On The Structured Manifold Optimization: Reduced-rank and Positive Definite Matrix Estimation

by

Ting Yuan
B.S. University of Science and Technology of China

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Chicago, 2015

Chicago, Illinois

Defense Committee:
Samad Hedayat, Advisor and Chair of the Committee
Junhui Wang, Advisor, City University of Hong Kong
Jie Yang
Jing Wang
Ryan Martin

ACKNOWLEDGMENTS

First and foremost, I want to express the sincere gratitude to my advisors, Professor Samad Hedayat and Professor Junhui Wang. They are introducers to my thesis topics and they have a great influence over the results of my work with their knowledge and insights. I appreciate the energy they spend throughout my study in statistics, as well as the profusion of their advices. I will pass on their enthusiasm on statistical machine learning.

I gained many benefits from several statisticians who raised useful comments and suggestions during the preparation of this work. In particular I thank Professor Jie Yang for the fruitful suggestions. Moreover, I thank Professor Jing Wang and Professor Ryan Martin as my thesis committee members.

I also appreciate the help from graduate students of Department of MSCS in UIC. The department provides an exciting environment of study. In particular I thank Yi Huang for her insightful discussion.

ACKNOWLEDGMENTS (Continued)

Contribution of Authors

Chapter 1 is an introduction to my dissertation topic, and it highlights the significance of my research problem. Chapter 2 represents an unpublished manuscript currently under review, and ultimately it will be a publication for which I am the primary author, and Professor Junhui Wang provides wonderful advices. Chapter 3 represents a published work (Yuan, T. and Wang, J. (2012) A coordinate descent algorithm for sparse positive definite matrix estimation. *Statistical Analysis and Data Mining*) where I am the first author and Professor Junhui Wang makes great suggestions. Chapter 4 concludes my thesis and discusses the possible future work. Chapter 5 is the appendix.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF ABBREVIATIONS

$\lvert\mathcal{A}\rvert$	Cardinality of $\mathcal{A}$
$\mathbf{A}^k$	The k-th row of matrix $\mathbf{A}$
$\mathbf{A}_k$	The k-th column of matrix $\mathbf{A}$
$\mathcal{R}^r$	The r-dimensional Euclidean space
$\lVert\mathbf{A}^k\rVert_2$	The Euclidean norm of row vector $\mathbf{A}^k$
$\mathbf{I}_r$	The r by r identity matrix
$\lVert\mathbf{A}\rVert_F$	The Frobenius norm of matrix $\mathbf{A}$
$\mathcal{A}_{\mathbf{C}}$	Active set or non-zero rows of matrix $\mathbf{C}$
$\widehat{\mathbf{A}}$	The optimizer with respect to $\mathbf{A}$ of some objective function
$\mathbf{x}_{(j)}$	The j-th covariate of random vector $\mathbf{x}$
$\mathbf{x}_{-j}$	All other covariates of random vector $\mathbf{x}$ except the j-th covariate
$1(\cdot)$	The indicator function

SUMMARY

This thesis mainly proposes optimization schemes regarding both types of structured matrices, as well as their applications in statistics.

The first structured matrix optimization is proposed under the reduced-rank constraint, and particularly it can be applied for conducting multi-label classification and variable selection simultaneously. To optimize the resultant cost function, an efficient alternating optimization scheme is developed, which alternates between the constrained manifold optimization algorithm and the gradient descent algorithm. The proposed algorithm is computationally efficient and delivers superior numerical performance in terms of both classification and variable selection accuracy. The asymptotic consistencies are also established to support the advantages of the proposed method.

The second structured matrix optimization proposes the estimation of sparse positive definite matrix generated by a generic coordinate descent (CD) algorithm, and particularly it can be applied to the estimation of the high-dimensional covariance matrix and inverse covariance or precision matrix with variable selection. To assure the positive definiteness of the estimated matrices, the proposed CD algorithm iteratively updates the current estimated matrix at either one diagonal entry or two symmetric off-diagonal entries, and appropriately determines the step size based on a simple sufficient and necessary condition. Furthermore, since each iteration updates only one or two coordinates, the sparsity in the estimated matrix can be achieved by early stopping the iteration. Extensive numerical experiments are conducted to demonstrate

the effectiveness of the proposed CD algorithm for estimation of the precision and covariance matrices.

CHAPTER 1

INTRODUCTION

Rapid developments of computer technology as well as Internet application bring us into a new information era, when the majority of individuals are capable of accessing and generating sets of massive information. "Big Data" technology is revolutionizing human society from all aspects, and the generation of super-scale quantity of data challenges people's ability to process it. Under the currently limited power of computing, statistical machine learning is developed as the cutting-edge technology to deal with this challenge. It researches a wide scope of topics aiming to improve efficiency in industrial and economic activity by extracting insights and summarizing conclusions from the large quantity of data.

A statistical machine learning modeler is to solve certain problem given some set of data. An appropriate family of models are selected as candidates to solve the formulated the problem, and the model is typically trained by solving a core optimization problem with respect to some contraints, and these contraints are presented in the form of structured matrices. In particular, many statistical machine learning problems can be reduced to the matrix optimization problems over a collection of structured matrices:

$$\widehat{\mathbf{M}} = \boldsymbol{argmin}_{\mathbf{M}\in\mathcal{M}} \ \ \mathcal{F}(\mathbf{M}),$$

where $\mathcal{M}$ is the collection of structured matrices, and $\mathcal{F}$ is the smooth objective function. It is likely that the core optimization problem is solved many times during model selection and model validation. Therefore the structured matrix optimizations are crucial for statistics and machine learning.

In this thesis, we discuss optimizations over two types of structured matrices: reduced-rank matrix, and sparse positive definite matrix. Consequently we apply them into practice.

The matrix optimization under reduced-rank constraint and sparse positive definite constraint are of central interest for both the statistics and machine learning communities, and solving many problems in both areas essentially comes down to these two matrix optimizations. The reduced-rank model relies on the basic assumptions where the underlying data approximately resides in a low-dimensional linear subspace, and it usually requires to estimate an unknown low-rank matrix from the limited set of observed entries. The reduced-rank method has a wide range of applications including computer vision, medical imaging, sensor networks; the sparse positive definite matrix estimation is frequently encountered in multivariate statistics, and sparse positive definite matrix also has a wide range of applications such as the functional magnetic resonance imaging, web-mining, bioinformatics, climate studies and risk managements.

We apply the idea of reduced-rank model into multi-label classification problem. The multi-label classification is an important research topic in statistics and machine learning (Zhang, 1998; Breiman and Frieman., 1997; Tsoumakas and Katakis, 2007). The primary task of the multi-label classification is to classify a given instance into multiple class labels. It differs from

the multi-class classification in that the multiple class labels are not exclusive, and one instance can be assigned into more than one class labels. Multi-label classification has a wide spectrum of applications in real life. For example, in text categorization (Dumais et al., 1998), one textual article can be associated with various topics, such as an article about "Lebron James" can be labeled by categories "Sports" and "Entertainment". In social network annotation (Peters et al, 2012), one individual ID can have multiple identities, such as "student", "salesman", "father". In medical biology (Barutcuoglu et al., 2006), one gene might be responsible for a number of biological functions. As opposed to the multi-class classification, the key challenge of the multi-label classification is to leverage the classification performance by incorporating the dependency structure among class labels. For example, an article about "Politics" is more likely to discuss "Economics" than "Sports" or "Entertainment".

To resolve this dependence structure issue in multi-label classification, Chapter 2 proposes a reduced-rank multi-label classification framework, equipped with the group lasso penalty for sparse estimation. The framework is motivated from the classical reduced-rank regression model (Izenman, 1975; Chen and Huang, 2012) in multivariate statistics. The model is popular in analyzing the multi-response regression problems, yet little seems to be done to extend it to multi-label classification. Here the reduced-rank model assumes that the classification functions corresponding to each class label reside in the same low-dimensional space. Clearly, the dependency structure among the multiple class labels is modeled through the common low-dimensional space, where each dimension can be regarded as one latent variable admitted into

the classification functions. A group lasso penalty is equipped with the proposed reduced-rank model to tackle the large-dimensional problem and identify the truly informative variables.

Sparse positive definite matrix estimation has attracted much attention in academics, such as the estimation of the precision matrix or the inverse covariance matrix (Edwards, 2000; Drton and Perlman, 2004; Meinshausen and Buhlmann, 2006; Friedman, et al., 2008) and the estimation of the covariance matrix (Rothman, 2012). Both the precision and the covariance matrices must be positive definite and have close connection to the Gaussian graphical models (Edward, 2000), since a multivariate Gaussian distribution can be fully characterized by its first two moments and the conditional or marginal dependence structure among the Gaussian variables can be determined by the precision or covariance matrix. Under the multivariate Gaussian distribution assumption, the sample covariance matrix is the maximum likelihood estimate of the covariance matrix, and its inverse (if invertible) can naturally serve as the estimator of the precision matrix. However, under scenarios with large dimension and small sample size, the sample covariance matrix is no longer invertible, and the precision and covariance matrices are sparse with many zero entries as it is generally believed that many covariates are either marginally or conditionally independent. To attain the sparsity, regularized likelihood functions (Drton and Perlman, 2004; Rothman, 2012) are often employed, where lasso-type penalties are used to encourage the sparsity of the resultant estimated matrix.

As an alternative to the regularized formulation for sparse matrix estimation, Chapter 3 proposes a generic coordinate descent (CD) framework which can be used for various optimizations with respect to the sparse positive definite matrix. The proposed CD algorithm iteratively

updates its current estimated matrix at either one diagonal entry or two symmetric off-diagonal entries, and appropriately determines the step size to assure the positive definiteness of the updated matrix. A simple necessary and sufficient condition is derived for determining the step size. Furthermore, since each iteration updates only one or two coordinates, the sparsity can be achieved by early stopping the iteration based on certain model selection criteria.

The rest of this thesis is organized as follows: Chapter 2 gives an introduction to multi-label classification and reduced-rank regression model, then proposes the reduced-rank multi-label classification framework, establishes the asymptotic consistencies, and presents the computing algorithm as well as the associated numerical analysis supporting the advantages of the framework; Chapter 3 examines the popularly used approaches to conduct sparse inverse covariance matrix estimation and sparse covariance matrix estimation, presents the generic coordinate descent framework with its applications in the precision matrix estimation and the covariance matrix estimation, and conducts the numerical analysis to demonstrate the superiority of the framework. Chapter 4 makes the conclusion remarks of the thesis and discusses the possible future work. Chapter 5 as the appendix collects the tables and graphs in numerical experiments for Chapter 2 and Chapter 3.

CHAPTER 2

REDUCED-RANK MATRIX MODEL FOR MULTI-LABEL CLASSIFICATION

In this chapter, we propose a reduced-rank multi-label classification framework, equipped with the group lasso penalty for sparse estimation. To optimize the resultant regularized likelihood, an efficient computing algorithm is developed, which alternates between a constrained manifold optimization algorithm and a standard coordinate descent algorithm. More importantly, the asymptotic behavior of the proposed multi-label method is quantified in terms of its estimation and variable selection consistencies. In addition, to optimize the tuning parameter selection, we adopt the Bayesian Information Criterion (BIC; Schwarz, 1978) and establish its model selection consistency. Extensive numerical experiments and real applications are examined to support the superior performance of the proposed method.

This chapter is organized as follows. Section 2.1 reviews the literature work on multi-label classification problem, and presents the necessary background on reduced-rank regression model, Stiefel manifold optimization, and curvilinear search algorithm on manifold optimization as the preparation for following sections. Section 2.2 presents the reduced-rank model used for multi-label classification. Section 2.3 establishes the asymptotic results for the proposed method in terms of the estimation and variable selection consistencies, as well as the model selection consistency via BIC. Section 2.4 develops an efficient computing algorithm alternating between the constrained manifold optimization and the coordinate descent algorithm. Section

2.5 conducts the numerical experiments on the simulated data sets, as well as the real examples. The effective comparison of the performance sufficiently demonstrates the advantage of our proposed model and method.

2.1 Introduction to Multi-label Classification and Reduced-rank Regression Model

Multi-label classification is a widely researched topic in machine learning and statistics, and there exists a large literature dealing with this problem. In this section, we introduce the formal definition of multi-label classification, and provide a timely review on the methods popularly utilized in the literature and in practice. Various approaches have been proposed for engineering multi-label classification problem, and most of the approaches can be categorized into two groups: problem transformation methods and algorithm adaptation methods. It is impossible to present the full set of methods, we select a few representatives in either group for the purpose of demonstration, and discuss their benefits and drawbacks.

Reduced-rank regression model is popularly used under the scenario where the response variables are predicted by a set of latent factors. In this section, we give a brief introduction to this model. In order to appropriately solve the optimization problem proposed by this model, we investigate a generalized optimization problem on a constrained manifold named the Stiefel manifold, and propose the computing algorithm to solve the optimization problem. The consequent algorithm is very useful in solving our proposed reduced-rank model for multi-label classification, and the work is cited when necessary.

2.1.1 Definition of Multi-label Classification

Suppose the instance $\mathbf{x}$ is in p-dimensional Euclidean space denoted by $\mathcal{R}^p$, and $\mathbf{y} = (y_1, y_2, ..., y_q)$ denotes the set with q labels, multi-lable classification is to fulfill the task of learning a classify function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_q(\mathbf{x}))$ from the multi-label training set $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}$ with $i = 1, ..., n$. For each instance $\mathbf{x}_i$, the binary response $\mathbf{y}_i$ is the set of labels associated with $\mathbf{x}_i$. For any given instance $\mathbf{x}$, the multi-label classifier $\mathbf{f}(\mathbf{x})$ is utilized to predict the set of proper labels, and $f_j(\mathbf{x})$ corresponds to the classify function for the j-th prediceted label.

2.1.2 Literature Review

2.1.2.1 Problem Transformation Methods

Under problem tranformation methods, the proposed approach transforms one multi-label classification problem into multiple binary classification problems, which can be solved by well-established algorithms. The following approaches are listed and examined: the Binary Relevance approach (Luaces et al., 2012) treats each class label separately and transforms a multi-label classification into multiple binary classification problems; the Chain Classifer approach (CC; Read et al., 2009) modifies the Binary Relevance approach and includes previous estimated class labels as covariates for classifying the current class label; the Label Powerset method (LP; Tsoumakas et al., 2008) translates a multi-label classification into a multi-class classification with each class representing one possible combination of multiple class labels.

- *Binary Relevance*

The basic idea of the Binary Relevance approach is to decompose a multi-label classification problem into multiple *indepedent* binary classification problems, and each binary classification problem corresponds to one possible label. The Binary Relevance approach first constructs a corresponding binary training set and utilizes the training set to induce a binary classifier by some binary learning algorithm.

Specifically, for the j-th label with $j = 1, ..., q$, some binary learning algorithm induces a binary classifier $g_j(\cdot) : \mathcal{R}^p \to \mathcal{R}$. For a given instance $\mathbf{x}$, the Binary Relevance approach predicts its associated label set $\widehat{\mathbf{y}}$ by querying labeling relevance on each individual binary classifier and then combine relevant labels following

$$\widehat{\mathbf{y}}(\mathbf{x}) = \{y_j | g_j(\mathbf{x}) > 0, 1 \leq j \leq q\}$$

The Binary Relevance approach is straightforward in handling the multi-label classification dataset, while the defects in this approach lie in the complete ignorance of potential correlations among labels, therefore this approach may suffer from the issue of class-imbalance when the number of class labels in the training set is large or the class labels are imbalanced as a consequence of correlations among labels in the training set.

- *Chain Classifier (CC)*

  The basic idea of CC is to transform a multi-label classification problem into a chain of binary classification problems, where the subsequent binary classifier in the chain relies on the prediction of preceding ones. For q possible class labels $\{y_1, y_2, ..., y_q\}$, CC first

permutes the ordering among the labels by $\tau : \{1, ..., q\} \rightarrow \{1, ..., q\}$, and constructs the binary training set by appending each instance with its relevance to those labels preceding the current estimated label. After that, CC employs some binary learning algorithm to induce a binary classifier.

Specifically, for the j-th label with $j = 1, ..., q$, the binary label classifier is induced as $g_{\tau(j)} : \mathcal{R}^p \times \{0, 1\}^{j-1} \rightarrow \mathcal{R}$, and its associated labels can be recursively derived as:

$$\widehat{y}_{\tau(1)}(\cdot) = sgn\left(g_{\tau(1)}(\cdot)\right),$$
$$\widehat{y}_{\tau(j)}(\cdot) = sgn\left(g_{\tau(j)}(\cdot, \widehat{y}_{\tau(1)}(\cdot), ..., \widehat{y}_{\tau(j-1)}(\cdot))\right) \quad (2 \leq j \leq q).$$

where $sgn(u) = 1$ if $u > 0$, and 0 otherwise.

It is clear that CC takes into account the potential interactions among labels by constructing the subsequent classifiers, however, its effectiveness is largely affected by the ordering specified by the permutation τ. To account for the effect of ordering, researchers propose an ensemble of the chain classifiers built with random permutations over the label spaces. Under that scenario, multiple chain classifiers are built with random permutations over the labels. For each permutation, a modified training set is used by permuting the original training set, CC is conducted, and the class labels are determined by voting on the ensemble of outcomes. However, under the scenario with large number of class labels, there exists overwhelming number of permutations, and this introduces an overly high computation complexity for both training and testing procedures.

- *Label Power (LP)*

  The basic idea of this method is to translate a multi-label classification problem into an ensemble of multi-class classification problems. LP translates the original multi-label training set into the multi-class training set by mapping each distinct label set into a new class. Afterwards, some multi-class classification algorithm is employed to induce a multi-class classifier, therefore LP reassigns the instance with the newly mapped single class and then participates in multi-class classifier induction.

  Unfortunately LP has two major drawbacks under consideration of practical feasibility. On the one hand it results in incompleteness, where this method confines its predicted label set to those appearing in the training set, and unable to generalize the label set outside; on the other hand, when the size of labels is large, there might be overwhelmingly many mapped classes leading to overly high complexity in training the multi-class classifiers.

  To overcome the drawbacks of this method while keeping its simplicity, the researchers propose two major classes of solutions. The first solution is Random k-Labelsets which combines the ensemble learning with LP to learn from multi-label data. The key strategy of this combination is to invoke LP only on random subsets of size k in the label set to guarantee the computational efficiency, and then ensemble various LP classifiers to achieve the predictive completeness. Under the scenario with large number of class labels, it still induces the polynomial order of computational complexity for both training and testing, and its performance largely depends on the partition of random subsets. The second

solution relies on the pruning classes and reducing the training label sets, however the class pruning may depend on the specific problems and result in the incompleteness of learned labels.

2.1.2.2 Algorithm Adaptation Methods

As for algorithm adaptation methods, a huge number of classification algorithms have been adapted for multi-label classification, such as nearest neighbor (Zhang and Zhou, 2007), decision tree (Clare and King, 2001), maximum margin approach (Elisseeff and Weston, 2002), neural network (Zhou and Zhang, 2006), Curds and Whey (CW; Breiman and Friedman, 1997), Partial Least Square for Discriminant Analysis (PLSDA; Barker and Rayens, 2003), Multi-label k-Nearest Neighbor (ML-kNN; Zhang and Zhou, 2007), and Collective Multi-label Classifier (Ghamrawi and McCallum, 2005). Besides, an extension of the reduced-rank model to multi-label classification is done in Yu et al. (2014), where a nuclear norm penalty is employed to attain the low rank structure. Extensive study shows that the performance of selected method in multi-label learning mainly relies on the problem, and the selecting methods are suggested based on the multi-label learning motivations (Madjarov et al., 2012). In this section we examine the following methods: Curds and Whey, Partial Least Square for Discriminant Analysis, Multi-label k-Nearest Neighbor, and Collective Multi-label Classifier for the purpose of demonstration.

- *Curds and Whey (CW)*

CW is originated from the multivariate linear model in dealing with the intercorrelated response variables, and it is extended to deal with multi-label data. In order to effectively utilize the dependence structure among labels, CW method conducts a two-step analysis. The first step applies Binary Relevance to estimate q classify functions $g_1(\cdot), ..., g_q(\cdot)$, and the second step regresses the binary responses in training datatset over q classify functions to achieve estimates. Specifically, the classifiers for a given instance $\mathbf{x}$ are

$$\widehat{\mathbf{y}}(\mathbf{x}) = \{\widehat{y}_j | \widehat{y}_j = 1\left((g_1(\mathbf{x}), ..., g_q(\mathbf{x}))(G^T G)^{-1} G^T Y_j > 0.5\right)\},$$

provided $G_{ij} = g_j(\mathbf{x}_i)$, Y_j denotes the j-th label of the training set as a column vector, $i = 1, ..., n$ and $j = 1, ..., q$. It treats the categorical response as continuous one and it may result in undesirable predictive performance.

- *Partial Least Square for Discriminant Analysis (PLSDA)*

  PLSDA consists in a classical partial least square regression where the response variable is categorical. It can be extended to deal with multi-label data by conducting q independent analysis. For the j-th label, it splits the sample cross validation to determine the number of latent variables by minimizing the predicted residual sum of squares. Afterwards it conducts the Partial Least Square regression to compute the scores for latent factors on the training data, and applies the binary logistic regression for classification. The

predicted probability is computed by logistic function given instance $\mathbf{x}$. Specifically, the probability of the instance $\mathbf{x}$ is labelled by j-th label is computed by

$$\left(1+\exp\left(-(\widehat{\beta}_{j0}+\widehat{\beta}_{j1}\tilde{x}_{j1}+...+\widehat{\beta}_{jr_j}\tilde{x}_{jr_j})\right)\right)^{-1}$$

where $(\tilde{x}_{j1},...,\tilde{x}_{jr_j})$ are selected latent factors, r_j is the number of selected latent factors, $\widehat{\beta}_{j1},...,\widehat{\beta}_{jr_j}$ are correspondent coefficients, and $\widehat{\beta}_{j0}$ is the intercept term. It is obvious that this approach treats each label independently, and the associated latent factors are also independently determined by each label.

- *Multi-label k-Nearest Neighbor (ML-kNN)*

  The basic idea of this algorithm is to adapt the k- nearest neighbor techniques to multi-label data. For any instance $\mathbf{x}$, let $\mathcal{N}(\mathbf{x})$ be its k nearest neighbors identified in training set $\mathcal{D}$, in which the similarity is generally measured with the Euclidean distance. For the j-th label, ML-kNN computes O_j recording the number of $\mathbf{x}$'s neighbors with label y_j. Let $P\left(y_j(\mathbf{x})=1|O_j\right)$ be the posterior probablity given a instance $\mathbf{x}$, and ML-kNN transforms the problem into estimation of this posterior probability.

  By the Bayes theorem,

$$\frac{P\left(y_j(\mathbf{x})=1|O_j\right)}{P\left(y_j(\mathbf{x})=0|O_j\right)}=\frac{P(y_j(\mathbf{x})=1)P(O_j|y_j(\mathbf{x})=1)}{P(y_j(\mathbf{x})=0)P(O_j|y_j(\mathbf{x})=0)}$$

and therefore it suffices to estimate the prior probability as well as the likelihood for predictions. ML-kNN fulfills the estimation procedures via the frequency counting strategy. The prior probabilities are estimated by counting the number of training examples associated with each label. Specifically,

$$P\left(y_j(\mathbf{x})=1\right)=\frac{1}{n}\sum_{i=1}^{n}1\left(y_j(\mathbf{x}_i)=1\right);P\left(y_j(\mathbf{x})=0\right)=1-P\left(y_j(\mathbf{x})=1\right)\quad\left(1\leq j\leq q\right).$$

For the j-th label, the likelihoods are estimated by maintaining two arrays $\kappa(r)$ and $\tilde{\kappa}(r)$: $\kappa(r)$ counts the number of training examples with j-th label and exact r neighbors with label y_j, and $\tilde{\kappa}(r)$ counts the number of training examples without j-th label and have exact r neighbors with j-th label. Afterwards, the likelihoods can be estimated based on κ_j and $\tilde{\kappa}$ where

$$\begin{aligned}P(O_j|y_j(\mathbf{x})=1)&=\frac{\kappa_j(O_j)}{\sum_{r=0}^{k}\kappa_j(r)},\\P(O_j|y_j(\mathbf{x})=0)&=\frac{\tilde{\kappa}_j(O_j)}{\sum_{r=0}^{k}\tilde{\kappa}_j(r)},\quad\left(1\leq j\leq q,0\leq O_j\leq k\right)\end{aligned}$$

where $k+1$ is the number of elements contained in κ_j, thereafter the predicted label set naturally follows.

ML-kNN has the advantage of inheriting the merits of Bayesian reasoning and Nearest Neighbor algorithm, where the class imbalance issue is mitigated due to the prior probability estimated for each class labels, as well as the varying neighbors identified for one

given instance adjust the decision boundary adaptively. However it fails to exploit label correlations by reasoning the relevance of each label separately.

- *Collective Multi-label Classifier*

  The basic idea of this approach is to adapt the model where the correlations among labels are encoded as constraints in terms of the probability distribution, and thereby it predicts the set of class labels collectively by estimating the parameters specified in the probability distribution. Frequently, the conditional probability mass function of class labels $\mathbf{y}$ given an instance $\mathbf{x}$ is assumed to be

$$P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}) = \frac{1}{Z_\Lambda(\mathbf{x})} \exp\left(\sum_k \lambda_k u_k(\mathbf{x}, \mathbf{y})\right),$$

  where λ_k is a set of parameters to be determined, $u_k(\mathbf{x}, \mathbf{y})$'s are the weighting functions to discriminate among q labels, and $Z_\Lambda(\mathbf{x})$ is the partition function serving as the normalization factor with $Z_\Lambda(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left(\sum_k \lambda_k u_k(\mathbf{x}, \mathbf{y})\right)$. One typical example of this approach may include Conditional Ising model with Covariates (Cheng et al., 2014), where the probability mass function is specified as

$$P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}) = \frac{1}{Z(\lambda(\mathbf{x}))} \exp\left(\sum_{j=1}^{q} \lambda_{jj}(\mathbf{x}) y_j + \sum_{1 \le j < k \le q} \lambda_{jk}(\mathbf{x}) y_j y_k\right),$$

  and each component of $\lambda(\mathbf{x})$ is characterized by a linear function of $\mathbf{x}$. Cheng et al. (2014) shows the theoretical analysis of consistency in estimated parameters, after equip-

ping the negative log-likelihood function of marginal distribution with sparsity-induced regularization terms.

In general this model is unable to tackle the optimization of model parameters directly from the probability mass function due to the complexity of computing $Z(\lambda(\mathbf{x}))$. Instead it fulfills the task by approximating the likelihood function with the marginal distributions of $y_j's$ given $\mathbf{x}$ such that the computation of the likelihood function is not overly complicated. However, even with the simplest pairwise correlation between labels as specified in the Conditional Ising Model with Covariates, the computational complexity of optimization remains at least $O(p^2 + q^2)$. This is computationally infeasible in practice, especially under scenarios with large p and large q.

2.1.2.3 Summary And Discussion

Most of the existing literature attempts to solve the multi-label classification problem by methods that either treat the class labels independently or model the dependency structure among labels with overly high order of computational complexity. In problem transformation methods, the interpretation of the Binary Revelance approach, CC and LP is simple and straightward. However, the Binary Relevance treats the labels independently and ignores the possible correlations among labels; the performance of CC is highly affected by the permutation of labels, and the upgraded combination between CC and ensemble learning mitigates the permutation affection while it introduces the complexity increasing exponentially with the number of labels; LP suffers from overwhelming number of classes when the number of labels is large, one proposed adaptation Random k-Labelsets is largely affected in the predictive performance

by the partition of random subset, and another proposed pruning strategy results in the incompleteness of learned labels. In algorithm adaptation methods, many classical classification algorithms are extended to deal with the multi-label data. Also either they fail to account for the dependency structure among labels, or they introduce a high level of computational complexity when model the label correlations. CW has the issue in treating the catergorical response as continuous one; PLSDA treats each label separately; MLk-NN also fails to exploit the label correlations by reasoning the label relavance independently; Collective Multi-label Classifier encodes the correlation in the specified probability distribution, while its computation complexity grows exponentially with the order of interaction it accounts for, and under most scenarios it tries to solve a modified approximated problem, so the estimated parameters may contain natural biasness.

The proposed reduced-rank model for multi-label classification framework has many advantages to overcome the drawbacks mentioned above. In this model, the label dependencies are captured by assuming the the classification functions corresponding to each class label reside in the same low-dimensional space, and the dependency structure is modeled through a common low-dimensional space, without accounting for the order of interaction among labels. The proposed method does not require replicating the subset of training set for ensemble learning, and at most the model has to estimatea linearly growing number of parameters with the number of class labels, therefore the computational complexity is much lower than the methods mentioned above. Furthermore, the computing algorithm is proposed to alternate between a fast

constrained manifold optimization and a coordinate descent algorithm, which makes optimization very efficient.

2.1.3 Reduced-rank Regression Model

This section presents the classical reduced rank regression model, and specifies the motivation for us to induce the reduced-rank model in dealing with multi-label classification problems.

The multivariate regression is presented as the well-known formalism

$$\mathbf{Y} = \mathbf{XC} + \mathbf{E}$$

where $\mathbf{E}$ is the error matrix with zero mean and finite variance, $\mathbf{Y}$ is the response matrix, $\mathbf{X}$ is the design matrix, and $\mathbf{C}$ is the coefficient matrix. This multivariate regression problem leads to the minimization of the least square criterion

$$RSS(\mathbf{C}) = \|\mathbf{Y} - \mathbf{XC}\|_F^2,$$

and its solution as the ordinary least square(OLS) estimate of $\mathbf{C}$ is $\widehat{\mathbf{C}}_{ols} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$. In fact, it can be noted that the OLS estimate each response variable separately without considering the possible inter-correlation between response variables, where the response variables might be highly correlated. In order to take into account the inter-correlation between response variables, many methods are proposed to overcome this drawback, such as the linear factor regression where the response variables are regressed against the linear combination of the predictors as latent factors, and the number of latent factors is much smaller than the

number of variables. The examples includes the principal compenent analysis, canonical correlation analysis and partial least square. Another class of approaches is to impose the linear vector space constraints by assuming that $\mathbf{C}$ is in low rank. It immediately follows that $\mathbf{C}$ can be factorized as the product between two low-rank matrices $\mathbf{C} = \mathbf{BA}$, $\mathbf{B}$ and $\mathbf{A}$ are low rank matrices. The problem is presented as

$$\widehat{\mathbf{C}} = \boldsymbol{argmin}_{\mathbf{C}} \quad \|\mathbf{Y} - \mathbf{XC}\|_F^2, \quad \text{rank}(\mathbf{C}) = r, \tag{2.1}$$

where $\mathbf{C}$ is a $p \times q$ matrix, $\mathbf{B}$ is a $p \times r$ matrix, $\mathbf{A}$ is an $r \times q$ matrix, and $r << \{p, q\}$. Under this constraint, the efficiency of the variable selection is improved, and the dimensionality reduction is achieved.

One immediate observation is the non-identifiability of the factorization, that is, with any fully ranked $r \times r$ square matrix $\mathbf{Q}$, the transformation $\mathbf{C} = (\mathbf{BQ})(\mathbf{Q}^{-1}\mathbf{A}) = \mathbf{BA}$ is not unique. However, under the orthogonal constraint $\mathbf{B}^T\mathbf{B} = \mathbf{I}_r$, Chen and Huang (2012) shows that the solution to Equation 2.1 is unique up to an $r \times r$ orthogonal matrix.

One key observation is the potential similarity between the interaction among response variables in reduce-rank regression model and the dependencies among class labels in multi-label classification data. This motivates us to incorporate the reduced-rank structure in multi-label classification framework to account for the label dependencies, and improve the performance of predicting class labels.

2.1.4 Stiefel Manifold Optimization

In Section 2.1.3, under the constraint $\mathbf{B}^T\mathbf{B} = \mathbf{I}_r$, Equation 2.1 follows as

$$(\widehat{\mathbf{B}}, \widehat{\mathbf{A}}) = \boldsymbol{argmin}_{\mathbf{B},\mathbf{A}} \quad \|\mathbf{Y} - \mathbf{XBA}\|_F^2, \quad \mathbf{B}^T\mathbf{B} = \mathbf{I}_r,$$

which becomes an optimization problem with a orthogonality constraint on $\mathbf{B}$ if $\mathbf{A}$ is fixed. In fact the generalization of the objective function in this optimization problem leads to an optimization problem on a special manifold named the Stiefel manifold, and it becomes useful in our proposed reduced-rank model for multi-label classification.

The Stiefel manifold (Edelman et al., 1998), is defined as $\mathcal{M}_r^p = \{\mathbf{B} \in \mathcal{R}^{p\times r} : \mathbf{B}^T\mathbf{B} = \mathbf{I}_r\}$. The optimization problem with orthogonality constraints presents:

$$\min_{\mathbf{B}\in\mathcal{R}^{p\times r}} \mathcal{F}(\mathbf{B}), \quad \text{s.t. } \mathbf{B}^T\mathbf{B} = \mathbf{I}_r, \tag{2.2}$$

where $\mathbf{I}_r$ is the r-dimensional identity matrix, and $\mathcal{F}$ is a differentiable objective function. Note that the Stiefel manifold under $r = 1$ scenario reduces to the unit-sphere manifold $Sp^{p-1} := \{\mathbf{B} \in \mathcal{R}^p : \|\mathbf{B}\|_2 = 1\}$.

It is generally difficult to solve the optimization problem since usually the constraint leads to many local minimizers. Under most of the scenarios, there is no guarantee to obtain the

global minimizer except for a few simple cases. Given a feasible point $\mathbf{B}$, as well as the gradient $\mathbf{G} := \mathcal{D}\mathcal{F}(\mathbf{B}) = \left(\frac{\partial \mathcal{F}(\mathbf{B})}{\partial \mathbf{B}}\right)$. We can define the anti-symmetric matrix in the form of constraints:

$$\mathbf{W} := \mathbf{G}\mathbf{B}^T - \mathbf{B}\mathbf{G}^T,$$

and we have the following lemmas for the Stiefel manifolds. These lemmas are stated without proof, and they can be found in Wen and Yin (2013).

Lemma 2.1.1. *Suppose that* $\widehat{\mathbf{B}}$ *is a local minimizer of Equation 2.2, then* $\widehat{\mathbf{B}}$ *must satisfy the first-order optimality condition, i.e., KKT condition,* $\widehat{\mathbf{G}} - \widehat{\mathbf{B}}\widehat{\mathbf{G}}^T\widehat{\mathbf{B}} = \mathbf{0}$ *and* $\widehat{\mathbf{B}}^T\widehat{\mathbf{B}} = \mathbf{I}_r$. *Moreover, define*

$$\bigtriangledown \mathcal{F}(\mathbf{B}) := \mathbf{G} - \mathbf{B}\mathbf{G}^T\mathbf{B},$$

then $\bigtriangledown \mathcal{F} = \mathbf{W}\mathbf{B}$, *and* $\bigtriangledown \mathcal{F} = 0$ *is the necessary and sufficient condition of that if* $\mathbf{W} = 0$ *given that* $\mathbf{W} = \mathbf{G}\mathbf{B}^T - \mathbf{B}\mathbf{G}^T$.

In order to descend the objective function $\mathcal{F}(\cdot)$, we need to modify the classical steepest descent to accomodate the orthogonality constraints. Since the gradient $\bigtriangledown \mathcal{F} = \mathbf{W}\mathbf{B}$, the natural idea is to compute the next iteration as $\mathcal{Y} = \mathbf{B} - \tau\mathbf{W}\mathbf{B}$, where τ is step-size. The key

challenge here is to accomodate the new trial point $\mathcal{Y}$ such that $\mathcal{Y} \in \mathcal{M}_p^r$. The new trial point is determined by the Crank-Nicolson-like scheme (Wen and Yin, 2013)

$$\mathcal{Y}(\tau) = \mathbf{B} - \tau \mathbf{W} \left(\mathbf{B} + \mathcal{Y}(\tau) \right),$$

where $\mathcal{Y}(\tau)$ is given in the closed form as

$$\mathcal{Y}(\tau) = (\mathbf{I} + \tau \mathbf{W})^{-1} (\mathbf{I} - \tau \mathbf{W}) \mathbf{B}, \tag{2.3}$$

which is known as the Cayley transformation. The definition of $\mathcal{Y}(\tau)$ leads to the nice properties that $\mathcal{Y}^T(\tau)\mathcal{Y}(\tau) = \mathbf{B}^T\mathbf{B} = \mathbf{I}_r$ for any $\tau \in \mathcal{R}$, as well as that $\mathcal{Y}'(0)$ equals the projection of $-\mathbf{G}$ onto the tangent space of $\mathcal{M}_p^r$ at $\mathbf{B}$, and hence $\mathcal{Y}(\tau)$ is a descent path. The line search algorithm can be conducted naturally to search for an appropriate step-size τ and guarantee the algorithm to converge to a stationary point.

And we have the following lemma

Lemma 2.1.2. *Given* $\mathbf{W} = \mathbf{G}\mathbf{B}^T - \mathbf{B}\mathbf{G}^T$*, then* $\mathcal{Y}(\tau)$ *in Equation 2.3 is a descent curve at* $\tau = 0$*. Moreover,*

$$\mathcal{F}'(\mathcal{Y}(0)) = \frac{\partial \mathcal{F}(\mathcal{Y}(\tau))}{\partial \tau}|_{\tau=0} = -\frac{1}{2}\|\mathbf{W}\|_F^2 < 0.$$

The decending property of curve $\mathcal{Y}(\tau)$ is obviously obtained by Lemma 2.1.2.

2.1.4.1 Curvilinear Search Algorithm

This section provides the general algorithm to solve Equation 2.2 numerically. Currently the matrix re-orthogonalization or generating points along geodesics of $\mathcal{M}_p^r$ is utilized by most of the existing constraint-preserving algorithms. Both methods are with drawbacks. The matrix re-orthgonalization must involve the matrix factorization such as the singular value decomposition. The generation of points along geodesics must compute the matrix exponentials or solve the partial differential equations. In either case, the complexity of computation is immense.

Wen and Yin (2013) introduces an efficient curvilinear search algorithm for optimization over the Stiefel manifold. Note that the tangent matrices of $\mathcal{M}_r^p$ at $\mathbf{B}$, denoted as $\mathcal{T}_{\mathbf{B}}\mathcal{M}_r^p$, must satisfy

$$\mathcal{T}_{\mathbf{B}}\mathcal{M}_r^p = \{\mathbf{Z} \in \mathcal{R}^{p\times r} : \mathbf{B}^T\mathbf{Z} + \mathbf{Z}^T\mathbf{B} = \mathbf{0}\}. \tag{2.4}$$

The Equation 2.4 motivates the adaptation of the curvilinear search algorithm on the Stiefel manifold, where it computes certain $\mathbf{Z}$ such that $\bigtriangledown_{\mathbf{B}}\mathcal{F} = \mathbf{Z}^T\mathbf{B}$, and update each iteration using $\mathbf{Z}$ instead of $\bigtriangledown_{\mathbf{B}}\mathcal{F}$.

The curvilinear search algorithm then minimizes $\mathcal{F}(\cdot)$ over the Stiefel manifold by conducting a line-search for τ to satisfy the Armijo-Wolfe criterion (Edelman et al., 1998).

Algorithm 2.1 (Curvilinear algorithm):

Step 1. Initialize $\mathbf{B}_{(0)}$ such that $\mathbf{B}_{(0)}^T\mathbf{B}_{(0)} = \mathbf{I}_r$. Set two constants $0 < \rho_1 \leq \rho_2 \leq 1$.

Step 2. Set $\tau = 1$, compute $\mathcal{Y}(\tau)$ as in Equation 2.3, and conduct a line search for $\hat{\tau}$ such that

$$\mathcal{F}'_\tau\left(\mathcal{Y}(\hat{\tau})\right) \geq \rho_1 \hat{\tau} \mathcal{F}'_\tau\left(\mathcal{Y}(0)\right), \ \mathcal{F}\left(\mathcal{Y}(\hat{\tau})\right) \leq \mathcal{F}\left(\mathcal{Y}(0)\right) + \rho_2 \hat{\tau} \mathcal{F}'_\tau\left(\mathcal{Y}(0)\right).$$

Update $\mathbf{B}_{(t+1)} = \mathcal{Y}(\hat{\tau})$.

Step 3. Set a small $\epsilon > 0$, repeat *Step 2* until $\|\mathbf{B}_{t+1} - \mathbf{B}_t\| < \epsilon$.

As showed in Wen and Yin (2013), the curvilinear algorithm is guaranteed to decrease the objective function and converge eventually, provided that there exists at least one solution of Equation 2.2. The line search scheme in *Step 2* can be further augmented by the trusted region algorithm (Nocedal and Yuan, 1998), which iteratively determines the current step-size based on the performance of last step. Based on our limited numerical experience, the curvilinear search algorithm appears to be more efficient than the algorithms in Wang and Wang (2010) and Chen and Huang (2012), as it optimizes with respect to $\mathbf{B}$ in the Stiefel manifold directly rather than to each column of $\mathbf{B}$ separately.

The complexity of computing the matrix inverse $(\mathbf{I} + \tau\mathbf{W})^{-1}$ dominates the computation of $\mathcal{Y}(\tau)$. While in particular it becomes cheap for the computation. The Sherman-Morrison-Woodbury (SMW) formalism formulates a fast way for computing the matrix inverses of a certain form.

$$\left(\mathbf{B} + \mathbf{U}\mathbf{V}^T\right)^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{U}\left(\mathbf{I} + \mathbf{V}^T\mathbf{B}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{B}^{-1}$$

The SMW formalism is important since it allows to update the inverse of $\mathbf{B}$ to the inverse of $\left(\mathbf{B}+\mathbf{U}\mathbf{V}^T\right)$ efficiently. If $\mathbf{U},\mathbf{V}$ are $p\times 2r, 2r<p$ matrices, assuming $\mathbf{B}^{-1}$ available, then the computation of $\left(\mathbf{B}+\mathbf{U}\mathbf{V}^T\right)^{-1}$ only requires inverting $\left(\mathbf{I}+\mathbf{V}^T\mathbf{B}^{-1}\mathbf{U}\right)$, which is a $2r\times 2r$ matrix, and we have the following lemma,

Lemma 2.1.3. *Given that* $\mathbf{W}=\mathbf{G}\mathbf{B}^T-\mathbf{B}\mathbf{G}^T$, *where* $\mathbf{B},\mathbf{G}\in\mathcal{R}^{p\times r}$, *then rewrite* $\mathbf{W}=\mathbf{U}\mathbf{V}^T$ *for* $\mathbf{U}=[\mathbf{G},\mathbf{B}]$ *and* $\mathbf{V}=[\mathbf{B},-\mathbf{G}]$, *then*

$$\mathcal{Y}(\tau)=\mathbf{B}-\tau\mathbf{U}\left(\mathbf{I}+\tau\mathbf{V}^T\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{B}.$$

In general, if $r<<p$, then inverting $\mathbf{I}+\tau\mathbf{V}^T\mathbf{U}\in\mathcal{R}^{2r\times 2r}$ is much easier than inverting $\mathbf{I}+\tau\mathbf{W}\in\mathcal{R}^{p\times p}$. $(\mathbf{I}+\tau\mathbf{W})^{-1}$ can be efficiently computed by taking advantage of the special form and applying the Sherman-Morrison-Woodbury formalism or Lemma 2.1.3.

2.2 Reduced-rank Model For Multi-label Data

As we recall, in multi-label classification, assume that the training set consists of $(\mathbf{x}_i,\mathbf{y}_i); i=1,...,n$, where $\mathbf{x}_i\in\mathcal{R}^p$ and $\mathbf{y}_i\in\{0,1\}^q$ with y_{ij} denoting whether $\mathbf{x}_i$ can be labeled by the j-th label. The primary task of multi-label classification is to construct a classification function vector $\mathbf{f}(\mathbf{x})=(f_1(\mathbf{x}),\ldots,f_q(\mathbf{x}))$ with $f_j(\mathbf{x})$ corresponding to the j-th label.

To model multi-label classification, one natural idea is to estimate f_j based on $(\mathbf{x}_i,y_{ij})$ for each individual label separately. However, this separate estimating method completely ignores the dependency structure among the labels and thus the classification performance can be suboptimal. Therefore, the key challenge of multi-label classification is to appropriately

integrate the dependency structure among labels into modeling so that improved classification performance can be attained.

In literature, it is common to assume that

$$P(\mathbf{Y}_{(j)} = 1|\mathbf{X} = \mathbf{x}) = g\left(\mathbf{x}^T\mathbf{C}_j^* + c_{0j}^*\right), \tag{2.5}$$

where $\mathbf{Y}_{(j)}$ denotes the j-th label in $\mathbf{Y}$, $f_j^*(\mathbf{x}) = \mathbf{x}^T\mathbf{C}_j^* + c_{0j}^*$ is the true classification function for the j-th label, and $g(u)$ is a link function that maps $\mathcal{R}$ to $[0, 1]$. Popular choices of $g(u)$ include the logistic function $g(u) = (1 + \exp(-u))^{-1}$ and the probit function $g(u) = \Phi(u)$ with Φ being the cumulative distribution function of standard normal distribution. In order to integrate the dependency structure among $\mathbf{Y}_{(j)}$'s, the reduced-rank model further assumes that $\text{rank}(\mathbf{C}^*) = r^* \leq \min(p, q)$, or more specifically,

$$\mathbf{C}^* = (\mathbf{C}_1^*, \ldots, \mathbf{C}_q^*) = \mathbf{B}^*\mathbf{A}^*, \tag{2.6}$$

where $\mathbf{B}^*$ is a $p \times r^*$ rank reduction matrix with $\mathbf{B}^{*T}\mathbf{B}^* = \mathbf{I}_{r^*}$, and $\mathbf{A}^* = (\mathbf{A}_1^*, \ldots, \mathbf{A}_q^*)$ is a $r^* \times q$ coefficient matrix. Here, $\mathbf{B}^*$ serves as the projection matrix mapping $\mathbf{x}$ from large-dimensional $\mathcal{R}^p$ to low-dimensional $\mathcal{R}^{r^*}$, and $\mathbf{A}_j^*$ is the corresponding coefficient vector for the j-th label in $\mathcal{R}^{r^*}$. Note that the factorization in Equation 2.6 is not unique as $\mathbf{B}^*\mathbf{A}^* = \mathbf{B}^*\mathbf{\Lambda}\mathbf{\Lambda}^T\mathbf{A}^*$ for any orthogonal matrix $\mathbf{\Lambda}$. However, it may not be an issue for multi-label classification since its goal is to construct the estimated classification function $\hat{\mathbf{f}}(\mathbf{x}) = \mathbf{x}^T\widehat{\mathbf{C}} + \widehat{\mathbf{c}}_0$, with $\widehat{\mathbf{C}} = \widehat{\mathbf{B}}\widehat{\mathbf{A}}$ and $\widehat{\mathbf{c}}_0 = (\widehat{c}_{01}, \ldots, \widehat{c}_{0q})^T$.

It is worth to re-emphasize that the model assumes the classification functions corresponding to each class label reside in the same low dimensional space as the instance space, and therefore the dependency structure is modeled through the common low-dimensional space. In other words, that the model assumes there exists joint indepedence among the $\mathbf{Y}'_{(j)}s$ given the projected space $\mathbf{x}^T\mathbf{B}^*$, while such joint independence is not necessarily existing given $\mathbf{x}$, therefore the depedency structure remains among $\mathbf{Y}'_{(j)}s$ given $\mathbf{x}$.

Consequently the negative log-likelihood function for the model Equation 2.5 with the reduced-rank structure Equation 2.6 is

$$\begin{aligned} l(\mathbf{C}, \mathbf{c}_0) &= l(\mathbf{B}, \mathbf{A}, \mathbf{c}_0) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{q}\Big(- y_{ij}\log g\big(\mathbf{x}_i^T\mathbf{B}\mathbf{A}_j + \mathbf{c}_{0j}\big) - (1-y_{ij})\log\big(1 - g(\mathbf{x}_i^T\mathbf{B}\mathbf{A}_j + \mathbf{c}_{0j})\big)\Big). \end{aligned} \tag{2.7}$$

Here g is assumed to be proper so that the estimate $(\widehat{\mathbf{C}}, \widehat{\mathbf{c}}_0)$ is unique. Furthermore, when the data dimension p becomes relatively large, it is generally believed that only a small proportion of the covariates in $\mathbf{x}$ are informative for the class labels and the remaining covariates are noise. To enforce the sparsity in $\widehat{\mathbf{C}}$, it is desirable to equip the likelihood criterion with a sparsity-induced regularization term. Therefore, the proposed multi-label classification framework becomes

$$\min_{\mathbf{B},\mathbf{A}} \; l_p(\mathbf{B}, \mathbf{A}, \mathbf{c}_0), \quad \text{s.t.} \;\; \mathbf{B}^T\mathbf{B} = \mathbf{I}_r, \tag{2.8}$$

where $l_p(\mathbf{B}, \mathbf{A}, \mathbf{c}_0) = l(\mathbf{B}, \mathbf{A}, \mathbf{c}_0) + \lambda J(\mathbf{B})$, $J(\mathbf{B})$ is a sparsity-induced penalty, λ is an adaptive tuning parameter, and the rank r is assumed to be small compared with p and q, suggesting

the low-rank structure of the multi-label classification model. In this paper, we set $J(\mathbf{B})$ as the adaptive group lasso penalty (Yuan and Lin, 2006; Wang, 2008),

$$J(\mathbf{B}) = \sum_{k=1}^{p} \frac{\|\mathbf{B}^k\|_2}{\|\mathbf{B}_0^k\|_2}, \tag{2.9}$$

where $\mathbf{B}_0$ is an initial consistent estimate of $\mathbf{B}^*$ that can be obtained by minimizing $l(\mathbf{B}, \mathbf{A}, \mathbf{c}_0)$. The group lasso penalty screens out the noise covariate $\mathbf{x}_k$ only when all components of $\mathbf{B}^k$ are zero.

To solve the constrained optimization in Equation 2.8, we develop a constrained manifold optimization algorithm in the following section, which is designed to conduct optimization over a manifold such as the one defined by $\mathbf{B}^T\mathbf{B} = \mathbf{I}_r$. It is also worth pointing out that although the solution to Equation 2.8 is only unique up to an orthogonal matrix, the resultant $\widehat{\mathbf{C}}$ is unique and the non-zero rows of $\widehat{\mathbf{B}}$ are unique as well (Chen and Huang, 2012).

2.3 Asymptotic Theories

This section establishes the asymptotic results for the proposed multi-label classification, in terms of the estimation and variable selection consistencies of $\widehat{\mathbf{C}} = \widehat{\mathbf{B}}\widehat{\mathbf{A}}$, as well as the model selection consistency via BIC.

2.3.1 Assumptions

For simplicity, we assume only the first p_0 variables are truly informative, implying only the first p_0 rows of $\mathbf{B}^*$ are non-zero. In addition, the following technical assumptions are made.

Assumption C1. For any u, $(g'(u))^2 \geq \max\left(g''(u)g(u), g''(u)(g(u)-1)\right)$.

Assumption C2. The set $\mathcal{C} = \{(\mathbf{C}, \mathbf{c}_0) : E\left(l_1(\mathbf{C}, \mathbf{c}_0, \mathbf{X}, \mathbf{Y})\right) < \infty\} \neq \emptyset$, where it has $l_1(\mathbf{C}, \mathbf{c}_0, \mathbf{X}, \mathbf{Y}) = \sum_{j=1}^{q} \big(- \mathbf{Y}_j \log g(\mathbf{X}^T\mathbf{C}_j + \mathbf{c}_{0j}) - (1 - \mathbf{Y}_j) \log \big(1 - g(\mathbf{X}^T\mathbf{C}_j + \mathbf{c}_{0j})\big)\big)$.

Assumption C3. There exists $0 < c_1 \leq c_2$ such that $c_1 \leq \phi_{\min}\left(I_1(\mathrm{vec}(\mathbf{B}^*, \mathbf{A}^*, \mathbf{c}_0^*))\right) \leq \phi_{\max}\left(I_1(\mathrm{vec}(\mathbf{B}^*, \mathbf{A}^*, \mathbf{c}_0^*))\right) \leq c_2$, where $I_1\left(\mathrm{vec}(\mathbf{B}^*, \mathbf{A}^*, \mathbf{c}_0^*)\right)$ is the Fisher information matrix induced by $\mathrm{vec}(\mathbf{B}^*, \mathbf{A}^*, \mathbf{c}_0^*)$, and $\phi_{min}(\cdot)$ and $\phi_{max}(\cdot)$ denote the minimal and maximal eigenvalues.

Assumption C1 implies that $-y_{ij}(\log g(u))'' - (1 - y_{ij})\left(\log(1 - g(u))\right)'' \geq 0$ for any $y_{ij} \in \{0, 1\}$, and thus the second derivative of Equation 2.7 on $(\mathbf{C}, \mathbf{c}_0)$ must be positive definite. Consequently, it assures that Equation 2.7 is a convex programming problem, and has a unique solution $\left(\widehat{\mathbf{C}}, \widehat{\mathbf{c}}_0\right)$. Assumption C1 is satisfied by many popular link functions, including the logistic function and the probit function. Assumption C2 assures that $(\mathbf{C}^*, \mathbf{c}_0^*) = \boldsymbol{argmin}_{\mathbf{C}, \mathbf{c}_0} E(l_1(\mathbf{C}, \mathbf{c}_0, \mathbf{X}, \mathbf{Y}))$ is also unique (Shao, 2003). Assumption C3 is a bounded eigenvalue assumption, and has been popularly used in literature to guard the Fisher information matrix from degeneration (Rothman et al., 2008). Theorem 2.3.1 establishes the estimation consistency of $(\widehat{\mathbf{C}}, \widehat{\mathbf{c}}_0)$ in estimating $(\mathbf{C}^*, \mathbf{c}_0^*)$.

2.3.2 Estimation Consistency

Theorem 2.3.1. *(Estimation consistency) Under Assumptions C1-C3, if r^* is known, $\lambda/\sqrt{n} \to 0$, there exists a local minimizer $(\widehat{\mathbf{B}}, \widehat{\mathbf{A}}, \widehat{\mathbf{c}}_0)$ of Equation 2.8, such that $\widehat{\mathbf{C}} = \widehat{\mathbf{B}}\widehat{\mathbf{A}}$ and $\widehat{\mathbf{c}}_0$ are $\sqrt{n}$-consistent estimates of $\mathbf{C}^*$ and $\mathbf{c}_0^*$.*

Proof of Theorem 2.3.1 : Since the factorization of $(\mathbf{B}^*, \mathbf{A}^*)$ in Equation 2.6 is not unique as $\mathbf{B}^*\mathbf{A}^* = \mathbf{B}^*\mathbf{\Lambda}\mathbf{\Lambda}^T\mathbf{A}^*$ for any orthogonal matrix $\mathbf{\Lambda}$, we denote by $\mathcal{T}_{\mathbf{C}^*}$ the collection of all such $(\mathbf{B}^*, \mathbf{A}^*)$'s. In the sequel, $(\mathbf{B}^*, \mathbf{A}^*)$ refers to any given pair in $\mathcal{T}_{\mathbf{C}^*}$. Let $\mathbf{\Gamma} = \mathrm{vec}(\mathbf{B}, \mathbf{A}, \mathbf{c}_0)$,

$\mathbf{\Gamma}^* = \text{vec}(\mathbf{B}^*, \mathbf{A}^*, \mathbf{c}_0^*)$, $T(\mathbf{\Gamma}) = l(\mathbf{B}, \mathbf{A}, \mathbf{c}_0)$, $T_p(\mathbf{\Gamma}) = l_p(\mathbf{B}, \mathbf{A}, \mathbf{c}_0)$. The Taylor expansion of $T(\mathbf{\Gamma})$ at $\mathbf{\Gamma}^*$ implies

$$T(\mathbf{\Gamma}) = T(\mathbf{\Gamma}^*) + \frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}^T}(\mathbf{\Gamma} - \mathbf{\Gamma}^*) + (\mathbf{\Gamma} - \mathbf{\Gamma}^*)^T \frac{1}{2} H(\tilde{\mathbf{\Gamma}})(\mathbf{\Gamma} - \mathbf{\Gamma}^*),$$

where $H(\tilde{\mathbf{\Gamma}})$ is the Hessian matrix, $\tilde{\mathbf{\Gamma}}$ is a matrix between $\mathbf{\Gamma}$ and $\mathbf{\Gamma}^*$.

The proof proceeds as follows. We first construct a neighbourhood of $\mathbf{\Gamma}^*$ as $N_n(\gamma, \mathbf{\Gamma}^*) = B_n(\gamma, \mathbf{\Gamma}^*) \bigcap \mathcal{M}_{\mathbf{B}}$, where $B_n(\gamma, \mathbf{\Gamma}^*) = \{\mathbf{\Gamma} : \|I_1(\mathbf{\Gamma}^*)^{1/2}(\mathbf{\Gamma} - \mathbf{\Gamma}^*)\|_2 \leq \gamma/\sqrt{n}\}$, and $\mathcal{M}_{\mathbf{B}} = \{\mathbf{\Gamma} : \mathbf{B}^T\mathbf{B} = \mathbf{I}_{r^*}\}$. Note that $B_n(\gamma, \mathbf{\Gamma}^*)$ is a connected and closed ellipsoid, and $\mathcal{M}_{\mathbf{B}}$ is identical with the manifold $\{\mathbf{B}, \mathbf{B}^T\mathbf{B} = \mathbf{I}_{r^*}\}$ times $\mathcal{R}^{qr^*+q}$, therefore it implies $N_n(\gamma, \mathbf{\Gamma}^*)$ is a closed and connected set. Then we show that $T_p(\mathbf{\Gamma}^*)$ is smaller than $T_p(\mathbf{\Gamma})$ for any $\mathbf{\Gamma}$ on the boundary of $N_n(\gamma, \mathbf{\Gamma}^*)$, which implies that there exists a local minimizer within $N_n(\gamma, \mathbf{\Gamma}^*)$. Finally, the desired result follows from the fact that $\mathbf{\Gamma}^* \in N_n(\gamma, \mathbf{\Gamma}^*)$ and thus that the distance between the local minimizer and $\mathbf{\Gamma}^*$ is upper bounded by $\gamma/\sqrt{n}$.

Let $\bar{N}_n(\gamma, \mathbf{\Gamma}^*)$ be the boundary of $N_n(\gamma, \mathbf{\Gamma}^*)$, then for any $\mathbf{\Gamma} \in \bar{N}_n(\gamma, \mathbf{\Gamma}^*)$,

$$T_p(\mathbf{\Gamma}) - T_p(\mathbf{\Gamma}^*) = T(\mathbf{\Gamma}) - T(\mathbf{\Gamma}^*) + \sum_{k=1}^{p} \lambda \|\mathbf{B}_0^k\|_2^{-1} \left(\|\mathbf{B}^k\|_2 - \|\mathbf{B}^{*k}\|_2 \right).$$

It follows from the fact $\|\mathbf{B}^{*k}\|_2 = 0$ for $k > p_0$, and the Cauchy-Schwarz inequality that

$$\begin{aligned}\sum_{k=1}^{p} \lambda \|\mathbf{B}_0^k\|_2^{-1} \left(\|\mathbf{B}^k\|_2 - \|\mathbf{B}^{*k}\|_2 \right) &\geq \sum_{k=1}^{p_0} \lambda \|\mathbf{B}_0^k\|_2^{-1} \left(\|\mathbf{B}^k\|_2 - \|\mathbf{B}^{*k}\|_2 \right) \\ &\geq -\sum_{k=1}^{p_0} \lambda \|\mathbf{B}_0^k\|_2^{-1} \|\mathbf{B}^k - \mathbf{B}^{*k}\|_2.\end{aligned}$$

Then we have

$$\begin{aligned}T_p(\mathbf{\Gamma}) - T_p(\mathbf{\Gamma}^*) &\geq T(\mathbf{\Gamma}) - T(\mathbf{\Gamma}^*) - \sum_{k=1}^{p_0} \lambda \|\mathbf{B}_0^k\|_2^{-1} \|\mathbf{\Gamma} - \mathbf{\Gamma}^*\|_2 \\ &\geq \frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}^T} (\mathbf{\Gamma} - \mathbf{\Gamma}^*) + \frac{1}{2} (\mathbf{\Gamma} - \mathbf{\Gamma}^*)^T H(\tilde{\mathbf{\Gamma}}) (\mathbf{\Gamma} - \mathbf{\Gamma}^*) \\ &\quad - \frac{\lambda p_0}{\sqrt{n} \min_{1 \leq k \leq p_0} \|\mathbf{B}_0^k\|_2} \|I_1(\mathbf{\Gamma}^*)^{-1/2}\|_2 \|\sqrt{n} I_1(\mathbf{\Gamma}^*)^{1/2} (\mathbf{\Gamma} - \mathbf{\Gamma}^*)\|_2.\end{aligned}$$

Next we bound each term separately. The first term can be bounded as

$$\begin{aligned}\frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}^T} (\mathbf{\Gamma} - \mathbf{\Gamma}^*) &= \left(\sqrt{n} I_1(\mathbf{\Gamma}^*)^{1/2} (\mathbf{\Gamma} - \mathbf{\Gamma}^*) \right)^T (n I_1(\mathbf{\Gamma}^*))^{-1/2} \frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}} \\ &\geq -\gamma \left\| n^{-1/2} I_1(\mathbf{\Gamma}^*)^{-1/2} \frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}} \right\|_2.\end{aligned}$$

By the Markov's inequality,

$$\begin{aligned}P\left(\left\| n^{-1/2} I_1(\mathbf{\Gamma}^*)^{-1/2} \frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}} \right\|_2 \leq \frac{\gamma}{2} \right) &\geq 1 - \frac{4}{\gamma^2} E \left\| n^{-1/2} I_1(\mathbf{\Gamma}^*)^{-1/2} \frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}} \right\|_2^2 \\ &= 1 - \frac{4}{\gamma^2} E \left(\frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}^T} n^{-1} I_1(\mathbf{\Gamma}^*)^{-1} \frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}} \right) \\ &= 1 - \frac{4}{\gamma^2} \dim(\mathbf{\Gamma}^*) = 1 - \frac{4(p+q)r^* + 4q}{\gamma^2},\end{aligned}$$

where the last equality follows from the fact that $I_1(\mathbf{\Gamma}^*)$ is the Fisher-information matrix, $T(\mathbf{\Gamma})$ is the log-likelihood. Then it follows that $P\left(\frac{\partial T(\mathbf{\Gamma}^*)}{\partial \mathbf{\Gamma}^T}(\mathbf{\Gamma}-\mathbf{\Gamma}^*) > -\frac{\gamma^2}{2}\right) \geq 1 - \frac{4(p+q)r^*+4q}{\gamma^2}$.

Since $\frac{1}{n}H(\tilde{\mathbf{\Gamma}}) \xrightarrow{P} I_1(\mathbf{\Gamma}^*)$ as $n \to \infty$, the second can be bounded as

$$\frac{1}{2}(\mathbf{\Gamma}-\mathbf{\Gamma}^*)^T H(\tilde{\mathbf{\Gamma}})(\mathbf{\Gamma}-\mathbf{\Gamma}^*) =$$
$$\frac{1}{2}\left(\sqrt{n}I_1(\mathbf{\Gamma}^*)^{1/2}(\mathbf{\Gamma}-\mathbf{\Gamma}^*)\right)^T I_1(\mathbf{\Gamma}^*)^{-1/2}\frac{1}{n}H(\tilde{\mathbf{\Gamma}})I_1(\mathbf{\Gamma}^*)^{-1/2}\left(\sqrt{n}I_1(\mathbf{\Gamma}^*)^{1/2}(\mathbf{\Gamma}-\mathbf{\Gamma}^*)\right) \xrightarrow{P} \frac{\gamma^2}{2}.$$

The last term can be bounded as follows. Since $\mathbf{B}_0$ is the consistent estimate to some $\mathbf{B}^*$, $\min_{1\leq k\leq p_0}\|\mathbf{B}_0^k\|_2^1 \geq c_3$ for certain $c_3 > 0$. By Assumption C3 there exists $c_4 > 0$ such that $\|I_1(\mathbf{\Gamma}^*)^{-1/2}\|_2 \leq c_4$. Along with $\lambda/\sqrt{n} \to 0$ as $n \to \infty$,

$$\frac{\lambda p_0}{\sqrt{n}\min_{1\leq k\leq p_0}\|\mathbf{B}_0^k\|_2}\|I_1(\mathbf{\Gamma}^*)^{-1/2}\|_2\|\sqrt{n}I_1(\mathbf{\Gamma}^*)^{1/2}(\mathbf{\Gamma}-\mathbf{\Gamma}^*)\|_2 \leq c_4 p_0\gamma\lambda(\sqrt{n}c_3)^{-1} \xrightarrow{P} 0.$$

Combining the above bounds, for any $\eta > 0$, we can select γ sufficiently large such that for any $\mathbf{\Gamma} \in \bar{N}_n(\gamma, \mathbf{\Gamma}^*)$, $P\left(T_p(\mathbf{\Gamma}) - T_p(\mathbf{\Gamma}^*) > 0\right) > 1-\eta$, therefore there exists at least one local minimizer $\widehat{\mathbf{\Gamma}}$ of $T_p(\cdot)$ inside $N_n(\gamma, \mathbf{\Gamma}^*)$, and it follows $\|\widehat{\mathbf{\Gamma}} - \mathbf{\Gamma}^*\|_2 \leq O(\gamma/\sqrt{n})$, $\|\widehat{\mathbf{c}}_0 - \mathbf{c}_0^*\| \leq O(\gamma/\sqrt{n})$, $\|\widehat{\mathbf{A}} - \mathbf{A}^*\|_F \leq O(\gamma/\sqrt{n})$, as well as $\|\widehat{\mathbf{B}} - \mathbf{B}^*\|_F \leq O(\gamma/\sqrt{n})$. It completes the proof of Theorem 2.3.1.

2.3.3 Variable Selection Consistency

Next, to quantify the variable selection performance, denote the active set $\mathcal{A}_{\mathbf{C}} = \{\mathbf{x}_{(k)} : \|\mathbf{C}^k\|_2 \neq 0\}$, where $\mathbf{x}_{(k)}$ denotes the k-th variable and $\mathbf{C}^k$ denotes the k-th row of $\mathbf{C}$. Clearly, $\mathcal{A}_{\mathbf{C}}$ contains the set of informative variables that contribute to at least one classification function

defined by $\mathbf{C}$. Theorem 2.3.2 shows that the estimated active set induced by $\widehat{\mathbf{C}}$ recovers the true active set induced by $\mathbf{C}^*$ with probability tending to 1.

Theorem 2.3.2. *(Variable selection consistency) Under Assumptions C1-C3, if r^* is known, $\lambda/\sqrt{n} \to 0$ and $\lambda \to \infty$, $P(\mathcal{A}_{\widehat{\mathbf{C}}} = \mathcal{A}_{\mathbf{C}^*}) \to 1$.*

Proof of Theorem 2.3.2: First we note that the active set induced by $\widehat{\mathbf{C}}$ is the same as that induced by $\widehat{\mathbf{B}}$ in the sense that $\|\widehat{\mathbf{C}}^k\| = 0$ if and only if $\|\widehat{\mathbf{B}}^k\| = 0$. We now prove this theorem by contradiction. Suppose that there exists some $k > p_0$ such that $\|\widehat{\mathbf{B}}^k\|_2 > 0$. Denote $\mathbf{G} = \frac{\partial l_p(\cdot)}{\partial \mathbf{B}}$, then the first order Karush-Kuhn-Tucker condition on $\widehat{\mathbf{B}} \in \mathcal{M}_{r^*}^p$ yields $\widehat{\mathbf{G}}\widehat{\mathbf{B}}^T = \widehat{\mathbf{B}}\widehat{\mathbf{G}}^T$ by Lemma 2.1.1, leading to $\widehat{\mathbf{G}} = \widehat{\mathbf{B}}\widehat{\mathbf{G}}^T\widehat{\mathbf{B}}$ given that $\widehat{\mathbf{B}}^T\widehat{\mathbf{B}} = \mathbf{I}_{r^*}$. That is, for any k, $\widehat{\mathbf{G}}^k = \widehat{\mathbf{B}}^k\widehat{\mathbf{G}}^T\widehat{\mathbf{B}}$, where $\widehat{\mathbf{G}}^k$ and $\widehat{\mathbf{B}}^k$ are the k-th rows of $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{B}}$, respectively. We will then show that $\|\widehat{\mathbf{G}}^k\|_2$ and $\|\widehat{\mathbf{B}}^k\widehat{\mathbf{G}}^T\widehat{\mathbf{B}}\|_2$ are of different magnitudes, leading to contradiction.

One one hand, we have

$$\begin{aligned}\widehat{\mathbf{G}}^k = \frac{\partial l_p(\widehat{\mathbf{B}}, \widehat{\mathbf{A}}, \widehat{\mathbf{c}}_0)}{\partial \mathbf{B}^k} &= \frac{\partial l(\widehat{\mathbf{B}}, \widehat{\mathbf{A}}, \widehat{\mathbf{c}}_0)}{\partial \mathbf{B}^k} + \lambda \frac{\partial J(\widehat{\mathbf{B}})}{\partial \mathbf{B}^k} \\ &= \frac{\partial l(\mathbf{B}^*, \widehat{\mathbf{A}}, \widehat{\mathbf{c}}_0)}{\partial \mathbf{B}^k} + \big(\widehat{\mathbf{B}}^k - \mathbf{B}^{*k}\big) H(\tilde{\mathbf{B}}^k) + \lambda \frac{\partial J(\widehat{\mathbf{B}})}{\partial \mathbf{B}^k},\end{aligned}$$

where the k-th row $\frac{\partial J(\widehat{\mathbf{B}})}{\partial \mathbf{B}^k} = \|\mathbf{B}_0^k\|_2^{-1} \frac{\widehat{\mathbf{B}}^k}{\|\widehat{\mathbf{B}}^k\|_2}$, and $\tilde{\mathbf{B}}^k$ is between $\widehat{\mathbf{B}}^k$ and $\mathbf{B}^{*k}$. By Theorem 2.3.1, $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{A}}$ are the $\sqrt{n}$-consistent estimates of some $\mathbf{B}^*$ and $\mathbf{A}^*$ in $\mathcal{T}_{\mathbf{C}^*}$, and $\widehat{\mathbf{c}}_0$ is the $\sqrt{n}$-estimate

of $\mathbf{c}_0^*$, then $n^{-1}H(\tilde{\mathbf{B}}^k) = I_1(\mathbf{B}^{*k}) + O_p(1/\sqrt{n})$, and $n^{-1}\frac{\partial l(\mathbf{B}^*, \widehat{\mathbf{A}}, \widehat{\mathbf{c}}_0)}{\partial \mathbf{B}^k} - S_1(\mathbf{B}^{*k}) = O_p(1/\sqrt{n})$, where the score function $S_1(\mathbf{B}^{*k}) = \frac{1}{n}\frac{\partial}{\partial \mathbf{B}^k} E\left(l(\mathbf{B}^*, \mathbf{A}^*, \mathbf{c}_0^*)\right) = 0$. Consequently, we have

$$\widehat{\mathbf{G}}^k = O_p(\sqrt{n}) + \widehat{\mathbf{B}}^k \left(nI_1(\mathbf{B}^{*k}) + O_p(\sqrt{n})\right) + \frac{\lambda \widehat{\mathbf{B}}^k}{\|\mathbf{B}_0^k\|_2 \|\widehat{\mathbf{B}}^k\|_2}.$$

Furthermore, as $\|\mathbf{B}_0^k\| = O_p(n^{-1/2})$ and $I_1(\mathbf{B}^*)$ is positive definite, $\|\widehat{\mathbf{G}}^k\|_2$ is of the same order as $O_p(n)\|\widehat{\mathbf{B}}^k\|_2$.

On the other hand,

$$\widehat{\mathbf{B}}^k \widehat{\mathbf{G}}^T \widehat{\mathbf{B}} = \widehat{\mathbf{B}}^k \left(O_p(\sqrt{n}) + \left(nI_1(\mathbf{B}^*) + O_p(\sqrt{n})\right)(\widehat{\mathbf{B}} - \mathbf{B}^*)^T + \lambda \frac{\partial J(\widehat{\mathbf{B}})}{\partial \mathbf{B}^T} \right) \widehat{\mathbf{B}}.$$

Then $\|\widehat{\mathbf{B}}^k \widehat{\mathbf{G}}^T \widehat{\mathbf{B}}\|_2 \leq \|\widehat{\mathbf{B}}^k\|_2 \left\| \left(O_p(\sqrt{n}) + (nI_1(\mathbf{B}^*) + O_p(\sqrt{n}))(\widehat{\mathbf{B}} - \mathbf{B}^*)^T + \lambda \frac{\partial J(\widehat{\mathbf{B}})}{\partial \mathbf{B}^T} \right) \widehat{\mathbf{B}} \right\|_F$. By Theorem 2.3.1, we have $\|\widehat{\mathbf{B}} - \mathbf{B}^*\|_F = O_p(1/\sqrt{n})$, and thus $\|\widehat{\mathbf{B}}^k \widehat{\mathbf{G}}^T \widehat{\mathbf{B}}\|_2 \leq O_p(\lambda\sqrt{n})\|\widehat{\mathbf{B}}^k\|_2$. Since $\|\widehat{\mathbf{B}}^k\|_2 > 0$ and $\lambda/\sqrt{n} \to 0$, it can be concluded that $\|\widehat{\mathbf{B}}^k \widehat{\mathbf{G}}^T \widehat{\mathbf{B}}\|_2$ is of smaller magnitude than $\|\widehat{\mathbf{G}}^k\|_2$, which contradicts with the fact that $\widehat{\mathbf{G}}^k = \widehat{\mathbf{B}}^k \widehat{\mathbf{G}}^T \widehat{\mathbf{B}}$. This implies that $\|\widehat{\mathbf{B}}^k\|_2 = 0$ for all $k > p_0$ and completes the proof.

2.3.4 Bayesian Information Criterion Selection Consistency

Note that the true r^* is assumed known in both Theorems 2.3.1 and 2.3.2, which may not be available in practice. We now establish the model selection consistency in selecting the tuning

parameter (λ, r) via BIC. The BIC criterion for the reduced-rank multi-label classification model is defined as

$$\mathrm{BIC}_{\lambda,r} = \frac{l\left(\widehat{\mathbf{C}}_{\lambda,r}, (\widehat{\mathbf{c}}_0)_{\lambda,r}\right)}{n} + \frac{\log n}{n} \cdot \mathrm{df}_{\widehat{\mathbf{C}}_{\lambda,r}},$$

where $\left(\widehat{\mathbf{C}}_{\lambda,r}, (\widehat{\mathbf{c}}_0)_{\lambda,r}\right)$ is the estimated coefficient matrix by solving Equation 2.8 subject to $\mathrm{rank}(\mathbf{C}) = r$, and $\mathrm{df}_{\widehat{\mathbf{C}}_{\lambda,r}} = r(|\mathcal{A}_{\widehat{\mathbf{C}}_{\lambda,r}}| + q - (r+1)/2)$ with $|\mathcal{A}|$ denoting the cardinality of $\mathcal{A}$.

Lemma 2.3.3. *Under Assumption C1-C3, let $\lambda = \lambda_n = \log n$, then as $n \to \infty$,*

$$BIC_{\lambda_n, r^*} - BIC_{\mathbf{C}^*, \mathbf{c}_0^*} \xrightarrow{P} 0$$

Proof of Lemma 2.3.3 : First we have

$$\mathrm{BIC}_{\lambda,r} - \mathrm{BIC}_{\mathbf{C}^*,\mathbf{c}_0^*} = \frac{l\left(\widehat{\mathbf{C}}_{\lambda,r}, (\widehat{\mathbf{c}}_0)_{\lambda,r}\right)}{n} - \frac{l\left(\mathbf{C}^*, \mathbf{c}_0^*\right)}{n} + \frac{\log n}{n}\left(\mathrm{df}_{\widehat{\mathbf{C}}_{\lambda,r}} - \mathrm{df}_{\widehat{\mathbf{C}}^*}\right). \tag{2.10}$$

Since $\lambda = \lambda_n = \log n, r = r^*$ satisfies the conditions in Theorem 2.3.1 and Theorem 2.3.2, $(\widehat{\mathbf{c}}_0)_{\lambda_n,r^*}$ is the consistent estimate of $\mathbf{c}_0^*$, $\widehat{\mathbf{C}}_{\lambda_n,r^*} = \widehat{\mathbf{B}}_{\lambda_n,r^*}\widehat{\mathbf{A}}_{\lambda_n,r^*}$ is the consistent estimate of $\mathbf{C}^*$, then $P\left(\mathrm{df}_{\widehat{\mathbf{C}}_{\lambda_n,r^*}} = \mathrm{df}_{\widehat{\mathbf{C}}^*}\right) \to 1$, as well as $\frac{l(\widehat{\mathbf{C}}_{\lambda_n,r^*},(\widehat{\mathbf{c}}_0)_{\lambda_n,r^*})}{n} - \frac{l(\mathbf{C}^*,\mathbf{c}_0^*)}{n} \xrightarrow{P} 0$, then it completes the proof.

Lemma 2.3.3 shows that the value of the BIC criterion for $(\widehat{C}, \widehat{c}_0)$ with $\lambda = \log n$ and $r = r^*$ converges to that of the true model in probability.

Next, for simplicity, we denote

$$\begin{aligned}\Omega_+ &= \{(\lambda, r) : \mathrm{df}_{\widehat{\mathbf{C}}_{\lambda,r}} > \mathrm{df}_{\mathbf{C}^*}\},\\ \Omega_- &= \{(\lambda, r) : \mathcal{A}_{\widehat{\mathbf{C}}_{\lambda,r}} \not\supseteq \mathcal{A}_{\mathbf{C}^*}, \text{or } \mathcal{A}_{\widehat{\mathbf{C}}_{\lambda,r}} \supseteq \mathcal{A}_{\mathbf{C}^*} \text{ and } r < r^*\}.\end{aligned}$$

Note that $\Omega_+ \cup \Omega_-$ forms the collection of (λ, r)'s failing to identify the true model. In other words, $(\Omega_+ \cup \Omega_-)^c = \{(\lambda, r) : \mathcal{A}_{\widehat{\mathbf{C}}_{\lambda,r}} = \mathcal{A}_{\mathbf{C}^*} \text{ and } r = r^*\}$.

Lemma 2.3.4. *Under Assumptions C1-C3, as $n \to \infty$,*

$$P\left(\inf_{(\lambda,r)\in\Omega_+\cup\Omega_-} BIC_{\lambda,r} > BIC_{\mathbf{C}^*,\mathbf{c}_0^*}\right) \to 1.$$

Proof of Lemma 2.3.4: The proof proceeds by cases:

(i) For $(\lambda, r) \in \Omega_+$ such that $\mathrm{df}_{\widehat{\mathbf{C}}_{\lambda,r}} > \mathrm{df}_{\mathbf{C}^*}$, from Equation 2.10 we have

$$\begin{aligned}\mathrm{BIC}_{\lambda,r} - \mathrm{BIC}_{\mathbf{C}^*,\mathbf{c}_0^*} &\geq \frac{l\left(\widehat{\mathbf{C}}_{\lambda,r}, (\widehat{\mathbf{c}}_0)_{\lambda,r}\right) - l\left(\mathbf{C}^*, \mathbf{c}_0^*\right)}{n} + \frac{\log n}{n}\\ &\geq \frac{l(\mathbf{C}_m, \mathbf{c}_{0m}) - l(\mathbf{C}^*, \mathbf{c}_0^*)}{n} + \frac{\log n}{n},\end{aligned}$$

where $(\mathbf{C}_m, \mathbf{c}_{0m})$ denote the minimizer of $l(\cdot)$. By the classical asymptotic theory, since p and q are fixed, as $n \to \infty$, $-2l\left(\mathbf{C}_m, \mathbf{c}_{0m}\right) + 2l\left(\mathbf{C}^*, \mathbf{c}_0^*\right) \xrightarrow{D} \chi^2_{(p+1)q}$ and $\log n \to \infty$, then it follows

$$P\left(\inf_{(\lambda,r)\in\Omega_+} \mathrm{BIC}_{\lambda,r} - \mathrm{BIC}_{\mathbf{C}^*,\mathbf{c}_0^*} > 0\right) \to 1.$$

(ii) For $(\lambda, r) \in \Omega_-$, we denote $\mathcal{C}_- = \{(\mathbf{C}, \mathbf{c}_0) : \mathcal{A}_{\mathbf{C}} \not\supseteq \mathcal{A}_{\mathbf{C}^*}, \text{or } \mathcal{A}_{\mathbf{C}} \supseteq \mathcal{A}_{\mathbf{C}^*} \text{ and } r < r^*\}$, then $(\widehat{\mathbf{C}}_{\lambda,r}, (\widehat{\mathbf{c}}_0)_{\lambda,r}) \in \mathcal{C}_-$. In Equation 2.10 for any $(\mathbf{C}, \mathbf{c}_0) \in \mathcal{C}_-$, since the degree of freedom terms are finite, as $n \to \infty$,

$$\begin{aligned} \mathrm{BIC}_{\mathbf{C},\mathbf{c}_0} - \mathrm{BIC}_{\mathbf{C}^*,\mathbf{c}_0^*} &\xrightarrow{P} E(l_1(\mathbf{C}, \mathbf{c}_0, \mathbf{x}, \mathbf{y})) - E(l_1(\mathbf{C}^*, \mathbf{c}_0^*, \mathbf{X}, \mathbf{Y})) \\ &= \mathrm{vec}(\mathbf{C} - \mathbf{C}^*, \mathbf{c}_0 - \mathbf{c}_0^*)^T \frac{\partial^2 E\left(l_1(\tilde{\mathbf{C}}, \tilde{\mathbf{c}}_0, \mathbf{X}, \mathbf{Y})\right)}{\partial(\mathbf{C}, \mathbf{c}_0)^2} \mathrm{vec}(\mathbf{C} - \mathbf{C}^*, \mathbf{c}_0 - \mathbf{c}_0^*), \end{aligned}$$

where $(\tilde{\mathbf{C}}, \tilde{\mathbf{c}}_0)$ is between $(\mathbf{C}, \mathbf{c}_0)$ and $(\mathbf{C}^*, \mathbf{c}_0^*)$. Next we will show $\inf_{(\mathbf{C},\mathbf{c}_0)\in\mathcal{C}_-} \|\mathbf{C} - \mathbf{C}^*\|_F > 0$ by considering the following two cases.

(a) For those $(\mathbf{C}, \mathbf{c}_0) \in \mathcal{C}_-$ with $\mathcal{A}_{\mathbf{C}} \not\supseteq \mathcal{A}_{\mathbf{C}^*}$, then $\|\mathbf{C}^* - \mathbf{C}\|_F \geq \|\mathbf{C}^{*k} - \mathbf{C}^k\|_2 = \|\mathbf{C}^{*k}\|_2 > 0$, for k such that $\mathbf{x}_{(k)} \in \mathcal{A}_{\mathbf{C}}^c \bigcap \mathcal{A}_{\mathbf{C}^*}$, then $\inf_{\mathcal{A}_{\mathbf{C}} \not\supseteq \mathcal{A}_{\mathbf{C}^*}} \|\mathbf{C} - \mathbf{C}^*\|_F > 0$.

(b) For those $(\mathbf{C}, \mathbf{c}_0) \in \mathcal{C}_-$ with $\mathcal{A}_{\mathbf{C}} \supseteq \mathcal{A}_{\mathbf{C}^*}$, and $\mathrm{rank}(\mathbf{C}) < r^*$, it immediately implies $\|\mathbf{C}^* - \mathbf{C}\|_2 > 0$, and then $\|\mathbf{C} - \mathbf{C}^*\|_F \geq \|\mathbf{C} - \mathbf{C}^*\|_2 > 0$, and $\inf_{\mathcal{A}_{\mathbf{C}} \supseteq \mathcal{A}_{\mathbf{C}^*}, \mathrm{rank}(\mathbf{C}) < r^*} \|\mathbf{C} - \mathbf{C}^*\|_F > 0$.

Combining both cases, we have $\inf_{(\mathbf{C},\mathbf{c}_0)\in\mathcal{C}_-} \|\mathbf{C} - \mathbf{C}^*\|_F > 0$, which, together with the fact that $\frac{\partial^2 E\left(l_1(\tilde{\mathbf{C}}, \tilde{\mathbf{c}}_0, \mathbf{X}, \mathbf{Y})\right)}{\partial(\mathbf{C}, \mathbf{c}_0)^2}$ is positive definite, implies the desired results.

Combining Lemmas 2.3.3 and 2.3.4, it is clear that with probability tending to 1, the BIC criterion value for $(\widehat{\mathbf{C}}, \widehat{\mathbf{c}}_0)$ with $\lambda = \log n$ and $r = r^*$ is asymptotically smaller than that for $(\widehat{\mathbf{C}}, \widehat{\mathbf{c}}_0)$ with parameters in $\Omega_+ \cup \Omega_-$. As a consequence, those (λ, r)'s in $\Omega_+ \cup \Omega_-$ will not be selected by the BIC criterion asymptotically. Equivalently, the selected tuning parameter by the BIC criterion must be contained in $(\Omega_+ \cup \Omega_-)^c$ yielding the true model asymptotically.

Theorem 2.3.5. *(Model selection consistency via BIC) Under Assumptions C1-C3, as* $n \to \infty$,

$$\|\widehat{\mathbf{C}}_{\hat{\lambda},\hat{r}} - \mathbf{C}^*\|_F + \|(\widehat{\mathbf{c}}_0)_{\hat{\lambda},\hat{r}} - \mathbf{c}_0^*\|_2 \xrightarrow{P} 0.$$

Proof of Theorem 2.3.5 :

We just need to show for any $\epsilon > 0$, $P\left(E(l_1(\widehat{\mathbf{C}}_{\hat{\lambda},\hat{r}}, (\widehat{\mathbf{c}}_0)_{\hat{\lambda},\hat{r}}, \mathbf{X}, \mathbf{Y})) - E(l_1(\mathbf{C}^*, \mathbf{c}_0^*, \mathbf{X}, \mathbf{Y})) \leq \epsilon\right) \to 1$. Then by the proof of Lemma 2,

$$P\left(\|\widehat{\mathbf{C}}_{\hat{\lambda},\hat{r}} - \mathbf{C}^*\|_F + \|(\widehat{\mathbf{c}}_0)_{\hat{\lambda},\hat{r}} - \mathbf{c}_0^*\|_2 \leq \phi_{\min}\left(\frac{\partial^2 E\left(l_1(\tilde{\mathbf{C}}, \tilde{\mathbf{c}}_0, \mathbf{X}, \mathbf{Y})\right)}{\partial(\mathbf{C}, \mathbf{c}_0)^2}\right)^{-1} \epsilon^{1/2}\right) \to 1,$$

where $(\tilde{\mathbf{C}}, \tilde{\mathbf{c}}_0)$ is between $(\mathbf{C}, \mathbf{c}_0)$ and $(\mathbf{C}^*, \mathbf{c}_0^*)$. Since ϵ is arbitrary, and $\frac{\partial^2 E\left(l_1(\tilde{\mathbf{C}}, \tilde{\mathbf{c}}_0, \mathbf{X}, \mathbf{Y})\right)}{\partial(\mathbf{C}, \mathbf{c}_0)^2}$ is positive definite with finite eigenvalues, then it completes the proof.

From Lemma 2.3.3, for any $\epsilon > 0$, as $n \to \infty$, $P\left(\mathrm{BIC}_{\hat{\lambda},\hat{r}} \leq \mathrm{BIC}_{\lambda_n, r^*} \leq \mathrm{BIC}_{\mathbf{C}^*, \mathbf{c}_0^*} + \epsilon/3\right) \to 1$. Furthermore,

$$\begin{aligned}
&E\left(l_1(\widehat{\mathbf{C}}_{\hat{\lambda},\hat{r}}, (\widehat{\mathbf{c}}_0)_{\hat{\lambda},\hat{r}}, \mathbf{X}, \mathbf{Y})\right) - E\left(l_1(\mathbf{C}^*, \mathbf{c}_0^*, \mathbf{X}, \mathbf{Y})\right) \\
&= E\left(l_1(\widehat{\mathbf{C}}_{\hat{\lambda},\hat{r}}, (\widehat{\mathbf{c}}_0)_{\hat{\lambda},\hat{r}}, \mathbf{X}, \mathbf{Y})\right) - \mathrm{BIC}_{\hat{\lambda},\hat{r}} + \mathrm{BIC}_{\hat{\lambda},\hat{r}} - \mathrm{BIC}_{\mathbf{C}^*, \mathbf{c}_0^*} + \mathrm{BIC}_{\mathbf{C}^*, \mathbf{c}_0^*} - E\left(l_1(\mathbf{C}^*, \mathbf{c}_0^*, \mathbf{X}, \mathbf{Y})\right).
\end{aligned}$$

When n is large enough, with probability tending to 1, $E\left(l_1(\widehat{\mathbf{C}}_{\hat{\lambda},\hat{r}}, (\widehat{\mathbf{c}}_0)_{\hat{\lambda},\hat{r}}, \mathbf{X}, \mathbf{Y})\right) - \mathrm{BIC}_{\hat{\lambda},\hat{r}} \leq \epsilon/3$, $\mathrm{BIC}_{\mathbf{C}^*, \mathbf{c}_0^*} - E\left(l_1(\mathbf{C}^*, \mathbf{c}_0^*, \mathbf{X}, \mathbf{Y})\right) \leq \epsilon/3$, and it implies the results.

Theorem 2.3.5 assures that the model with the tuning parameter selected by the BIC criterion can well approximate the true model. It is also worth pointing out that $\widehat{\mathbf{C}}_{\hat{\lambda},\hat{r}}$ in Theorem 2.3.5 refers to the local estimate achieving the estimation consistency in Theorem 2.3.1.

2.4 Computing Algorithms

Now we turn to the optimization in Equation 2.8, which can be rewritten as

$$\min_{\mathbf{B},\mathbf{A}} \; l(\mathbf{B}, \mathbf{A}, \mathbf{c}_0) \text{ s.t. } J(\mathbf{B}) \leq s, \; \mathbf{B}^T\mathbf{B} = \mathbf{I}_r, \tag{2.11}$$

where the constraints are induced by the adaptive group lasso regularization as well as the orthogonality of $\mathbf{B}$. We then propose to solve Equation 2.11 for $\mathbf{B}$ and $(\mathbf{A}, \mathbf{c}_0)$ separately pretending the other party is fixed. Specifically, when $\mathbf{B}$ is fixed, Equation 2.11 becomes unconstrained and can be efficiently solved by the standard gradient descent algorithm; when $(\mathbf{A}, \mathbf{c}_0)$ is fixed, we develop a constrained manifold optimization algorithm to solve Equation 2.11 for $\mathbf{B}$.

To solve for $\mathbf{B}$, we modify the curvilinear search algorithm to accommodate the sparsity constraint $J(\mathbf{B}) \leq s$. The details are given as follows.

Algorithm 2.2 (Alternating algorithm):

Step 1. Initialize $\mathbf{B}_{(0)}$ such that $\mathbf{B}_{(0)}^T\mathbf{B}_{(0)} = \mathbf{I}_r$ and $J(\mathbf{B}_{(0)}) \leq s$. Set $\mathcal{F}(\cdot) = l(\cdot)$, and two constants $0 < \rho_1 \leq \rho_2 \leq 1$.

Step 2. Given $\mathbf{B}_{(t)}$, solve Equation 2.11 for $\left(\mathbf{A}_{(t+1)}, \mathbf{c}_{0(t+1)}\right)$ via a coordinate descent algorithm.

Step 3. Given $\left(\mathbf{A}_{(t+1)}, \mathbf{c}_{0(t+1)}\right)$, conduct a modified *Step 2* in Algorithm 2.1 with an additional requirement $J(\mathcal{Y}(\hat{\tau})) \leq s$, and update $\mathbf{B}_{(t+1)} = \mathcal{Y}(\hat{\tau})$.

Step 4. Set small $\epsilon > 0$, repeat *Step 2* and *Step 3* until $\|\text{vec}\left(\mathbf{B}_{(t+1)}, \mathbf{A}_{(t+1)}, \mathbf{c}_{0(t+1)}\right) - \text{vec}\left(\mathbf{B}_{(t)}, \mathbf{A}_{(t)}, \mathbf{c}_{0(t)}\right)\|_2 < \epsilon$.

Similar as Wen and Yin (2013), Algorithm 2.2 also decreases the objective function value in Equation 2.11 and will converge eventually. The computational cost for adding the additional sparsity constraint does not increase much as it may only slightly increase the number of steps in the line search of *Step 3*. Furthermore, *Step 2* can be easily accelerated by parallelizing the coordinate descent algorithm (Richtrik and Tak, 2012).

2.5 Numerical Simulations

This section examines the performance of the proposed reduced-rank multi-label classification method by conducting a variety of numerical experiments, and comparing its performance against several popular alternatives in literature.

For the proposed method, we set the link function as the logistic function in Equation 2.8 for illustration. Four existing multi-label classification methods are included for comparison: separate binary classification (SB) as the Binary Relevance with logistic classify function, the Chain Classifier (CC; Read et al., 2009), the "curds and whey" approach (CW; Breiman and Friedman, 1997), and the partial least square discriminant analysis (PLSDA; Barker and Rayens, 2003). For fair comparison, all methods are equipped with the lasso or group lasso penalty to attain the sparsity. Specifically, SB predicts each class labels separately, and the coefficients on j-th label is estimated from a separate binary classification formulation,

$$\widehat{\mathbf{C}}_j = \boldsymbol{argmin}_{\mathbf{C}_j} \sum_{i=1}^{n} -y_{ij}(\mathbf{x}_i^T\mathbf{C}_j + \mathbf{c}_{0j}) + \log(1 + \exp(\mathbf{x}_i^T\mathbf{C}_j + \mathbf{c}_{0j})) + \lambda\|\mathbf{C}_j\|_2. \tag{2.12}$$

where the regularization term $\|\mathbf{C}_j\|_2$ is used to achieve sparsity in SB. CC computes the each label cofficients by sequentially incorporating previous labels into the covariates, where similar regularization term as in Equation 2.12 is used as well. CW is a two-step method. The first step is the same as SB and produces the predicted function values, and the second step re-conducts the classification by regressing the labels on the predicted function values. PLSDA applies the partial least square discriminant analysis on each label separately. All the tuning parameters $(\hat{r}, \hat{\lambda})$ are selected by the BIC criterion via grid search, where the grid points are set as $r \in \{1, \ldots, \min(p, q)\}$ and $\lambda \in \{10^{3(t-1)/49}; t = 1, \ldots, 50\}$.

To compare the performance in multi-label classification, a number of evaluation metrics are proposed in literature (Tsoumakas and Katakis, 2007). In this paper, we adopt the Hamming loss (HL) computing the proportion of the mis-predicted labels,

$$\mathrm{HL}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{mq} \sum_{i=1}^{m} \sum_{j=1}^{q} I\left(\hat{y}_{ij} \neq y_{ij}\right),$$

where m is the size of testing set. The averaged Frobenius norm error and the variable selection error (Ravikumar et al., 2011),

$$\begin{aligned} \mathrm{F}(\widehat{\mathbf{C}}, \mathbf{C}^*) &= \|\widehat{\mathbf{C}} - \mathbf{C}^*\|_F + \|\widehat{\mathbf{c}}_0 - \mathbf{c}_0^*\|_2, \\ \mathrm{VS}(\widehat{\mathbf{C}}, \mathbf{C}^*) &= \frac{\dim\left(\mathcal{A}^c_{\widehat{\mathbf{C}}} \cap \mathcal{A}_{\mathbf{C}^*}\right)}{2p_0} + \frac{\dim\left(\mathcal{A}_{\widehat{\mathbf{C}}} \cap \mathcal{A}^c_{\mathbf{C}^*}\right)}{2(p - p_0)}, \end{aligned}$$

are also reported to evaluate the estimation and variable selection performance.

2.5.1 Simulation Study

The following simulation examples are examined. First, $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{p\times p})$, where $\mathbf{\Sigma}_{kl} = \rho + (1-\rho)I(k=l)$ with $\rho = 0.2$. Each row of $\mathbf{A}^*$ is randomly generated from $\mathcal{N}(\mathbf{0}, (\mathbf{\Sigma}_b)_{q\times q})$, where $(\mathbf{\Sigma}_b)_{kl} = \rho^{|k-l|}$ if $k \neq l$, and $(\mathbf{\Sigma}_b)_{kl} = k$ if $k = l$. The first p_0 rows of $\mathbf{B}^*$ are randomly generated from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{r^*})$, and the rest $p - p_0$ rows are set to be 0. And the first p_0 rows of $\mathbf{B}^*$ are normalized so that $\mathbf{B}^{*T}\mathbf{B}^* = \mathbf{I}_{r^*}$. Each $\mathbf{c}^*_{0j}$ is randomly generated from the uniform distribution in $[-1, 1]$. Next, $y_{ij} \sim \text{Bern}\left((1+\exp(-\mathbf{x}_i^T\mathbf{C}^*_j + \mathbf{c}^*_{0j}))^{-1}\right)$ for $j = 1, \ldots, q$. Five scenarios with different p, p_0, q, r^*, n and m are examined:

Scenario 1: $(p, p_0, q, r^*, n, m) = (10, 5, 5, 3, 500, 3000)$;

Scenario 2: $(p, p_0, q, r^*, n, m) = (100, 50, 10, 3, 500, 3000)$;

Scenario 3: $(p, p_0, q, r^*, n, m) = (100, 50, 10, 3, 1500, 3000)$;

Scenario 4: $(p, p_0, q, r^*, n, m) = (100, 30, 100, 30, 1500, 3000)$;

Scenario 5: $(p, p_0, q, r^*, n, m) = (100, 50, 50, 50, 2000, 3000)$;

The first three scenarios are designed with low-rank structure, and the last two scenarios are of full-rank structure. Each scenario is replicated 50 times. The averaged evaluation metrics over the 50 replications are summarized in Tables I-III.

Tables I-III about here

Clearly, Tables I-III suggest that the performance of the proposed reduced-rank multi-label classification is competitive in all the scenarios. More attractively, the smaller misclassification errors are often achieved via simpler models in most scenarios. PLSDA delivers superior classi-

fication accuracy in the low-dimensional scenario, whereas the performance is sub-optimal when data dimension becomes relatively high. The Frobenius norm error and the variable selection error of PLSDA are not reported in Tables II and III, since PLSDA estimates the partial least square regression coefficients instead of $\mathbf{C}^*$ and selects the subset of the linearly combination of all variables instead of the variables themselves. CW delivers better classification accuracy than SB, as the second step of CW attempts to incorporate the dependency structure among the class labels that was completely ignored by SB. It is also interesting to note that CC is less competitive, even compared with SB, although it indeed makes use of the dependency structure to some extent.

2.5.2 Real Examples

This section applies the proposed reduced-rank multi-label classification method to analyze two real datasets, Birds and Scene. Both datasets are publicly available and can be downloaded at http://mulan.sourceforge.net/datasets.html.

The Birds dataset collects the audio signal information to classify the species of vocalizing birds. Briggs et al. (2012) uses the representation of a 2D time-frequency segmentation of the audio signal to separate bird sounds that overlap in time, and summarizes the representation information into the Birds dataset. This dataset contains 645 instances with 258 numerical attributes and 19 labels, among which 323 instances are used for the training set.

The Scene dataset collects the images of sematic scenes, and catergorizes images into sematic classes, such as “sunset”, “beaches”, and so on. Boutell et al. (2004) proposed a multi-label learning method in scene label learning, and implemented their method on the summarized

Scene dataset. This data set contains 2,407 instances with 294 numerical attributes and 6 labels, among which 1,211 instances are used for the training set.

To better predict the class labels, a threshold strategy is adopted such that the density of "1" in the testing set is as close to the one in the training set as possible (Fan and Lin, 2007). As the truth is unknown in real datasets, the F-norm error and the variable selection error can not be computed. The averaged Hamming loss and number of selected variables are reported in Table IV.

Table IV about here

The superiority of the proposed method is evident in Table IV, where the proposed method delivers smaller Hamming loss than the other four methods and yields a simpler model with smaller number of variables. Furthermore, the Hamming loss and the BIC criterion are graphed as functions of s and r in Figure Figure 1, respectively. It is clear that the performance of the BIC criterion appears to be satisfactory and selects the model with small Hamming loss.

Figure 1 about here

CHAPTER 3

SPARSE POSITIVE DEFINITE MATRIX ESTIMATION

Parts of this chapter was previously published as Yuan, T. and Wang, J. (2012) A coordinate descent algorithm for sparse positive definite matrix estimation. *Statistical Analysis and Data Mining.*

This chapter presents the generic framework of coordinate descent (CD) algorithm for solving the optimization problem with respect to sparse positive definite matrix, and it is organized as follows. Section 3.1 introduces the sparse positive definite matrix estimation, discusses the connection between the Gaussian graphical model and the precision and covariance matrices. Section 3.2 presents the generic coordinate descent algorithm, and applies it to precision matrix estimation and covariance matrix estimation respectively. Section 3.3 presents the numerical experiments in both the simulations and real examples. The effective comparison of the performance sufficiently demonstrates the advantage of our proposed model and method. Section 3.4 extends the proposed algorithm to a block CD algorithm in the context of graph clustering.

3.1 Introduction To Sparse Positiive Definite Matrix Estimation

This section reviews the used methods of sparse precision matrix estimation and sparse covariance matrix estimation in literature. The techiniques for estimating a sparse precision matrix are very different from the ones for estimating a sparse covariance matrix, while the sparsity in the precision or covariance matrix can be interpreted by Gaussian graphical model,

and the sparsity is equivalent with the conditional or marginal independence among random variables following multivariate Gaussian distribution.

3.1.1 Gaussian Graphical Model

As an interesting view of the multivariate Gaussian distribution, a Gaussian grahical model (Edward, 2000) for a multivariate Gaussian random vector $\mathbf{x}$ is represented by an undirected graph with vertices and undirected edges. The p-dimensional Gaussian multivariate probability density function follows the well-known covariance form

$$f(\mathbf{x}) = (2\pi)^{-p/2} \det(\Sigma_0)^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T {\Sigma_0}^{-1}(\mathbf{x}-\mu)\right\},$$

and another presentation is the information form

$$f(\mathbf{x}) = (2\pi)^{-p/2} \det(\Omega_0)^{1/2} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \Omega_0(\mathbf{x}-\mu)\right\}.$$

Under the interpretation of Gaussian graphical model, the off-diagonal entries of Σ_0 and Ω_0 represents dependence structure between variables. We present the following two results without citation.

Lemma 3.1.1. $(\Omega_0)_{jk} = 0$ *is the necessary and sufficient condition of that* $\mathbf{x}_{(j)}$ *and* $\mathbf{x}_{(k)}$ *are conditionally independent given other variables* $\mathbf{x}_{-j,-k}$.

Lemma 3.1.2. $(\Sigma_0)_{jk} = 0$ *is the necessary and sufficient condition of that* $\mathbf{x}_{(j)}$ *and* $\mathbf{x}_{(k)}$ *are marginally independent.*

These two lemmas guarantee that estimating parameters and identifying zeros in the precision or covariance matrix are identical with the parameter estimation and variable selection in Gaussian graphical model. The absence of edges between graph vertices corresponds to the nullity of off-diagonal entries in the precision or covariance matrix.

3.1.2 Positive Definite Matrix Estimation Setup

Assume that a training sample $\mathbf{X} = (X_1, ..., X_n)^T$ is available, where $X_i = (X_{i1}, ..., X_{ip})^T \in \mathcal{R}^p$ follows a multivariate Gaussian distribution with zero-mean and covariance matrix Σ_0. The inverse covariance matrix is denoted as $\Omega_0 = \Sigma_0^{-1}$. To estimate Ω_0, one natural choice is S^{-1}, where

$$S = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^T$$

is the sample covariance matrix, and $\overline{X}$ is the sample mean. However, S is not always invertible. For example, in scenarios with large dimension and small sample size, S is only nonnegative definite and thus non-invertible. Even when S is invertible, S^{-1} is not sparse and cannot be used to infer the conditional dependence structure in the corresponding Gaussian graphical model.

3.1.3 Literature Review

3.1.3.1 Precision Matrix Estimation

By Section 3.1.1, the estimation of a large precision matrix captures the conditional independence among random variables. In order to induce the sparsity in precision matrix, the

formulation is to minimize the regularized negative log-likelihood function (Friedman et al., 2008),

$$\widehat{\Omega}_\lambda = \boldsymbol{argmin}_\Omega \; l_p(\Omega) + \lambda J(\Omega), \tag{3.1}$$

with negative log-likelihood of multivariate normal distribution

$$l_p(\Omega) = \mathrm{tr}(\Omega S) - \log\det(\Omega)$$

after dropping the constant terms, where $\mathrm{tr}(\cdot)$ and $\det(\cdot)$ are the trace and determinant of a matrix, $J(\Omega)$ is the regularization term encouraging sparsity in Ω, such as the lasso penalty (Tibshirani, 1994), the SCAD penalty (Fan and Li, 2001), the adaptive lasso penalty (Zou, 2006), and the L_q penalty (Liu et al., 2007).

The practical interpretation of the regularization method is self-evident. On the one hand, under the scenario that $\lambda = 0$ and S is invertible, $\widehat{\Omega}_\lambda = S^{-1}$ presents the maximum likelihood estimator of Ω, while this estimator tends to be with many noisy covariates, and fails to capture the conditional independence structure among random variables. On the other hand under the scenario $\lambda = \infty$, $\widehat{\Omega}_\lambda$ presents the null matrix containing no information. Under appropriate selection of tuning parameter λ, $\widehat{\Omega}_\lambda$ is expected to be a "balanced" estimator with an appropriate level of sparsity and credential.

In literature, a number of optimization algorithms are proposed for minimizing Equation 3.1 with the lasso penalty $J(\Omega) = \|\Omega\|_1 = \sum_{j,k=1}^{p} |\Omega_{jk}|$ or $J(\Omega) = \|\Omega^-\|_1 = \sum_{j\neq k} |\Omega_{jk}|$, such as the max-det optimization algorithm (Yuan and Lin, 2007), the graphical lasso (Friedman et al.,

2008) and the Cholesky decomposition (Rothman et al., 2008). Note that the max-det optimization algorithm and the Cholesky decomposition are computationally demanding, graphical lasso is computationally efficient but only assures the positive definiteness for Equation 3.1 with lasso penalty. In addition, the performance of the regularized formulation in Equation 3.1 largely relies on the tuning parameter λ, so the aforementioned optimization algorithms need to be run multiple times for various values of λ and the computation cost can be significantly increased.

Another approach to estimate the precision matrix is through the column-by-column estimation method (Yuan, 2010), which exploits the relationship between conditional multivariate normal distribution and linear regression. Inspired by the conditional distribution of $\mathbf{x}_{(j)}$ given $\mathbf{x}_{-j}$ satisfying

$$\mathbf{x}_{(j)}|\mathbf{x}_{-j} \sim N\left(\alpha_j'\mathbf{x}_{-j}, \sigma_j^2\right),$$

where $\alpha_j = \left(\Sigma_{-j,-j}\right)^{-1}\Sigma_{-j,j} = -\left(\Omega_{jj}\right)^{-1}\Omega_{-j,j}$ and $\sigma_j^2 = \Sigma_{jj} - \Sigma_{-j,j}\left(\Sigma_{-j,-j}\right)^{-1}\Sigma_{-j,j} = \left(\Omega_{jj}\right)^{-1}$, the resultant linear regression model can be interpreted as

$$\mathbf{X}_{(j)} = \alpha_j'\mathbf{X}_{-j} + \epsilon_j,$$

where ϵ_j is random noise and independent with $\mathbf{X}_{-j}$, therefore it is desirable to recover the precision matrix Ω by extracting the coefficients from regressing $\mathbf{X}_{(j)}$ on $\mathbf{X}_{-j}$ in sequence. In

order to obtain the sparsity, Meinshausen and Bhlmann (2006) proposes the Lasso regression problem:

$$\widehat{\alpha_j} = \boldsymbol{argmin}_{\alpha_j \in \mathcal{R}^{p-1}} \|\mathbf{X}_{(j)} - \alpha_j' \mathbf{X}_{-j}\|_2^2 + \lambda_j \|\alpha_j\|_1$$

to estimate each α_j. To estimate the Ω, $\widehat{\sigma}_j^2 = n^{-1}\|\mathbf{X}_{(j)} - \widehat{\alpha}_j' \mathbf{X}_{-j}\|_2^2$ are used to estimate σ_j^2's.

Additionally, Yuan (2010) proposes the graphical Dantzig selector to estimate α_j,

$$\widehat{\alpha}_j = \boldsymbol{argmin}_{\alpha_j \in \mathcal{R}^{p-1}} \|\alpha_j\|_2 \quad \text{subject to} \quad \|S_{-j,j} - S_{-j,-j}\alpha_j\|_\infty \leq \gamma_j$$

where γ_j is a tuning parameter. Given $\widehat{\alpha}_j$, it is shown σ_j^2 can be estimated by

$$\widehat{\sigma}_j^2 = \left(1 - 2\widehat{\alpha}_j' S_{-j,j} + \widehat{\alpha}_j' S_{-j,-j}\widehat{\alpha}_j\right)^{-1}.$$

Yuan (2010) shows that the the l_1 norm distance between optimal estimator $\widehat{\Omega}$ and Ω_0 obtains the minimax optimality over certain model space.

Similarly Cai et al. (2011) proposes the "constrained l_1-Minimization for Inverse Matrix Estimation" (CLIME) estimator. The j-th column of this estimator is by solving

$$\widehat{\Omega}_{(j)} = \boldsymbol{argmin}_{\Omega_{(j)}} \|\Omega_{(j)}\|_1$$
$$\text{subject to} \quad \|S\Omega_{(j)} - \mathbf{e}_j\|_\infty \leq \delta_j,$$

where $j = 1, ..., p$, $\mathbf{e}_j$ is the unit vector with j-th entry being 1 and remaining entries being 0, and δ_j is a tuning parameter. Under certain regularity conditions, Cai et al. (2011) shows that the estimator $\widehat{\Omega}$ is asymptotically positive definite, and its rate convergence is also derived. In a close related work with Cai et al. (2011), Liu and Luo (2014) proposes the SCIO estimator, which can be efficiently computed by the pathwise coordinate descent algorithm (Friedman et al., 2007). Similar to regularization methods, the column-by-column estimation methods also conduct the regularization or its equivalence requiring multiple times of running for various tuning parameters, and the associated computational cost is immense.

3.1.3.2 Covariance Matrix Estimation

The estimation of the sparse covariance matrix Σ_0 mainly follows the route of component-wise thresholding, such as hard thresholding (Bickel et al., 2008; El Karoui, 2008), soft thresholding (Rothman et al., 2009), adaptive thresholding (Cai et al., 2011). Rothman et al. (2011) generalizes the thresholding methods and propose the generalized thresholding method of covariance matrix. Furthermore, Rothman et al. (2011) shows that the thresholding methods are equivalent with a regularized square loss with various penalty functions. Generally, the thresholding method may carry low computation burden while it delivers an estimated covariance matrix which cannot guarantee the positive definiteness. In order to achieve the positive definiteness, it requires careful selection of thresholding constants (Fan et al., 2013).

For any $\lambda > 0$, a generalized thresholding operator $t_\lambda : \mathcal{R} \to \mathcal{R}$ is defined if following conditions are satisfied for all $u \in \mathcal{R}$:

(i) $|t_\lambda(u)| \leq |u|$;

(ii) $|t_\lambda(u)| = 0$ for $|u| \leq \lambda$;

(iii) $|t_\lambda(u) - u| \leq \lambda$.

The simplest example of generalized thresholding is hard thresholding rule $|s_\lambda^H(u)| = u1\,(|u| > \lambda)$. Soft-thresholding gives the rule $|s_\lambda^S(u)| = sgn(u)(|u| - \lambda)_+$, and it is shown to correspond to the maximum amount of shrinkage allowed by condition (iii), while hard thresholding allows no shrinkage. The adaptive thresholding presents the rule $|s_\lambda^A(u)| = sgn(u)(|u| - \lambda^{\eta+1}|u|^{-\eta})_+$ where η is some pre-specified positive constant, and it can unifies the previously proposed thresholding methods by specifying the parameter values.

Furthermore, the researchers alternatively propose a simpler method to apply thresholding on the correlation matrix, and recover the estimated covariance matrix using the diagonal components of S. This method is argued to be more appropriate than the simple thresholding since it is thresholded on standardized scale. However, thresholding on neither sample covariance matrix nor correlation matrix can naturally guarantee the positive definiteness of the estimators in practice, even though the asymptotic estimator may exhibit the positive definiteness under certain conditions.

Another approach is the so called "nearest correlation matrix projection" (Qi and Sun, 2006), the key idea of which is to find the nearest positive definite correlation matrix close to the thresholded correlation matrix. In order to assure the sparsity of the covariance matrix, Liu et al. (2014) further generalizes the problem as

$$\widehat{\Theta}_\lambda = \boldsymbol{argmin}_{\phi_{min}(\Theta)\geq\tau}\ \frac{1}{2}\|\Theta - S\|_F^2 + \lambda\|\Theta^-\|_1, \tag{3.2}$$

and develops an efficient algorithm to solve it.

Rothman (2012) uses an additional penalty on the determinant of the estimated matrix to assure positive definiteness and develops an optimization algorithm similar as the graphical lasso, it proposes an estimation method based on the regularized cost function,

$$\widehat{\Theta}_{\lambda_1,\lambda_2} = \boldsymbol{argmin}_{\Theta} \frac{1}{2}\|\Theta - S\|_F^2 - \lambda_1 \log(\det(\Theta)) + \lambda_2 \|\Theta^-\|_1. \tag{3.3}$$

where $\|M\|_F = (\sum_{i,j} M_{ij}^2)^{1/2}$ is the Frobenius norm of M, λ_1 and λ_2 are two tuning parameters, and the penalty $\log(\det(\Theta))$ is used to guard Θ from nonnegative definiteness. A similar computational algorithm as the graphical lasso is also developed for solving Equation 3.3. Although it is argued in Rothman (2012) that λ_1 can be set as a small constant, the value of λ_2 still requires careful tuning at the cost of increasing computation burden.

3.1.4 Summary And Discussion

Most of the literature use the regularization methods for sparse positive precision matrix estimation. The regularization method equips the negative log-likelihood function with a sparsity-encouraged penalty term, and estimates the resultant precision matrix based on the appropriate selection of tuning parameter, therefore multiple times of running the optimizations are necessary to deliver an appropriate estimator; regarding sparse covariance matrix estimation, majority of the literature suggests to use the component-wise thresholding strategy and this simple method may yield an estimator without the positive definiteness, and as well the

regularization method in sparse covariance matrix estimation requires multiple trials to select an appropriate tuning parameter, therefore the computational burden increases.

In our proposed method, we are able to unify both sparse precision matrix estimation and sparse covariance matrix estimation under the simple coordinate descent framework. The techniques only differ at the selection of objective functions, and the regularization terms are not needed. Furthermore, our framework is generic in the sense that it can estimate any sparse positive definite matrix provided the objective function. Instead of multiple trials on the selection of tuning parameters, the CD algorithm generates a chain of positive definite matrices. The updating rule is simple enough to update a diagonal or both symmetric off-diagonal entries at each iteration, and the step size is easily determined by a closed formula. Furthermore, our algorithm in covariance matrix estimation guarantees the estimated matrix to stay positive definite, while the component-wise thresholding methods fail in this aspect.

3.2 The Generic CD Algorithm For Positive Definite Estimation

The CD algorithm has been widely used in convex optimization literature due to its simple implementation and superior computational efficiency. After it was introduced to the statistics community by Friedman et al.(2007), the CD algorithm has soon gained its popularity in statisticians (Wu and Lange, 2008; Mazumder et al., 2011). This section extends the CD algorithm to the sparse positive definite matrix optimization,

$$\widehat{M} = \boldsymbol{argmin}_{M \in \mathcal{M}} \; s(M),$$

where $s(M)$ is convex in M and $\mathcal{M}$ is the collection of all positive definite matrices.

The key idea of the proposed CD algorithm in matrix optimization is to update one diagonal entry or two symmetric off-diagonal entries of the current estimated matrix at each iteration. The step size along the selected entries needs to be appropriately determined so that the updated matrix at each iteration remains positive definite. The details of the generic CD algorithm is given as follows.

Algorithm 3.1 (A generic CD algorithm):

Step 1. Initialize $M_1 = I_p$, the $p \times p$ identity matrix.

Step 2. At the t-th step, compute the matrix gradient $D_t = \nabla s(M_t)$ and set

$$(a, b) = \boldsymbol{argmax}_{j \leq k} \, |(D_t)_{jk}|.$$

Step 3. Denote W_t as a $p \times p$ matrix with all entries being 0 except that $(W_t)_{ab} = (W_t)_{ba} = (D_t)_{ba}$, and update $M_{t+1} = M_t - v_t W_t$ with v_t being the step size.

Step 4. Repeat *Steps 2* and *3* for T times, where T is a pre-specified number of iterations.

Algorithm 3.1 is generic as it can be adapted to optimize any convex objective function with respect to the positive definite matrix. To assure the positive definiteness of the generated M_t's, the step size v_t in *Step 3* needs to be appropriately determined based on the following Theorem 3.2.1.

Theorem 3.2.1. *Given that M_t is positive definite, a necessary and sufficient condition for $M_{t+1} = M_t - v_t W_t$ being positive definite is*

$$\det(M_{t+1}) > 0.$$

Furthermore, $\det(M_{t+1}) > 0$ *is the necessary and sufficient condition of that if* $v_t \in (v_t^d, v_t^u)$ *where*

$$v_t^u = \begin{cases} \frac{-(D_t)_{ab}(M_t^{-1})_{ab} + |(D_t)_{ab}|\sqrt{(M_t^{-1})_{aa}(M_t^{-1})_{bb}}}{(D_t)_{ab}^2 \Delta_t}, & \text{if } a \neq b; \\ \frac{1}{(D_t)_{aa}(M_t^{-1})_{aa}}, & \text{if } a = b,\ (D_t)_{aa} > 0; \\ +\infty, & \text{if } a = b,\ (D_t)_{aa} \leq 0, \end{cases} \tag{3.4}$$

and

$$v_t^d = \begin{cases} \frac{-(D_t)_{ab}(M_t^{-1})_{ab} - |(D_t)_{ab}|\sqrt{(M_t^{-1})_{aa}(M_t^{-1})_{bb}}}{(D_t)_{ab}^2 \Delta_t}, & \text{if } a \neq b; \\ -\infty, & \text{if } a = b,\ (D_t)_{aa} > 0; \\ \frac{1}{(D_t)_{aa}(M_t^{-1})_{aa}}, & \text{if } a = b,\ (D_t)_{aa} \leq 0, \end{cases} \tag{3.5}$$

and $\Delta_t = (M_t^{-1})_{aa}(M_t^{-1})_{bb} - (M_t^{-1})_{ab}^2$.

Proof of Theorem 3.2.1: The necessity is trivial, and we just prove the sufficiency. The proof is done for $a = b$ and $a \neq b$ respectively.

(1) When $a = b$, $M_{t+1} = M_t - v_t(D_t)_{aa}\mathbf{1}_a\mathbf{1}_a^T$, where $\mathbf{1}_a = (0, \ldots, 1, \ldots, 0)^T$ is a vector of 0's except the a-th entry being 1. Then VV^T is a rank-1 diagonal matrix with diagonal V. By the matrix determinant lemma (Harville, 1997),

$$\det(M_{t+1}) = \det(M_t)\Big(1 - v_t(D_t)_{aa}(M_t^{-1})_{aa}\Big).$$

Therefore, $\det(M_{t+1}) > 0$ is equivalent to $1 - v_t(D_t)_{aa}(M_t^{-1})_{aa} > 0$, which assures that the equation

$$v_t(D_t)_{aa} + 2\xi + \xi^2(M_t^{-1})_{aa} = 0$$

has a real root ξ^*. Therefore, $-v_t(D_t)_{aa} = 2\xi^* + (\xi^*)^2(M_t^{-1})_{aa}$, and

$$M_{t+1} = M_t + (2\xi^* + (\xi^*)^2(M_t^{-1})_{aa})\mathbf{1}_a\mathbf{1}_a^T.$$

Re-organizing the right hand side yields that

$$M_{t+1} = (I + \xi^*\mathbf{1}_a\mathbf{1}_a^T M_t^{-1})M_t(I + \xi^* M_t^{-1}\mathbf{1}_a\mathbf{1}_a^T),$$

and hence that M_{t+1} is positive definite as long as M_t is positive definite.

(2) When $a \neq b$, denote two $n \times 2$ matrices,

$$U = \begin{pmatrix} \mathbf{1}_a \\ \\ \mathbf{1}_b \end{pmatrix}^T \text{ and } V = \begin{pmatrix} \mathbf{1}_b \\ \\ \mathbf{1}_a \end{pmatrix}^T,$$

then $M_{t+1} = M_t - v_t(D_t)_{ab}UV^T$, where UV^T is a $n \times n$ matrix with all entries being 0 except the (a, b) and (b, a)-th entries being 1. Similar as in the case of $a = b$, the positive definiteness of M_{t+1} can be concluded if there exists a matrix A such that

$$M_{t+1} = (I + A)M_t(I + A). \tag{3.6}$$

In fact, M_{t+1} can be decomposed as

$$M_{t+1} = M_t - v_t(D_t)_{ab}\Big(\mathbf{1}_b\mathbf{1}_a^T + \mathbf{1}_a\mathbf{1}_b^T\Big) = M_t + \eta\eta^T - \zeta\eta^T - \eta\zeta^T,$$

where $\eta = -(2v_t)^{1/2}(D_t)_{ab}\mathbf{1}_b$, and $\zeta = -(v_t/2)^{1/2}\Big((D_t)_{ab}\mathbf{1}_b + \mathbf{1}_a\Big)$. If denote $A = -\eta(\alpha\eta + \zeta)^T M_t^{-1}$ with some unknown α, then Equation 3.6 is true if there exists α such that

$$M_t - \eta(\alpha\eta + \zeta)^T - \eta^T(\alpha\eta + \zeta) + \eta(\alpha\eta + \zeta)^T M_t^{-1}(\alpha\eta + \zeta)\eta^T = M_t - \zeta\eta^T - \eta\zeta^T + \eta\eta^T,$$

which can be simplified as

$$(\alpha\eta + \zeta)^T M_t^{-1}(\alpha\eta + \zeta) - 2\alpha = 1.$$

The above equation has real roots if and only if

$$(\eta^T M_t^{-1}\zeta - 1)^2 - (\eta^T M_t^{-1}\eta)(\zeta^T M_t^{-1}\zeta - 1) = \Big(1 - v_t(D_t)_{ab}M_{ab}^{-1}\Big)^2 - v_t^2(D_t)_{ab}^2 M_{aa}^{-1}M_{bb}^{-1} > 0.$$

This is true since by the matrix determinant lemma again,

$$\det(M_{t+1}) = \det(M_t)\det\Big(I - v_t(D_t)_{ab}V^T M_t^{-1}U\Big),$$

therefore $\det(M_{t+1}) > 0$ is equivalent to

$$\det\Big(I - v_t(D_t)_{ab}V^T M_t^{-1}U\Big) = \Big(1 - v_t(D_t)_{ab}(M_t)_{ab}^{-1}\Big)^2 - v_t^2 D_t^2 (M_t)_{aa}^{-1}(M_t)_{bb}^{-1} > 0.$$

This completes the proof.

In general, positive determinant is only a necessary condition for positive definiteness, so Theorem 3.2.1 is interesting as it shows, under the framework of coordinate descent algorithm, that positive determinant is as well a sufficient condition for positive definiteness. Theorem 3.2.1 also provides an explicit working interval for the step size v_t, where the upper bound v_t^u relies only on M_t^{-1} and $(D_t)_{ab}$. As computational remarks, the matrix inverse $M_t^{-1} = (M_{t-1} - v_{t-1}W_{t-1})^{-1}$ can be efficiently computed according to Miller (1981). In specific, if W is a rank-1 matrix,

$$(M - vW)^{-1} = M^{-1} + \frac{v}{1 - v\mathrm{tr}(WM)}M^{-1}WM^{-1},$$

and if W is a rank-2 matrix, one can decompose W as a sum of two rank-1 matrices, and $(M - vW)^{-1}$ can be obtained by applying the above formulation twice.

Furthermore, the number of iteration T in *Algorithm 3.1* can be treated as a tuning parameter that controls the balance between the model fitting and the model sparsity. In this paper, we adopt two classical model selection criteria to determine T: Akaike information criterion (AIC; Akaike, 1974) and Bayesian information criterion (BIC; Schwarz, 1978),

$$\begin{aligned} \mathrm{AIC}(M) &= s(M) + 2 \cdot \mathrm{df}(M), \\ \mathrm{BIC}(M) &= s(M) + \log(n) \cdot \mathrm{df}(M), \end{aligned}$$

where $\mathrm{df}(M) = n^{-1}\#\{(j,k) : j < k, M_{jk} \neq 0\}$. Generally speaking, AIC puts less penalty upon the model complexity than BIC, and thus minimizing AIC often leads to a relatively denser estimated matrix than BIC (Yang, 2005).

It is worth pointing out that the proposed generic CD algorithm can be regarded as an analogy of the forward stagewise regression in the context of positive definite matrix estimation. When the step size $v_t \to 0$, a LARS-type of matrix estimation algorithm can also be derived as in Efron et al. (2004). However, as argued in Efron et al. (2004) and Hastie et al. (2007), it remains unclear whether there exists an equivalence between the proposed CD algorithm and a global optimization formulation as in Equation 3.1.

3.2.1 Precision Matrix Estimation

This section applies the generic CD algorithm to the precision matrix estimation. Particularly, we set the objective function as the negative log-likelihood function $l_p(\Omega)$ in Equation 3.1. Note that $l_p(\Omega)$ is convex in Ω, since its first and second derivatives with respect to Ω are

$$\frac{\partial l_p(\Omega)}{\partial \Omega} = S - \Omega^{-1}, \quad \frac{\partial^2 l_p(\Omega)}{\partial \Omega^2} = \Omega^{-1} \otimes \Omega^{-1},$$

where $\otimes$ is the Kronecker product. The precision matrix estimation algorithm proceeds as follows.

Algorithm 3.2 (precision matrix estimation):

Step 1. Initialize $\Omega_1 = (diag\{S\})^{-1}$, where $diag\{S\}$ is a diagonal matrix composed by the diagonal entries of S.

Step 2. At the t-th step, compute $D_t = S - \Omega_t^{-1}$, and select the coordinate (a, b) by

$$(a, b) = \boldsymbol{argmax}_{j \leq k} \, |S_{jk} - (\Omega_t^{-1})_{jk}|.$$

Steps 3 & 4. The same as *Steps 3 and 4* in *Algorithm 3.1.*

In *Step 3*, v_t is given by $v_t = \alpha v_t^m$, where $0 < \alpha \leq 1$ is a pre-specified constant, and

$$
\begin{aligned}
v_t^m &= \boldsymbol{argmin}_v \; l_p(\Omega_t - vW_t) \\
&= \begin{cases}
\frac{1}{S_{aa}(\Omega_t^{-1})_{aa}}, & \text{if } a = b; \\
\frac{-(\Delta_t + 2(\Omega_t^{-1})_{ab} S_{ab}) + \sqrt{\Delta_t^2 + 4S_{ab}^2 \Delta_t + 4S_{ab}^2((\Omega_t^{-1})_{ab})^2}}{2\Delta_t (D_t)_{ab} S_{ab}}, & \text{if } a \neq b, S_{ab} \neq 0; \\
\frac{1}{\Delta_t}, & \text{if } a \neq b, S_{ab} = 0,
\end{cases}
\end{aligned} \tag{3.7}
$$

where $\Delta_t = (\Omega_t^{-1})_{aa}(\Omega_t^{-1})_{bb} - (\Omega_t^{-1})_{ab}^2$. Here v_t^m minimizes the negative log-likelihood function along the direction W_t and thus specifies the greediest step size, and α scales down the greediest step size for a conservative updating. Clearly, *Algorithm 3.2* is a standard coordinate descent algorithm with a Gauss-Southwell updating rule (Tseng, 1991), and Corollary 1 assures its convergence as well as the positive definiteness of the estimated precision matrix.

Corollary 1. *Algorithm 3.2 always converges, and the generated precision matrices Ω_t are positive definite.*

Proof of Corollary 1: Direct calculation yields that $v_t^m < v_t^u$ in Equation 3.5, and then Theorem 3.2.1 assures that all the matrices Ω_t's generated by *Algorithm 3.2* are positive definite. In addition, *Algorithm 3.2* essentially uses a Gauss-Southwell coordinate updating Rule (Tseng, 1991) and thus always converges according to Theorem 2.1 in Luo and Tseng (1992).

More interestingly, *Algorithm 3.2* yields a solution path for the positive definite Ω_t. This generated path starts from the sparsest diagonal matrix Ω_1, and gradually adds in nonzero off-diagonal entries. As a consequence, the selection of tuning parameter T becomes much

more efficient than the regularized methods, where the precision matrix needs to be estimated multiple times for various tuning parameters. Figure 2 (Yuan, T. and Wang, J. (2012) A coordinate descent algorithm for sparse positive definite matrix estimation. *Statistical Analysis and Data Mining*) displays the solution paths of some randomly selected coordinates of Ω_t's and the smallest eigenvalues of Ω_t's as functions of T.

Figure 2 about here

Whereas the fast numerical convergence speed of each coordinate is evident in Figure 2, the speeds vary from coordinate to coordinate. It is also clear that the smallest eigenvalues of all Ω_t's are always positive, which confirms the positive definiteness of Ω_t.

3.2.2 Covariance Matrix Estimation

To estimate the covariance matrix, we set the objective function as

$$l_c(\Theta) = \frac{1}{2}\|\Theta - S\|_F^2 - \lambda \log(\det(\Theta)), \tag{3.8}$$

where λ is fixed as a small positive constant (Rothman, 2012). It is convex in Θ as its first and second derivatives are

$$\frac{\partial l_c(\Theta)}{\partial \Theta} = \Theta - S - \lambda\Theta^{-1}, \quad \frac{\partial^2 l_c(\Theta)}{\partial \Theta^2} = \Gamma + \lambda\Theta^{-1} \otimes \Theta^{-1}, \tag{3.9}$$

where the semi-definite matrix $\Gamma = \sum_{i,j=1}^{p} \delta_{ij} \otimes \delta_{ij}$, and δ_{ij} is a $p \times p$ matrix with 1 at the (i,j)-th entry and 0 elsewhere. Setting $s(M) = l_c(\Theta)$ in Algorithm 3.1, the covariance estimation algorithm proceeds as follows.

Algorithm 3.3 (covariance matrix estimation):

Step 1. Initialize $\Theta_1 = diag\{S\}$.

Step 2. At the t-th step, compute $D_t = \Theta_t - \lambda\Theta_t^{-1} - S$, and select the coordinate (a,b) by

$$(a,b) = \boldsymbol{argmax}_{j \le k} \; |(\Theta_t)_{jk} - \lambda(\Theta_t^{-1})_{jk} - S_{jk}|.$$

Steps 3 & 4. The same as *Steps 3 and 4* in *Algorithm 3.1.*

In *Step 3*, v_t is given by $v_t = \alpha v_t^m$, where $0 < \alpha \le 1$ is pre-specified, and

$$v_t^m = \boldsymbol{argmin}_{v>0} \; l_c(\Theta_t - vW_t). \tag{3.10}$$

Specifically, let $\Delta_t = (\Theta_t^{-1})_{aa}(\Theta_t^{-1})_{bb} - ((\Theta_t^{-1})_{ab})^2$ and $d = -(D_t)_{ab}$, and then v_t^m in Equation 3.10 can be obtained as the smallest positive root of

$$\Theta_{ab} - S_{ab} + dv = \begin{cases} \frac{\lambda((\Theta_t^{-1})_{ab} - \Delta_t dv)}{1 + 2(\Theta_t^{-1})_{ab} dv - \Delta_t d^2 v^2}, & \text{if } a \neq b; \\ \frac{\lambda(\Theta_t^{-1})_{aa}}{1 + (\Theta_t^{-1})_{aa} dv}, & \text{if } a = b, \end{cases} \tag{3.11}$$

which leads to solving a cubic or quadratic equation in each step. More importantly, Corollary 2 assures the convergence of *Algorithm 3.3* and the positive definiteness of the estimated covariance matrix.

Corollary 2. *Algorithm 3.3 always converges, and the generated covariance matrices* Θ_t *are positive definite.*

Proof of Corollary 2: To prove the positive definiteness, it suffices to show the smallest positive root of Equation 3.11, v_t^m, is upper bounded by v_t^u. In fact, $\det(\Theta_t - v_t^u W_t) = 0$ leads to that $l_c(\Theta_t - v_t^u W_t) = +\infty > l_c(\Theta_t)$. By the continuity of $l_c(\Theta_t - vW_t)$ in v and the fact that $\frac{\partial l_c(\Theta_t - vW_t)}{\partial v}|_{v=0} < 0$, we have $v_t^m \in (0, v_t^u)$.

The convergence is guaranteed if the eigenvalues of the second derivative of $l_c(\Theta_t)$ are bounded away from 0 and $+\infty$ (Bertsekas, 1999). In Equation 3.9, Γ is semi-definite with finite nonnegative eigenvalues, so it suffices to show that the smallest and largest eigenvalue of Θ_t are bounded away from 0 and $+\infty$ respectively. Since Θ_t is positive definite, it is easy to see that its largest coordinate must be in its diagonal, say (m, m). Simple algebra yields that

$$\log(\det(\Theta_t)) \leq p \log(\phi_{max}(\Theta_t)) \leq p \log(p(\Theta_t)_{mm}),$$

where $\phi_{max}(\Theta_t)$ denotes the largest eigenvalue of Θ_t. Note that

$$((\Theta_t)_{mm} - S_{mm})^2 - \lambda p \log(p(\Theta_t)_{mm}) < l_c(\Theta_t) \leq l_c(\Theta_0) < +\infty, \tag{3.12}$$

which implies that $(\Theta_t)_{mm} < +\infty$, and thus $\phi_{max}(\Theta_t) < +\infty$. Therefore, the smallest eigenvalue of Θ_t^{-1}, $\phi_{min}(\Theta_t^{-1}) = (\phi_{max}(\Theta_t))^{-1}$, which is bounded away from 0, and also the largest eigenvalue of Θ_t^{-1}, $\phi_{max}(\Theta_t^{-1}) = (\phi_{min}(\Theta_t))^{-1}$ is bounded away from $+\infty$. This completes the proof.

Again, *Algorithm 3.3* generates a solution path for positive definite Θ_t with non-decreasing complexity, and thus the selection of tuning parameter T can be conducted efficiently.

3.3 Numerical Experiments

This section examines the performance of the proposed CD algorithm in a variety of numerical experiments, and compares it against the popular competitors for estimating the precision matrix and the covariance matrix, respectively.

3.3.1 Numerical Experiment I: Precision Matrix Estimation

3.3.1.1 Simulated Examples

Employed in the comparison of the precision matrix estimation are four simulated multivariate Gaussian models with zero mean and different covariance matrices:

Model P1 (AR(1)): $(\Sigma_0)_{jk} = \rho^{|j-k|}$ with $\rho = 0.5$, and $\Omega_0 = (\Sigma_0)^{-1}$;

Model P2 (AR(3)): $(\Omega_0)_{jk} = \mathbf{I}(|j-k| = 0) + 0.5\mathbf{I}(|j-k| = 1) + 0.2\mathbf{I}(|j-k| = 2) + 0.1\mathbf{I}(|j-k| = 3)$;

Models P3 & **P4** (Randomly generated matrix): $(\Omega_0)_{jk} \sim \text{Bern}(\gamma)$ when $j \neq k$, with $\gamma = 0.1$ for Model P3 and $\gamma = 0.5$ for Model P4, and $(\Omega_0)_{jj}$'s are set so that the smallest eigenvalue of Ω_0 is 0.1.

For each model with $p = 25$, 50 or 100, 100 training observations are generated from $N_p(0, \Sigma_0)$, and three precision matrix estimation methods are compared: the graphical lasso, *Algorithm 3.2* with $\alpha = 0.2$ and $\alpha = 1$. To optimize the performance of each estimation method, the tuning parameter in the graphical lasso method is selected by BIC as suggested in Rothman et al. (2008), and the tuning parameter in *Algorithm 3.2* is selected by AIC and BIC. For simplicity, *glasso* denotes the graphical lasso method, αCD_AIC and αCD_BIC denote *Algorithm 3.2* with $\alpha = 0.2$ and tuned by AIC or BIC, and CD_AIC and CD_BIC denote *Algorithm 3.2* with $\alpha = 1$ and tuned by AIC or BIC.

To evaluate the estimation performance, the Kullback-Leibler (KL) loss and the Frobenius-norm (F-norm) loss are used,

$$\begin{aligned} \mathrm{KL}(\widehat{\Omega}, \Omega_0) &= \mathrm{tr}(\Sigma_0\widehat{\Omega}) - \log|\Sigma_0\widehat{\Omega}| - p, \\ \mathrm{F}(\widehat{\Omega}, \Omega_0) &= \|\Omega_0 - \widehat{\Omega}\|_F. \end{aligned}$$

Also the signed variable selection loss (Ravikumar et al., 2011) is used for assessing the variable selection performance,

Each model is replicated 100 times, and the averaged performance measures are summarized in Tables V-VII (Yuan, T. and Wang, J. (2012) A coordinate descent algorithm for sparse positive definite matrix estimation. *Statistical Analysis and Data Mining*).

Tables V-VII about here

It is evident that the proposed CD algorithm delivers superior estimation and variable selection performance. In specific, αCD_AIC shows the best performance in terms of the averaged KL loss and the F-norm loss, whereas it tends to overestimate the precision matrix by including the redundant non-zero entries. CD_BIC and αCD_BIC yield slightly worse estimation performance, but they produce much sparser estimated precision matrices than CD_AIC and αCD_AIC in Table VII. Furthermore, as a more fair comparison based on the same tuning criterion, CD_BIC still outperforms the graphical lasso in most scenarios.

3.3.1.2 Colon Tumor Classification

We now integrate the proposed precision matrix estimation algorithm with Fisher's linear discriminant analysis (LDA) in a real application of colon tumor classification based on their gene expression profiles (Alon et al., 1999). The dataset consists of 62 colon adenocarcinoma tissue samples, among which 40 are tumor tissues and 22 are non-tumor tissues. All tissue samples were analyzed using an Affymetrix oligonucleotide array, and the raw data were processed, filtered and reduced to a subset of 2,000 gene expression profiles with the largest minimal intensity across the 62 tissue samples (Rothman, 2008). The data is publicly accessible in the "plsgenomics" package of R.

To examine the estimation performance with various dimensions, we select the p most significant gene profiles among the original dataset with $p = 25$, 50 or 200, where the significance is measured by the two-sample t-statistics as in Rothman et al. (2008). For each p-dimensional dataset, the group mean and the homogeneous precision matrix are first estimated, and then a classification model is constructed based on Fisher's LDA with $k = 1$ (non-tumor) and 2

(tumor), where $\hat{\pi}_k$ is the proportion of observations from class k, $\hat{\mu}_k$ is the sample mean for class k, and $\hat{\Omega}$ is the estimated homogeneous precision matrix.

Since the true precision matrix is unknown in the real application, the testing errors are used to compare the precision matrix estimation methods. Specifically, the 62 tissues are randomly split into a training set of size 42 and a testing set of size 20. The classification model is trained based on the 42 training tissues, and the testing error is measured by its corresponding misclassification error on the testing set. The splitting is randomly replicated for 100 times, and the averaged testing errors are summarized in Table VIII (Yuan, T. and Wang, J. (2012) A coordinate descent algorithm for sparse positive definite matrix estimation. *Statistical Analysis and Data Mining*).

Table VIII about here

Among the five precision matrix estimation methods, it is clear that CD_BIC and αCD_BIC perform better than the graphical lasso, but CD_AIC and αCD_AIC appear to be less competitive.

3.3.2 Numerical Experiment II: Covariance Matrix Estimation

3.3.2.1 Simulated Examples

Employed in the comparison of the covariance matrix estimation are two simulated multivariate Gaussian models quoted from Rothman (2012):

Model C1: $(\Sigma_0)_{ij} = 0.4\mathbf{I}(|i-j|=1) + \mathbf{I}(i=j)$.

Model C2: Parition indices $1, ..., p$ into K ordered blocks of equal size, with $K = 3, 5$ or 10 and $p = 30, 100$ or 200, respectively. Set $(\Sigma_0)_{ii} = 1$ for all $1 \leq i \leq p$, and $(\Sigma_0)_{ij} = 0.4$ if i and j are in the same block, or if i and j are in adjacent blocks and $\min(i, j)$ is the maximum index of a block.

Three methods are compared, including Rothman's method (Rothman, 2012), *Algorithm 3.3* with $\alpha = 1.0$ and $\alpha = 0.2$. To evaluate the estimation performance, we report the F-norm loss and the signed variable selection loss in Section 6.1, as well as the spectral norm,

$$\mathrm{S}(\widehat{\Theta}, \Sigma_0) \quad = \quad \|\Sigma_0 - \widehat{\Theta}\|_2,$$

which is the maximal absolute eigenvalue of difference matrix. The averaged performance measures over 500 replications are summarized in Tables IX-XI for both models with $n = 50$ and $p = 30, 100, 200$ (Yuan, T. and Wang, J. (2012) A coordinate descent algorithm for sparse positive definite matrix estimation. *Statistical Analysis and Data Mining*).

Tables IX-XI about here

Clearly, the proposed covariance matrix estimation algorithm yields comparable performance as Rothman's method. In specific, CD_BIC yields the best performance for model C2 in terms of all error measures, and αCD_BIC and Rothman's method are among the best performers for model C1. The performance of CD_AIC and αCD_AIC appear to be less competitive for the covariance matrix estimation .

3.3.2.2 Speech Signal Classification

We now apply the covariance matrix estimation algorithm to a real example of speech signal classification (Little et al., 2009). The data is available at the UCI machine data repository, which consists of 195 speech signals among which 147 are Parkinson's disease patients. For each speech signal, 22 numerical features are provided. The data is randomly split into a training set with 65 signals and a testing set with 130 signals, where 49 of the training signals and 98 of the testing signals are from the Parkinson's desease patients. To conduct the classification, we use Fisher's quadratic discriminant analysis (QDA), where the classification decision function is given by

$$\delta(x) = \boldsymbol{argmax}_k \ \frac{1}{2}\log|\widehat{\Sigma}_k^{-1}| - (x_i - \hat{\mu}_k)^T\widehat{\Sigma}_k^{-1}(x_i - \hat{\mu}_k) + \log\hat{\pi}_k,$$

and $\hat{\mu}_k$ and $\hat{\Sigma}_k$ are the estimated mean and covariance matrix for group k. The splitting is replicated 500 times, and the averaged testing errors are summarized in Table XII (Yuan, T. and Wang, J. (2012) A coordinate descent algorithm for sparse positive definite matrix estimation. *Statistical Analysis and Data Mining*). Among all five covariance matrix estimation methods, αCD_BIC yields the smallest testing error, and three variants of the CD algorithms outperform Rothman's method in terms of classification.

Table XII about here

3.4 Block CD Algorithm And Graph Clustering

In this section, we extend the proposed CD algorithm to a block CD algorithm, which can be directly applied in graph clustering (Schaeffer, 2007). In graph clustering, each variable is represented as a vertex, and two vertexes are connected by a weighted edge where the weight relies on the dissimilarity of the vertexes. The goal of graph clustering is to cluster the vertexes into a number of subgroups so that the vertexes in the same subgroup are similar to each other. Suppose that the variables follow a multivariate Gaussian distribution, then a graph clustering problem boils down to estimation of a block diagonal precision matrix,

$$\boldsymbol{argmin}_{\Omega}\ \mathrm{tr}(\Omega S) - \log\det(\Omega),$$

subject to Ω is a block diagonal matrix, where S is the sample correlation matrix. The block CD algorithm starts from a diagonal matrix where each variable is a cluster, and repeatedly merges two clusters into a larger cluster or simply updates the current estimated precision matrix. The details of the block CD algorithm for graph clustering are given as follows.

Algorithm 3.4: (graph clustering algorithm)

Step 1. Initialize $\Omega_1 = (diag\{S\})^{-1}$ and the desired number of cluster r.

Step 2. Denote $\Omega_t = diag\{K_1, K_2, ..., K_m\}$ with m being the current number of clusters and $K_1, ...,K_m$ being m symmetric matrices. Set $D_t = S - \Omega_t^{-1}$, and the block coordinate (a, b) as

$$(a, b) = \boldsymbol{argmax}_{j \le k}\ \frac{\|(D_t)_{jk}\|_1}{dim(K_j)dim(K_k)}.$$

where $dim(K_j)$ is the number of rows in K_j and $\|(D_t)_{jk}\|_1$ is the sum of absolute values of all the entries in the (j,k)-th block of D_t.

Step 3. Let $(W_t)_{ab} = (W_t)_{ba}^T = (D_t)_{ab}$ and 0 elsewhere, compute

$$v_t^u = \left(\phi_{max}(K_a^{-1}(D_t)_{ab}K_b^{-1}(D_t)_{ab}^T)\right)^{-\frac{1}{2}}, \tag{3.13}$$

and update $\Omega_{t+1} = \Omega_t - v_t W_t$ with $v_t = \alpha v_t^u$.

Step 4. Repeat *Steps 2* and *3* until r clusters are generated.

Algorithm 3.4 turns out to be very similar to the popular agglomerative clustering algorithm, but with a new selection criterion in *Step 2*. In fact, the selection criterion in *Step 2* is to find the block with largest absolute group average, which is analogous to the group average for the agglomerative clustering. Other selection criteria can also be used, such as the single linkage or the full linkage (Hastie et al., 2009) . In *Step 3*, if $a \neq b$, updating Ω_{t+1} along W_t implies a merge of subgroups a and b; and if $a = b$, updating Ω_{t+1} along W_t does not merge any subgroups and only updates the current estimated precision matrix. Furthermore, Corollary 3 assures the positive definiteness of the estimated precision matrix.

Corollary 3. *In Algorithm 3.4, the generated precision matrices Ω_t are positive definite.*

Proof of Corollary 3: The proof is done for $a = b$ and $a \neq b$ respectively.

(1) When $a = b$, $(W_t)_{aa} = (D_t)_{aa}$, and then $\widetilde{K}_a = K_a - v_t(D_t)_{aa}$, where K_a and $\widetilde{K}_a$ are the (a,a)-th block in Ω_t and Ω_{t+1}, respectively. Since K_a is positive definite, the sufficient and

necessary condition for $\widetilde{K}_a$ to be positive definite is that the largest eigenvalue of $v_t K_a^{-1}(D_t)_{aa}$ is less than 1, which is equivalent to $v_t < (\phi_{max}(K_a^{-1}(D_t)_{aa}))^{-1} = v_t^u$ in Equation 3.13.

(2) When $a \neq b$, since only the (a, b)-th block will be updated, we denote Ω_t and Ω_{t+1} as

$$\Omega_t = \begin{pmatrix} K_a & 0 \\ 0 & K_b \end{pmatrix}, \qquad \Omega_{t+1} = \begin{pmatrix} K_a & -v_t(D_t)_{ab} \\ -v_t(D_t)_{ab}^T & K_b \end{pmatrix},$$

for simplicity. The positive definiteness of Ω_{t+1} is implied if there exists a matrix A such that

$$\Omega_{t+1} = (I + A\Omega_t^{-1})\Omega_t(I + A\Omega_t^{-1})^T.$$

The above equation can be simplified as

$$U + U^T + UK_a^{-1}U^T + v_t^2(D_t)_{ab}K_b(D_t)_{ab}^T = 0. \tag{3.14}$$

if assuming

$$A = \begin{pmatrix} U & -v_t(D_t)_{ab} \\ 0 & 0 \end{pmatrix},$$

where $dim(U) = dim(K_a)$, $dim(A) = dim(\Omega_{t+1})$. Denote $H = UK_a^{-1/2}$, and then Equation 3.14 can be rewritten as

$$(H + K_a^{1/2})(H + K_a^{1/2})^T = K_a - v_t^2(D_t)_{ab}(K_b)^{-1}(D_t)_{ab}^T. \tag{3.15}$$

Since the left hand side of Equation 3.15 is positive definite, $K_a - v_t^2 (D_t)_{ab} (K_b)^{-1} (D_t)_{ab}^T$ must be positive definite as well, which is equivalent to $v_t < v_t^u = (\phi_{max}((K_a)^{-1}(D_t)_{ab} K_b^{-1} (D_t)_{ab}^T))^{-1/2}$. This completes the proof.

3.4.1 Simulations

Next, we apply the proposed block CD algorithm to two graph clustering examples where the covariates follow multivariate Gaussian distribution.

Model G1 : The true covariance matrix Σ_0 consists of five blocks Σ_i with equal size $p = 10$, where $(\Sigma_0)_i = (V_i V_i^T + 0.2 I_{10})$ for $i = 1, ..., 5$, and $(V_i)_{jk} \sim [Unif(-1, 1)]_+$ for $j, k = 1, \ldots, p$.

Model G2 : The true covariance matrix Σ_0 consists of three blocks sizes $p_1 = 25$, $p_2 = 15$ and $p_3 = 10$, where $(\Sigma_0)_i = (V_i V_i^T + 0.2 I_{p_i})$ for $i = 1, 2, 3$, and V_i's are the same as in Model G1.

For each model, $n = 500$ observations are generated, and the corresponding sample correlation matrices are computed and supplied to *Algorithm 3.4*. Figure 3 displays the estimated clustering structures, which are very close to the clustering structure in the true covariance matrix (Yuan, T. and Wang, J. (2012) A coordinate descent algorithm for sparse positive definite matrix estimation. *Statistical Analysis and Data Mining*).

Figure 3 about here

CHAPTER 4

CONCLUSION REMARKS AND FUTURE WORK

In this chapter we summarize and draw conclusion remarks from this thesis, and discuss the possible future work for both types of structured matrix optimizations with their applications in statistics.

In this thesis we propose two kinds of the structured matrix optimizations, and apply them in statistics consequently. As the first remark, we propose a reduced-rank multi-label classification framework that can conduct multi-label classification and variable selection simultaneously. To optimize the resultant cost function, an efficient alternating optimization scheme is developed, which alternates between the constrained manifold optimization algorithm and the coordinate descent algorithm. The proposed algorithm is computationally efficient and delivers superior numerical performance in terms of both classification and variable selection. The asymptotic consistencies are also established to support the advantage of the proposed method. Compared with other methods proposed to deal with multi-label classification problem, our proposed model has the advantages of capturing the dependency structure by a simple reduced-rank constraint without specifically accounting for the order of interaction among labels, and that the consequent optimization problem can be efficiently solved by the proposed algorithm alternating between a fast constrained manifold optimization algorithm and a coordinate descent algorithm.

Note that in our proposed model, the asymptotic consistencies are established under fixed number of dimensions p, fixed number of labels q, and letting the number of instances $n \to \infty$.

It will be interesting to see whether the consistencies still hold under the scenarios $n < p$ or $n < q$ with $n \to \infty$. Intuitively, the conclusions may heavily depend on the behavior of rank r^* and cardinality p_0 of active set $\mathcal{A}_{\mathbf{C}^*}$, as well as the divergence rates of p, q and n. We believe that the asymptotic consistencies still hold under certain scenarios, whereas the proof may require more advanced techniques than what this thesis presents.

As the second remark, we propose a generic coordinate descent framework that can be used for optimization with respect to sparse positive definite matrix. The proposed algorithm in the framework iteratively updates the current estimated matrix at either one diagonal entry or two symmetric off-diagonal entries, and the step size is appropriately determined to assure the positive definiteness of the estimated matrix. The algorithm is applied to the estimation of the precision matrix and the covariance matrix, and yields superior numerical performance and computational efficiency against the popular competitors. The sparsity of the estimated matrix is achieved by early stopping the algorithm, and hence that no regularization term is needed. Furthermore, our generic coordinate descent framework is simple enough and generic enough to conduct the sparse precision matrix estimation and the sparse covariance matrix estimation.

It would be of interest to establish a connection between the proposed coordinate descent algorithm and the regularization method for profound theoretical analysis. Note that when the step size is close to zero, our algorithm is much like a forward stage-wise updating algorithm, the solution of which resembles a description by an ordinary differential equation (Wu, 2011), and this may imply that the regularization term follows some complicated differential or integral equation. Furthermore, in order to better clarify the connection between the proposed

coordinate descent algorithm and the regularization method, one may construct a roadmap composed of two parts. The first part attempts to demonstrate the solution set by all possible regularizers is identical with the solution set obtained from selecting various step size at each iteration; the second part attempts to find a path of the step sizes for each optimizer in regularization method. It seems difficult to tackle either part in the roadmap, however once this task is fulfilled, it is likely to establish the asymptotic consistencies for the estimators from our proposed coordinate descent algorithm.

CHAPTER 5

APPENDIX

TABLE I

AVERAGED HAMMING LOSS FOR VARIOUS METHODS AND THE STANDARD ERRORS (IN PARENTHESIS) IN VARIOUS SIMULATION SCENARIOS BASED ON 50 REPLICATIONS. THE BEST PERFORMER IN EACH SCENARIO IS BOLDFACED.

Scenario	RR	SB	CC	CW	PLSDA	Truth
1	.21(.026)	.27(.027)	.27(.020)	.27(.028)	**.14(.039)**	.06(.008)
2	.14(.013)	.15(.017)	.22(.017)	**.14(.006)**	.16(.001)	.14(.005)
3	**.05(0.01)**	.06(0.01)	.06(0.01)	.06(0.01)	.10(0.01)	.04(.005)
4	**.02(.005)**	.03(.008)	.04(.008)	.02(.005)	.06(.001)	.01(.000)
5	**.03(.004)**	.13(.015)	.16(.016)	.08(.010)	.06(.001)	.00(.000)

TABLE II

AVERAGED FROBENIUS NORM ERROR FOR VARIOUS METHODS AND THE STANDARD ERRORS (IN PARENTHESIS) IN VARIOUS SIMULATION SCENARIOS BASED ON 50 REPLICATIONS. THE BEST PERFORMER IN EACH SCENARIO IS BOLDFACED.

Scenario	RR	SB	CC	CW
1	**0.039(0.002)**	0.043(0.004)	0.045(0.005)	0.043(0.004)
2	**0.728(0.061)**	1.633(0.167)	0.932(0.159)	1.633(0.167)
3	**0.529(0.052)**	0.806(0.062)	0.684(0.057)	0.806(0.062)
4	**24.735(0.531)**	26.837(1.263)	29.646(1.443)	26.837(1.263)
5	39.672(1.266)	39.616(1.315)	**37.776(1.192)**	39.616(1.315)

TABLE III

AVERAGED VARIABLE SELECTION ERROR FOR VARIOUS METHODS AND THE STANDARD ERRORS (IN PARENTHESIS) IN VARIOUS SIMULATION SCENARIOS BASED ON 50 REPLICATIONS. THE BEST PERFORMER IN EACH SCENARIO IS BOLDFACED.

Scenario	RR	SB	CC	CW
1	**.38(.042)**	.56(.053)	.58(.091)	.56(.053)
2	**.30(.040)**	.44(.078)	.51(.102)	.44(.078)
3	**.36(.053)**	.52(.052)	.49(.052)	.52(.052)
4	**.17(.050)**	.48(.127)	.50(.148)	.48(.127)
5	**.17(.043)**	.33(.085)	.35(.072)	.33(.085)

TABLE IV

AVERAGED HAMMING LOSS AND NUMBER OF SELECTED VARIABLES OF VARIOUS METHODS IN THE REAL EXAMPLES. THE BEST PERFORMER IN EACH EXAMPLE IS BOLDFACED.

	RR	SB	CC	CW	PLSDA
Birds					
Ham. Loss	**.05**	.25	.28	.16	.14
Dimension	127	199	215	199	-
Scene					
Ham. Loss	**.09**	.27	.31	.16	.14
Dimension	183	237	259	237	-

TABLE V

AVERAGED KL LOSSES AND THEIR ESTIMATED STANDARD ERRORS IN PARENTHESES FOR VARIOUS PRECISION MATRIX ESTIMATION ALGORITHMS BASED ON 100 REPLICATIONS. THE BEST PERFORMER IN EACH COMPARISON IS BOLDFACED.

Model	p	glasso	CD_AIC	CD_BIC	αCD_AIC	αCD_BIC
P1	25	1.61(.037)	1.91(.032)	1.70(.034)	**1.13(.023)**	1.25(.029)
	50	4.12(.062)	5.38(.049)	4.33(.036)	**3.73(.031)**	6.13(.076)
	100	10.91(.080)	16.53(.090)	11.31(.071)	**9.64(.046)**	13.58(.052)
P2	25	2.34(.031)	2.43(.026)	2.34(.025)	**1.89(.021)**	2.05(.025)
	50	5.78(.050)	7.02(.036)	5.59(.033)	**5.11(.033)**	6.88(.083)
	100	13.59(.100)	20.07(.079)	13.69(.072)	**12.53(.051)**	15.26(.045)
P3	25	**1.84(.036)**	2.26(.026)	2.34(.029)	1.90(.022)	2.30(.034)
	50	6.17(.084)	5.73(.044)	5.36(.043)	**4.90(.031)**	8.93(.104)
	100	23.67(.305)	18.81(.082)	19.69(.074)	**16.33(.053)**	36.32(.122)
P4	25	4.23(.074)	**2.97(.025)**	5.03(.029)	3.00(.026)	5.31(.049)
	50	12.45(.169)	9.30(.046)	11.78(.043)	**7.99(.042)**	16.76(.347)
	100	31.80(.279)	25.86(.085)	27.49(.065)	**20.68(.059)**	36.05(.093)

Figure 1. The Hamming losses as well as the BIC criterion value as functions of $\log s$ or r in the real examples.

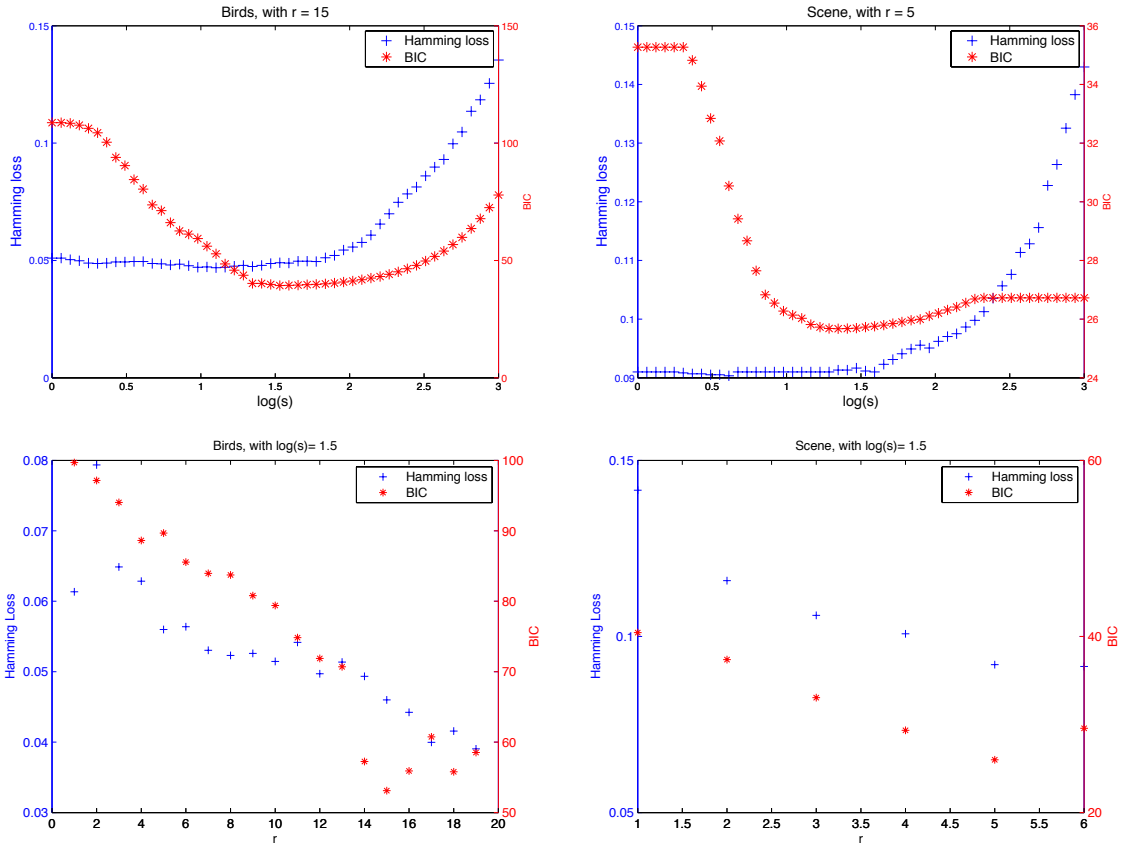

Figure 2. Solution paths of the smallest eigenvalue $\phi_{min}(\Omega_t)$ (the thick line) and a number of randomly selected entries of Ω_t (the thin lines) in the simulated model P1 with $p = 100$, $n = 1000$ and $\alpha = 0.2$.

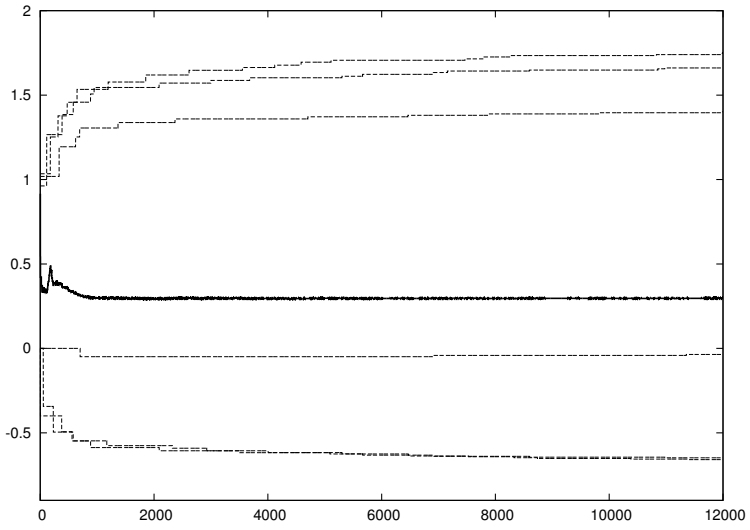

Figure 3. Heat map of the estimated clustering structure for models G1 and G2. The left column displays the sample correlation matrices, and the right column displays the estimated precision matrices.

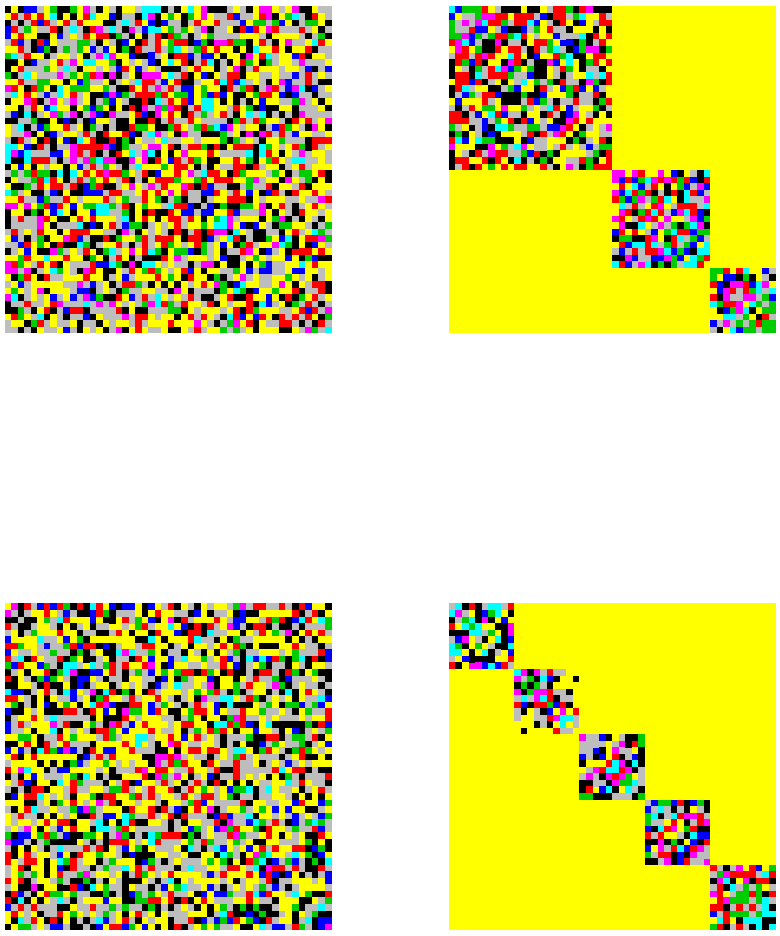

TABLE VI

AVERAGED F-NORM LOSSES AND THEIR ESTIMATED STANDARD ERRORS BASED ON 100 REPLICATIONS. THE BEST PERFORMER IN EACH COMPARISON IS BOLDFACED.

Model	p	glasso	CD_AIC	CD_BIC	αCD_AIC	αCD_BIC
P1	25	3.03(.034)	3.01(.045)	3.06(.047)	**1.90(.019)**	1.98(.022)
	50	4.87(.029)	**4.23(.016)**	4.45(.019)	4.27(.038)	6.23(.069)
	100	7.82(.021)	**6.93(.017)**	7.69(.014)	7.25(.035)	9.24(.016)
P2	25	3.12(.015)	2.51(.013)	2.84(.013)	**2.31(.010)**	2.40(.009)
	50	4.79(.014)	**3.94(.010)**	4.37(.008)	4.22(.033)	5.11(.059)
	100	7.17(.017)	**6.03(.008)**	6.33(.008)	6.87(.023)	7.69(.007)
P3	25	3.00(.029)	3.22(.024)	3.37(.026)	**2.86(.015)**	2.97(.016)
	50	7.16(.035)	**6.38(.023)**	6.46(.024)	6.48(.021)	8.70(.030)
	100	17.05(.077)	**13.55(.023)**	14.72(.027)	14.02(.038)	20.29(.027)
P4	25	8.93(.065)	7.44(.025)	9.47(.020)	**7.42(.021)**	8.90(.013)
	50	19.77(.062)	17.29(.028)	19.08(.023)	**15.70(.020)**	21.76(.090)
	100	41.39(.064)	36.57(.024)	38.49(.028)	**36.48(.088)**	43.07(.052)

TABLE VII

AVERAGED SIGNED VARIABLE SELECTION LOSSES AND THEIR ESTIMATED STANDARD ERRORS BASED ON 100 REPLICATIONS. THE BEST PERFORMER IN EACH COMPARISON IS BOLDFACED.

Model	p	glasso	CD_AIC	CD_BIC	αCD_AIC	αCD_BIC
P1	25	.447(.0008)	.469(.0010)	.442(.0006)	.457(.0010)	**.439(.0004)**
	50	.324(.0005)	.377(.0010)	.324(.0006)	.410(.0023)	**.315(.0005)**
	100	.186(.0003)	.252(.0003)	.195(.0004)	.266(.0015)	**.181(.0001)**
P2	25	.192(.0022)	.262(.0023)	**.174(.0014)**	.251(.0027)	.176(.0014)
	50	.103(.0010)	.205(.0015)	.101(.0007)	.253(.0051)	**.089(.0008)**
	100	.052(.0005)	.141(.0003)	.062(.0004)	.150(.0027)	**.045(.0002)**
P3	25	.110(.0030)	.175(.0022)	**.074(.0018)**	.169(.0038)	.076(.0017)
	50	.083(.0009)	.146(.0015)	**.057(.0007)**	.255(.0021)	.078(.0014)
	100	.094(.0007)	.126(.0007)	**.067(.0003)**	.238(.0016)	.076(.0003)
P4	25	.423(.0030)	**.376(.0018)**	.445(.0023)	.385(.0018)	.448(.0019)
	50	.459(.0011)	**.439(.0011)**	.462(.0009)	.454(.0010)	.466(.0012)
	100	.482(.0003)	**.472(.0005)**	.476(.0004)	.478(.0005)	.481(.0002)

TABLE VIII

AVERAGED TESTING ERRORS AND THEIR ESTIMATED STANDARD ERRORS OVER 100 REPLICATIONS. THE BEST PERFORMER IN EACH COMPARISON IS BOLDFACED.

p	glasso	CD_AIC	αCD_AIC	CD_BIC	αCD_BIC
25	16.25(.506)	18.25(.601)	17.25(.652)	**14.75(.433)**	15.25(.406)
50	20.75(.507)	20.75(.742)	20.75(.723)	**17.75(.451)**	20.50(.605)
100	23.25(.939)	19.65(.540)	17.75(.868)	17.00(.350)	**16.75(.524)**

TABLE IX

AVERAGED F-NORM LOSSES AND THEIR ESTIMATED STANDARD ERRORS IN PARENTHESES FOR VARIOUS COVARIANCE MATRIX ESTIMATION ALGORITHMS BASED ON 500 REPLICATIONS WITH $N = 50$ AND $\lambda = 0.0001$. THE BEST PERFORMER IN EACH COMPARISON IS BOLDFACED.

Model	p	CD_AIC	αCD_AIC	CD_BIC	αCD_BIC	Rothman
C1	30	3.76(0.010)	2.97(0.010)	3.32(0.010)	2.41(0.012)	**2.39(0.010)**
	100	10.88(0.064)	7.99(0.062)	7.96(0.064)	5.62(0.050)	**4.93(0.010)**
	200	16.67(0.100)	12.25(0.118)	11.88(0.120)	8.75(0.023)	**7.34(0.010)**
C2	30	4.16(0.030)	4.00(0.023)	4.03(0.019)	3.88(0.019)	**3.83(0.020)**
	100	11.58(0.055)	11.38(0.050)	**10.64(0.032)**	10.92(0.044)	10.73(0.040)
	200	19.57(0.064)	17.98(0.064)	**16.57(0.050)**	16.74(0.056)	16.76(0.040)

TABLE X

AVERAGED SPECTRAL NORM LOSSES AND THEIR STANDARD ERRORS BASED ON 500 REPLICATIONS WITH $N = 50$ AND $\lambda = 0.0001$. THE BEST PERFORMER IN EACH COMPARISON IS BOLDFACED.

Model	p	CD_AIC	αCD_AIC	CD_BIC	αCD_BIC	Rothman
C1	30	1.40(0.010)	0.95(0.010)	1.10(0.010)	**0.77(0.010)**	0.85(0.000)
	100	2.91(0.019)	2.19(0.015)	2.02(0.018)	0.98(0.010)	**0.96(0.000)**
	200	4.75(0.025)	3.24(0.019)	2.76(0.019)	1.87(0.012)	**1.00(0.000)**
C2	30	2.37(0.022)	2.35(0.026)	**2.25(0.018)**	2.38(0.015)	2.40(0.020)
	100	5.25(0.025)	5.19(0.025)	**5.04(0.024)**	5.53(0.019)	5.48(0.030)
	200	7.09(0.034)	6.01(0.023)	**5.86(0.187)**	6.44(0.033)	6.38(0.020)

TABLE XI

AVERAGED SIGNED VARIABLE SELECTION LOSSES AND THEIR ESTIMATED STANDARD ERRORS BASED ON 500 REPLICATIONS WITH $N = 50$ AND $\lambda = 0.0001$. THE BEST PERFORMER IN EACH COMPARISON IS BOLDFACED.

Model	p	CD_AIC	αCD_AIC	CD_BIC	αCD_BIC	Rothman
C1	30	0.31(0.001)	0.49(0.001)	**0.16(0.001)**	0.17(0.001)	–
	100	0.34(0.000)	0.45(0.000)	0.10(0.001)	**0.07(0.000)**	–
	200	0.32(0.000)	0.44(0.000)	0.08(0.000)	**0.05(0.000)**	–
C2	30	0.26(0.000)	0.46(0.001)	**0.17(0.000)**	0.30(0.000)	–
	100	0.37(0.000)	0.42(0.001)	**0.15(0.000)**	0.21(0.000)	–
	200	0.43(0.000)	0.42(0.001)	**0.13(0.000)**	0.14(0.000)	–

TABLE XII

AVERAGED CLASSIFICATION ERRORS BASED ON 500 REPLICATIONS WITH $\lambda = 0.0001$. THE BEST PERFORMER IN EACH COMPARISON IS BOLDFACED.

CD_AIC	αCD_AIC	CD_BIC	αCD_BIC	Rothman
0.209(0.0765)	0.192(0.0668)	0.245(0.0801)	**0.190(0.0674)**	0.219(−)

CITED LITERATURE

1. Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716-723.

2. Alon, U., Barkai, N., Notterman., D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, **96**, 6745-6750.

3. Barker, M., and Rayens, W. (2003). Partial least squares for discrimination. *Journal of Chemometrics*, **17**, 166-173.

4. Barutcuoglu, Z., Schapire, R. E. , and Troyanskaya, O. G. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, **22**, 830-836.

5. Bertsekas, D. P. (1999). Nonlinear Programming, 2nd Edition, *Athena Scientific, Belmont, Massachusetts.*

6. Bickel, P. J. and Levaina, E. (2008a). Regularized estimation of large covariance matrices. *Annals of Statistics*, **36**, 199-227.

7. Bickel, P. J. and Levina, E. (2008b). Covariance regularization by thresholding. *Annals of Statistics*, **36**, 2577-2604.

8. Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multilabel scene classiffication. *Pattern Recognition*, **37**(9): 1757-1771.

9. Breiman, L., and Friedman, J. H. (1997). Predicting Multivariate Responses In Multiple Linear Regression, *Journal of the Royal Statistical Society: Series B (Statistical Methodology*, **59**, 354.

10. Briggs, F., Lakshminarayanan, B., Neal, L., Fern, X. Z., Raich, R., Hadley, S. J. K., Hadley, A. S., and Betts, M. G. (2012) New methods for acoustic classification of multiple simultaneous bird species in a noisy environment. *IEEE International Workshop on Machine Learning for Signal Processing.*

11. Cai, T. and Liu, W. (2011). Adaptive Thresholding for Sparse Covariance Matrix Estimation. *Journal of the American Statistical Association*, **494**, 672-684.

12. Chen, L. S., and Huang, J. H. (2012). Sparse Reduced-Rank Regression for Simultaneous Dimension Reduction and Variable Selection. *Journal of the American Statistical Association*, **107**, 1533-1545.

13. Cheng, J., Levina, E., Wang, P., and Zhu, J. (2014). A sparse ising model with covariates. *Biometrics*, **70**, 943-953.

14. Clare, A.,J. and King, R. D. (2001). Knowledge discovery in multi-label phenotype data. *The 5th European Conference on Principles of Data Mining and Knowledge Discovery: Lecture Notes in Artificial Intelligence*, **2168**, 42-53.

15. Drton, M. and Perlman, M.D. (2004). Model selection for Gaussian concentration graphs. *Biometrika*, **91**, 591-602.

16. Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. *Proceedings of the seventh international conference on Information and knowledge management*, 148-155.

17. Edelman, A., Arias, T. A., and Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*. **20**, 303-353.

18. Edwards, D. (2000). *Introduction to Graphical Modeling, 2nd ed.* Springer, New York.

19. Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, **32**, 407-499.

20. El Karoui, N. (2008). Operator norm consistent estimation of large dimensional sparse covariance matrices. *Annals of Statistics*, **36**, 2717-2756.

21. Elisseeff, A., and Weston, J. (2002). A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems*, **14**.

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

23. Fan, J., Liao, Y., and Mincheva, V. (2013). Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society, Series B*, **75**, 603-680.

24. Fan, R. E., and Lin, C. J. (2007). A study on threshold selection for multi-label classification. *Technical Report, National Taiwan University.*

25. Friedman, J. H., Hastie,T. and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, **1**, 302-332.

26. Friedman, J. H., Hastie,T. and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**, 432-441.

27. Friedman, J. H., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33**, 1-22.

28. Ghamrawi, N. and McCallum, T. (2005). Collective multi-label classification. *Pro-*

ceedings of the 14th ACM International Conference on Information and Knowledge Management. Germany, 22-30.

29. Harville, D. (1997). *Matrix Algebra From a Statisticians Perspective.* Springer-Verlag, New York.

30. Hastie, T., Taylor, J., Tibshirani, R. and Walther, G. (2007). Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, **1**, 1-29.

31. Hastie, T., Tibshirani, R. and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd Edition).* Springer-Verlag, New York.

32. Izenman A. J. (1975). Reduced-Rank Regression for the Multivariate Linear Model. *Journal of Multivariate Analysis*, **5**, 248-264.

33. Little, M., Mcsharry, P., Hunter, E., Spielman, J. and Ramig, L. (2009). Suitability of dysphonia measurements for telemonitoring of Parkinsons disease. *IEEE Transactions on Biomedical Engineering*, **56**, 1015-1022.

34. Liu, H., and Luo, X. (2014) High-dimensional sparse precision matrix estimation via sparse column inverse operator. *Journal of Multivariate Analysis*, To appear.

35. Liu, H., Wang, L., and Zhao, T. (2014). Sparse covariance matrix estimation with

eigenvalue constraints. *Journal of Computational and Graphical Statistics*, **23**, 439-459.

36. Liu, Y., Zhang, H., Park, C., and Ahn, J. 2007. Support vector machines with adaptive L_q penalty. *Computational Statistics and Data Analysis*, **51**, 6380-6394.

37. Luaces, O., Dez, J., Barranquero, J., Jos del Coz, J., and Bahamonde, A. (2012). Binary relevance efficacy for multilabel classification, *Progress in Artificial Intelligence*, **4**, 303-313.

38. Luo, Z. and Tseng, P. (1992). On the Convergence of the Coordinate Descent Method for Convex Differentiable Minimization. *Journal of Optimization Theory and Applications*, **72**, 7-35.

39. Madjarov G., Kocev D., Gjorgjevikj D., and Deroski S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition,*,**45**, 3084-3104.

40. Mazumder, R., Friedman, J.H., and Hastie, T. (2011). SparseNet: Coordinate Descent With Nonconvex Penalties. *Journal of the American Statistical Association*, **106**, 1125-1138.

41. Meinshausen, N. and Buhlmann, P. (2006). High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, **34**, 1436-1462.

42. Miller, K. (1981). On the Inverse of the Sum of Matrices. *Mathematics Magazine*, **54**, 67-72.

43. Nocedal, J., and Yuan, Y.X. (1998). Combining trust region and line search techniques. *Advances in nonlinear programming*, 153-175.

44. Peters, S., Jacob, Y., Denoyer, L., and Gallinari, P. (2012). Iterative multi-label multi-relational classification algorithm for complex social networks. *Social Network Analysis and Mining*, **2**, 17-29.

45. Qi, H., and Sun, D. (2006). A quadratically convergent newton method for computing the nearest correlation matrix. *SIAM journal on matrix analysis and applications*, **28**, 360-385.

46. Ravikumar, P., Wainwright, S., Raskutti, G., Yu, B. (2011). High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, **5**, 935-980.

47. Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2009). Classifier Chains for Multi-label Classification, *Machine Learning and Knowledge Discovery in Databases, Springer*, 254-269.

48. Richtrik, P., Tak, M. (2012). Parallel Coordinate Descent Methods for Big Data Optimization. *arXiv preprint arXiv: 1212.0873*.

49. Rothman, A. (2012). Positive definite estimators of large covariance matrices. *Biometrika*, to appear.

50. Rothman, A., Bickel, P., Levina, E. and Zhu, J. (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, **2**, 494515.

51. Rothman, A., Levina, E. and Zhu, J. (2009). Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, **104**, 177-186.

52. Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461464.

53. Schaeffer, S. (2007). Graph clustering. *Computer Science Review*, **1**, 27-64.

54. Shao, J. (2003). *Mathematical Statistics*, 2nd Edition, Springer.

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

56. Tseng, P. (1990). Dual Ascent Methods for Problems with Strictly Convex Costs and Linear Constraints: A Unified Approach. *SIAM Journal on Control and Optimization*, **28**, 214-242.

57. Tsoumakas, G., Katakis, I. (2007). Multi-label classification: An overview. *International*

Journal of Data Warehousing and Mining, **3**, 113

58. Tsoumakas, G., Katakis, I., and Vlahavas, I. (2008). Random k-labelsets for multi-label classification, *IEEE Trans. Knowledge and Data Engineering*, **23**, 1079-1089.

59. Wang, H. L. (2008). A note on adaptive group lasso. *Computational Statistics & Data Analysis*, **52**, 5277-5286.

60. Wang, J. and Wang, L. (2010). Sparse supervised dimension reduction in high dimensional classification. *Electronic Journal of Statistics*, **4**, 914-931.

61. Wang, J. (2010). Consistent selection of the number of clusters via cross validation. *Biometrika*, **97**, 893-904.

62. Wen, Z. and Yin, W. (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, **142**, 397-434.

63. Wu, Y. C. (2011). An ordinary differential equation based solution path algorithm. *Journal of Nonparametric Statistics*, **23**, 185-199.

64. Wu, T. and Lange, K. (2008). Coordinate descent algorithm for lasso penalized regression. *Annals of Applied Statistics*, **2**, 224-244.

65. Yang, Y. (2005). Can the strengths of AIC and BIC be shared? A conflict between model

indentification and regression estimation. *Biometrika*, **92**, 937-950.

66. Yu, H. F., Jain, P., Kar P., and Dhillon I.S. (2014). Large-scale Multi-label Learning with Missing Labels. *Proceedings of the 31st International Conference on Machine Learning*.

67. Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, **68**, 49-67.

68. Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, **94**, 19-35.

69. Yuan, M. (2010) High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, **11**, 2261-2286.

70. Zhang, H., P. (1996). Classification trees for multiple binary responses. *Journal of the American Statistical Association*, **93**, 180-193.

71. Zhang, M. L., and Zhou, Z. H. (2007). ML-KNN A lazy learning approach to multi-label learning. *Pattern Recognition*, **40**, 2038-2048.

72. Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of American Statistical Association*, **101**, 1418-1429.

73. Zhou, Z. H., and Zhang M. L. (2006). Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, **18**, 1338-1351.

VITA

Ting Yuan

Research Interests

Large data mining, Statistical machine learning.

Education

University of Illinois Chicago (UIC) Aug 2009-present

Ph.D. Candidate in Statistics **2012-2015 (expected)**

M.S., Statistics Dec 2011

M.S., Theoretical Physics May 2011

University of Science and Technology of China (USTC), Hefei, Anhui, China

B.S., Special Class of Gifted Young (SCGY)

Thesis: Quantum Computation and Unitary Decomposition, Sept 2005-June 2009

Work Experience

Research Assistant, Department of Math, CS, & Statistics, UIC Aug 2012-present

Research Assistant, Department of Physics, UIC Aug 2010-June 2012

Honors and Awards

Graduate Student Assistantship, UIC, 2009-present.

Publications

Ting Yuan and Junhui Wang. (2013) A coordinate descent algorithm for sparse positive definite matrix estimation, *Statistical Analysis and Data Mining*, DOI: 10.1002/sam.11185.

Ting Yuan, Jeremy Figgins and Dirk K Morr. (2011) Hidden order transition in URu2Si2: Evidence for the emergence of a coherent Anderson lattice from scanning tunneling spectroscopy, *Physics Review B*, 86, 035129.

Talk & Presentation

- Statistical Learning and Data Mining Conference, 2012, University of Michigan, Ann Arbor, MI.
- American Physics Society, 2011, Dallas, TX.

Selected Courses

Advanced statistical theory	Applied statistics models	Stochastic process and models
Topics in machine learning	Computational statistics	Principles of microcomputer
Object-oriented programming	Software design	Data structure and algorithm

Selected Projects

- Jump Trading Challenge: Develop the algorithm based on the latent-variable Markov model in solving the open challenge, see *http://www.jumptrading.com/challenge/*.
- Network Analysis: Analyzed the grouping map of S&P 500 companies based on the self-realized sparse graphical model inference on stock price. Predicted 90% of group relationships correctly.

- Large financial data learning: Analyzed the daily trading data of size over 2 GB with highly performed classification algorithm.
- Design an iPhone app of smart calculator with X-code.

Computer Skills

- Programming Languages: C++, R, MatLab, Smalltalk, Objective-C, Mathematica.
- Applications: LaTeX, Microsoft Office, Origin.
- Operating Systems: Linux, Windows, MacOS.

Expected completion date: Sep 2015

Expected size (number of pages): 110

Requestor Location: Ting Yuan, 1200 west harrison street, CHICAGO, IL 60607, United States