive definite matrix K. Then the decision function can equivalently be expressed as

$$f(x) = sign(\sum_i \alpha_i y_i K(x_i, x) + \beta_0). \tag{2.6}$$

By choosing different kernel functions, the training data $X$ can be projected onto a different space such that hyperplanes in the new space correspond to more complex decision boundaries in the original space $X$.

**Multi-class SVM** To perform SVM where there are more than two classes or labels, a popular method is called the one-against-one method, introduced in [34], where $k(k-1)/2$ classifiers are constructed. Each classifier is trained on data from two classes. For training data from the $i$-th and the $j$-th classes, the optimal criterion is to solve the binary classification problem:

$$\begin{aligned} \min_{\beta^{ij}, b^{ij}, \xi^{ij}} & \left( \frac{||\beta^{ij}||^2}{2} + C \sum_t \xi^{ij} \right) \\ (\beta^{ij})^T \phi(x_t) + \beta_0^{ij} &\geq 1 - \xi^{ij}, \text{ if } y_t = i, \\ (\beta^{ij})^T \phi(x_t) + \beta_0^{ij} &\geq -1 + \xi^{ij}, \text{ if } y_t = j, \\ \xi_t^{ij} &\geq 0, \end{aligned} \tag{2.7}$$

where the first term measures the margin size and the second term is the training error. The quantities $\xi^{ij}$ measures the degree of misclassification.

There are different methods for using the $k(k-1)/2$ constructed classifiers for predicting new labels. One method is to use a voting strategy suggested in [19]: if the $i-j$ classifier $sign((\beta^{ij})^T \phi(x) + \beta_0^{ij})$ indicates that $x$ is in the $i$-th class (as opposed to the $j$-th class), then the vote for the $i$-th class is increased by one. Otherwise, the $j$-th is increased by one. All the votes are added and the decision is to predict $x$ to

be in the class with the largest vote. The implementation and the application on real data is shown in Sections 2.3 and 2.4.

**Large Margin Ordinal Classifier (LMOC)** Like SVM, the Large Margin Ordinal Classifier [60] also seeks a maximal margin hyperplane to separate classes. However, there are two differences: (1) LMOC uses parallel hyperplanes to separate data points with different labels, hence only the different intercepts for the hyperplanes are needed to categorize new data instances; (2) LMOC integrates the relationships between features (predictors) into a set of linear constraints on the classification coefficients, resulting in a large reduction in the parameter space. The following is a brief introduction of the model.

Suppose we have training data $x_1, ..., x_n$ that are vectors in some space $x \subseteq \mathbb{R}^d$, with labels $\{c_1, .., c_n\}$. Assume we have K outcomes with an ordering $1 \prec \cdots \prec K$, so that $c_i \in \{1, ..., K\}$. Let $\mathbf{f} = (f_1, ..., f_{K-1})^T$ be a classification function with $f_k(x) = \boldsymbol{\beta}^T x + \beta_{0k}$ representing the ordered classes up to class $k$, i.e., $\{1, ..., K\}$, for $k = 1, ..., K-1$ with the constraint that $\beta_{01} \leq \beta_{02} \leq \cdots \leq \beta_{0,K-1}$. This functional representation yields $K-1$ parallel hyperplanes, where $f_1, ..., f_{K-1}$ differ only in intercepts $\beta_{0j}$'s, and $\{x : f_k(x) = 0\}$ is the partition boundary separating classes $\{1, ..., k\}$ and $\{k+1, ..., K\}$. In this situation, $f_k(x) < 0$ implies that $f_j(x) < 0$ for all $j \leq k$, which indicates that the predictor vector $x$ has a response $c$ larger than $k$. Therefore, the decision function rule for $x$ is:

$$\Phi(x) = \begin{cases} K & \text{if } f_k(x) < 0 \text{ for all } k = 1, ..., K-1. \\ \min\{k : f_k(x) \geq 0\} & \text{otherwise.} \end{cases} \tag{2.8}$$

Similar with SVM (as shown in Figure 2.2), for each of the $K-1$ hyperplanes, the geometric margin separating any two classes should be at least $2/||\boldsymbol{\beta}||_2$ in $L_2$ norm. In other words, the $L_2$ norm distance between two hyperplanes with $f_k(x) = \pm 1$

is $2/||\boldsymbol{\beta}||_2$. The optimal criterion is to maximize the margin, and simultaneously minimize the misclassification. Hence the classification problem can be formulated as

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \left( \frac{||\boldsymbol{\beta}||^2}{2} + \lambda \sum_{i=1}^{n} \sum_{k=1}^{K-1} \xi_{ik} \right) \\ &\text{subject to } \boldsymbol{\beta} \in \mathcal{B},\ sign(k - c_i)(x_i^T \boldsymbol{\beta} + \beta_{0k}) \geq 1 - \xi_{ik}, \\ &\xi_{ik} \geq 0, \beta_{01} \leq \beta_{02} \leq ... \leq \beta_{0,K-1}, \end{aligned} \tag{2.9}$$

where $\lambda$ is the regularization parameter that trades off margin size and training error, $\xi_{ik}$ the slack variables measuring degrees of misclassification, and $\mathcal{B}$ represents the relational constraints on the coefficients.

Figure 2.3 is an illustration of the Large Margin Ordinal Classification with $K = 5$, with $L_2$ norm distance between $f_3(x) = \pm 1$ being the geometric margin separating the ordered classes {1,2,3} and {4,5}. The decision boundary occurs at $f_3(x) = 0$. In this figure, there are three misclassified instances by $f_3(x)$, denoted as slack variables $\xi_{13}$, $\xi_{23}$, $\xi_{33}$. More details and numerical studies on real data are provided in Sections 2.3 and 2.4.

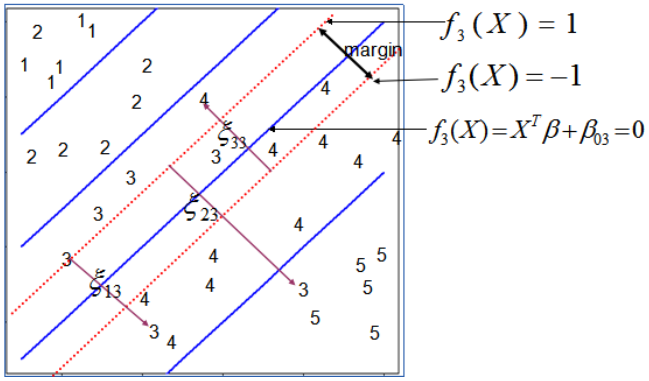



**Figure 2.3:** An illustration plot of LMOC.

### 2.1.5 Statistical Test for Two Classifiers

There are many different tests to compare the performance of two classifiers. One way is the $k$-fold Cross-Validated paired $t$ test [40].

Let the accuracy scores of the test folds for classifier A be $S_{A1}, ..., S_{Ak}$, and $S_{B1}, ..., S_{Bk}$ for classifier B, where each test set is independent of the others. Assume that the two classifiers have the same variance. The $k$-fold Cross-Validated paired $t$ test is simply a paired $t$ test on the paired differences

$$d_i = S_{Ai} - S_{Bi}, \ i = 1, 2, ...k,$$

for the hypotheses

$$H_0 : \mu = 0$$

$$H_1 : \mu > 0.$$

on $k-1$ degrees of freedom. A significant result indicates that classifier A has greater accuracy than classifier B.

## 2.2 Methodology

This section presents the proposed approach. Figure 2.4 gives an overview of the structure of our sentiment analysis procedure.

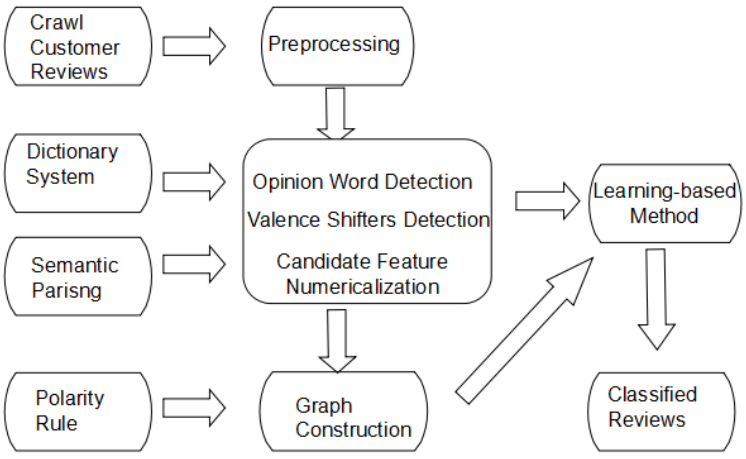



**Figure 2.4:** Structure of sentiment analysis procedure.

### 2.2.1 Reviews Data

Reviews are obtained from websites and the raw data set consists of a large number of reviews, each with a set of sentences and a rating score. For example, hotel reviews at TripAdvisor.com is a 5-star rating system, where customers describe their hotel experience with regards to criteria such as location, environment and food services, and finally give their rating score.

Before performing sentiment analysis, the raw data has to be processed. More specifically, this involves word and sentence and word segmentation, i.e., breaking down each review into a sequence of sentences and breaking down the sentences into an unordered list of meaningful words, disregarding grammatical structure.

### 2.2.2 Dictionary System

A dictionary system is needed to provide meaning, in our case positive or negative sentiment. Our dictionary system consists of two parts. One is a basic opinion dictionary (OD), which contains the list of positive and negative words. We use the English opinion lexicon dictionary [28] which contains 6,790 positive and negative words commonly used in opinion surveys. For example, the dictionary lists "good","great","nice","friendly" as positive words, and "bad", "absurd", "affront" as negative words.

The second is a valence-shifter dictionary (VSD), which contains the list of words that are negations, valence intensifiers, or valence diminishers. In ordinary language, we generally use words to modify the sentiment expressed by other words. For example, in a hotel review we might use expressions such as *not satisfied*, *less satisfied*, *satisfied* or *very satisfied* to show different levels of satisfaction with a hotel. Note that there are generally three ways of modifying sentiment expressed in the original word (*satisfied* in the example above). These are either negations (e.g., *not*), valence-intensifiers (e.g., *very*) or valence-diminishers (e.g., *less*). Negations are terms

that reverse the sentiment of a certain word [44]. Valence-intensifiers and valence-diminishers are terms that strengthen or weaken the degree of the expressed sentiment.

In order to accurately capture sentiment, we need to identify the presence of such terms in the sentiment analysis process. We do this using the General Inquirer system(GI) [54] which is available at http://www.wjh.harvard.edu/ inquirer/homecat. htm. In the GI system, valence-intensifiers and diminishers are known as overstatements and understatements respectively. The GI system contains 696 overstatements and 319 understatements. However, some words have several different definitions and so appear multiple times in the GI system. Also, a few of the words appear as overstatements and understatements. In these cases, we assign the word to the category it appears most often. Finally, we obtain 578 overstatements as valence-intensifiers, and 266 understatements as valence-diminishers.

The processed data for each review is matched with the two parts of the dictionary system to find the number of occurrences of each positive/negative word as well as any valence-shifters attached to these words. This is done through semantic parsing, described below.

**Semantic Parsing** The preprocessed reviews are parsed using the Stanford parser [7] to obtain the list of opinion words and valence shifters present in the data, often referred to as the "scope". The Stanford parser takes each word of the input text and works out any dependency information between words. As an example, consider the following review: *The room is not fancy, but very clean.* Figure 2.5 shows how parser works on this sentence[1]. Here *not fancy* is a negation of the opinion word *fancy.* The parser detects and assigns to it the *neg* dependency type. Similarly, *very* increases the degree of the sentiment word *clean*, and the parser identifies it as the dependency type, *advmod.*

[1]A demo is available at http://nlp.stanford.edu:8080/parser/index.jsp

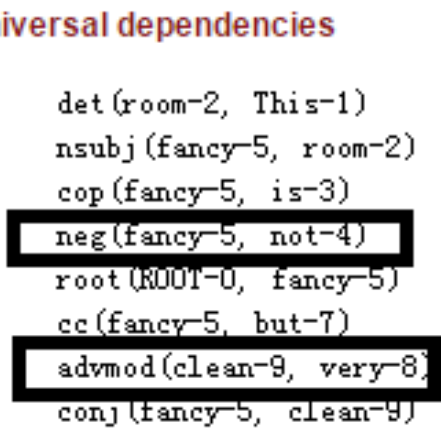



**Figure 2.5:** An example of the Stanford parser in action.

### 2.2.3 Candidate Feature Selection, and Numericalization

We describe here the construction of features using individual words (called unigrams) and combinations of words and valence shifters (called bigrams), and of the numericalization process of converting the features into numerical vectors.

**Candidate Feature Selection** The unigrams consist of individual words from the opinion dictionary OD, which has 6,790 words. The bigrams are combination of valence shifter with individual opinion word, such as "not good" or "rather efficient". The number of bigrams is the opinion dictionary size multiplied by the size of the valence shifter dictionary VSD, resulting in a total of about 6 million bigrams. However, if all these unigrams and bigrams are used as candidate features in the classification model, the resulting matrix will be very large but also very sparse.

To reduce the amount of noise, we need to perform additional feature integration. We follow the procedure proposed by [44]. First, we retain all the unigrams as basic candidate features. Instead of using all the bigrams, for each specific unigram word $w$, we construct 3 bigrams, denoted by $neg\_w$, $more\_w$, $less\_w$. Here $neg\_w$ represents any negations of opinion word $w$, $more\_w$ any intensifier of $w$ and $less\_w$ any diminisher of $w$. This greatly reduces the number of bigrams, since e.g.,

all negations are treated as identical etc. In this way, the final candidate feature set consists of 6,790 units (corresponding to the opinion dictionary size), with each unit having four terms, 1 unigrams and 3 bigrams, ($w$, $neg\_w$, $less\_w$, $w$, $more\_w$). Hence, the total feature size is 27,160.

**Candidate Feature Detection and Numericalization** Candidate feature detection and numericalization consists of going through the actual data and identifying which of the 27,160 elements in the candidate feature set are present, so that numerical vectors for the features are generated for use in the classification model.

We first use the semantic parser to detect all unigrams the opinion dictionary found in the reviews. For each review where a unigram $w$ occurs, we assign a value of 1 to this feature, otherwise it is set to 0. If the unigram is found, a further step is performed to check if there are valence shifters around this unigram. If for example, a valence intensifier belonging to the VSD is detected with this unigram, we assign a value of 1 to the bigram $more\_w$, otherwise we set it 0. This is done similarly for negations and diminishers. Note that we make our features binary. [43] showed that using the frequency of word occurrence instead may actually degrade the performance of the learning-based model.

### 2.2.4 Directed Graph Construction

The proposed method integrates the sentiment relations among words or phrases into the classification process, and we use a directed graph to express the relative sentiment strengths among candidate features.

The nodes of the graph represent sentiment candidate features including unigrams and bigrams, while the the directed edges $\rightarrow$ indicates the relative sentiment strength between two nodes. For example, a bigram *more beautiful*, representing an increasing sentiment, has a higher polarity than *beautiful*. Then we use *more beautiful*

$\rightarrow$ *beautiful* in the graph to indicate that the former has a higher strength than the latter.

In our setup, the bigrams are not a specific phrase but represents the general notion of negation, intensified or diminished valence. In other words, we only know they have different strength with unigrams (i.e., reversing, increasing or decreasing), but cannot directly measure their polarity. Hence, we construct the graph in two stages. The first proposed stage is to find a reliable statistic of polarity for any word or phrase. The second stage is to use the information contained in the two dictionary systems, the opinion dictionary and the valence shifter dictionary, to get the polarity of bigrams in nodes.

**Stage I: Sentiment Polarity for Word or Phrase** The sentiment polarity $SP(w)$ of a word or phrase $w$ is based on the difference of its semantic word similarity with two reference words, "good" and "bad":

$$SP(word) = \frac{sim(word, good) - sim(word, bad)}{sim(good, bad)}. \tag{2.10}$$

It essentially measures where each word/phrase $w$ stands on the spectrum between "good" and "bad". The semantic word similarity, in turn, is obtained using a statistic developed by [27], given in (2.1), which combines the results of two semantic analysis methods, LSA and Wordnet, applied on the Stanford Webbase Corpus. Therefore, based on the textual data contained in the corpus, each word $w$ can be measured for its similarity with "good" and with "bad", denoted as $sim(w, good)$ and $sim(w, bad)$ respectively.

We use their online service to obtain $sim(w, good)$ and $sim(w, bad)$ for every word or phrase $w$ in our candidate feature set and hence compute the sentiment polarity $SP(w)$.

**Stage II: Sentiment Polarity for Candidate Features** The sentiment polarity for each word can be computed using the procedure described above. To compute the sentiment polarity for our candidate features, we need to determine the polarity of each word $w$ as well as its associated bigrams, i.e., $more\_w$, $less\_w$, $neg\_w$.

In their linguistic analysis study, [44] proposed to do the following: all positive sentiment terms are assigned a value of 2, i.e., $w$=2 if $w$ is a positive word. If the word is preceded by an valence-intensifier in the same clause then a value of 3 is assigned, i.e., $more\_w$=3. If a valence-diminisher is in the same clause instead, then $less\_w$ is set to 1. On the other hand, negative sentiment terms are given a value of -2, i.e., $w$=-2 if $w$ is a negative word, and $less\_w$ is set to -1, or $more\_w$ set to -3 if $w$ is preceded by a valence-diminisher or a valence-intensifier respectively.

However, this method cannot recognize a difference in sentiment between words like "nice" and "good" which are both assigned the same polarity. Hence, we use the following method which avoids this drawback, and can provide different values for different opinion words. We do this using the dictionary systems and the polarity statistics given in (2.10).

We first introduce some notation. If $intv$ denotes a valence-intensifier, then the phrase with $intv$ combined with an opinion word $w$ is denoted by $intv\_w$, and its polarity by $SP(intv\_w)$. Similarly, the phrase $dimv\_w$ consisting of a valence-diminisher $dimv$ and the opinion word $w$ has polarity $SP(dim\_w)$.

We explain the scoring procedure using an example. For the word *good*, we compute the polarity of the bigram *more_good*, denoted by *SP(more_good)*, by averaging the polarities of the phrases which are valence-intensifiers combined with the word *good*. That is, the polarity of *more_good* is

$$SP(more\_good) = E(SP(intv\_good)).$$

Similarly,

$$SP(less\_good) = E(SP(dimv\_good)).$$

In addition, as [44] indicates, since a negation of a word is to reverse the sentiment of that word, the polarity of *neg_good* is the negative of the value of the polarity of *good*. i.e.,

$$SP(neg\_good) = -SP(good).$$

With the above procedure, we can compute the polarities of all the nodes in the graph, and the graph can be constructed by the partial orderings of each pair of different words or phrases as follows

$$phrase_1 \rightarrow phrase_2 \text{ if } SP(phrase_1) \geq SP(phrase_2).$$

For example, the graph will have $more\_satisfied \rightarrow satisfied$. In the next section, we use the graph in a classification system to reduce the parameter space and improve predictive accuracy.

### 2.2.5 Machine Learning Classifier

In our sentiment analysis, we use the candidate features together with the graph information in a learning-based model, like LMOC, SVM or RF to classify the reviews. In order to evaluate the effectiveness of adding contextual valence shifters and graph information into a classifier, we consider six classification systems. They are the basic LMOC, enhanced LMOC, basic SVM and enhanced SVM, basic RF, enhanced RF.

**Basic LMOC** The basic LMOC uses only unigrams as features, but incorporates the relational information between the unigrams as expressed in a graph.

Here we use only the unigrams from reviews that appear in opinion dictionary $D = \{w_1, .., w_d\}$ as positive or negative terms. Given $n$ textual documents $\mathbf{s}_i, i =$

$1, \ldots, n$ and their sentiment scores $c_i, i = 1, \ldots n$, we convert $\mathbf{s}_i$ to a numerical vector $\boldsymbol{x}(\mathbf{s}_i) = (x_1(\mathbf{s}_i), ..., x_d(\mathbf{s}_i))^T$, where $x_j(\mathbf{s}_i) = I(w_j \in \mathbf{s}_i)$ indicates absence and presence of $w_j$ in $\mathbf{s}_i$. The English opinion lexicon dictionary has 6,790 words. To incorporate the relational information expressed in graph into the classification system, we write the relational constraints as $\boldsymbol{\beta} \in \{\boldsymbol{\beta} : B\boldsymbol{\beta} \geq 0\}$, where $B$ is a constant matrix with 6,790 by 6,790 containing partial orderings between features. Then (2.9) can be formulated to solve:

$$\begin{gathered} \min_{\boldsymbol{\beta}} \left( \frac{||\boldsymbol{\beta}||^2}{2} + \lambda \sum_{i=1}^{n} \sum_{k=1}^{K-1} \xi_{ik} \right) \\ \text{subject to } B\boldsymbol{\beta} \geq 0,\ sign(k - c_i)(x_i^T \boldsymbol{\beta} + \beta_{0k}) \geq 1 - \xi_{ik}, \\ \xi_{ik} \geq 0, \beta_{01} \leq \beta_{02} \leq ... \leq \beta_{0,K-1}. \end{gathered} \tag{2.11}$$

**Enhanced LMOC** The Enhanced LMOC classifier extends the basic LMOC classifier by including bigrams which are combinations of valence shifters with the single words, as described in Sub-Section 2.2.4. That is, the candidate feature includes 6,790 units, each of which consists of four terms, $neg\_w$, $less\_w$, $w$, and $more\_w$. The total feature size is 27,160. This classification system also uses the relational information expressed in the graph, so the optimization procedure is formulated as function (2.11) as well, except that the matrix $B$ is 27,160 by 27,160 instead.

**Basic SVM** The feature set of the basic SVM classifier is the same as the basic LMOC, i.e., the features are only the 6,790 unigrams of all the positive and negative terms in the opinion dictionary. The values of the features are boolean, with a value of 1 if the feature word appears in the review, and 0 otherwise. However, any relational information in the graph is not used. The SVM optimization criterion is formulated in function (2.7).

**Table 2.3** Evaluation System Setup

| Parameters | Enhanced LMOC | Basic LMOC | Enhanced SVM | Basic SVM | Enhanced RF | Basic RF |
|---|---|---|---|---|---|---|
| Unigrams | Yes | Yes | Yes | Yes | Yes | Yes |
| Bigrams | Yes | No | Yes | No | Yes | No |
| Graph | Yes | Yes | No | No | No | No |

**Enhanced SVM** The feature set of the enhanced SVM classifier like that of the enhanced LMOC, consist of unigrams and their corresponding bigrams for a total of 27,160 features. However, the relational information is not used. Hence, the SVM optimization criterion is given by function (2.7) as well.

**Basic Random Forest (RF)** The feature set of the basic RF classifier is the same as the basic LMOC, i.e., the features are only the 6,790 unigrams of all the positive and negative terms in the opinion dictionary. The values of the features are boolean, with a value of 1 if the feature word appears in the review, and 0 otherwise. Similarly, any relational information in the graph is not used. The RF optimization criterion is given in Sub-Section 2.1.3.

**Enhanced RF** The feature set of the enhanced RF classifier like that of the enhanced LMOC, consist of unigrams and their valence shifting bigrams for a total of 27,160 features, but relational information is not used. Therefore, the RF optimization criteration is given by Sub-Section 2.1.3 as well.

Hence, Tables 2.3 and 2.4 summarize the components of these methods.

**Table 2.4** Evaluation System Setup for LMOC

| Systems | Graph Node | Graph Edge |
|---|---|---|
| Enhanced LMOC | Unigram + Bigram | semantic relationship between unigram and valence shifting bigram |
| Basic LMOC | Unigram | semantic relationship between unigram |

### 2.2.6 Test Error

In order to compare the classification methods, we need a method to evaluate the classification results. Suppose there are $K$ categories. The test error $TE$ is given by

$$\begin{aligned} TE &= \frac{1}{n_{test}} \sum_{1}^{n_{test}} l(C_i, C_i^{predicted}), \text{ where} \\ l(C_i, C_i^{predicted}) &= \frac{1}{K-1} |C_i - C_i^{predicted}|. \end{aligned} \tag{2.12}$$

Here, $n_{test}$ is the size of the test sample, $C_i$ is the true label for instance $x_i$. For SVM, $C_i^{predicted}$ is obtained using the voting strategy described in Sub-Section 2.1.4. The quantity $l(C_i, C_i^{predicted})$ represents the evaluation loss.

For LMOC, $C_i^{predicted}$ can be represented by the function $f(x_i)$, and so can be expressed in a more specific form:

$$\begin{aligned} TE &= \frac{1}{n_{test}} \sum_{1}^{n_{test}} l(C_i, f(x_i)) \text{ where} \\ l(C_i, f(x_i)) &= \frac{1}{K-1} \sum_{k=1}^{K-1} EI[f_k\{x_i\} sign(k - C_i) \leq 0]. \end{aligned} \tag{2.13}$$

Here, $f_k\{x\} sign(k - C_i) \leq 0$ implies that $x$ is misclassified by $f_k$.

## 2.3 Asymptotic Property

In this section, we give a finite-sample sentiment error bound for the proposed method, which is obtained by the minimizer of function (2.11) with both bigrams and unigrams as features.

The general form of (2.11) is

$$\min_{\boldsymbol{\beta}} \left( \frac{1}{n(K-1)} \sum_{i=1}^{n} \sum_{k=1}^{K-1} V(sign(k-c_i) f_k(\boldsymbol{x}(s_i))) + \lambda J(\boldsymbol{\beta}) \right) \quad \text{subject to } \boldsymbol{\beta} \in \mathcal{B},\ \beta_{01} < \beta_{02} < \cdots < \beta_{0,K-1}, \tag{2.14}$$

where $V(z) = \min(1, (1-z)_+)$ is the large-margin loss. Before proceeding, we introduce some notation. Let $L(\boldsymbol{y}, \boldsymbol{f}(\boldsymbol{x}(s))) = (K-1)^{-1} \sum_{k=1}^{K-1} V(y_k f_k(\boldsymbol{x}(s)))$ be the cost function in function (2.14) with $y_k = \text{sign}(k-c)$. Let $e_v(\boldsymbol{f}, \boldsymbol{f}^0) = (K-1)^{-1} \sum_{k=1}^{K-1} (EV(Y_k f_k(\boldsymbol{x}(S)))) - EV(Y_k f_k^0(\boldsymbol{x}(S)))$ be the error induced by using a margin loss V. Denote by $\mathcal{F}_{\mathcal{B}} = \{\boldsymbol{f} = (f_1, \cdots, f_{K-1}) : f_k(\boldsymbol{x}(s)) = \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}(s) + \beta_{0k}, \boldsymbol{\beta} \in \mathcal{B}, \beta_{01} < \beta_{02} < \cdots < \beta_{0,K-1}\}$ the parameter space, where the sentiment strength graph is built into the parameter space through relational constraints $\boldsymbol{\beta} \in \mathcal{B} = \{\boldsymbol{\beta} : B\boldsymbol{\beta} \geq 0\}$. The following technical assumptions are made.

**Assumption 1.** *For some positive sequence $\xi_{n,d,K} \to 0$, there exists $\boldsymbol{f}^* = (f_1^*, \cdots, f_{K-1}^*) \in \mathcal{F}_{\mathcal{B}}$ with $f_k^*(\boldsymbol{x}(s)) = (\tilde{\boldsymbol{\beta}}_k^*)^{\mathrm{T}} \tilde{x}^{\mathrm{T}}(s)$ such that $e_V(\boldsymbol{f}^*, \boldsymbol{f}^0) \leq \xi_{n,d,k}$.*

Assumption 1 requires that the ideal sentiment prediction function $\boldsymbol{f}^0$ can be well-approximated by $\mathcal{F}_{\mathcal{B}}$.

Next, we define a truncated $V^{\mathrm{T}}(y_k f_k(\boldsymbol{x}(s))) = \min(V(y_k f_k(\boldsymbol{x}(s))), T)$ for some constant $T \geq 0$ such that $\max(V(Y_k f_k^0(\boldsymbol{x}(S))), V(Y_k f_k^*(\boldsymbol{x}(S)))) \leq T$ almost surely. Further, let $L^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}(\boldsymbol{x}(s))) = (K-1)^{-1} \sum_{k=1}^{K-1} V^{\mathrm{T}}(y_k f_k(\boldsymbol{x}(s)))$ and $e_{L^{\mathrm{T}}}(\boldsymbol{f}, \boldsymbol{f}^0) = EL^{\mathrm{T}}(Y_k f_k(\boldsymbol{x}(S))) - EL^{\mathrm{T}}(Y_k f_k^0(\boldsymbol{x}(S)))$.

**Assumption 2.** *There exist constants $\alpha > 0$, $\gamma > 0$, $a_1 > 0$ and $a_2 > 0$ such that for any sufficiently small $\delta > 0$ and $\mathcal{F}_{\mathcal{B},\delta} = \{\boldsymbol{f} \in \mathcal{F}_{\mathcal{B}} : e_{L^{\mathrm{T}}}(\boldsymbol{f}, \boldsymbol{f}^0) \leq \delta\}$,*

$$\sup_{\boldsymbol{f} \in \mathcal{F}_{\mathcal{B},\delta}} e(\boldsymbol{f}, \boldsymbol{f}^0) \leq a_1 \delta^a, \tag{2.15}$$

$$\sup_{\boldsymbol{f} \in \mathcal{F}_{\mathcal{B},\delta}} \text{Var}\left( L^{\mathrm{T}}(Y_k, f_k(\boldsymbol{x}(S))) - L(Y_k, f_k^0(\boldsymbol{x}(S))) \right) \leq a_2 \delta^{\gamma}. \tag{2.16}$$

Assumption 2 describes the local smoothness of $e(\boldsymbol{f}, \boldsymbol{f}^0)$ and $\mathrm{Var}(L^{\mathrm{T}}(Y_k, f_k(\boldsymbol{x}(S))) - L(Y_k, f_k^0(\boldsymbol{x}(S))))$ within a neighborhood of $\boldsymbol{f}^0$. The exponents $\alpha$ and $\gamma$ depend on the joint distribution of $(S, C)$ as well as the loss function $L$. Moreover, inequality (2.16) is implied by the low noise assumption [36].

Let $H_{\mathcal{B}}(\epsilon, \mathcal{F}_{\mathcal{B}})$ be defined as the logarithm of the cardinality of the smallest $\epsilon$-bracketing function set of $\mathcal{F}_{\mathcal{B}}$.

Let $\mathcal{F}_{\mathcal{B}}(t) = \{\boldsymbol{f} \in \mathcal{F}_{\mathcal{B}} : J(\tilde{\boldsymbol{\beta}}) \leq J_{d,K}^* t\}$, $\mathcal{F}_{\mathcal{B}}^L(t) = \{(K-1)^{-1} \sum_{k=1}^{K-1} V^{\mathrm{T}}(y_k f(\boldsymbol{x}(s))) : \boldsymbol{f} \in \mathcal{F}_{\mathcal{B}}(t)\}$, and $\bar{J}_{d,K} = \max(J(\tilde{\boldsymbol{\beta}}^*), 1)$ that may depend on $(d, K)$.

**Assumption 3.** *For some constants $a_i > 0$; $i = 3, \cdots, 5$ and $\epsilon_{n,d,K} > 0$,*

$$\sup_{t \geq 2} \phi(\epsilon_{n,d,K}, t) \leq a_5 n^{1/2}, \tag{2.17}$$

*where $\phi(\epsilon, t) = \int_{a_4 D}^{a_2^{1/2} D^{\boldsymbol{\beta}/2}} H_B^{1/2}(w, \mathcal{F}_{\mathcal{B}}^L(t)) dw / D$, and $D = D(\epsilon, \lambda, t) = \min(\epsilon^2 + \lambda(t/2 - 1)\bar{J}_{d,K}, 1)$.*

**Theorem 1.** *Under Assumptions 1-3, for any large margin sentiment prediction rule $\hat{\phi}$ defined by function (2.8), there exists a constant $a_6 > 0$ such that, for any $\eta \geq 1$,*

$$\mathbb{P}\left(e(\hat{\phi}, \phi^0) \geq a_2 \eta \delta_{n,d,K}^{2\alpha}\right) \leq 3.5 \exp(-a_6 \eta^{2-\min(\boldsymbol{\beta},1)} n (\lambda \bar{J}_{d,K})^{2-\min(\boldsymbol{\beta},1)}), \tag{2.18}$$

*provided that $\lambda^{-1} \geq 2\delta_{n,d,K}^{-2} \bar{J}_{d,K}$, where $\delta_{n,d,K}^2 = \min(\epsilon_{n,d,K}^2 + 2\xi_{n,d,K}, 1)$, and $\alpha, \boldsymbol{\beta}, \epsilon_{n,d,K}$, $\xi_{n,d,K}$ are defined in Assumptions 1-3.*

The proof of Theorem 1 is similar to that in [60] and is given in Appendix A. The upper bound indicates the importance of the imposed relational constraints since they may reduce the size of the candidate function class $\mathcal{F}_{\mathcal{B}}$. Hence, we expect to realize better sentiment prediction accuracy than its alternative like SVM which does not incorporate such a graph among the words.

## 2.4 Simulation Study

Here, we describe our simulation study to compare the six classification systems, LMOC, SVM, RF and their enhanced versions, and report the results.

We consider a dictionary of size $d$, and each simulated review is represented by a $4d$-element vector $x$ consisting of 0's and 1's to represent the absence or presence of each word $w$ in the dictionary as well as its 3 associated bigrams, $neg\_w$, $less\_w$, and $more\_w$. We generate these values from the Bernoulli distribution independently, with probability 0.1 of getting a 1. The resulting matrix $X$ is $n \times 4d$ in size.

Next, we set the coefficients that capture the polarity of each word $w$ as a value in $(-1, 1)$. Without loss of generality, we assume that for the $d$ unigrams we have $\beta_1 \geq ... \geq \beta_d$. Then, for each coefficient $\beta_i$, we generate three more coefficients $\beta_{i1}, \beta_{i2}, \beta_{i3}$ for its corresponding bigrams, whose values would be constrained by the value of $\beta_i$. Specifically, we set

$$\begin{cases} \beta_{i1} = -\beta_i \\ 1 \geq \beta_{i3} \geq \beta_i \geq \beta_{i2} \geq 0 \qquad \text{if } \beta_i \geq 0 \\ -1 \leq \beta_{i3} \leq \beta_i \leq \beta_{i2} \leq 0 \quad \text{otherwise.} \end{cases} \tag{2.19}$$

That is, the polarity of $neg\_w_i$ is opposite sign to $w_i$, indicating that it represents the opposite sentiment. From the second line of function (2.14), if $\beta_i$ is a positive value, implying the unigram $w_i$ is a positive word, then the $more\_w_i$ bigram has a higher polarity $\beta_{i3}$ than $w_i$, while $less\_w_i$ has a lower polarity $\beta_{i2}$ than $w_i$. These values are positive, since both of these latter bigrams still represent postive sentiment. Similarly, from the third line of (2.14), if $\beta_i$ is negative, implying that the unigram $w_i$ is a negative word, then $more\_w_i$ has a lower polarity value than $w_i$, and $less\_w_i$ has a higher polarity value than $w_i$, both less than 0.

We consider both binary ratings as well as ratings with ordinal categories. For the binary case, the rating score $c_i$ is given by $\text{sign}(x_i^T \beta)$. For ordinal categories, we need to generate intercepts to separate the categories. For example, for $K = 5$ categories, we have four intercepts $\beta_{01}, \beta_{02}, \beta_{03}, \beta_{04}$ such that $-2 \leq \beta_{01} ... \leq \beta_{04} \leq 2$.

The rating score $c_i$ is then determined according to the following formulation:

$$c_i = \min\{k : x_i^T \boldsymbol{\beta} \geq -\beta_{0k}\},$$

which corresponds the decision rule function (2.8) for LMOC.

The full data then consists of $(x_i, c_i),\ i = 1, ..., n$, and we use $x_i$ to predict $c_i$. In the simulation study we use $(n, 4d) = (1000, 2000)$, and $(4000, 16000)$ for binary and ordinal sentiment analysis, respectively, so that the number of unigrams, $d$ is 500, and 4000, respectively. We classify the data using the six classification systems and measure their performance using functions (2.12) and (2.13). We use an independent sample of size $n$ for cross-validation with categories equally distributed to obtain the value of the tuning parameter $\lambda$ in function (2.11), minimizing the test error functions (2.12) and (2.13) over a set of values of $\lambda$ given by $\lambda = \{10^{-3+t}\},\ t = 0, ..., 10$. We then use this value of $\lambda$ with another independent sample of size $10^4$. Each test error is averaged over 100 simulation replications. Table 2.5, Figure 2.6, and Figure 2.7 show the test error obtained from our study.

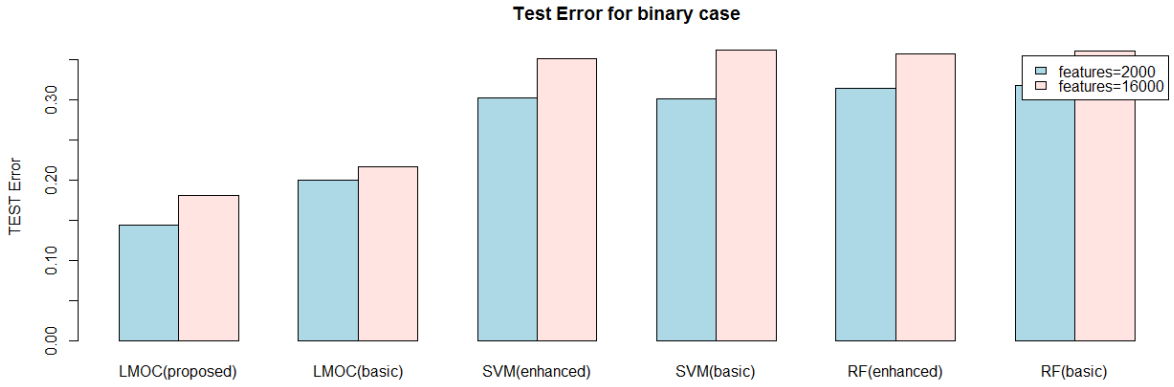



**Figure 2.6:** Test Error for binary case in simulation study.

From Table 2.5, Figure 2.6 and Figure 2.7, comparing basic SVM system with enhanced SVM system for each experiment, and comparing basic RF system with enhanced RF system for each experiment, we find that the enhanced SVM

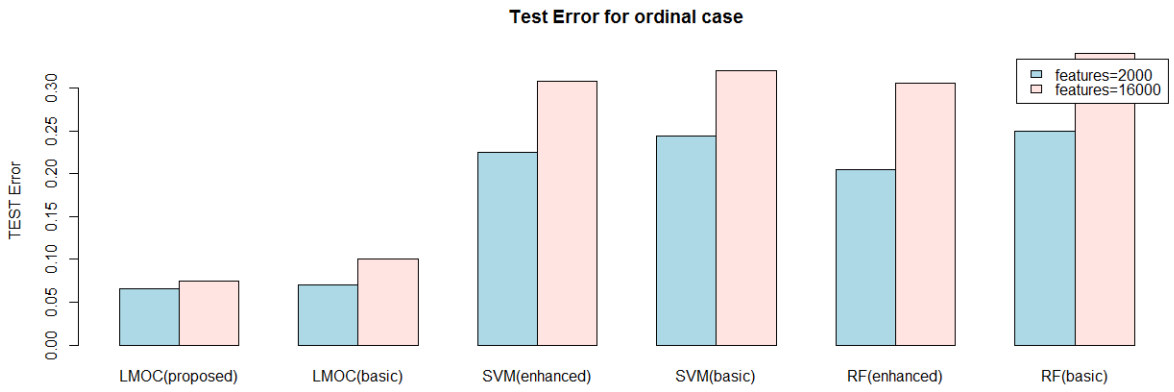



**Figure 2.7:** Test Error for ordinal case in simulation study.

**Table 2.5** The Mean Test Errors as well as their Standard Errors (in parentheses) Over 100 Simulation Replications in Simulated Examples for 6 Classification Systems: Enhanced LMOC (proposed), Basic LMOC, Enhanced SVM, Basic SVM, Enhanced Random Forest, Basic Random Forest.

| | | **Test Error (Simulation)** | | | | | |
|---|---|---|---|---|---|---|---|
| | Dim (4d) | Enhanced LMOC | Basic LMOC | Enhanced SVM | Basic SVM | Enhanced RF | Basic RF |
| Binary | 2000 | 0.144 (0.0050) | 0.200 (0.0070) | 0.303 (0.0022) | 0.310 (0.0010) | 0.314 (0.0034) | 0.318 (0.0030) |
| Binary | 16000 | 0.181 (0.0050) | 0.217 (0.0090) | 0.352 (0.0007) | 0.362 (0.0008) | 0.358 (0.0025) | 0.361 (0.0031) |
| Ordinal | 2000 | 0.066 (0.0004) | 0.070 (0.0007) | 0.225 (0.0007) | 0.244 (0.0006) | 0.205 (0.0011) | 0.206 (0.0009) |
| Ordinal | 16000 | 0.075 (0.0008) | 0.100 (0.0005) | 0.308 (0.0003) | 0.320 (0.0005) | 0.305 (0.0009) | 0.322 (0.0007) |

and enhanced RF systems which use valence shifting bigrams for classification have better performance (lower test error) than the corresponding basic classifier system. Moreover, the improvement is statistically significant at the level $\alpha = 0.05$. Similarly, comparing the basic LMOC with the enhanced LMOC systems, the enhanced LMOC has lower test error, with an improvement of at least 5%, which suggests that using bigrams based on valence shifters can improve classification accuracy. In addition, comparing the LMOC system with the SVM and RF system, we find that the LMOC system, which uses relational information as represented by constraints in the $\boldsymbol{\beta}$'s, performs better than SVM and RF, for both the basic and enhanced versions. This suggests that using relational information can offer a statistically significant improvement in classification accuracy.

## 2.5 Real Data Analysis

### 2.5.1 Data Set

In order to better evaluate the performance of proposed method, we use 3 different data sets of reviews with their corresponding rating system.

**Hotel Reviews** The first data set we use consists of TripAdvisor reviews obtained from the website described in [1], with 15,763 hotel reviews. For each textual review, an integer rating between 1 and 5 represents the degree of customer satisfaction.

**Movie Reviews** The second data set is classified movie reviews prepared by [42]. This data set contains 2000 movie reviews: 1000 positive and 1000 negative. The reviews were originally collected from the Internet Movie Database (IMDb) archive rec.arts.movies.reviews. Their classification as positive or negative is automatically extracted from the ratings, as specified by the original reviewer. They are currently available at http://www.cs.cornel.edu/people/pabo/movie-review-data/.

Only reviews where the author indicated the movie's rating with either stars or some numerical system were included.

**Restaurant Reviews** Another data set is classified restaurant reviews collected from OpenTable.com. This data set contains 21,000 reviews. This website aggregates user opinions on various restaurants including a text-based review and 5-star ratings. The restaurant reviews are provided exclusively by customers who have used the site to make a reservation at a particular restaurant. Since the reviews are voluntary, the comments are more likely to be unbiased.

### 2.5.2 Classification Analysis

We first extracted the valence words and shifters from the above data sets and formed unigrams and bigrams to use as potential features for prediction. We then used the statistic (2.10) to obtain the relative polarities of these features and construct the corresponding directed graph. In total there were 27,160 features when referred to the opinion dictionary. In addition, to consider the impact of dictionary system on the performance of proposed method, we use another dictionary, the General Inquirer for comparison. The General Inquirer contains 1,915 positive words and 2,291 negative words. Hence, in total there were 16,824 features with GI system. The evaluation system setup was the same as that used in the simulation study. The reviews were classified using LMOC, with coefficients constrained according to the directed graph. Finally, to compare the effectiveness of using the directed graph, we also classified the reviews using SVM as well as RF on unigrams alone and with unigrams and bigrams under each dictionary. For classification, reviews were chosen randomly with categories equally distributed in the training data. The same tuning scheme as described Section 2.3 was applied to optimize the performance of each method. The experiment was replicated 100 times. Test error for each classifier

system and each dictionary system were computed as described in Section 2.2.6 and the result are shown in Figure 2.8 - Figure 2.10 and in Table 2.6.

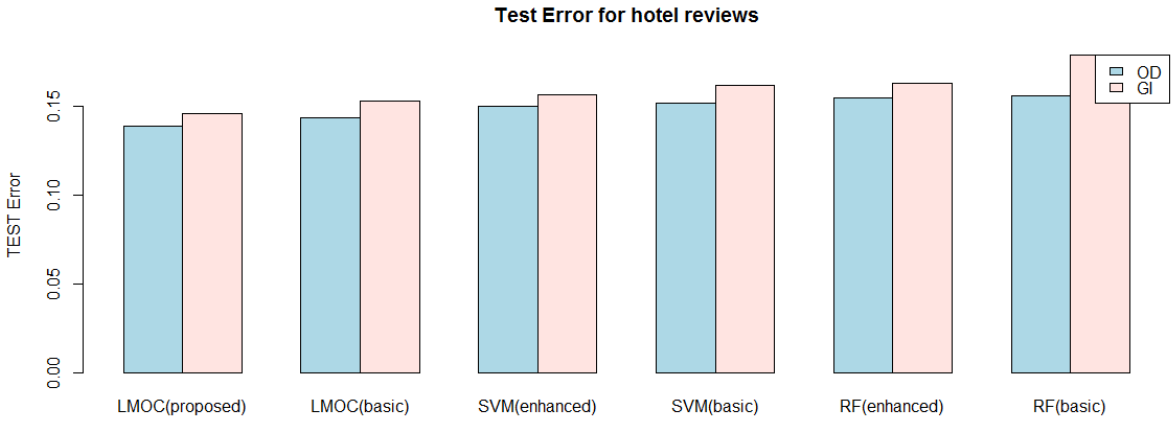



**Figure 2.8:** Test Error for hotel reviews using different dictionary systems. Note: OD is Opinion dictionary. GI is General Inquirer.

**Feature Effectiveness** From the numerical result, the proposed enhanced LMOC method yields about a 4%-32% improvement over standard SVM and RF in terms of test error, with relatively small standard errors both under Opinion Dictionary and General Inquirer. The improvement, however, appears to be less substantial than in the simulated examples, partially due to incomplete dictionary information, ie., words used by reviewers that are not in the dictionary. In addition, as shown in Figures 2.8 -2.10, the Opinion Dictionary with 6.790 opinion words, provided more accurate classification compared with the GI system, which has only 4,206 opinion words.

Although the Opinion dictionary and General Inquirer dictionary contain a large number of common positive words and negative words, they are not complete and may not cover important keywords in different domains. This is especially so with the restaurant reviews - although there were obvious differences between

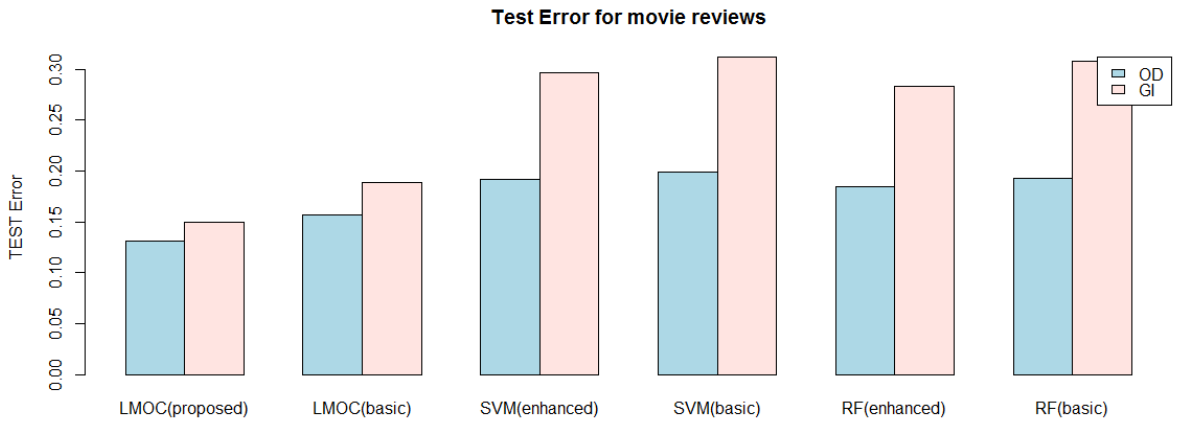



**Figure 2.9:** Test Error for movie reviews using different dictionary systems. Note: OD is Opinion dictionary. GI is General Inquirer.

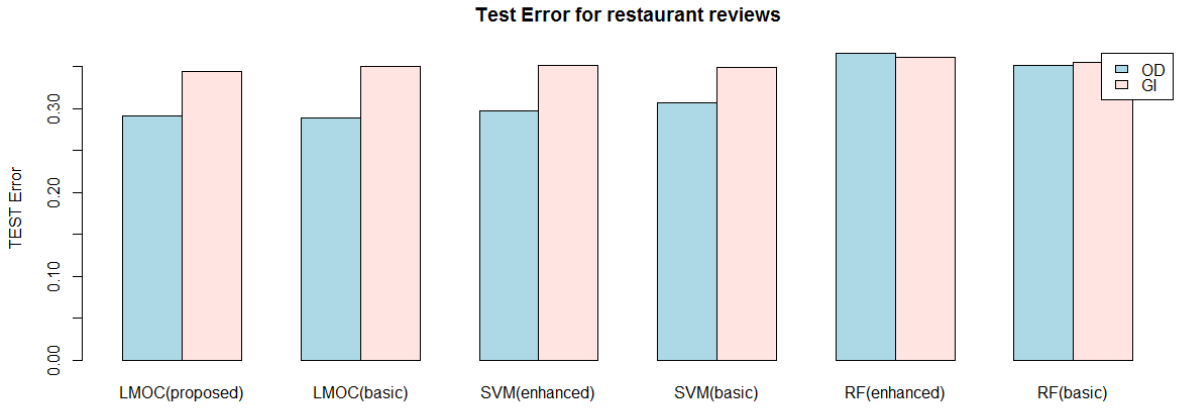



**Figure 2.10:** Test Error for restaurant reviews using different dictionary systems. Note: OD is Opinion dictionary. GI is General Inquirer.

**Table 2.6** The Mean Test Errors as well as their Standard Errors (in parentheses) over 100 Simulation Replications for 6 Classification Systems: Enhanced LMOC (proposed), Basic LMOC, Enhanced SVM, Basic SVM, Enhanced Random Forest, Basic Random Forest Using three Different Real Datasets and Two Different Dictionary Systems (Opinion Dictionary, General Inquirer).

| | | **Test Error (Real Data)** | | | | | |
|---|---|---|---|---|---|---|---|
| | Dictionary System | Enhanced LMOC | Basic LMOC | Enhanced SVM | Basic SVM | Enhanced RF | Basic RF |
| Hotels | OD | 0.139 (0.0003) | 0.144 (0.0004) | 0.150 (0.0008) | 0.152 (0.0007) | 0.155 (0.0006) | 0.156 (0.0008) |
| | GI | 0.146 (0.0008) | 0.153 (0.0006) | 0.157 (0.0009) | 0.162 (0.0009) | 0.163 (0.0004) | 0.179 (0.0015) |
| Movies | OD | 0.131 (0.0005) | 0.157 (0.0009) | 0.192 (0.0007) | 0.199 (0.0008) | 0.185 (0.0005) | 0.193 (0.0003) |
| | GI | 0.150 (0.0004) | 0.189 (0.0008) | 0.297 (0.0007) | 0.312 (0.0006) | 0.283 (0.0004) | 0.308 (0.0004) |
| Restaurant | OD | 0.291 (0.0004) | 0.289 (0.0007) | 0.297 (0.0011) | 0.302 (0.0010) | 0.366 (0.0011) | 0.352 (0.0009) |
| | GI | 0.344 (0.0003) | 0.350 (0.0007) | 0.352 (0.0008) | 0.349 (0.0011) | 0.361 (0.0003) | 0.355 (0.0007) |

methods for the hotel and movie reviews, (Figure 2.8 and Figure 2.9), there was little difference between methods for restaurants. However, when we added 100 high frequency opinion words such as "fresh","tasty","filling" and "spicy" found in the reviews that were not in the Opinion Dictionary or General Inquirer System, there was marked improvement in test error for all methods, especially for our proposed Enhanced LMOC method, where test error fell from 0.305 to 0.210 (Figure 2.11).

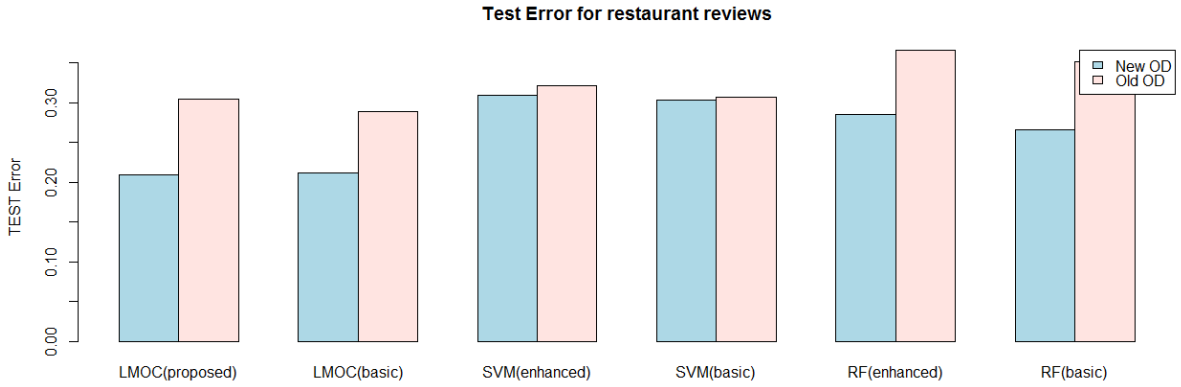



**Figure 2.11:** Test Error for restaurant reviews using different dictionary systems. Note: New OD is Opinion Dictionary after adding top 100 high frequency words. Old OD is original Opinion Dictionary system.

To explore the importance of valence shifting bigrams, we also ran the same system for hotel reviews on a smaller set of 1,000 features pre-selected using random forest. The results are shown in Tables 2.7 and 2.8.

Of the 1000 features selected by Random Forest, almost 400 of them were bigrams. In the SVM system, the pre-selected feature sets including these 400 bigrams provided better performance even than the one with the complete set of unigrams and bigrams. Moreover, in the LMOC system, the pre-selected feature sets with these bigrams achieved the same level of performance as the one with full unigram and bigrams. Hence, we believe that the use of bigrams in the form of valence shifters with

**Table 2.7** Mean Test Errors as well as Their Standard Errors (in parentheses) over 100 Simulation Replications for the TripAdvisor.com Data Example. Results are All Systems with SVM classifier. The Basic System Counts Positive and Negative Terms with Full Feature Size of 6,790 and Reduced Feature Size of 1,000 Selected Using Random Forest. The Enhanced System Adds Contextual Valence Shifters.

| **System** | **Feature size** | **Test Errors** |
|---|---|---|
| Basic: SVM, full unigrams | 6,790 | 0.152(0.0007) |
| Basic: SVM, selected unigrams | 1,000 | 0.152(0.0012) |
| Enhanced: SVM, full unigrams+bigrams | 27,160 | 0.150(0.0008) |
| Enhanced: SVM, selected unigrams+bigrams | 1,000 | 0.149(0.0009) |

**Table 2.8** Mean Test Errors as well as their Standard Errors (in parentheses) Over 100 Simulation Replications for the TripAdvisor.com Data Example. Results for All Systems with Large Margin Ordinal Classifier. The Basic System Counts Positive and Negative Terms with Full Feature Size of 27,160 and Reduced Feature Size of 1,000 Selected Using Random Forest. The Enhanced System Adds Contextual Valence Shifters.

| **System** | **Feature size** | **Test Errors** |
|---|---|---|
| Basic LMOC, full unigrams | 6,790 | 0.144(0.0004) |
| Basic LMOC, selected unigrams | 1,000 | 0.144(0.0009) |
| Enhanced LMOC, full unigrams+bigrams | 27,160 | 0.139(0.0003) |
| Enhanced LMOC, selected unigrams+bigrams | 1,000 | 0.139(0.0004) |

**Table 2.9** Top Words with Large Absolute Coefficients

| | |
|---|---|
| Positive | more awesome, awesome, more comfortable, comfortable, more recommended, more impressed, appreciated, more good, good, nice |
| Negative | ignorant, more pitiful, tainted, more desolate, desolate, more insulting, more appalling, appalling, crafty, false |

opinion words better captures the sentiment of the reviews, improving classification performance.

**Graph Effectiveness** For all three sets of hotel, movie and restaurant reviews, we find that the proposed Enhanced LMOC method performs better than enhanced SVM or enhanced RF, especially after including additional high frequency words into the dictionary system. In Table 2.9 we list the top 20 unigrams and valence shifting bigrams with the largest absolute coefficients in the sentiment function for hotel reviews. These words tend to have strong sentiment polarity, which contributed more to the classification of overall polarity of hotel reviews.

### 2.5.3 Discussion

In this chapter, we propose combining valence shifters and individual opinion words into bigrams to use in an ordinal margin classifier. The classifier is designed to utilize the relational information between features expressed in the form of directed graph. This is achieved by constructing relational constraints from an existing Semantic similarity measure statistic. Our numerical experiment suggests that the proposed method performs well and compares favorably with strong competitors in the literature. An application to hotel reviews, movie data, and restaurant data demonstrate the utility of the proposed method.

# CHAPTER 3

# PROPOSED PENALIZED AND DATA-DRIVEN BASED METHOD FOR THE ESTIMATION OF TWO GROUPS OF INDIVIDUAL NETWORKS

## 3.1 Method

In this study, we consider the problem of estimating two groups of individual networks simultaneously. We assume that the individual networks in the same group tend to have similar structure, while individuals from different groups are allowed to exhibit quite different structures. Such scenarios have been frequently seen in practice. For example, a group of healthy individuals and a group of patients. Our goal is to infer the structure of a graph and obtain an estimate of the partial correlation matrix describing the relationship among random variables within each of samples in healthy group and each of samples in diseased group. We first introduce the basic framework for estimating one group of individual networks, then we extend it to jointly estimate two groups of individual networks.

### 3.1.1 Single Group Multiple Networks Estimation

For one group of $K$ samples, we denote that the $k$-th sample $\boldsymbol{Y}^k = (\boldsymbol{Y}_1^k, \cdots, \boldsymbol{Y}_P^k)$ contains $n_k$ observations, which are independently sampled from a $P$-dimensional multivariate normal distribution with mean $\mu^k$ and covariance $\Sigma^k$. The structure of each distribution can be conveniently represented by an undirected graph $G_k = (\Gamma, E_k)$ with its nodes in $\Gamma = \{1, \cdots, P\}$, and edges in $E_k \subseteq \Gamma \times \Gamma$. We denote the $k$-th sample of the $p$-th variable as $\boldsymbol{Y}_p^k = (y_{p1}^k, \cdots, y_{pn_k}^k)^{\mathrm{T}} \in \mathbb{R}^{n^k \times 1}$; $p = 1, \cdots, P$; $k = 1, \cdots, K$.

To estimate the partial correlation matrix is to explore the conditional independence structure for every pair of variables, given all other remaining variables.

We propose to perform cyclic linear regression, that is, for each node $p$, consider that variable $\boldsymbol{Y}_p^k$ is the response, and $\{\boldsymbol{Y}_q^k; 1 \leq q \leq P, q \neq p\}$ are the predictor variables. Namely, $P$ linear regressions are needed for each matrix. It is related to the neighborhood selection approach proposed by Meinshausen and Bühlmann [41], where LASSO regression for each node is performed separately on the remaining of the variables. Here, we denote the design matrix for the $p$-th regression of the $k$-th matrix to be $\boldsymbol{X}_p^k = (\boldsymbol{Y}_1^k, \cdots, \boldsymbol{Y}_{p-1}^k, \boldsymbol{Y}_{p+1}^k, \cdots, \boldsymbol{Y}_P^k) \in \mathbb{R}^{n^k \times (P-1)}$, excluding the $p$-th column of $\boldsymbol{Y}^k$. We define the model coefficients for the $p$-th linear regression to be $\boldsymbol{\beta}_p^k = \left(\beta_{p1}^k, \cdots, \beta_{p(p-1)}^k, \beta_{p(p+1)}^k, \cdots, \beta_{pP}^k\right)^{\mathrm{T}} \in \mathbb{R}^{(P-1)\times 1}$. Then, the partial correlation matrices are defined by $\boldsymbol{B}^k = (\boldsymbol{\beta}_1^k, \cdots, \boldsymbol{\beta}_P^k)^{\mathrm{T}}$; $k = 1, \cdots, K$. To encourage sparsity and to account for the networks relatedness we introduce the adaptive lasso penalty [69] into the least squares approach. To encourage the commonality of individual networks in the group, we consider element-wise clustering of matrices via $\ell_2$ penalty. When $P$ is large, the number of extracted features for each node through function (3.1) is typically huge. Chances are high that many of generated features may be non-discriminative across all individuals. Therefore, it is better to impose a feature selection indicator to make the learned features discriminative only on some dimensions in the regression for each node. We denote the feature selection indicator, $\boldsymbol{\alpha}_p = (\alpha_{p1}, \cdots, \alpha_{p(p-1)}, \alpha_{p(p+1)}, \cdots, \alpha_{pP})^{\mathrm{T}}$ where $\alpha_{ij} \geq 0$, $i, j = 1, \cdots, P$, $i \neq j$ to be the learned importance score of the remaining variables in the regression for the $p$-th node. This yields the following general penalized loss function

$$\min_{\boldsymbol{B}^1 \cdots \boldsymbol{B}^K} \sum_{k=1}^{K} \sum_{p=1}^{P} (\|\boldsymbol{X}_p^k \boldsymbol{\beta}_p^k - \boldsymbol{Y}_p^k\|_2^2 + \lambda_1 \|w_p^{-\gamma^{\mathrm{T}}} \boldsymbol{\beta}_p^k\|_1) + \lambda_2 \sum_{k,k'=1}^{K} \sum_{p=1}^{P} \|\boldsymbol{\beta}_p^k - \boldsymbol{\beta}_p^{k'}\|_{\boldsymbol{A}_p}^2, \quad (3.1)$$

where $\boldsymbol{A}_p = \mathrm{diag}(\boldsymbol{\alpha}_p)$. We denote $\|\boldsymbol{\beta}\|_M = (\boldsymbol{\beta}^{\mathrm{T}} M \boldsymbol{\beta})^{1/2}$ for a vector $\boldsymbol{\beta} \in \mathbb{R}^d$, and a symmetric $d \times d$ positive definite matrix $M$. There are two tuning parameters, $\lambda_1$ which controls the sparsity across $\boldsymbol{B}^1, \cdots, \boldsymbol{B}^K$, and $\lambda_2$ which encourages $\boldsymbol{B}^1, \cdots, \boldsymbol{B}^K$ to share common structure. The adaptive weights are $w_p \in \mathbb{R}^{(P-1)}$; $p = 1, \cdots, P$. We

use shared adaptive weights $\boldsymbol{W} = (w_1, w_2, \cdots, w_P) \in \mathbb{R}^{(P-1)\times P}$ to regularize over all the K samples to obtain their partial correlation matrix $\boldsymbol{B}^1, \cdots, \boldsymbol{B}^K$, simultaneously.

### 3.1.2 Two Groups Joint Penalized Estimation

Suppose we are given $K_1$ samples in healthy group and $K_2$ samples in diseased group. Within each specific group, we consider element-wise clustering of matrices using $\ell_2$ penalty to encourage the common structure. Between two groups, we restrict the element-wise distance of matrices in a different degree to identity and reconstruct the heterogeneity. The prior knowledge concerning group is specified in an undirected graph $U = (V, \mathcal{E})$, where $V = \{1, \cdots, (K_1 + K_2)\}$ is a set of individual networks from two groups, and $\mathcal{E}$ denote a set of edges that represent the connection between two networks. The corresponding two nodes are connected if two individuals are in the same group. In the model of estimating two groups of individual networks, we consider adding the feature selection indicator as well, which is crucial for differentiating the network pattern of each individual between different groups. To estimate and improve this model, we propose the following joint sparse regression penalized method through incorporating one penalized criterion into (3.1),

$$\begin{aligned} \min_{\boldsymbol{B}} & \sum_{k=1}^{K_1+K_2} \sum_{p=1}^{P} (\|\boldsymbol{X}_p^k \boldsymbol{\beta}_p^k - \boldsymbol{Y}_p^k\|_2^2 + \lambda_1 \|w_p^{-\gamma^{\mathrm{T}}} \boldsymbol{\beta}_p^k\|_1) \\ & + \lambda_2 \sum_{\mu,v=1}^{K_1+K_2} s_{\mu v} \sum_{p=1}^{P} \|\boldsymbol{\beta}_p^{\mu} - \boldsymbol{\beta}_p^{v}\|_{\boldsymbol{A}_p}^2 \\ & \text{s.t. } w_p \geq 0, w_p \cdot 1 = \omega_p; \sum_{p=1}^{P} tr(\boldsymbol{A}_p) = 1, \end{aligned} \tag{3.2}$$

where $\boldsymbol{B} = (\boldsymbol{B}^1 \cdots \boldsymbol{B}^{K_1+K_2})$. The two tuning parameters, $\lambda_1$ controls the sparsity across $\boldsymbol{B}^1, \cdots, \boldsymbol{B}^{K_1+K_2}$, and $\lambda_2$ encourages $\boldsymbol{B}^1, \cdots, \boldsymbol{B}^{K_1+K_2}$ belonging to the same group to share certain characteristics. We set $s_{\mu v} = 1$ if $(\mu, v) \in \mathcal{E}$, otherwise $s_{\mu v} = a$; $0 \leq a \leq 1$. Specifically, as $a = 1$, which indicates the network distance of individuals

between groups is the same with that within group, we believe that the model for single group networks estimation defined in function (3.1) is the special case of our proposed model. The weight parameter $a$ is used to detect the heterogeneity between two groups. We propose two different ways to tune the value of $a$. The first commonly used way is combining $\lambda_2$ as $\ell_2$ penalty. That is, the identification of heterogeneity is learned through the cross-validation, which is the most popular approach for the regularized model to select the value of penalty. The second way is using a data-driven based adaptive method to tune the value of $a$, which is set to be the ratio of expected network distance within each group to that between two groups. We update it by continuously training the data and the obtained updated partial correlation matrices. We expect it to favorably reconstruct the true heterogeneity via learning the ratio based on fixed $\lambda_2$. The detailed description about this adaptive method is presented in Section 3.4.

## 3.2 Optimization Algorithm

We now detail an iterative optimization procedure for the first proposed method which takes $a$ as the part of the $\ell_2$ penalty by interactively performing the feature learning and feature screening by learning the importance of features.

### 3.2.1 Fix $\boldsymbol{A}_p$ and Iteratively Solve for $w_p$ and $\boldsymbol{\beta}_p^k$

Suppose we have an initial $\boldsymbol{A}_p$, which states that some features for the $p$-th regression are more important. Then we can solve $w_p$ and $\boldsymbol{\beta}_p^k$ iteratively, such that features are discriminative on selected dimensions specified by $\boldsymbol{A}_p$.

In this step, the $P$ sub-problems can be separated. Therefore, we can optimize over $p = 1, \cdots, P$, independently:

$$\begin{aligned} &\min_{\boldsymbol{B}} \sum_{k=1}^{K} (\|\boldsymbol{X}_p^k \boldsymbol{\beta}_p^k - \boldsymbol{Y}_p^k\|_2^2 + \lambda_1 \|w_p^{-\gamma^{\mathrm{T}}} \boldsymbol{\beta}_p^k\|_1) + \lambda_2 \sum_{\mu v} s_{\mu v} \|\boldsymbol{\beta}_p^{\mu} - \boldsymbol{\beta}_p^{v}\|_{\boldsymbol{A}_p}^2 \\ &\text{s.t. } w \geq 0, w_p \cdot 1 = \omega_p. \end{aligned} \tag{3.3}$$

For this problem, we need to solve it iteratively among the two sets of variables, $w_p$ and $\boldsymbol{\beta}_p^k$ .

**(a). Fix $w_p$ and optimize $\boldsymbol{\beta}_p^k$** One needs to repeatedly optimize over the two groups of individual network $k$ until convergence, $k = 1, \cdots, K_1 + K_2$. To derive this, suppose we want to update $\boldsymbol{\beta}_p^k$. Then the objective function can be written as

$$\begin{aligned} \min_{\boldsymbol{B}} \|\boldsymbol{X}_p^k \boldsymbol{\beta}_p^k - \boldsymbol{Y}_p^k\|_2^2 + \lambda_1 \|w_p^{-\gamma^{\mathrm{T}}} \boldsymbol{\beta}_p^k\|_1 + 2\lambda_2 \sum_{i \neq k} s_{ki} \|\boldsymbol{\beta}_p^k - \boldsymbol{\beta}_p^i\|_{\boldsymbol{A}_p}^2 \\ + \text{const.} \end{aligned} \tag{3.4}$$

Then this can be turned to

$$\min \ \left( \boldsymbol{\beta}_p^{k^{\mathrm{T}}} \boldsymbol{Q}_p^k \boldsymbol{\beta}_p^k - 2(\boldsymbol{b}_p^k)^{\mathrm{T}} \boldsymbol{\beta}_p^k + \lambda_1 \|w_p^{-\gamma^{\mathrm{T}}} \boldsymbol{\beta}_p^k\|_1 \right) \tag{3.5}$$

where

$$\boldsymbol{Q}_p^k = (\boldsymbol{X}_p^k)^{\mathrm{T}} \boldsymbol{X}_p^k + \boldsymbol{A}_p \cdot 2\lambda_2 \sum_{i \neq k} s_{ki}$$

$$\boldsymbol{b}_p^k = (\boldsymbol{X}_p^k)^{\mathrm{T}} \boldsymbol{y}_p^k + 2\lambda_2 \sum_{i \neq k} s_{ki} \boldsymbol{A}_p \boldsymbol{\beta}_p^k.$$

Solving (3.5) can be converted to a standard adaptive LASSO problem by taking singular value decomposition of the Gram matrix $\boldsymbol{Q}_p^k$. We suppose $\boldsymbol{Q}_p^k = U_p^k \Sigma_p^k U_p^{k^{\mathrm{T}}}$. Let $H_p^k = \Sigma_p^{k^{1/2}} U_p^{k^{\mathrm{T}}}$ and $Z_p^k = (H_p^k H_p^{k^{\mathrm{T}}})^{-1} \cdot H_p^k \cdot \boldsymbol{b}_p^k$, then (3.5) can be written as

$$\|H_p^k \cdot \boldsymbol{\beta}_p^k - Z_p^k\|_2^2 + \lambda_1 \|w_p^{-\gamma^{\mathrm{T}}} \boldsymbol{\beta}_p^k\|_1,$$

which is the adaptive LASSO.

**(b). Fix $\boldsymbol{\beta}_p^k$ and optimize $w_p$** Our optimization problem becomes

$$\begin{aligned} \min \ & \sum_{k=1}^{K_1+K_2} \|w_p^{-\gamma^{\mathrm{T}}} \cdot \boldsymbol{\beta}_p^k\|_1 \\ \text{s.t. } & w_p \geq 0, w_p \cdot 1 = \omega_p, \end{aligned} \tag{3.6}$$

which has a closed form solution,

$$w_p = \left( \frac{\boldsymbol{\theta}_p^{\frac{1}{\gamma+1}}}{\|\boldsymbol{\theta}_p^{\frac{1}{\gamma+1}}\|_1} \right) \omega_p,$$

where $\boldsymbol{\theta}_p = \sum_{k=1}^{K_1+K_2} |\boldsymbol{\beta}_p^k|$.

### 3.2.2 Fix $\boldsymbol{W}$, $\boldsymbol{B}$ and Solve $\boldsymbol{A}$

Let $\boldsymbol{A} = diag(\boldsymbol{A}_1, \cdots, \boldsymbol{A}_P)$. We can fix $\boldsymbol{W}$ and $\boldsymbol{B}$, and then solve for $\boldsymbol{A}$ by solving the objective function

$$\begin{aligned} &\min_A \sum_{\mu,v=1}^{K_1+K_2} s_{\mu v} (\boldsymbol{B}^\mu - \boldsymbol{B}^v)^{\mathrm{T}} \boldsymbol{A} (\boldsymbol{B}^\mu - \boldsymbol{B}^v) \\ &\text{s.t. } tr(\boldsymbol{A}) = 1, \end{aligned} \tag{3.7}$$

which is a simple linear programming problem.

### 3.2.3 Algorithm

Putting them together, we have Algorithm 1.

**Algorithm 1:** Joint Sparse Regression Penalized Method

**Input:** $\boldsymbol{Y}^1, \cdots, \boldsymbol{Y}^{K_1+K_2}, \lambda_1, \lambda_2, a, \omega_1, \cdots, \omega_P$

**Output:** solution $\boldsymbol{B} = (\boldsymbol{B}^1, \cdots, \boldsymbol{B}^{K_1+K_2})$ to (3.2)

1 Initialize $\boldsymbol{A}^{(0)}$ to be identity matrices and $w_p^{(0)}$ with equal entries for $p = 1, \cdots, P$;

2 **for** $i = 1, 2, \cdots$ **do**

3  **for** $p = 1, \cdots, P$ **do**

4  Fix $\boldsymbol{A}^{(i-1)}$ and update $\boldsymbol{\beta}_p^{1(i)}, \cdots, \boldsymbol{\beta}_p^{K_1+K_2(i)}$ and $w_p^{(i)}$ via solving problem (3.3)

5  Fix $\boldsymbol{B}^{(i)}, \boldsymbol{W}^{(i)}$ and update $\boldsymbol{A}^{(i)}$ via solving problem (3.7);

6  **if** *converge* **then**

7  return $\boldsymbol{B} = \boldsymbol{B}^{(i)}, \boldsymbol{A} = \boldsymbol{A}^{(i)}, \boldsymbol{W} = \boldsymbol{W}^{(i)}$;

### 3.2.4 Model Selection

In practice, the choice of the tuning parameter is important for balancing between the goodness-of-fit and complexity of the model and optimizing the predictive power. Commonly used approaches are Bayesian Information Criterion (BIC), Akaike Information Criterion (AIC) and cross validation. Here, we propose using the cross-validation method to select the tuning parameter, which is expected to be more accurate based on the result of Guo et al. [26] who compared the performance for different approaches. We define the predictive criterion as

$$\mathrm{CV}(\boldsymbol{\lambda}) = \sum_{k=1}^{K_1+K_2} \sum_{p=1}^{P} \sum_{i=1}^{n} (\boldsymbol{x}_{pi}^k \boldsymbol{\beta}_p^k(\boldsymbol{\lambda}) - y_{pi}^k)^2, \tag{3.8}$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, a)$ and $\boldsymbol{\beta}_p^k(\boldsymbol{\lambda})$ is the estimated partial correlation for the $p$-th node on the $k$-th network using the fixed tuning parameter $\boldsymbol{\lambda}$. The quantities $y_{pi}^k$ and $\boldsymbol{x}_{pi}^k$ are the corresponding response and predictors for the $i$-th observation. A grid search can be performed to select $\boldsymbol{\lambda}$ over its domain through $\boldsymbol{\lambda}^\star = \mathrm{argmin}_{\boldsymbol{\lambda}} \mathrm{CV}(\boldsymbol{\lambda})$.

## 3.3 Properties of the Proposed Procedure

### 3.3.1 The Grouping Effect

We show in this section that the estimates of (3.2) can lead to desirable grouping effects for individual networks that are in the same group. If we consider the simple case when only two individual networks are in the same group and the rest are in the another group, the following theorem provides an upper bound on the difference of the estimates between these two networks from (3.2).

**Theorem 2.** *Given dataset $\boldsymbol{Y}^k$ for each $k = 1, \cdots, K_1 + K_2$ and three fixed scalars $(\lambda_1, \lambda_2, a)$, the response $\boldsymbol{Y}_p^k$ for each $p$ and $k$ is centered and predictors $\boldsymbol{X}_p^k$ for each $p$ and $k$ are standardized. Let $\hat{\boldsymbol{B}}(\lambda_1, \lambda_2, a)$ be the solution to (3.2). Suppose that for $j \in \{1, \cdots, p-1, p+1, \cdots, P\}$, $\hat{\boldsymbol{\beta}}_{pj}^{\mu}(\lambda_1, \lambda_2, a)\hat{\boldsymbol{\beta}}_{pj}^{v}(\lambda_1, \lambda_2, a) > 0$, and in a group, individual networks $\mu$ and $v$ are linked only to each other. Define*

$$D_{\lambda_1,\lambda_2,a}(\mu, v, p, j) = \frac{1}{\|\boldsymbol{y}\|_2} |\hat{\boldsymbol{\beta}}_{pj}^{\mu}(\lambda_1, \lambda_2, a) - \hat{\boldsymbol{\beta}}_{pj}^{v}(\lambda_1, \lambda_2, a)|,$$

*then*

$$D_{\lambda_1,\lambda_2,a}(\mu,v,p,j) \leq \frac{1}{(2\lambda_2+\lambda_2 aK_2)\alpha_{pj}}\sqrt{2(1-\rho)}, \tag{3.9}$$

*where* $\|\boldsymbol{y}\|_2 = \sqrt{\sum_k \sum_p \sum_{i=1}^n |y_{pi}^k|^2}$ *and* $\rho = \boldsymbol{x}_{pj}^{\mu\,\mathrm{T}} \boldsymbol{x}_{pj}^v$. $\boldsymbol{x}_{pj}^k$ *is the* $j$*-th feature for the* $p$*-th regression on the* $k$*-th individual network.*

The upper bound in (3.9) gives a quantitative description for the grouping effect of method (3.2). For the simple case where $\boldsymbol{x}_{pj}^{\mu}$ and $\boldsymbol{x}_{pj}^{v}$ are highly correlated, i.e., $\rho = 1$, then the difference between the coefficient paths of features j of the $p$-th node for network $\mu$ and $v$ is almost 0. Furthermore, if Theorem 2 holds for any $j \in \{1, \cdots, p-1, p+1, \cdots, P\}$, the difference of estimated matrices between network $\mu$ and network $v$ is approximately 0.

### 3.3.2 Asymptotic Property

In this section, we investigate the theoretical aspects of the ideal version, $\ell_0$-constrained method and its computationally surrogate, our proposed method.

**$\ell_0$-Constrained version** First consider a $\ell_0$-constrained version of (3.2):

$$\begin{aligned}
&\min_{\boldsymbol{\beta}} \boldsymbol{Q}(\boldsymbol{\beta}) = \sum_{k=1}^{K_1+K_2} \sum_{p=1}^{\mathrm{P}} (\|\boldsymbol{X}_p^k \boldsymbol{\beta}_p^k - \boldsymbol{Y}_p^k\|_2^2, \text{ subject to} \\
&\sum_{k=1}^{K_1+K_2} \sum_{p=1}^{P} \mathbb{1}(|\beta_p^k| \neq 0) \leq C_1, \sum_{\{(\mu,v)\in\mathcal{E}\}} \sum_{p=1}^{P} \mathbb{1}(|\beta_p^\mu - \beta_p^v| \neq 0) \leq C_2, \\
&\sum_{\{(\mu,v)\notin\mathcal{E}\}} \sum_{p=1}^{P} \mathbb{1}(|\beta_p^\mu - \beta_p^v| = 0) \leq C_3,
\end{aligned} \tag{3.10}$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ and $C_1, C_2, C_3$ are the three non-negative tuning parameters. Similarly, $C_1$ controls the sparsity across all individual networks in two groups. $C_2$ encourages the clustering across individual networks from the same group. $C_3$ encourages the differentiating among individuals in two groups.

With regards to simultaneous separating and grouping pursuit and feature selection, we will prove that the global minimizer of function (3.10) recovers the "oracle estimator" provided that sparseness, grouping and differentiating structures are known ahead. We adopt the similar setup of [68] for the asymptotic analysis.

**Oracle Estimator and Consistent Graph** To define the oracle estimator, let $\mathcal{G}(\boldsymbol{\beta})$ denote a partition of $\mathcal{I} \equiv \{1, \cdots, d\}$ by the parameter $\boldsymbol{\beta}$ containing the clustering of individual networks in the same group and the differentiating of networks in two groups, that is, $\mathcal{G}(\boldsymbol{\beta}) = \{\mathcal{I}_0(\boldsymbol{\beta}), \cdots, \mathcal{I}_{S(\boldsymbol{\beta})}(\boldsymbol{\beta})\}$, with $\mathcal{I}_0(\boldsymbol{\beta}) = \mathcal{I} \backslash A(\boldsymbol{\beta})$ and $\mathcal{I}_s(\boldsymbol{\beta})$ satisfying $\boldsymbol{\beta}_j = \boldsymbol{\beta}_{j'}$; $j, j' \in \mathcal{I}_s(\boldsymbol{\beta})$; $s = 1, \cdots, S(\boldsymbol{\beta})$, where $S(\boldsymbol{\beta})$ is the number of nonzero groups and $A(\boldsymbol{\beta}) \equiv \{i : \beta_i \neq 0\}$ is the support of $\boldsymbol{\beta}$. Let $(\mu, v) \in \mathcal{E}$ denote that all the entries of the partial correlation matrices for network $\mu$ and $v$ are correspondingly grouped. Let $\mathcal{G}^0 = \mathcal{G}(\boldsymbol{\beta}^0)$ be the true partition induced by $\boldsymbol{\beta}^0$, the true parameter value and $\boldsymbol{\beta}^0 \in \mathbb{R}^d$.

**Definition 3.3.1.** Given $\mathcal{G}^0$, the oracle estimator is defined as: $\hat{\boldsymbol{\beta}}^{ol} = \text{argmin}_{\boldsymbol{\beta}:\mathcal{G}(\boldsymbol{\beta})=\mathcal{G}^0} Q(\boldsymbol{\beta})$, the corresponding ordinary least square estimator.

**Definition 3.3.2.** An undirected graph $\mathcal{U} = (\mathcal{I}, \mathcal{E})$ is consistent with the true group $\mathcal{G}^0 = \{\mathcal{I}_0^0, \cdots, \mathcal{I}_{s_0}^0\}$, if the subgraph restricted on node set $\mathcal{I}_j^0$ is connected; $j = 1, \cdots, S_0$.

Given a graph $\mathcal{U} = (\mathcal{I}, \mathcal{E})$, let $M = \{\boldsymbol{\beta} : C_1(\boldsymbol{\beta}) \leq d_0, C_2(\boldsymbol{\beta}) \leq c_2, C_3(\boldsymbol{\beta}) \leq c_3, \mathcal{G}(\boldsymbol{\beta}) \neq \mathcal{G}(\boldsymbol{\beta}^0)\}$, where $d_0 = |A^0|$ with $A^0 = A(\boldsymbol{\beta}^0)$. Given a partition $\mathcal{G}$, let $M_{\mathcal{G}} = \{\boldsymbol{\beta} \in M : \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}\}$. For an given index set $A \subseteq \mathcal{I}$, let $M_A = \{\boldsymbol{\beta} \in M : A(\boldsymbol{\beta}) = A\}$. Let $M_i = \cup_{A:|A^0 \backslash A|=i} M_A$, $M_i^* = \max_{A:|A^0 \backslash A|=i} |\{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\beta} \in M_A\}|$; $i = 0, \cdots, d_0$, and $M^* = \exp(\max_{0 \leq i \leq d_0} \frac{\log M_i^*}{\max(i,1)})$. $M^*$ quantifies the complexity of the space of candidate partial correlation matrices denoted by the number of nonzero entries. The degree-of-separation condition is stated as follows,

$$\mathrm{C}_{\min}(\boldsymbol{\beta}^0) \geq c_1 \frac{\log d + \log M^*}{n}, \tag{3.11}$$

where $c_1 > 0$, $\mathrm{C}_{\min}(\boldsymbol{\beta}^0) \equiv \inf_{\boldsymbol{\beta} \in M} \frac{e(\boldsymbol{\beta}, \boldsymbol{\beta}^0)}{n \cdot \max(|A^0 \setminus A|, 1)}$ and $e(\boldsymbol{\beta}, \boldsymbol{\beta}^0) = EQ(\boldsymbol{\beta}) - EQ(\boldsymbol{\beta}^0)$ is the excess risk. The measure $\mathrm{C}_{\min}$ denotes the degree of separation between $A^0$ and a least favorable candidate model for feature selection, clustering within the same group and differentiating among two groups pursuit. We now define a complexity measure for the size of a space $\mathcal{F}$. The bracketing metric entropy of $\mathcal{F}$, $H(\cdot, \mathcal{F})$, is defined by logarithm of the cardinality of the $\epsilon$-bracketing (of $\mathcal{F}$) of the smallest size. That is, for a bracket covering $S(\epsilon, m) = \{f_1^l, f_1^u, \cdots, f_m^l, f_m^u\} \subset \mathcal{L}_2$, which satisfies $\max_{1 \le j \le m} \|f_j^l - f_j^u\|_2 \le \epsilon$ and for $f \in \mathcal{F}$, there exists a $j$ such that $f_j^l \le f \le f_j^u$, $a.e.$P, i.e., $H(\epsilon, \mathcal{F})$ is denoted by $\log(\min\{m : S(\epsilon, m)\})$, where $\|f\|_2 = (\int f^2 d_\mu)^{1/2}$.

Denote $\mathcal{F}_{\mathcal{G}} = \{\boldsymbol{f} = (f_p^k)_{1 \le k \le K, 1 \le p \le P} : f_p^k = \boldsymbol{X}_p^k \boldsymbol{\beta}_p^k, \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}\}$ by any subset $\mathcal{G} \in \{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\beta} \in M\}$.

**Assumption 4.** *(uniformly subGaussian) For each each $k = 1, \cdots K$, $p = 1 \cdots P$, we assume that $\varepsilon_1, \cdots, \varepsilon_n$ are uniformly subGaussian: For some $\alpha > 0$, $\Lambda > 0$,*

$$\sup_n \max_{1 \le i \le n} E(\exp |\alpha \varepsilon_i|^2) \le \Lambda < \infty. \tag{3.12}$$

**Assumption 5.** *(size of parameter space) For any $0 < t < \epsilon \le 1$, $H(t, \mathcal{B}_{\mathcal{G}}(\epsilon)) \le \boldsymbol{O}(|A| \log(c'\epsilon/t))$ for some constant $c'$, where $\mathcal{B}_{\mathcal{G}}(\epsilon) = \mathcal{F}_{\mathcal{G}} \cap \{e(\boldsymbol{\beta}, \boldsymbol{\beta}^0) \le \epsilon^2\}$ is a local parameter space.*

The next theorem describes that a global minimizer of (3.10) $\hat{\boldsymbol{\beta}}^{L_0}$ can consistently recover the oracle estimator at a degree of separation level that is slightly higher than the lower bound in (3.11). Without loss of generality, we assume that the global minimizer of function (3.10) exists.

**Theorem 3** (Global minimizer of (3.10))**.** *Under Assumption 4 and 5, if $\mathcal{E}$ is consistent with respect to $\mathcal{G}^0$, then for a global minimizer of (3.10) $\hat{\boldsymbol{\beta}}^{L_0}$ with estimated clustering within the same group and differentiating among the two groups $\hat{\mathcal{G}}^{L_0} = \mathcal{G}(\hat{\boldsymbol{\beta}}^{L_0})$ at $(C_1, C_2, C_3) = (d_0, c_2, c_3)$, with $d_0 = C_1(\boldsymbol{\beta}), c_2 = C_2(\boldsymbol{\beta}, \mathcal{E}), c_3 = C_3(\boldsymbol{\beta}, \mathcal{E})$.*

$$\mathbb{P}(\hat{\boldsymbol{\beta}}^{L_0} \ne \hat{\boldsymbol{\beta}}^o) \le 2 \exp\left(-C_0 \mathrm{C}_{\min}(\boldsymbol{\beta}^0) + 2 \log(\frac{d+1}{2}) + \log M^*\right) \tag{3.13}$$

*Under (3.11),* $\mathbb{P}(\hat{\mathcal{G}}^{L_0}) \neq \mathcal{G}^0) = \mathbb{P}(\hat{\boldsymbol{\beta}}^{L_0} \neq \boldsymbol{\beta}^o) \to 0$ *as* $n, d \to \infty$.

**Constrained penalization** In a high-dimensional situation, it is computationally infeasible to minimize a discontinuous cost fuction involving the $\ell_0$-function in (3.10). As a surrogate, we investigate the asymptotic property of our proposed method.

Under the assumption that $p$ is fixed and the sample size $n \to \infty$, we derive results of asymptotic property for the estimates of our proposed method defined in (3.2), which is computationally efficient surrogate of $\ell_0$-constrained version. We adopt the setup of [35] for the asymptotic analysis. In the two groups of individual networks, for each network $k$ and each node $p$, we assume two conditions:

**(a)** $\boldsymbol{y}_{pi}^k = \boldsymbol{X}_{pi}^k {\boldsymbol{\beta}_{pi}^k}^0 + \varepsilon_{pi}^k$, where $\varepsilon_{p1}^k, \cdots, \varepsilon_{pn}^k$ are independent identically distributed (iid) random variables with mean 0 and variance ${\sigma_p^k}^2$;

**(b)** $\frac{1}{n} {\boldsymbol{X}_p^k}^{\mathrm{T}} \boldsymbol{X}_p^k \to \boldsymbol{C}_p^k$, where $\boldsymbol{C}_p^k$ is a positive definite matrix.

Let oracle $M^0 = \{j : \boldsymbol{\beta}_j^0 \neq 0\}$, and without loss of generality, assume that $M^0 = \{1, \cdots, p_0\}$ and $M_p^{k\,0} = \{1, \cdots, n_p^k\}, \sum_k \sum_p n_p^k = p_0$. For each $p$ and $k$, let

$$\boldsymbol{C}_p^k = \begin{bmatrix} {\boldsymbol{C}_{11}}_p^k & {\boldsymbol{C}_{12}}_p^k \\ {\boldsymbol{C}_{21}}_p^k & {\boldsymbol{C}_{22}}_p^k \end{bmatrix}$$

where ${\boldsymbol{C}_{11}}_p^k$ is $n_p^k \times n_p^k$ matrix. Recall that the penalized least squares criterion for two groups of individual networks is

$$\begin{aligned} &\sum_{k=1}^{K_1+K_2} \sum_{p=1}^{P} (\|\boldsymbol{X}_p^k \beta_p^k - \boldsymbol{Y}_p^k\|_2^2 + \lambda_n^{(1)} \|w_p^{-\gamma \mathrm{T}} \beta_p^k\|) \\ &+ \lambda_n^{(2)} \sum_{(\mu,v)\in\mathcal{E}} \sum_{p=1}^{P} \|\beta_p^\mu - \beta_p^v\|_{\boldsymbol{A}_p}^2 + \lambda_n^{(3)} \sum_{(\mu,v)\notin\mathcal{E}} \sum_{p=1}^{P} \|\beta_p^\mu - \beta_p^v\|_{\boldsymbol{A}_p}^2 \end{aligned} \tag{3.14}$$

where the Lagrange multipliers $\lambda_n^{(1)}$, $\lambda_n^{(2)}$, and $\lambda_n^{(3)}$ are functions of the sample size $n$. We have the following asymptotic theorem for the estimates:

**Theorem 4.** *Suppose that* $\lambda_n^{(l)}/\sqrt{n} \to 0$ *for* $l = 1, \cdots, 3$ *and* $\lambda_n^{(1)} n^{(\gamma-1)/2} \to \infty$*, Then*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}^n - \boldsymbol{\beta}^0) \xrightarrow{d} \text{argmin}(V),$$

*where*

$$V(\boldsymbol{u}) = \begin{cases} \sum_k \sum_p \boldsymbol{u}^k_{p M_p^k} C_{11_p}{}^k \boldsymbol{u}^k_{p M_p^k} - 2\boldsymbol{u}^k_{p M_p^k}{}^{\mathrm{T}} \boldsymbol{W}_p^k & \text{if } \boldsymbol{u}^k_{pj} = 0 \ \forall j \notin M_p^k \\ \infty & \text{otherwise,} \end{cases}$$

*and* $\boldsymbol{W}_p^k \sim N(0, \sigma_p^{k\,2} \boldsymbol{C}_p^k)$.

### 3.4 Adaptive Method

In the sections above, we mainly investigate using the commonly regularized approach to tune the value of $a$. In this section, we propose a data-driven based adaptive method to tune the value of $a$. In order to detect the true heterogeneity between two groups, we need to learn the value of $a$ from the $K_1 + K_2$ samples.

**Definition 3.4.1.** the oracle $a$ is defined as:

$$a^0 = h(\boldsymbol{B}^0), \tag{3.15}$$

where $h(\boldsymbol{B}^0) = \frac{E\{d(\boldsymbol{B}^{\mu 0}, \boldsymbol{B}^{v 0}) \mathbb{1}((\mu,v) \in \mathcal{E})\}}{E\{d(\boldsymbol{B}^{\mu 0}, \boldsymbol{B}^{v 0}) \mathbb{1}((\mu,v) \notin \mathcal{E})\}}$ the proportion of the expected networks distance within the same group to that between two groups and $\boldsymbol{B}^{k0}$ is the true partial correlation matrix for network $k$. Here $d(\cdot)$ is the Euclidean distance.

In the optimization, we learn the value of $\hat{a}$ in terms of estimated $\hat{\boldsymbol{\beta}}^n$ obtained from each last iteration. Then we have $\hat{a}^n = h(\hat{\boldsymbol{\beta}}^n)$. By the condition of Theorem 4, we have the following proposition on consistency of $\hat{\boldsymbol{\beta}}$.

**Proposition 3.4.1.** *Suppose that* $\lambda_n^{(l)}/\sqrt{n} \to 0$ *for* $l = 1, 2$ *and* $\lambda_n^{(1)} n^{(\gamma-1)/2} \to \infty$*, Then*

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}}^n(\lambda_n^{(1)}, \lambda_n^{(2)}, \lambda_n^{(2)}\hat{a}^n) - \boldsymbol{\beta}^0\right) \xrightarrow{d} \text{argmin}(V)$$

*where*

$$V(\boldsymbol{u}) = \begin{cases} \sum_k \sum_p \boldsymbol{u}^k_{p M^k_p} C_{11_p}{}^k \boldsymbol{u}^k_{p M^k_p} - 2\boldsymbol{u}^k_{p M^k_p}{}^{\mathrm{T}} \boldsymbol{W}^k_p & \text{if } \boldsymbol{u}^k_{pj} = 0 \ \forall j \notin M^k_p \\ \infty & \text{otherwise,} \end{cases}$$

*and* $\boldsymbol{W}^k_p \sim N(0, \sigma^{k\,2}_p \boldsymbol{C}^k_p)$.

### 3.4.1 Algorithm

Based on the second definition of $a$, following Algorithm 1, we directly derive Algorithm 2 via adding the learning procedure for $a$ after each iteration of estimation of $\hat{\boldsymbol{\beta}}$. The choice of tuning parameters in this adaptive method is the same as (3.8) except that the value of a is learned systematically from the estimated network matrices $\hat{\boldsymbol{B}}$ at each iteration.

**Algorithm 2:** Adaptive Method

**Input:** $\boldsymbol{Y}^1, \cdots, \boldsymbol{Y}^K, \lambda_1, \lambda_2, \omega_1, \cdots, \omega_P$

**Output:** solution $\boldsymbol{B} = (\boldsymbol{B}^1, \cdots, \boldsymbol{B}^K)$ to (3.2)

Initialize $\boldsymbol{B}^{(0)}$ with $\boldsymbol{B}^{k\,(0)} = (\boldsymbol{\beta}^k_1(ols), \cdots, \boldsymbol{\beta}^k_p(ols))^{\mathrm{T}}$ for $k = 1, \cdots, K$; $\boldsymbol{A}^{(0)}$ to be identity matrices; $w^{(0)}_p$ with equal entries for $p = 1, \cdots, P$; $a^{(0)} = h(\boldsymbol{B}^{(0)})$;

**for** $i = 1, 2, \cdots$ **do**

  **for** $p = 1, \cdots, P$ **do**

    Fix $\boldsymbol{A}^{(i-1)}$, $a^{(i-1)}$ and update $\boldsymbol{\beta}^{1\,(i)}_p, \cdots, \boldsymbol{\beta}^{K\,(i)}_p, w^{(i)}_p$ via solving problem (3.3)

  Update $a^{(i)} = h(\boldsymbol{B}^{(i)})$ ;

  Fix $\boldsymbol{B}^{(i)}, \boldsymbol{W}^{(i)}$ and update $\boldsymbol{A}^{(i)}$ via solving problem (3.7);

  **if** *converge* **then**

    return $\boldsymbol{B} = \boldsymbol{B}^{(i)}, \boldsymbol{A} = \boldsymbol{A}^{(i)}, \boldsymbol{W} = \boldsymbol{W}^{(i)}$;

### 3.5 Numerical Evaluation

We investigate the numerical performance of our proposed framework on two types of simulated networks: a chain network and a nearest-neighbor network. In each example, we compare our two proposed methods against three different methods.

**M1:** classical graphical LASSO with $\ell_1$ penalty that estimates single network (Lasso);

**M2:** inverse of the sample covariance matrix (Sample);

**M3:** regularized MLE with noncovex penalty for pursuit of sparseness and clustering by [68] (SC);

**M4:** our proposed joint sparse regression model in (3.2) with $\ell_1$ penalty for sparsity and two $\ell_2$ penalties for clustering and separating through cross-validation (Joint);

**M5:** our proposed adaptive model with $\ell_1$ penalty for sparsity, $\ell_2$ penalty for clustering and $a$, the learned metric for detecting the heterogeneity as described in Section 5 (Adaptive).

The Lasso method is the popular approach in estimating individual network, which pursues the sparsity of the network structure. The Sample method directly uses the inverse of the sample covariance matrix to estimate the network structure of the sample. The SC method is an approach to estimate a single group of individual networks. This method defines the nonconvex penalty in the form

$$\boldsymbol{J}_{ij}(\omega_{ij}^1, \cdots, \omega_{ij}^K) = \lambda_1 \sum_{k=1}^{K} \boldsymbol{J}_\tau(|\omega_{ij}^k|) + \lambda_2 \sum_{\{(\mu,v)\in\mathcal{E}\}} \boldsymbol{J}_\tau(|\omega_{ij}^\mu - \omega_{ij}^v|),$$

where $\boldsymbol{J}(z) = \min(|z|, \tau)$ is the truncated $\ell_1$ penalty of [52]. The precision matrix is estimated by solving

$$\begin{aligned} \max_{\boldsymbol{\Omega}>0} S(\Omega) = & \sum_{k=1}^{K} n_k \left(\log(\det(\boldsymbol{\Omega}^k)) - tr(S^k\boldsymbol{\Omega}^k)\right) \\ & - \sum_{i\neq j} \boldsymbol{J}_{ij}(\omega_{ij}^1, \cdots, \omega_{ij}^K) \end{aligned} \tag{3.16}$$

using difference convex programming with block-wise coordinate descent method. The optimal estimate is obtained through a grid search over the domain of the tuning parameters.

In the simulation study, each method described above is used in different ways to estimate two groups of individual networks. We apply the graphical Lasso method separately to fit each individual network of two groups as well as the procedure of Sample method. We use SC method to estimate each single group of individual networks separately. Our two proposed methods are performed to jointly estimate multiple graphical models corresponding to two groups of individual networks simultaneously.

In our experiment, we apply two metrics to evaluate the performance of competing methods.

**Metric 1:** area under the ROC curve (AUC) for the specific number of edges;

**Metric 2:** sensitivity with the false positive rate controlled at 5%.

The AUC area measures discrimination, that is, the ability of the test to correctly classify those with and without an edge. Metric 2 assesses the variability of the competing methods.

### 3.5.1 Simulation Settings

**Example 1: Chain networks** In this example, we generate tridiagonal precision matrices for estimation, which follows the simulation in Fan [17]. The covariance matrices $\Sigma^k$ is AR(1)-structured with $ij$-element $\sigma_{ij}^k = \exp(-|s_i^k - s_j^k|/2)$ and $s_1^k < s_2^k < \cdots < s_p^k$. Here, $s_i^k - s_{i-1}^k \sim \text{Unif}(0.5, 1)$; $i = 2, \cdots, p$, $k = 1, 2, \cdots, K$. Further, let the partial correlation matrix $\Omega^{-1^k} = \Sigma^k$. Initially, we generate two partial correlation matrices $\Omega'$ and $\Omega''$ by this procedure, so that they share a common structure (pattern of zeros) as shown in Figure 3.1, but with possibly different off-diagonal non-zero elements. There are twelve situations to be considered: $n_k = 60$,

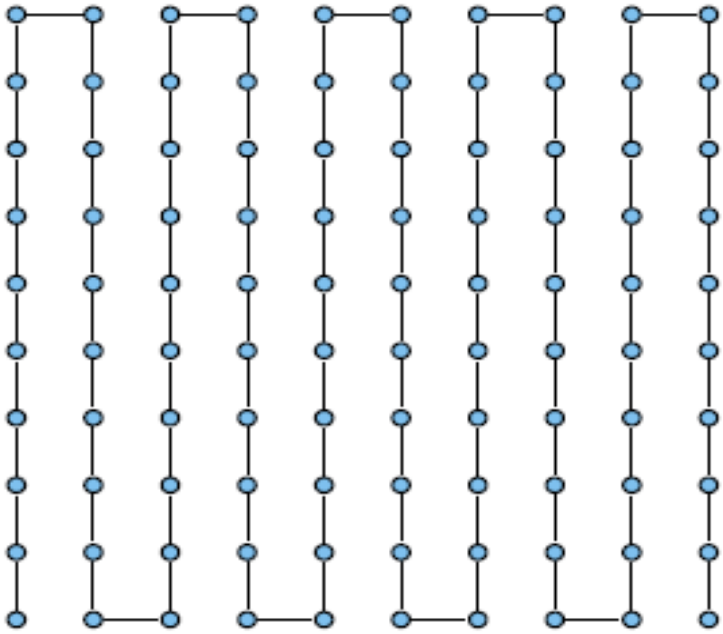


**Figure 3.1:** Chain Network by Guo et al. [26]

100, 300; $P = 50, 100$; $K = 50, 100$ initially with $\Omega^1 = \cdots = \Omega^{K/2} = \Omega'$ and $\Omega^{K/2+1} = \cdots = \Omega^K = \Omega''$. That is, we generated two groups of equal number of samples. Within each group, the samples are initially generated independently and identically from a multivariate normal distribution N(0,$(\Omega)^{-1}$) with the same generated partial correlation matrix. To study the performance of competing methods as the heterogeneity between the two groups varies, we gradually create additional individual links in the common structure. For $\Omega'$ and $\Omega''$, we generate values uniformly from [-1,-0.5]$\cup$[0.5,1] to replace the same number of randomly selected off-diagonal symmetric zero elements. Moreover, to fit the problem that the individual networks within each group are slightly different as well, we use the same method to randomly replace 1%-2% selected off-diagonal symmetric zero elements of each partial correlation matrix. Through this procedure, the true values of $a$, representing the ratio of expected network distance within each group to that between two groups as defined in function (3.15), can be set to vary from 1 down to 0. In the simulations,

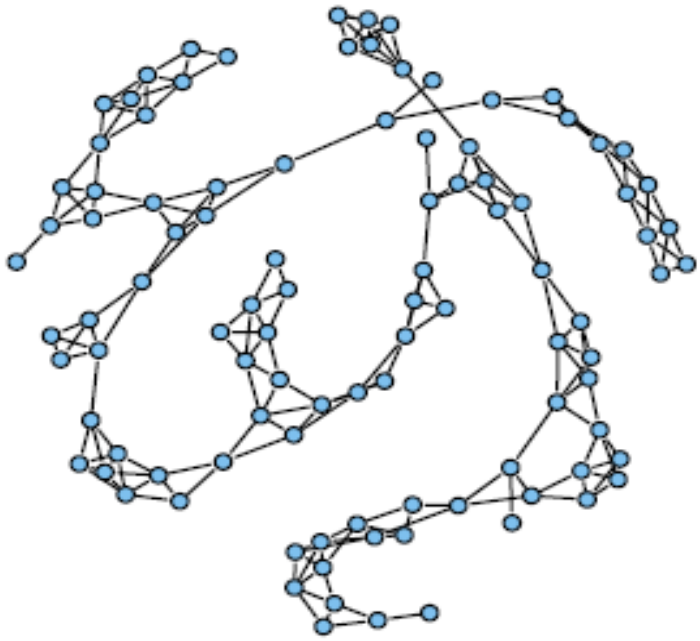


**Figure 3.2:** Nearest-neighbor Network by Guo et al. [26]

we consider the performance of the five different methods with respect to $a$ over 100 repetitions at the aforementioned twelve situations.

**Example 2: Nearest neighbor networks** In this example, we consider using the data generating scheme described in Li and Gui [38], which is a general sparse partial correlation matrix. In particular, we generate $P$ points randomly on a unit square, and compute all $P(P-1)/2$ pairwise Euclidean distances. For each point, we find the $m$ nearest neighbors based on the distances and link the corresponding points to construct the nearest neighbor network. The integer value of $m$ controls the degree of sparsity. In our study, we choose $m$=5. For each"edge" in the network, the corresponding off-diagonal element in the partial correlation matrix is generated uniformly over [-1,-0.5]∪[0.5,1]. The $i$-th diagonal value is defined as a multiple of the sum of the absolute values of the off-diagonal entries in the $i$-th row. Here the multiple chosen is 2, to ensure that the partial correlation matrix is positive definite. Finally, each row of the matrix is divided by the corresponding diagonal element so that the

value of the diagonal entries of the final partial correlation matrix is 1. The final common structure of the nearest-neighbor network is shown in Figure 3.2. Similarly to our procedure for adding heterogeneity mechanism to generate two different groups of individual networks in example 1, we add some individual links to the common structure for $\Omega'$ and $\Omega''$. Then we study the performance of the five different methods with respect to $a$ over 100 replications under twelve different settings of $n_k$, $P$ and $K$.

### 3.5.2 Simulation Results

In our experiment, the obtained estimated value of $a$ in the proposed adaptive model is much more close to the oracle value of $a$ than the obtained value via cross validation method. This domination also can be seen from the Figures 3.3 and 3.4, which show the estimated area under ROC curves (AUC) averaged over 100 repetitions for the two simulated examples. The AUC for a method shows its performance over all choices of the tuning parameter. The model selection procedure is better the closer its AUC value approaches 1. The results suggest that our two proposed methods, which both simultaneously estimate two groups of individual networks, especially the adaptive method does better than the other competing methods in making the trade off between the false positive rate and the true positive rate across all cases considered, especially when the value of $a$ is low. As $a$ becomes lower, the heterogeneity between two groups becomes larger. The SC method, which separately performing the networks estimation for each single group does not consider the differences between two groups only in its pursuit of clustering within the same group, performs increasingly worse than our proposed methods as $a$ gets smaller. Meanwhile, separated LASSO method becomes gradually closer to SC, but still performs poorly. That is just because separate LASSO method both ignores the common information shared within each group and the detection of difference between two groups. As $a$

increases, the overall structure of the networks across two groups become more and more similar, and we find that the SC method and our proposed methods perform more similarly, with the SC method close to ours at $a$=1 in most cases. On the other hand, the separate LASSO continues to do worse. This is expected, because the Joint, Adaptive and SC methods can take advantage of the greater overlap in the structures. Over all of the simulations, we find that the Sample method performs worse than any of the other four methods. We believe that is due to its lack of shrinkage toward the partial correlation matrix leading to a much more complicated network than the regularized ones.

In the twelve situation settings, we can examine how AUC changes with respect to $a$, $n$, $P$ and $L$, respectively. With the other quantities fixed, the methods perform better with larger $a$, $n$, and $L$, and with smaller $P$. Therefore, we further explore their stability under the settings where the performance is worst and best, i.e., $n = 60$, $P = 100$, $L = 50$ and $n = 300$, $P = 50$, $L = 200$. Figures 3.5 and 3.6 show the variability of sensitivity when the false positive rate is fixed at 5% under these two settings based on 100 simulations in each of the two examples. The results suggest our methods are more stable and significantly different than others.

## 3.6 Application

In the real data analysis, we apply the proposed adaptive method to the polychromatic flow cytometry data. Polychromatic flow cytometry allows the simultaneous measurement of many different proteins within thousands of individual cells. Sachs and coworkers [47] originally presented this experiment to infer a signaling network by quantitatively measuring protein expression levels in Figure 3.7. In this study, the amounts of eleven well-studied proteins were simultaneously measured from single cells after imposing a series of perturbations (stimulatory cues or inhibition) on the network. There were nine different stimuli to target specific proteins in the selected

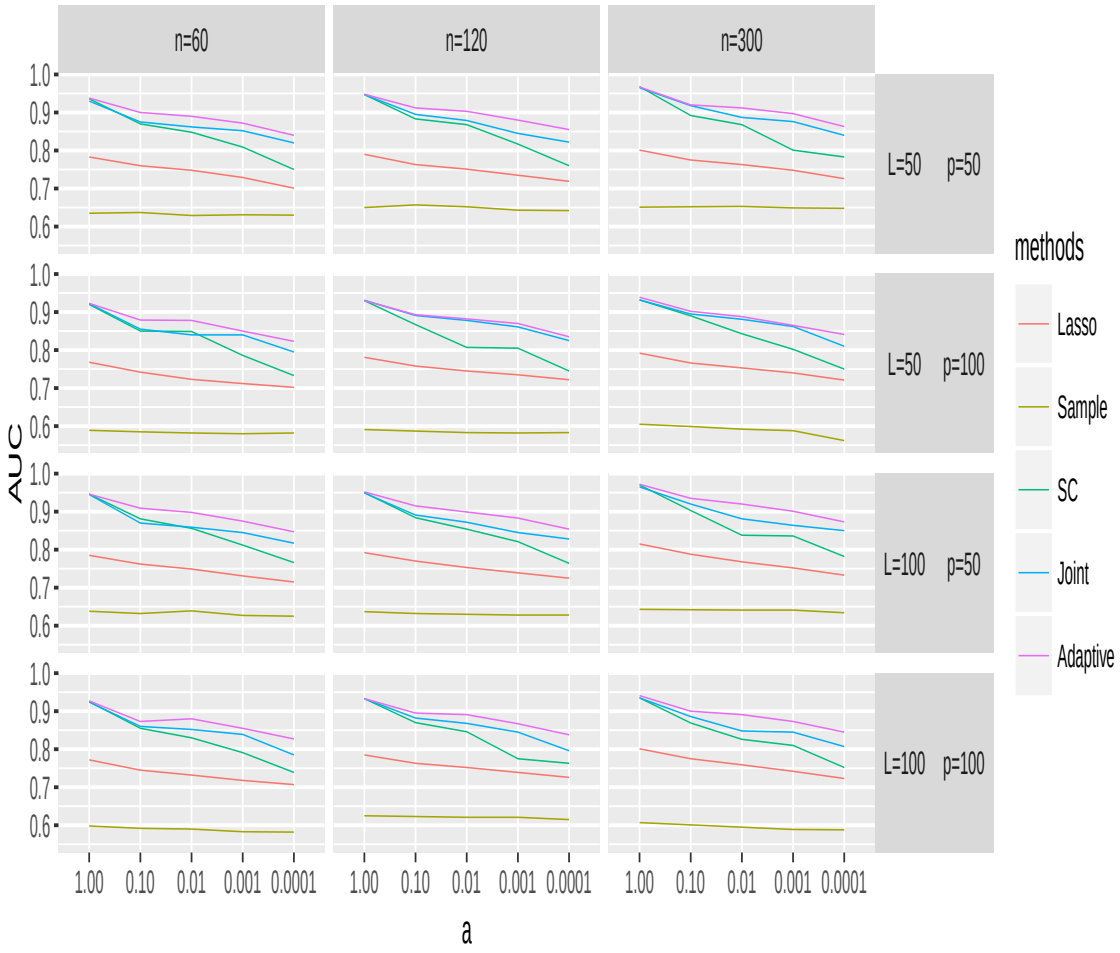



**Figure 3.3:** Average AUC based on 100 simulations for estimating networks in example 1.

pathway, the effects of which are summarized in Table 3.1. If a protein is stimulated or inhibited, then the downstream proteins of the corresponding pathway including this protein would grow or drop, but there would be no effect on the uncorrelated proteins. Sachs [47] employed nine experimental conditions to collect data from 7466 cells for model inference and also used another five experimental conditions to collect data from 4206 cells. Each of these two sets of conditions both consist of one stimulus or a combination of two or three different stimulus as described in Table 3.1. Based on these data, the structure of inferred networks are expected to be similar in some edges due to common stimuli, with some variation due to differences between conditions. To explore the performance of our method in detecting the heterogeneity between

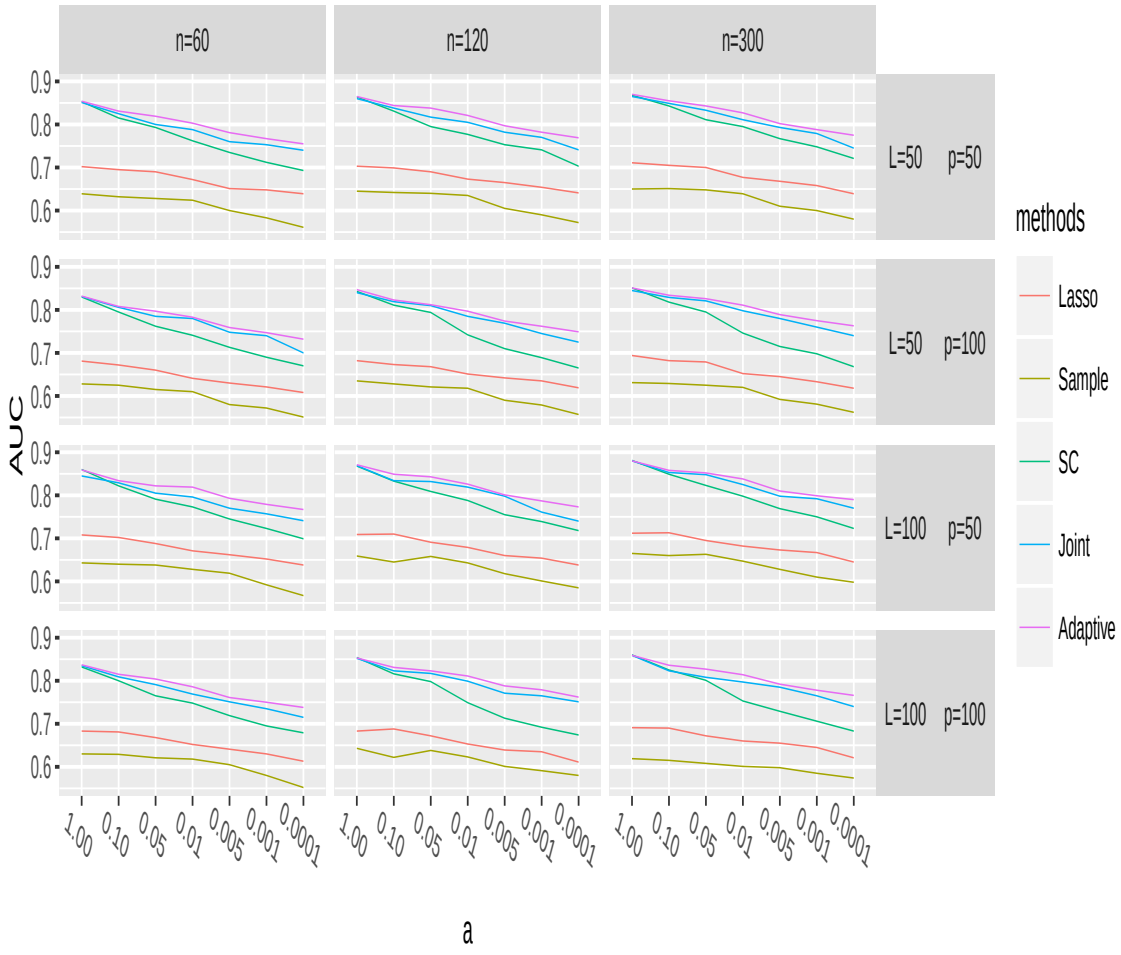



**Figure 3.4:** Average AUC based on 100 simulations for estimating networks in example 2.

the networks under different sets of conditions, we consider the two networks that belong in two groups. Moreover, within each group we specifically add extra sample of data sets for the networks inference. The extra sample in the first group is collected under eight conditions selected from the aforementioned nine conditions and the extra sample in the second group is collected under four conditions selected from the above five conditions. Hence, under those four sets of conditions, we finally obtain two groups of data sets on quantities amounts for 11 proteins, where the first group contains two samples with sample size $n_1 = 7466$ and $n_2 = 6667$, the second group contains two samples with sample size $n_3 = 4206$ and $n_4 = 3338$.

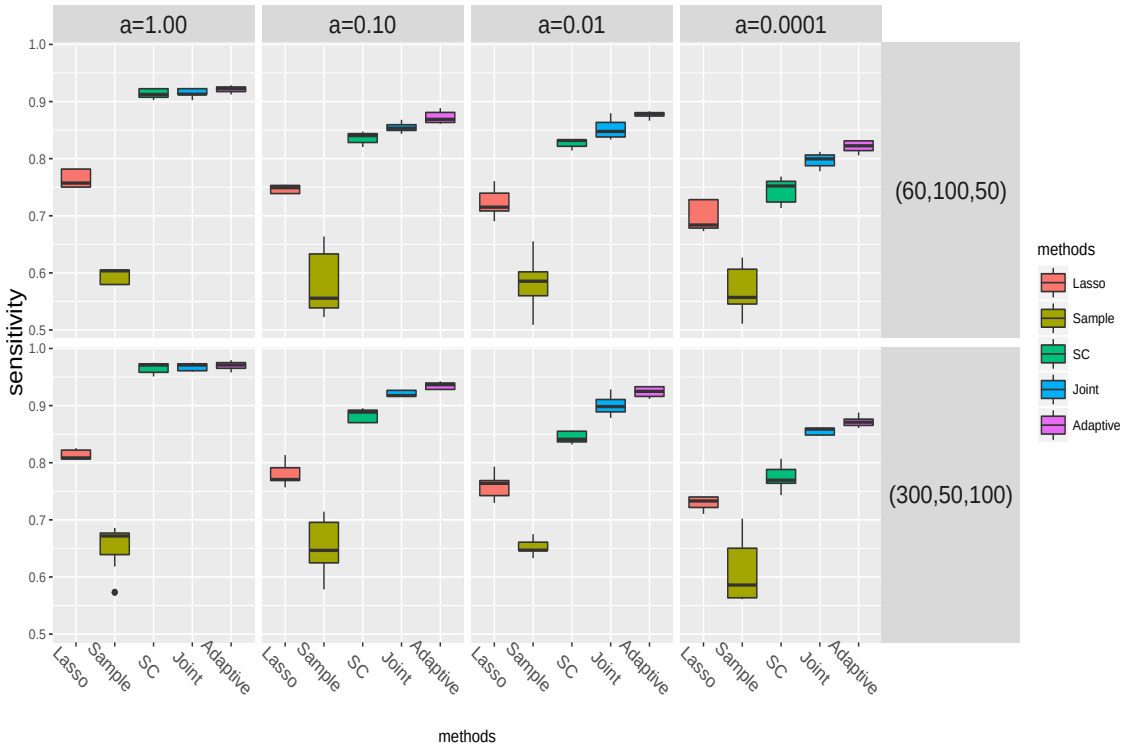



**Figure 3.5:** Boxplot of sensitivity under the false positive rate controlled at 5% in example 1. The parenthesis are the possible values of $(n, P, L)$.

We apply the proposed adaptive method to the two groups of normalized data using 10-fold cross validation to jointly estimate the two groups of individual networks. The four reconstructed undirected graphs are shown in Figure 3.8. For the estimated two graphs in the first group, we get twelve edges and eleven edges respectively among eleven protein nodes. For the estimated two graphs in the second group, we obtain nine edges and eight edges respectively. They are both the subset of the edges of currently accepted cell-signaling network [47] as shown in Figure 3.7. It can be seen that Figure 3.8.a and Figure 3.8.b both show the common network structure within each group. Moreover, the results also shows the detected heterogeneity between two groups. The inferred networks in the second group miss some edges compared with the first group. For instance, the links from protein "PKA" to "Erk", "Erk" to "Mek" and "PKA" to "Mek". This is possibly caused by inadequate information in the data

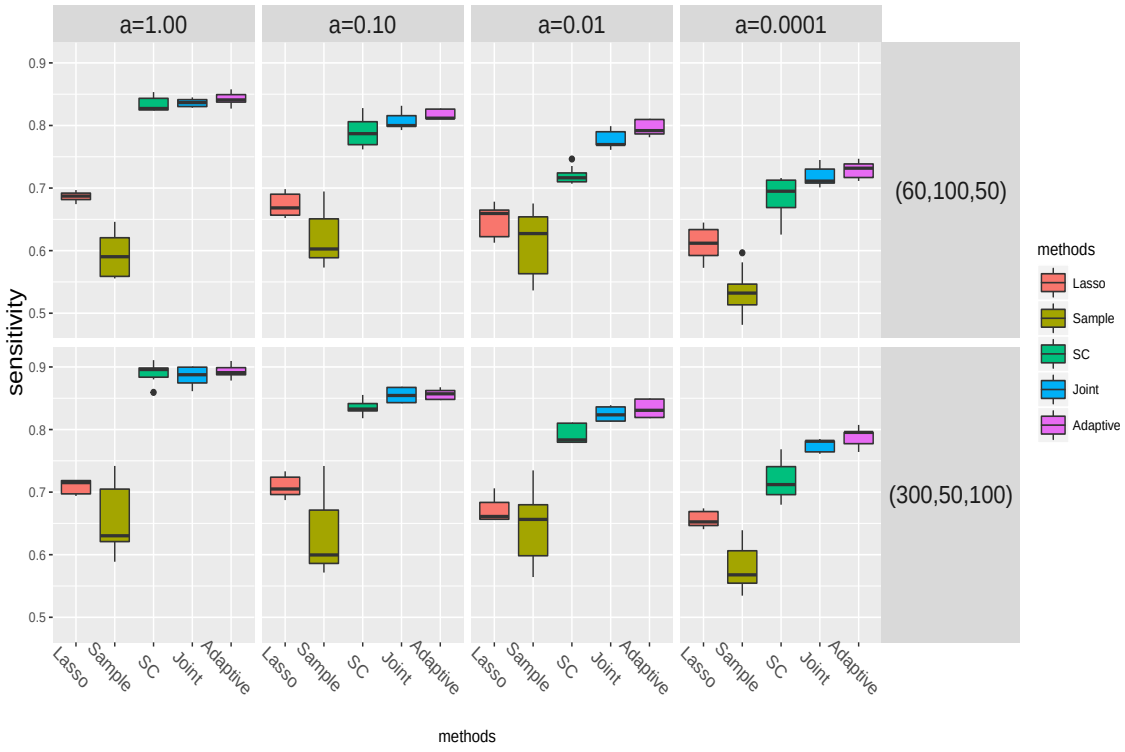



**Figure 3.6:** Boxplot of sensitivity under the false positive rate controlled at 5% in example 2. The parenthesis are the possible values of $(n, P, L)$.

brought about by not imposing direct specific perturbations on "PKA" in the five experimental settings. Specifically, our model also allows the commonality between two groups since the experimental settings in the two groups overlap to some degree. Altogether, our model appears to work well in that it can automatically capture the basic common structure of the proteins under different conditions, but also identify and recover their unique partial correlations among eleven proteins in two different groups.

## 3.7 Discussion

We proposed a joint sparse regression penalized model and an adaptive model that both jointly infer two groups of individual networks structure by obtaining the estimator of the partial correlations among random variables for each network

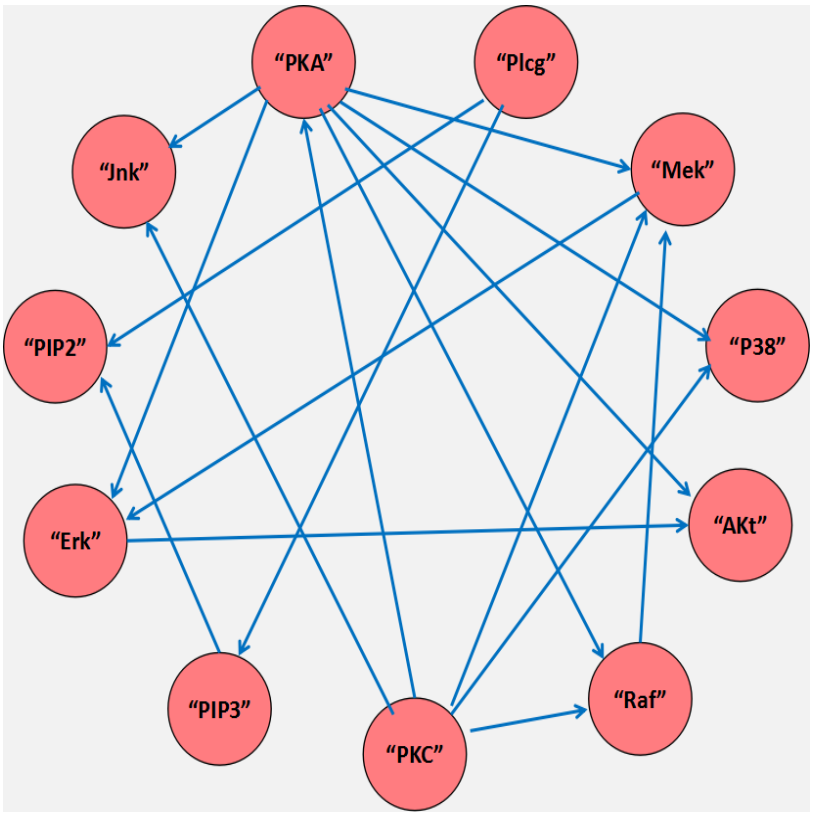
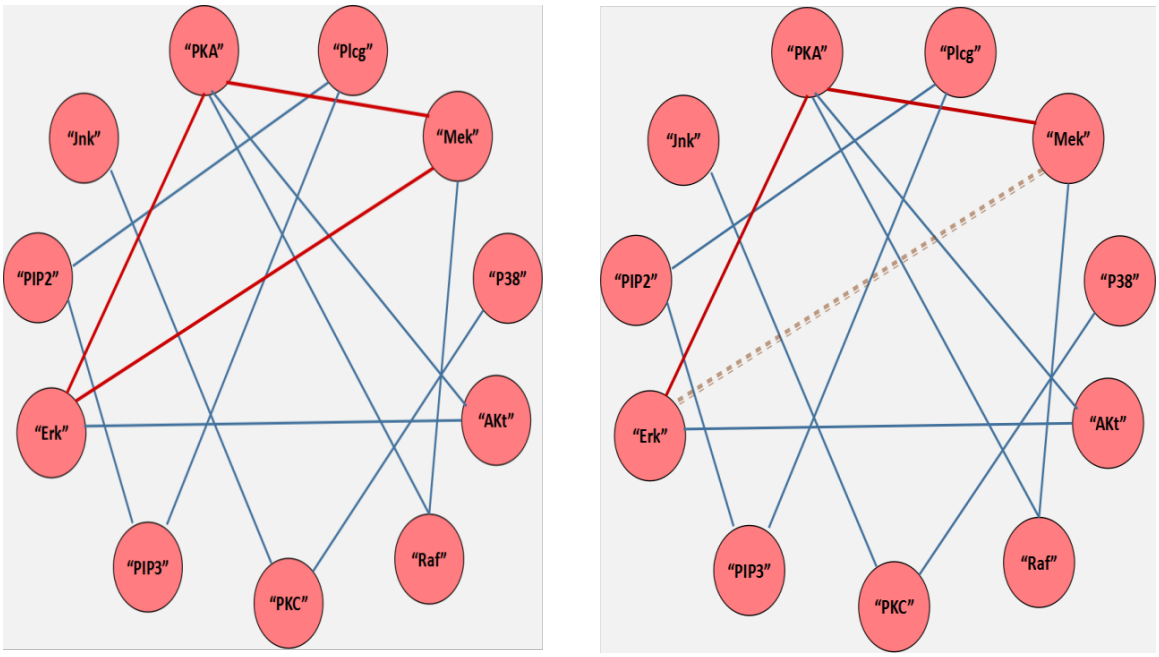


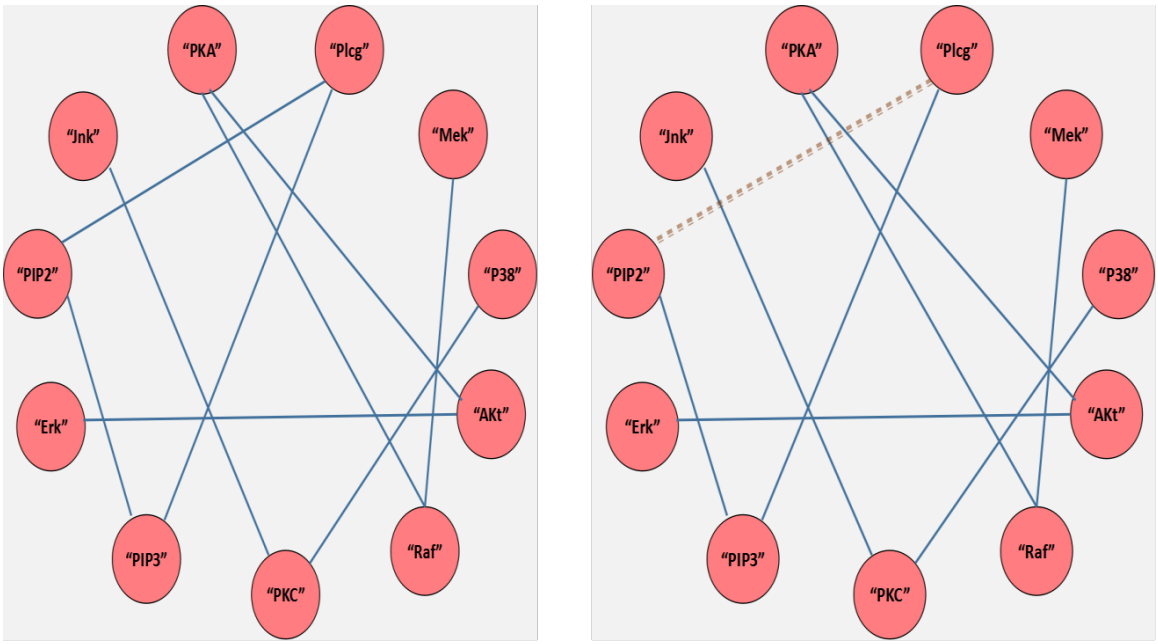

**Figure 3.7:** The directed graph shows the currently accepted cell-signaling network, reproduced from Sachs and coworkers [47]

**Table 3.1** A Summary of Nine Different Experimental Stimuli on Different Targets

| **Stimulus** | **Effect** |
|---|---|
| **CD3, CD28** | General stimulation |
| **ICAM-2** | General stimulation |
| **$\beta$2cAMP** | Specific stimulation: Actives PKA |
| **AKT-inhibitor** | Specific perturbation: Inhibits Akt |
| **U0126** | Specific perturbation: inhibits Mek |
| **PMA** | Specific perturbation: actives PKC |
| **G06976** | Specific perturbation: inhibits PKC |
| **Psitectorigenin** | Specific perturbation: inhibits PIP2 |
| **LY294002** | Specific perturbation: inhibits PI3K |




**(a)** Under nine conditions (left) and Under eight selected conditions (right)




**(b)** Under five conditions (left) and Under four selected conditions (right)

**Figure 3.8:** The reconstructed undirected graph by our proposed adaptive model. The blue lines are the links appearing in both groups, and red lines are the the links only appearing in one group compared with another group. The dash lines are the missing links compared with another inferred network within group.

simultaneously. The former utilizes the $\ell_2$ penalty to accommodate changes in network structure over two groups. The adaptive model explicitly employs a metric to iteratively learn the ratio of network distances within each group to that between two groups in the optimization procedure to detect the heterogeneity. Theoretically, under appropriate regularity conditions, we show asymptotic property both for the ideal $\ell_0$-constrained model and our proposed computationally surrogate model in consistently reconstructing the sparsity, group-specific commonality and heterogeneity between two groups. We expect to obtain some improvement over previous methods that do not include any group-wise learning procedure. With extensive simulation studies in finite samples, we demonstrate that our methods especially the data-driven adaptive method dominate other competing models in nonzero partial correlation selection, performing favorably with less variability. The application to polychromatic flow cytometry data sets also demonstrate the feasibility and effectiveness of the proposed adaptive method.

# APPENDIX A

# TECHNICAL PROOFS

## Proof of Theorem 1

**Proof 1.** *The proof is similar to that in [60]. We provide some key steps. Consider the case when* $\eta = 1$ *for simplicity. Denote* $\tilde{L}(\boldsymbol{y}, \boldsymbol{f}(x(s))) = L(\boldsymbol{y}, \boldsymbol{f}(x(s))) + \lambda J(\tilde{\boldsymbol{\beta}})$ *and* $\tilde{L}^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}(x(s))) = L^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}(x(s))) + \lambda J(\tilde{\boldsymbol{\beta}})$, *and* $E_n\left(\tilde{L}^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}^*(x(s)))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}(x(s)))\right) = n^{-1}\sum_{i=1}^{n}\left(\tilde{L}^{\mathrm{T}}(\boldsymbol{y}_i, \boldsymbol{f}^*(x(s_i))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{y}_i, \boldsymbol{f}(x(s_i)))\right) - E\left(\tilde{L}^{\mathrm{T}}(\boldsymbol{Y}, \boldsymbol{f}^*(x(S))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{Y}, \boldsymbol{f}(x(S)))\right)$ *be a scaled empirical process.*

*First, by assumption 2,* $\{|e(\hat{\boldsymbol{f}}, \boldsymbol{f}^0)| \geq a_1\delta_{n,d,K}^{2\alpha}\} \subset \{e_{L^{\mathrm{T}}}(\boldsymbol{f}, \boldsymbol{f}^0) \geq \delta_{n,d,K}^2\}$ *is a subset of*

$$\left\{\sup_{\boldsymbol{f}\in\mathcal{F}: e_{L^{\mathrm{T}}}(\boldsymbol{f},\boldsymbol{f}^0)\geq\delta_{n,d,K}^2}\sum_{i=1}^{n}(\tilde{L}^{\mathrm{T}}(\boldsymbol{y}_i, \boldsymbol{f}^*(x(s_i))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{y}_i, \boldsymbol{f}(x(s_i)))) \geq 0\right\}$$

*Therefore,* $\mathbb{P}(|e(\hat{\boldsymbol{f}}, \boldsymbol{f}^0)| \geq a_1\delta_{n,d,K}^{2\alpha})$ *is upper-bounded by*

$$I \equiv \mathbb{P}^*\left(\sup_{\boldsymbol{f}\in\mathcal{F}: e_{L^{\mathrm{T}}}(\boldsymbol{f},\boldsymbol{f}^0)\geq\delta_{n,d,K}^2}\sum_{i=1}^{n} n^{-1s}(\tilde{L}^{\mathrm{T}}(\boldsymbol{y}_i, \boldsymbol{f}^*(x(s_i))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{y}_i, \boldsymbol{f}(x(s_i)))) \geq 0\right) \leq I_1 + I_2$$

*where* $\mathbb{P}^*$ *is the outer probability, and*

$$I_1 = \sum_{i,j\geq 1}\mathbb{P}^*\left(\sup_{\boldsymbol{f}\in A_{ij}} E_n\left(\tilde{L}^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}^*(x(s)))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}(x(s)))\right) \geq M(i,j)\right)$$

$$I_2 = \sum_{i=1}^{\infty}\mathbb{P}^*\left(\sup_{\boldsymbol{f}\in A_{i0}} E_n\left(\tilde{L}^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}^*(x(s)))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{y}, \boldsymbol{f}(x(s)))\right) \geq M(i,0)\right)$$

*Here* $A_{ij} = \{\boldsymbol{f} \in \mathcal{F} : 2^{i-1}\delta_{n,d,K}^2 \leq e_{L^{\mathrm{T}}}(\boldsymbol{f}, \boldsymbol{f}^0) \leq 2^i\delta_{n,d,K}^2, 2^{j-1}max(\bar{J}_{d,K}, 1) \leq J(\tilde{\boldsymbol{\beta}}) \leq 2^j max(\bar{J}_{d,K}, 1)\}$, *and* $A_{i0} = \{\boldsymbol{f} \in \mathcal{F} : 2^{i-1}\delta_{n,d,K}^2 \leq e_{L^{\mathrm{T}}}(\boldsymbol{f}, \boldsymbol{f}^0) \leq 2^i\delta_{n,d,K}^2, J(\tilde{\boldsymbol{\beta}}) \leq max(\bar{J}_{d,K}, 1)\}$ *for* $i = 1, 2, \cdots$ *and* $j = 1, 2, \cdots$, *and Assumption 1 and* $\lambda^{-1} \geq 2\delta_{n,d,K}^{-2}\bar{J}_{d,K}$ *imply that*

$$\inf_{\boldsymbol{f}\in A_{ij}} E\left(\tilde{L}^{\mathrm{T}}(\boldsymbol{Y}, \boldsymbol{f}(x(S))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{Y}, \boldsymbol{f}^*(x(S)))\right) \geq 2^{i-1}\delta_{n,d,K}^2 + \lambda 2^{j-1}J(\tilde{\boldsymbol{\beta}}^*) \equiv M(i,j)$$

*for $i = 1, 2, \cdots$ and $j = 0, 1, 2, \cdots$. Similarly, for the variance, it follows from Assumption 2 that*

$$\sup_{\boldsymbol{f} \in A_{ij}} var\left(\tilde{L}^{\mathrm{T}}(\boldsymbol{Y}, \boldsymbol{f}(x(S))) - \tilde{L}^{\mathrm{T}}(\boldsymbol{Y}, \boldsymbol{f}^*(x(S)))\right) \leq 4a_2 M^{\gamma}(i,j),$$

*for $i = 1, 2, \cdots$ and $j = 0, 1, 2, \cdots$.*

*Next, an application of Theorem 3 of [53] yields, by Assumption 3 that*

$$\begin{aligned} I_1 &\leq \sum_{i,j:M(i,j\leq T)} 3\exp\left(-\frac{(1-\epsilon)n(M(i,j))^2}{2(4M^{\gamma}M(i,j) + M(i,j)T/2)}\right) \\ &\leq \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} 3\exp(-c_6 n M(i,j)^{2-\min(1,\gamma)}) \\ &\leq 3\exp(-a_6 n(\lambda \bar{J}_{d,K})^{2-\min(1,\gamma)})/[(1-\exp(-a_6 n(\lambda \bar{J}_{d,K})^{2-\min(1,\gamma)}))] \end{aligned} \tag{A.1}$$

*Similarly, $I_2$ can be bounded, and the desired result follows after some simple algebra.*

**Proof of Theorem 2**

**Proof 2.** *Since $\hat{\boldsymbol{\beta}}^{\mu}_{pj}(\lambda_1, \lambda_2, a)\hat{\boldsymbol{\beta}}^{v}_{pj}(\lambda_1, \lambda_2, a) > 0$, we have $sgn\{\hat{\boldsymbol{\beta}}^{\mu}_{pj}(\lambda_1, \lambda_2, a)\} = sgn\{\hat{\boldsymbol{\beta}}^{v}_{pj}(\lambda_1, \lambda_2, a)\}$. Because of (3.2), $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2, a)$ satisfies*

$$\left.\frac{\partial S(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_k}\right|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda_1,\lambda_2,a)} = 0 \text{ if } \hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2, a) \neq 0.$$

*Hence we have*

$$\begin{aligned} &-2{\boldsymbol{x}^{\mu}_{pj}}^{\mathrm{T}}\{y^{\mu}_p - x^{\mu}_p\hat{\boldsymbol{\beta}}^{\mu}_p\} + \lambda_1 w^{-\gamma}_{pj} sgn\{\hat{\boldsymbol{\beta}}^{\mu}_{pj}\} + \\ &2\lambda_2\sum_{s\sim\mu}(\hat{\boldsymbol{\beta}}^{\mu}_{pj} - \hat{\boldsymbol{\beta}}^{s}_{pj})\alpha_{pj} + 2\lambda_2 a\sum_{s\nsim\mu}(\hat{\boldsymbol{\beta}}^{\mu}_{pj} - \hat{\boldsymbol{\beta}}^{s}_{pj})\alpha_{pj} = 0 \end{aligned} \tag{A.2}$$

*and*

$$\begin{aligned} &-2{\boldsymbol{x}^{v}_{pj}}^{\mathrm{T}}\{y^{v}_p - x^{v}_p\hat{\boldsymbol{\beta}}^{v}_p\} + \lambda_1 w^{-\gamma}_{pj} sgn\{\hat{\boldsymbol{\beta}}^{v}_{pj}\} + \\ &2\lambda_2\sum_{t\sim v}(\hat{\boldsymbol{\beta}}^{v}_{pj} - \hat{\boldsymbol{\beta}}^{t}_{pj})\alpha_{pj} + 2\lambda_2 a\sum_{t\nsim v}(\hat{\boldsymbol{\beta}}^{v}_{pj} - \hat{\boldsymbol{\beta}}^{t}_{pj})\alpha_{pj} = 0 \end{aligned} \tag{A.3}$$

*By assumption, $sgn\{\hat{\boldsymbol{\beta}}^{\mu}_{pj}(\lambda_1, \lambda_2, a)\} = sgn\{\hat{\boldsymbol{\beta}}^{v}_{pj}(\lambda_1, \lambda_2, a)\}$, and $\mu$ and $v$ are only linked to each other. Substracting equation (A.2) from equation (A.3)) gives*

$$-{\boldsymbol{x}^{\mu}_{pj}}^{\mathrm{T}}\hat{\boldsymbol{r}}^{\mu}_p + {\boldsymbol{x}^{v}_{pj}}^{\mathrm{T}}\hat{\boldsymbol{r}}^{v}_p + (2\lambda_2 + \lambda_2 a(K-2))\alpha_{pj}(\hat{\boldsymbol{\beta}}^{\mu}_{pj} - \hat{\boldsymbol{\beta}}^{v}_{pj}) = 0,$$

*where* $\boldsymbol{r}_p^k = \boldsymbol{y}_p^k - \boldsymbol{X}_p^k \boldsymbol{\beta}_p^k$ *is the residual vector of the* $p-th$ *model for the* $k-th$ *subject. Hence*

$$\hat{\boldsymbol{\beta}}_{pj}^{\mu} - \hat{\boldsymbol{\beta}}_{pj}^{v} = \frac{\boldsymbol{x}_{pj}^{\mu\,\mathrm{T}} \hat{\boldsymbol{r}}_p^{\mu} - \boldsymbol{x}_{pj}^{v\,\mathrm{T}} \hat{\boldsymbol{r}}_p^{v}}{(2\lambda_2 + \lambda_2 a(K-2))\alpha_{pj}}. \tag{A.4}$$

*Since* $\boldsymbol{X}_p^k$ *are standarized for each* $p$ *and* $k$, $\|\boldsymbol{x}_{pj}^{\mu} - \boldsymbol{x}_{pj}^{v}\|_2^2 = 2(1-\rho)$ *where* $\rho = \boldsymbol{x}_{pj}^{\mu\,\mathrm{T}} \boldsymbol{x}_{pj}^{v}$. *By problem (3.2), we much have*

$$S(\lambda_1, \lambda_2, a, \hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2, a)) \leq S(\lambda_1, \lambda_2, a, \hat{\boldsymbol{\beta}} = 0).$$

*i.e.,*

$$\begin{aligned} &\|\hat{\boldsymbol{r}}_p^{\mu}\|_2^2 + \|\hat{\boldsymbol{r}}_p^{v}\|_2^2 + \lambda_1 \|\hat{\boldsymbol{\beta}}\|_1 + \lambda_2 \Big(\sum_{\mu \sim v} \|\hat{\boldsymbol{\beta}}^{\mu} - \hat{\boldsymbol{\beta}}^{v}\|_A^2 + a \sum_{\mu \nsim v} \|\hat{\boldsymbol{\beta}}^{\mu} - \hat{\boldsymbol{\beta}}^{v}\|_A^2\Big) \\ &\leq \|\boldsymbol{y}\|_2^2 \end{aligned} \tag{A.5}$$

*So* $\sqrt{\|\hat{\boldsymbol{r}}_p^{\mu}\|_2^2 + \|\hat{\boldsymbol{r}}_p^{v}\|_2^2} \leq \|\boldsymbol{y}\|_2$ *Then equation (A.4) implies that*

$$\begin{aligned} D_{\lambda_1, \lambda_2, a}(\mu, v, p, j) &\leq \frac{\|\boldsymbol{x}_{pj}^{\mu} - \boldsymbol{x}_{pj}^{v}\|_2 (\|\hat{\boldsymbol{r}}_{pj}^{\mu}\|_2^2 + \|\hat{\boldsymbol{r}}_{pj}^{v}\|_2^2)^{1/2}}{\|\boldsymbol{y}\|_2 (2\lambda_2 + \lambda_2(K-2))\alpha_{pj}} \\ &\leq \frac{1}{(2\lambda_2 + \lambda_2 a(K-2))\alpha_{pj}} \sqrt{2(1-\rho)}. \end{aligned} \tag{A.6}$$

**Proof of Theorem 3**

**Proof 3.** *The part of setup in this proof is similar with that of [68].* $M = \{\boldsymbol{\beta} : C_1(\boldsymbol{\beta}) \leq d_0, C_2(\boldsymbol{\beta}) \leq c_2, C_3(\boldsymbol{\beta}) \leq c_3, \mathcal{G}(\boldsymbol{\beta}) \neq \mathcal{G}(\boldsymbol{\beta}^0)\}$. $M_A = \{\boldsymbol{\beta} \in M : A(\boldsymbol{\beta}) = A\}$. *Let a class of candidate subsets be* $\{A : A \neq A^0, |A| \leq d_0\}$ *for feature selection. Note that* $A \subset \{1, \cdots, d\}$ *can be partitioned into* $(A \setminus A^0) \cup (A^0 \cap A)$. *Let* $B_{kj} = \{A : A \neq A^0, |A^0 \cap A| = k, |A \setminus A^0| = j,\ k = 0, \cdots, d_0 - 1,\ j = 1, \cdots, d_0 - k\}$. *Note that* $B_{kj}$ *consists of* $\binom{d_0}{k}\binom{d-d_0}{j}$ *different elements* $A$*'s of sizes* $|A^0 \cap A| = k$ *and* $|A \setminus A^0| = j$. $M_A = \cup_{\mathcal{G} \in \{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\beta} \in M_A\}} M_{\mathcal{G}}$, *where* $M_{\mathcal{G}} = \{\boldsymbol{\beta} \in M : \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}\}$. *So* $M = \cup_{k=0}^{d_0-1} \cup_{j=1}^{d_0-k} \cup_{A \in B_{kj}} M_A = \cup_{k=0}^{d_0-1} \cup_{j=1}^{d_0-k} \cup_{A \in B_{kj}} \cup_{\mathcal{G} \in \{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\beta} \in M_A\}} M_{\mathcal{G}}$. *For* $A \in B_{kj}$, *under degree-of-separation condition (3.11),* $M_{\mathcal{G}} \subseteq M_{\mathcal{G}}{}' = \{\boldsymbol{\beta} : (d_0 - k)\mathrm{C}_{\min}(\boldsymbol{\beta}^0) \leq e(\boldsymbol{\beta}, \boldsymbol{\beta}^0)/n\}$ *for* $\mathcal{G} \in \{g(\boldsymbol{\beta}) : \boldsymbol{\beta} \in M_A\}$. *To bound the error probability, note that if* $\hat{\mathcal{G}}^{L_0} = \mathcal{G}^0$, *then* $\hat{\boldsymbol{\beta}}^{L_0} = \hat{\boldsymbol{\beta}}^{ol}$. *Thus* $\{\hat{\mathcal{G}^{L_0}} = \mathcal{G}^0\} = \{\hat{\boldsymbol{\beta}}^{L_0} = \hat{\boldsymbol{\beta}}^{ol}\}$. *So* $\{\hat{\boldsymbol{\beta}}^{L_0} \neq \hat{\boldsymbol{\beta}}^{ol}\} \subseteq \{Q(\hat{\boldsymbol{\beta}}^{L_0}) - Q(\hat{\boldsymbol{\beta}}^{ol}) \leq 0\} \subseteq \{Q(\hat{\boldsymbol{\beta}}^{L_0}) - Q(\boldsymbol{\beta}^0) \leq 0\}$. *This together with* $\{\hat{\boldsymbol{\beta}}^{L_0} \neq \hat{\boldsymbol{\beta}}^{ol}\} \subseteq \{\hat{\boldsymbol{\beta}}^{L_0} \in M\}$ *implies that* $\{\hat{\boldsymbol{\beta}}^{L_0} \neq \hat{\boldsymbol{\beta}}^{ol}\} \subseteq \{Q(\hat{\boldsymbol{\beta}}^{L_0}) - Q(\boldsymbol{\beta}^0) \leq 0\} \cap \{\hat{\boldsymbol{\beta}}^{L_0} \in M\}$.

*Consequently,* $\mathbb{P}(\hat{\boldsymbol{\beta}}^{L_0} \neq \hat{\boldsymbol{\beta}}^{ol})$

$$
\begin{aligned}
&\leq \mathbb{P}^*_{\boldsymbol{\beta}\in M}(Q(\boldsymbol{\beta}) - Q(\hat{\boldsymbol{\beta}}^{ol}) \leq 0) \\
&\leq \mathbb{P}^*_{\boldsymbol{\beta}\in M}(Q(\boldsymbol{\beta}) - Q(\boldsymbol{\beta}^0 \leq 0) \\
&\leq \sum_{k=0}^{d_0-1}\sum_{j=1}^{d_0-k}\sum_{A\in B_{kj}}\sum_{\mathcal{G}\in\{\mathcal{G}(\boldsymbol{\beta}):\boldsymbol{\beta}\in M_A\}} \mathbb{P}^*_{\boldsymbol{\beta}\in M}(Q(\boldsymbol{\beta}) - Q(\boldsymbol{\beta}^0) \leq 0) \\
&\leq \sum_{k=0}^{d_0-1}\sum_{j=1}^{d_0-k}\sum_{A\in B_{kj}}\sum_{\mathcal{G}\in\{\mathcal{G}(\boldsymbol{\beta}):\boldsymbol{\beta}\in M_A\}} \mathbb{P}^*_{\boldsymbol{\beta}\in M_{\mathcal{G}'}}(Q(\boldsymbol{\beta}) - Q(\boldsymbol{\beta}^0) \leq 0)
\end{aligned}
$$

*where* $\mathbb{P}^*$ *is the outer measure and* $Q(\hat{\boldsymbol{\beta}}^{ol}) < Q(\boldsymbol{\beta}^0)$ *by definition.*

*Let* $V_n(\boldsymbol{\beta} - \boldsymbol{\beta}^0) = \sqrt{n}[\frac{Q(\boldsymbol{\beta}^0)}{n} - E\frac{Q(\boldsymbol{\beta}^0)}{n}] - \sqrt{n}[\frac{Q(\boldsymbol{\beta})}{n} - E\frac{Q(\boldsymbol{\beta})}{n}]$. *Thus* $\{Q(\hat{\boldsymbol{\beta}}^{L_0}) - Q(\boldsymbol{\beta}^0) \leq 0\} \subset \{V_n(\hat{\boldsymbol{\beta}}^{L_0} - \boldsymbol{\beta}^0) \geq \sqrt{n}e(\hat{\boldsymbol{\beta}}^{L_0}, \boldsymbol{\beta}^0)/n\}$. *Then we have*

$$
\begin{aligned}
&\mathbb{P}(\hat{\boldsymbol{\beta}}^{L_0} \neq \hat{\boldsymbol{\beta}}^{ol}) \leq \\
&\sum_{k=0}^{d_0-1}\sum_{j=1}^{d_0-k}\sum_{A\in B_{kj}}\sum_{\mathcal{G}\in\{\mathcal{G}(\boldsymbol{\beta}):\boldsymbol{\beta}\in M_A\}} \mathbb{P}^*_{\boldsymbol{\beta}\in M_{\mathcal{G}'}}\left(V_n(\boldsymbol{\beta} - \boldsymbol{\beta}^0) \geq \sqrt{n}\frac{e(\boldsymbol{\beta}, \boldsymbol{\beta}^0)}{n}\right) \\
&\equiv I
\end{aligned}
$$

*For* $I$, *we apply Lemma 3.4 and Theorem 4.1 of [59] to bound it. Towards this end, we verify the entropy condition (11) for the local entropy over* $\mathcal{B}_{\mathcal{G}}$. *Note under Assumption 5, then for* $L > 0$, $\delta > 0$, $\sqrt{n}\delta \geq 1$, *there some constant* $c_0 > 0$, *such that* $H\left(\mu L\delta, \mathcal{B}_{\mathcal{G}}(L\delta)\right) \leq c_0(logd)|A|log(\frac{c'}{\mu})$. *Then for* $\delta = \delta_{n,d_0,d} = (c'c_0)^{\frac{1}{2}}log^{\frac{1}{2}}d(\frac{d_0}{n})^{\frac{1}{2}}$ *and* $n$ *sufficiently large satisfy*

$$
\frac{\int_0^1 \sqrt{H\left(\mu L\delta_{n,d_0,d}, \mathcal{B}_{\mathcal{G}}(L\delta_{n,d_0,d})\right)}d\mu}{\sqrt{n}L\delta_{n,d_0,d}} \to 0, \text{ as } L \to \infty \tag{A.7}
$$

*By (3.11),* $\mathrm{C}_{\min}(\boldsymbol{\beta}^0) \geq L^2\delta^2_{n,d_0,d}$ *implies that (A.7), provided that* $c_1 \geq c_0c'L^2d_0$.

*Using the facts about binomial coefficients:* $\sum_{j=0}^{d_0-k}\binom{d-d_0}{j} \leq (d - d_0 + 1)^{d_0-k}$, $\binom{d_0}{k} \leq d_0^{d_0-k}$ *and* $(d_0(d - d_0 + 1)))^{d_0-k} \leq (\frac{d+1}{2}^2)^{d_0-k}$. *Let* $M_i^* = max_{\substack{A\in B_{(d_0-i)j}\\ 1\leq j\leq i}}|\{\mathcal{G}(\boldsymbol{\beta}) : \boldsymbol{\beta} \in M_A\}|$ *and* $logM^* = max_{1\leq i\leq d_0}\frac{log(M_i^*)}{i}$. *By Assumption 4, Lemma 3.4 and Theorem 4.2 of [59], we obtain, that for* $C_0 > 0$ *and* $L_0 > 0$ *depending on* $(\alpha, \Lambda)$, $I$ *is upper*

*bounded by*

$$
\begin{aligned}
I &\leq \sum_{k=0}^{d_0-1} \binom{d_0}{k} \sum_{j=0}^{d_0-k} \binom{d-d_0}{j} M^*_{(d_0-k)} \exp(-C_0 n(d_0-k)\mathrm{C}_{\min}(\boldsymbol{\beta}^0)) \\
&\leq \sum_{k=0}^{d_0-1} (\frac{d+1}{2}^2)^{d_0-k} M^*_{(d_0-k)} \exp(-C_0 n(d_0-k)\mathrm{C}_{\min}(\boldsymbol{\beta}^0)) \\
&\leq \sum_{k=0}^{d_0} (\frac{d+1}{2}^2)^{i} M^*_i \exp(-C_0 n i \mathrm{C}_{\min}(\boldsymbol{\beta}^0)) \\
&= \sum_{k=0}^{d_0} \exp(-i(C_0 n \mathrm{C}_{\min}(\boldsymbol{\beta}^0) - 2log(\frac{d+1}{2}) - logM^*)) \\
&= \boldsymbol{R}(\exp(-C_0 n \mathrm{C}_{\min}(\boldsymbol{\beta}^0) + 2log(\frac{d+1}{2}) + logM^*))
\end{aligned}
$$

*where* $\boldsymbol{R}(x) = x/(1-x)$. *Note* $I \leq 1$, *hence* $x/(1-x) \leq 1$ *implies* $x \leq \frac{1}{2}$. *Thus we have* $x/(1-x) \leq 2x$. *Then*

$$
I \leq 2\exp(-C_0 n \mathrm{C}_{\min}(\boldsymbol{\beta}^0) + 2log(\frac{d+1}{2}) + logM^*).
$$

**Proof of Theorem 4**

**Proof 4.** *Let* $\boldsymbol{\beta}_p^k = \boldsymbol{\beta}_p^{k0} + \frac{\boldsymbol{u}_p^k}{\sqrt{n}}$ *and* $\psi(\boldsymbol{u}) = \sum_k \sum_p \|\boldsymbol{y}_p^k - \boldsymbol{X}_p^k(\boldsymbol{\beta}_p^{k0} + \frac{\boldsymbol{u}_p^k}{\sqrt{n}})\|_2^2 + \lambda_n^{(1)} \sum_k \sum_p \|\boldsymbol{\beta}_p^{k0} + \frac{\boldsymbol{u}_p^k}{\sqrt{n}}\|_1 + \lambda_n^{(2)} \sum_{\mu \sim v} \sum_p \|\boldsymbol{\beta}_p^{\mu 0} + \frac{\boldsymbol{u}_p^\mu}{\sqrt{n}} - \boldsymbol{\beta}_p^{v0} + \frac{\boldsymbol{u}_p^v}{\sqrt{n}}\|_{\boldsymbol{A}_p}^2 + \lambda_n^{(3)} \sum_{\mu \nsim v} \sum_p \|\boldsymbol{\beta}_p^{\mu 0} + \frac{\boldsymbol{u}_p^\mu}{\sqrt{n}} - \boldsymbol{\beta}_p^{v0} + \frac{\boldsymbol{u}_p^v}{\sqrt{n}}\|_{\boldsymbol{A}_p}^2$. *Let* $\hat{\boldsymbol{u}}^n = \left(\hat{\boldsymbol{u}}_p^k\right)_{1 \leq k \leq K, 1 \leq p \leq P} = \mathrm{argmin}\psi_n(\boldsymbol{u})$. *Then* $\hat{\boldsymbol{\beta}}^n = \boldsymbol{\beta}^0 + \frac{\hat{\boldsymbol{u}}^n}{\sqrt{n}}$ *or* $\hat{\boldsymbol{u}}^n = \sqrt{n} \times (\hat{\boldsymbol{\beta}}^n - \boldsymbol{\beta}^0)$. *Note that* $\psi_n(\boldsymbol{u}) - \psi_n(0) = V_n(\boldsymbol{u})$, *where*

$$
\begin{aligned}
V_n(\boldsymbol{u}) = & \sum_k \sum_p \left( {\boldsymbol{u}_p^k}^{\mathrm{T}} (\frac{1}{n} {\boldsymbol{X}_p^k}^{\mathrm{T}} \boldsymbol{X}_p^k) \boldsymbol{u}_p^k - 2 \frac{{\boldsymbol{\varepsilon}_p^k}^{\mathrm{T}} \boldsymbol{X}_p^k}{\sqrt{n}} \boldsymbol{u}_p^k \right) \\
& + \frac{\lambda_n^{(1)}}{\sqrt{n}} \sum_k \sum_p \sum_{j=1}^{P-1} \hat{w}_{pj}^{-\gamma} \sqrt{n} (\|{\boldsymbol{\beta}_{pj}^k}^0 + \frac{\boldsymbol{u}_{pj}^k}{\sqrt{n}}\|_1 - \|{\boldsymbol{\beta}_{pj}^k}^0\|_1) \\
& + \lambda_n^{(2)} \sum_{\mu \sim v} \sum_p \left( \|\boldsymbol{\beta}_p^{\mu 0} - \boldsymbol{\beta}_p^{v0} + \frac{\boldsymbol{u}_p^\mu}{\sqrt{n}} - \frac{\boldsymbol{u}_p^v}{\sqrt{n}}\|_{\boldsymbol{A}_p}^2 - \|\boldsymbol{\beta}_p^{\mu 0} - \boldsymbol{\beta}_p^{v0}\|_{\boldsymbol{A}_p}^2 \right) \\
& + \lambda_n^{(3)} \sum_{\mu \nsim v} \sum_p \left( \|\boldsymbol{\beta}_p^{\mu 0} - \boldsymbol{\beta}_p^{v0} + \frac{\boldsymbol{u}_p^\mu}{\sqrt{n}} - \frac{\boldsymbol{u}_p^v}{\sqrt{n}}\|_{\boldsymbol{A}_p}^2 - \|\boldsymbol{\beta}_p^{\mu 0} - \boldsymbol{\beta}_p^{v0}\|_{\boldsymbol{A}_p}^2 \right)
\end{aligned}
\tag{A.8}
$$

*We know that* $\frac{1}{n} {\boldsymbol{X}_p^k}^{\mathrm{T}} \boldsymbol{X}_p^k \to \boldsymbol{C}_p^k$, *and* $\frac{{\boldsymbol{\varepsilon}_p^k}^{\mathrm{T}} \boldsymbol{X}_p^k}{\sqrt{n}} \xrightarrow{d} \boldsymbol{W}_p^k \sim N(0, {\sigma_p^k}^2 \boldsymbol{C}_p^k)$. *Now consider the limiting behavior of the second term of (A.8). If* ${\boldsymbol{\beta}_{pj}^k}^0 \neq 0$, *then*

$\sqrt{n}(\|\boldsymbol{\beta}_{pj}^{k\,0} + \frac{\boldsymbol{u}_{pj}^k}{\sqrt{n}}\|_1 - \|\boldsymbol{\beta}_{pj}^{k\,0}\|_1) \to \boldsymbol{u}_{pj}^k sgn(\boldsymbol{\beta}_{pj}^{k\,0})$. *By* $\frac{\lambda_n^{(1)}}{\sqrt{n}} \xrightarrow{p} 0$ *and* $\hat{w}_{pj} \xrightarrow{p} |\beta_{pj}^{k\,0}|$, *hence* $\frac{\lambda_n^{(1)}}{\sqrt{n}} \sum_k \sum_p \sum_{j=1}^{P-1} \hat{w}_{pj}^{-\gamma} \sqrt{n}(\|\boldsymbol{\beta}_{pj}^{k\,0} + \frac{\boldsymbol{u}_p^k}{\sqrt{n}}\|_1 - \|\boldsymbol{\beta}_{pj}^{k\,0}\|_1) \xrightarrow{p} 0$. *If* $\beta_{pj}^{k\,0} = 0$, *then* $\sqrt{n}(\|\boldsymbol{\beta}_{pj}^{k\,0} + \frac{\boldsymbol{u}_{pj}^k}{\sqrt{n}}\|_1 - \|\boldsymbol{\beta}_{pj}^{k\,0}\|_1) = |\boldsymbol{u}_{pj}^k|$ *and* $\frac{\lambda_n^{(1)}}{\sqrt{n}} \hat{w}_{pj}^{-\gamma} = \frac{\lambda_n^{(1)}}{\sqrt{n}} n^{\gamma/2} (\sum_k \sqrt{n}|\hat{\beta}_{pj}^k|)^{-\gamma}$, *where* $\sqrt{n}\hat{\boldsymbol{\beta}}_{pj}^k = O(1)$ *for all* $k$. *Next we consider the third term of (A.8).*

$$\lambda_n^{(2)} \sum_{\mu \sim v} \sum_p \left( \|\boldsymbol{\beta}_p^{\mu 0} - \boldsymbol{\beta}_p^{v0} + \frac{\boldsymbol{u}_p^\mu}{\sqrt{n}} - \frac{\boldsymbol{u}_p^v}{\sqrt{n}}\|_{\boldsymbol{A}_p}^2 - \|\boldsymbol{\beta}_p^{\mu 0} - \boldsymbol{\beta}_p^{v0}\|_{\boldsymbol{A}_p}^2 \right)$$

$$= 2\frac{\lambda_n^{(2)}}{\sqrt{n}} \sum_{\mu \sim v} \sum_p (\boldsymbol{\beta}_p^{\mu 0} - \boldsymbol{\beta}_p^{v0})^{\mathrm{T}} \boldsymbol{A}_p (\frac{\boldsymbol{u}_p^\mu}{\sqrt{n}} - \frac{\boldsymbol{u}_p^v}{\sqrt{n}}) + \frac{\lambda_n^{(2)}}{n} \sum_{\mu \sim v} \sum_p \|\frac{\boldsymbol{u}_p^\mu}{\sqrt{n}} - \frac{\boldsymbol{u}_p^v}{\sqrt{n}}\|_{\boldsymbol{A}_p}^2$$

$\to 0$ *as* $\frac{\lambda_n^{(2)}}{\sqrt{n}} \to 0$. *Similarly, the last term of (A.8) goes to zero as* $\frac{\lambda_n^{(3)}}{\sqrt{n}} \to 0$. *Thus, we see that* $V_n(\boldsymbol{u}) \xrightarrow{d} V(\boldsymbol{u})$ *for every* $\boldsymbol{u}$. *Since* $V_n(\boldsymbol{u})$ *is convex and* $V(\boldsymbol{u})$ *has a unique minimum, it follows [25] that*

$$\operatorname{argmin}(V_n(\boldsymbol{u})) = \sqrt{n}(\hat{\boldsymbol{\beta}}^n - \boldsymbol{\beta}^0) \xrightarrow{d} \operatorname{argmin}(V(\boldsymbol{u})).$$

**Proof of Proposition**

**Proof 5.** *The proof is similar to that of Theorem 4 with minor modifications. Since* $\hat{a}^n = \frac{E\{d(\hat{\boldsymbol{\beta}}^{\mu\,0}, \hat{\boldsymbol{\beta}}^{v\,0}) \mathbb{1}(\mu \sim v)\}}{E\{d(\hat{\boldsymbol{B}}^{\mu\,0}, \hat{\boldsymbol{B}}^{v\,0}) \mathbb{1}(\mu \nsim v)\}} \leq 1$, *then we satisfy the condition of Theorem 4 that* $\frac{\lambda_n^{(3)}}{\sqrt{n}} = \frac{\lambda_n^{(2)} \hat{a}^n}{\sqrt{n}} \to 0$ *as* $n \to 0$. *Hence by Theorem (4), the adaptive method achieves the properties of the oracle estimator.*

## APPENDIX B

## VERIFICATION OF ASSUMPTIONS

### Verification of Assumption 5

**Proof 6.** *Since* $e(\boldsymbol{\beta},\boldsymbol{\beta}^0) = EQ(\boldsymbol{\beta}) - EQ(\boldsymbol{\beta}^0) = \sum_k^K \sum_p^P \|\boldsymbol{X}_p^k\boldsymbol{\beta}_p^k - \boldsymbol{X}_p^k\boldsymbol{\beta}_p^{k^0}\|_2^2 \leq \sum_{k=1}^K \sum_{p=1}^P \|\boldsymbol{X}_p^k\|^2\|\boldsymbol{\beta}_p^k - \boldsymbol{\beta}_p^{k^0}\|^2$. *Hence* $e(\boldsymbol{\beta},\boldsymbol{\beta}^0) \leq \epsilon^2$ *implies that* $\|\boldsymbol{\beta}_p^k - \boldsymbol{\beta}_p^{k^0}\| \leq c'\epsilon$ *for some constant* $c' > 0$, *for* $p = 1,\cdots,P; k = 1,\cdots,K$. *Next we bound* $H(t,\mathcal{B}_\mathcal{G})$. *Let* $\mathcal{F}_{p\,\mathcal{G}}^k = \{f_p^k(\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}_p^k : \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}\}$. *Evidently,* $|f_p^k(\boldsymbol{X}) - f_p^{k^0}(\boldsymbol{X})| \leq \|\boldsymbol{X}_p^k\|_2\|\boldsymbol{\beta}_p^k - \boldsymbol{\beta}_p^{k^0}\|_2$. *Hence, the number of brackets needed to bracket* $\mathcal{F}_p^k \cap \{e(\boldsymbol{\beta},\boldsymbol{\beta}^0) \leq \epsilon^2\}$ *is no greater than that of balls of radius* $t/2$ *to cover the set* $\{\boldsymbol{\beta} : \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}, e(\boldsymbol{\beta},\boldsymbol{\beta}^0) \leq \epsilon^2\}$. *Therefore,* $H(t,\mathcal{F}_{p\,\mathcal{G}}^k \cap \{e(\boldsymbol{\beta},\boldsymbol{\beta}^0) \leq \epsilon^2\}) \leq O((P-1)log(\frac{c'\epsilon}{t}))$. *To connect* $\mathcal{F}_\mathcal{G}$ *with* $\mathcal{F}_{p\,\mathcal{G}}^k$, *note that* $\mathcal{F}_\mathcal{G} = \{\boldsymbol{f} : \boldsymbol{f} \in \Pi_{k=1}^K \Pi_{p=1}^P \mathcal{F}_{p\,\mathcal{G}}^k, \mathcal{G}(\boldsymbol{\beta}) = \mathcal{G}\}$. *Hence,* $H(t,\mathcal{F}_\mathcal{G} \cap \{e(\boldsymbol{\beta},\boldsymbol{\beta}^0) \leq \epsilon^2\}) \leq O((P-1)PKlog(\frac{c'\epsilon}{t}))$. *Since the number of nonzero* $\beta$ *is* $|A|$, *then we have* $H(t,\mathcal{F}_\mathcal{G} \cap \{e(\boldsymbol{\beta},\boldsymbol{\beta}^0) \leq \epsilon^2\}) \leq O(|A|log(\frac{c'\epsilon}{t}))$. *Therefore,* $H(t,\mathcal{B}_\mathcal{G}) \leq O(|A|log(\frac{c'\epsilon}{t})))$ *for some constant* $c' > 0$.

## REFERENCES


[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Multi-facet rating of product reviews. *In Proceedings of the 31st European Conference on Information Retrieval*, 2009.

[2] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

[3] Jake Bartlett and Russ Albright. Coming to a theater near you! Sentiment classification techniques using SAS Text Miner. *In SAS Global Forum 2008*, 2008.

[4] Peter J Bickel and Elizaveta Levina. Regularized estimation of large covariance matrices. *The Annals of Statistics*, pages 199–227, 2008.

[5] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[6] Pinvadee Chaovalit and Lina Zhou. A comparison between supervised and unsupervised classification approaches. *In Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.

[7] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. *Proceedings of EMNLP*, 2014.

[8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] Joachim Dahl, Lieven Vandenberghe, and Vwani Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. *Optimization Methods & Software*, 23(4):501–520, 2008.

[10] Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical LASSO for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

[11] Alexandre d'Aspremont, Onureena Banerjee, and Laurent El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.

[12] Edwards David. Introduction to graphical modelling, 2000.

[13] Arthur P Dempster. Covariance selection. *Biometrics*, pages 157–175, 1972.

[14] Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R Nevins, Guang Yao, and Mike West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004.

[15] Mathias Drton and Michael D Perlman. Model selection for Gaussian concentration graphs. *Biometrika*, 91(3):591–602, 2004.

[16] Susan T Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38:188, 2005.

[17] Jianqing Fan, Yang Feng, and Yichao Wu. Network exploration via the adaptive LASSO and SCAD penalties. *The annals of applied statistics*, 3(2):521, 2009.

[18] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[19] Jerome Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.

[20] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics*, 9(3):432–441, 2008.

[21] Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

[22] Karl J Friston. Functional and effective connectivity: a review. *Brain connectivity*, 1(1):13–36, 2011.

[23] Terrence S Furey, Nello Cristianini, Nigel Duffy, David W Bednarski, Michel Schummer, and David Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[24] Timothy S Gardner, Diego Di Bernardo, David Lorenz, and James J Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.

[25] Charles J Geyer. On the asymptotics of constrained m-estimation. *The Annals of Statistics*, pages 1993–2010, 1994.

[26] Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, pages 1–13, 2011.

[27] Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. Umbc ebiquity-core: Semantic textual similarity systems. *in Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, 1:44–52, 2013.

[28] Minqing Hu and Bing Liu. A list of positive and negative opinion words or sentiment words for english. *Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2004.

[29] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. *In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, 2004.

[30] Jianhua Z Huang, Naiping Liu, Mohsen Pourahmadi, and Linxu Liu. Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika*, 93(1):85–98, 2006.

[31] Shuai Huang, Jing Li, Liang Sun, Jun Liu, Teresa Wu, Kewei Chen, Adam Fleisher, Eric Reiman, and Jieping Ye. Learning brain connectivity of alzheimer's disease from neuroimaging data. In *Advances in Neural Information Processing Systems*, pages 808–816, 2009.

[32] Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

[33] Alistair Kennedy and Diana Inkpen. Sentiment classification of movie and product reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125, 2006.

[34] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing: Algorithms, Architectures and Applications. Sprintger-Verlag*. Springer, 1990.

[35] Keith Knight and Wenjiang Fu. Asymptotics for LASSO-type estimators. *Annals of statistics*, pages 1356–1378, 2000.

[36] Vladimir Koltchinskii et al. Local rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006.

[37] Thomas K Landauer and Susan T Dumais. A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[38] Hongzhe Li and Jiang Gui. Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317, 2006.

[39] Yuhua Li, Zuhair A Bandar, and David McLean. An approach for measuring semantic similarity between words using multiple information sources. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4):871–882, 2003.

[40] Christopher D Manning and Hinrich Schütze. Foundations of statistical natural language processing. *MIT Press*, 1999.

[41] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the LASSO. *The annals of statistics*, pages 1436–1462, 2006.

[42] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. 2004.

[43] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. *In Proceedings of the Conference on Empirical Methods in NLP*, pages 79–86, 2002.

[44] Livia Polanyi and Annie Zaenen. Contextual lexical valence shifters. *In Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications.*, 2004.

[45] Gabriel Recchia and Michael N Jones. More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. *Behavior Research Methods*, 41(3):647–656, 2009.

[46] Adam J Rothman, Peter J Bickel, Elizaveta Levina, Ji Zhu, et al. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

[47] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[48] Franco Salvetti, Christoph Reichenbach, and Stephen Lewis. Opinion polarity identification of movie reviews. *Computing Attitude and Affect in Text: Theory and Applications*, pages 303–316, 2006.

[49] Juliane Schäfer and Korbinian Strimmer. An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

[50] Eran Segal, Michael Shapira, Aviv Regev, Dana Pe'er, David Botstein, Daphne Koller, and Nir Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature genetics*, 34(2):166–176, 2003.

[51] Mostafa Shaikh, Helmut Prendinger, and Ishizuka Mitsuru. Assessing sentiment of text by semantic dependency and contextual valence analysis. *In Proc 2nd Intl Conf on Affective Computing and Intelligent Interaction*, (ACII07), 2007.

[52] Xiaotong Shen, Wei Pan, and Yunzhang Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232, 2012.

[53] Xiaotong Shen and Wing Hung Wong. Convergence rate of sieve estimates. *The Annals of Statistics*, pages 580–615, 1994.

[54] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. *The MIT Press*, 1966.

[55] Maite Taboada, Caroline Anthony, and Kimberly Voll. Creating semantic orientation dictionaries. *In Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, pages 427–432, 2006.

[56] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307, 2011.

[57] Jesper Tegner, MK Stephen Yeung, Jeff Hasty, and James J Collins. Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciences*, 100(10):5944–5949, 2003.

[58] Peter Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *In Proceedings of 40th Meeting of the Association for Computational Linguistics*, pages 417–424, 2002.

[59] Sara Van de Geer. Estimating a regression function. *The Annals of Statistics*, pages 907–924, 1990.

[60] Junhui Wang, Xiaotong Shen, Yiwen Sun, and Annie Qu. Classification with unstructured predictors and an application to sentiment analysis. *Journal of the American Statistical Association*, (just-accepted), 2015.

[61] Janyce Wiebe. Learning subjective adjectives from corpora. *In Proceedings of 17th National Conference on Artificial Intelligence (AAAI)*, pages 735–740, 2000.

[62] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. *In Empirical Methods in Natural Language Processing (EMNLP)*, 2005.

[63] Masanao Yajima, Donatello Telesca, Yuan Ji, and Peter Muller. Differential patterns of interaction and Gaussian graphical models. 2012.

[64] Sen Yang, Lei Yuan, Ying-Cheng Lai, Xiaotong Shen, Peter Wonka, and Jieping Ye. Feature grouping and selection over an undirected graph. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 922–930. ACM, 2012.

[65] Ainur Yessenalina and Claire Cardie. Compositional matrix-space models for sentiment analysis. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 172–182, 2011.

[66] Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[67] Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. Combining lexicon-based and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 2011.

[68] Yunzhang Zhu, Xiaotong Shen, and Wei Pan. Structural pursuit over multiple undirected graphs. *Journal of the American Statistical Association*, 109(508):1683–1696, 2014.

[69] Hui Zou. The adaptive LASSO and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.