STATISTICAL ANALYSIS OF HIGH DIMENSIONAL DATA

A Thesis
Presented to
The Academic Faculty

by

Lingyan Ruan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H.Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2010

STATISTICAL ANALYSIS OF HIGH DIMENSIONAL DATA

Approved by:

Professor Ming Yuan, Advisor
H.Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor JC Lu
H.Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Xiaoming Huo
H.Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Nicoleta Serban
H.Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Yixin Fang
Department of Mathematics and Statistics, Georgia State University
Georgia Institute of Technology

Date Approved: November 2010

To my parents and husband

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Ming Yuan, for his introduction to the field of statistical learning. Without his guidance and encouragement, my research could not have been in shape. I really appreciate his patience in the long process and I feel very lucky to have him as my advisor. I learned from him not only the research abilities, but also his wisdom, which I believe will benefit me a lot in my whole life.

I am profoundly thankful for the support from Professor JC Lu. JC shares generously with me his brilliance and humor. He takes care of me personally in a very considerate way. It is fun to work with him. Also, I am extremely grateful to my thesis committee members, Professor Xiaoming Huo, Professor Nicoleta Serban and Professor Yixin Fang, for their service on my dissertation committee and their helpful comments and suggests.

I would also thank all my friends for their help to me. I enjoy the stay with Leihong Li and Chunpeng Xiao. It is a fruitful, pleased and memorable experience with them. The discussion with Xinwei deng inspired me a lot when I was puzzled. I also thank my friends Lulu Kang, Huijing Jiang, Xing Wang, Yibiao Lu, Yijie Wang, Huizhi Xie and Heeyoung Kim to share a great time with me in school.

Finally, I owe my parents and husband more than I can pay back. Their love always accompanies me and supports me anywhere anytime.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This century is surely the century of data (Donoho, 2000). Data analysis has been an emerging activity over the last few decades. High dimensional data is in particular more and more pervasive with the advance of massive data collection system, such as microarrays, satellite imagery, and financial data. However, analysis of high dimensional data is of challenge with the so called curse of dimensionality (Bellman 1961). This research dissertation presents several methodologies in the application of high dimensional data analysis.

The first part discusses a joint analysis of multiple microarray gene expressions. Microarray analysis dates back to Golub et al. (1999). It draws much attention after that. One common goal of microarray analysis is to determine which genes are differentially expressed. These genes behave significantly differently between groups of individuals.However, in microarray analysis, there are thousand of genes but few arrays (samples, individuals) and thus relatively low reproducibility remains. It is natural to consider joint analyses that could combine microarrays from different experiments effectively in order to achieve improved accuracy. In particular, we present a model-based approach for better identification of differentially expressed genes by incorporating data from different studies. The model can accommodate in a seamless fashion a wide range of studies including those performed at different platforms, and/or under different but overlapping biological conditions. Model-based inferences can be done in an empirical Bayes fashion. Because of the information sharing among studies, the joint analysis dramatically improves inferences based on individual analysis. Simulation studies and real data examples are presented to demonstrate the effectiveness of the proposed approach under a variety of complications that often

arise in practice.

The second part is about covariance matrix estimation in high dimensional data. First, we propose a penalised likelihood estimator for high dimensional t-distribution. The student t-distribution is of increasing interest in mathematical finance, education and many other applications. However, the application in t-distribution is limited by the difficulty in the parameter estimation of the covariance matrix for high dimensional data. We show that by imposing ℓ_1 penalty on the Cholesky factors of the covariance matrix, EM algorithm can efficiently compute the estimator and it performs much better than other popular estimators.

Secondly, we propose an estimator for high dimensional Gaussian mixture models. Finite Gaussian mixture models are widely used in statistics thanks to its great flexibility. However, parameter estimation for Gaussian mixture models with high dimensionality can be rather challenging because of the huge number of parameters that need to be estimated. For such purposes, we propose a penalized likelihood estimator to specifically address such difficulties. The ℓ_1 type penalty we impose on the inverse covariance matrices encourages sparsity on its entries and therefore helps reducing the dimensionality of the problem. We show that the proposed estimator can be efficiently computed via an Expectation-Maximization algorithm. To illustrate the practical merits of the proposed method, we consider its application in model-based clustering and mixture discriminant analysis. Numerical experiments with both simulated and real data show that the new method is a valuable tool in handling high dimensional data.

Finally, we present structured estimators for high dimensional Gaussian mixture models. The graphical representation of every cluster in Gaussian mixture models may have the same or similar structure, which is an important feature in many applications, such as image processing, speech recognition and gene network analysis. Failure to consider the sharing structure would deteriorate the estimation accuracy.

To address such issues, we propose two structured estimators, hierarchical Lasso estimator and group Lasso estimator. An EM algorithm can be applied to conveniently solve the estimation problem. We show that when clusters share similar structures, the proposed estimator perform much better than the separate Lasso estimator.

CHAPTER I

AN EMPIRICAL BAYES APPROACH TO JOINT ANALYSIS OF MULTIPLE MICROARRAY GENE EXPRESSION STUDIES

1.1 Introduction

Microarray technology has presented unprecedented opportunities in genomic studies of complex diseases. It allows researchers to simultaneously monitor thousands of transcripts and discover novel bio-markers and genes. Despite their successes, these studies are often hampered by their relatively low reproducibility. This deficiency is often attributed to the high variability of gene expression measurements. Sources of distortion and noise are involved in almost every step along the process of taking gene expression measurements. It has long been recognized (e.g., Lee et al., 2000; Mukherjee et al., 2003) that such problem could be alleviated through increased sample size. However, experiments with limited sample sizes remain common due to economic considerations. The recent explosion of popularity of high-throughput gene expression studies offers a more cost-effective alternative to this problem. With studies of the same diseases carried out independently by different research groups, it is natural to consider efficient ways of combining these data and jointly analyzing them. Through information sharing across studies, the accuracy of inferences could be greatly improved.

Because of its great potential, joint analysis of multiple experiments has attracted much attention in recent years. It is most commonly done through cross-experiment data normalization and transformation, which aims at translating and normalizing measurements from different sources on a common scale to allow for integration. Jiang

et al. (2004) present a gene shaving method based on random forests (Breiman, 2001) and Fisher's linear discrimination analysis. Warnat et al. (2005) and Shabalin et al. (2008) also discuss different ways of integrating data through cross-experiment transformation. And Parmigiani et al. (2002), Shen et al. (2004), Choi et al. (2007) translated all the observations into a probability of expression (poe) as a new scale. In general, however, it is difficult to integrate data without information loss and this would heavily bias each study. For example, van't Veer et al. (2002) and Wang et al. (2005) ended up with different predictive gene subsets with only three genes in common. and there is no clear guidelines as to how it can be performed efficiently.

Alternatively, one can also combine individual analysis results summarized by t-statistic, p-value, scored gene list and so on (e.g., Choi et al., 2003; Rhodes et al., 2002; Garrett-Mayer, 2007; Ghosh et al., 2003; Pyne et al., 2006). In particular, Choi et al., (2003) propose to combine the effect size of genes from each study and conduct a permutation test to determine the significance level. Rhodes et al. (2002) and Pyne et al. (2006) consider ways of combining p-values of each study. Due to the small sample size of each study, the summary statistics obtained inevitably have high variations and subsequently these methods are subject to loss of efficiency in information sharing. This happens such as the studies of van't Veer et al. (2002) and Wang et al. (2005) mentioned above. It is also demonstrated by Mah et al. (2004) that detected genes on different platforms could have poor overlap. See Hong et al. (2008) for a comparison of methods and Rhodes et al. (2004), Parmigiani et al. (2004) for other approaches discussions.

There are also several major practical hurdles to joint analysis. In particular, there is no general consensus on how gene expression experiments should be conducted. As a result, the choice of sample cohorts (e.g., age, ethnicity, and phase of disease), experiment platforms (e.g., cDNA or oligonucleotide), and processing facilities may all be different, and the scale of observations may not be comparable. These variations

among experiments prohibit us from treating them as if they were simple replicates from a single study. In particular, a recent study in Kuo et al. (2002) compared Affymetrix and spotted cDNA and it was claimed that the correlation between the measurements from the two platforms was fairly low so it was unlikely that the two types of data could be transformed or normalized into a common standardized index. In practice, integrating multiple studies can be further complicated by missing data, gene set mismatch and some times, mismatch in biological conditions. The concepts of missing data and gene set mismatch are the same when combining all data from the studies. Specifically, we define missing data as incomplete observations for one gene present in the study. Gene set mismatch happens when there is no observations for one gene in the study when combining the studies.

Consider, for illustration purpose, the study of prostate cancer, the most diagnosed cancer in men. There are a host of gene expression studies of prostate cancer. To motivate our work, Microarray data were collected from four publicly available prostate cancer gene expression datasets generated independently by Dhanasekaran et al. (2001), Luo et al. (2001), Magee et al. (2001) and Welsh et al. (2001) respectively. One of the goals common to all four studies is in determining which genes are differentially expressed between locally advanced prostate cancer and benign tissue. The experiments, however, are done with different technologies, Dhanasekaran et al. (2001) and Luo et al. (2001) studies used spotted cDNA microarrays (Schena, 2000); while the other two experiments utilized Affymetrix technology (Lipshutz et al. 1999) to focus on RNA and cRNA gene chip respectively. Furthermore, these studies were performed on different but overlapping sets of genes. To overcome this problem, existing methods (see, e.g., Rhodes et al., 2002; Ghosh et al., 2003; Warnat et al., 2005) focus only on genes that are present in all studies. As we shall see in Section 4, such practice may result in more than 75% of the genes being discarded in some studies. Moreover, the remaining 25% of genes contain missing data, i.e.,

not all genes have complete observations from the samples tested. If the methods applied can not allow missing data, this will reduce to only 1 gene (satisfying both intersection and complete data). This is clearly not an effective way of using the data. Another complication in combining the four experiments is the mismatch in biological conditions. Although all four studies include comparisons between locally advanced prostate cancer and benign prostate, Dhanasekaran et al. (2001) and Magee et al. (2001) also included a third biological condition: metastatic prostate cancer. Earlier attempts to combine these studies have either chosen to discard data collected from this condition or combining it with locally advanced cancer to form a new hypothesis.

These aforementioned limitations prompt us to develop a new technique. In this chapter, we propose a model-based method to integrate information from multiple experiments for the purpose of identifying differentially expressed genes among multiple biological conditions. Following Newton et al. (2001) and Kendziorski et al. (2003), we model the data from each individual study by a parametric empirical Bayes model to share information across transcripts. These separate models are flexible to be applicable to different platforms and multiple biological conditions. Latent variables are then introduced to model the pattern of expression for a particular transcript and to share information across experiments. The modeling framework is fairly flexible and can handle a variety of practical issues including those mentioned above with ease. Within this framework, all data present in every study can be used for analysis, not only intersection genes. Without loss of generality, let us assume that the genes into analysis are concordant. For disconcordant genes in the union data, apply methods discussed in Garrett-Mayer et al. (2007) to remove them.

In the next sections, we introduce first the joint analysis methods in literature. Then we present general modeling framework and show how statistical inferences can be efficiently conducted. Section 4 presents simulation studies to demonstrate the merits and versatility of the proposed method. We revisit the prostate cancer

examples in Section 5 as well as another real data example of liver cancer before concluding with some remarks and discussions in Section 6.

1.2 Overview of Joint Analysis Methods

1.2.1 Data Normalization Methods

Probability of Expression (POE) is the measure proposed by Shen et al.(2004) to integrate multiple data into one set. It is computed based on a Bayesian mixture modeling approach. For any gene, there are three possibilities: over expressed, under expressed and normally expressed. Then, given the j-th gene expression x_{ij} in the i-th data, POE is calculated as

$$p^*(x_{ij}) = p^+(x_{ij}) - p^-(x_{ij}),$$

where p^+ and p^- are defined as

$$p^+(x_{ij}) = \Pr(e_{ij} = 1) = \Pr(\text{gene } j \text{ is over expressed in sample } i),$$

$$p^-(x_{ij}) = \Pr(e_{ij} = -1) = \Pr(\text{gene } j \text{ is under expressed in sample } i).$$

and they satisfy the constraint

$$p^+(x_{ij}) + p^-(x_{ij}) + p^0(x_{ij}) = 1,$$

where

$$p^0(x_{ij}) = \Pr(e_{ij} = 0) = \Pr(\text{gene } j \text{ is normally expressed in sample } i),$$

and e_{ij} represents the latent categories where the raw expression x_{ij} falls into. Then $p^*(x_{ij})$ is a signed probability of differential expression for gene j in sample i. As it ranges in $[-1, 1]$, whatever the scale of the gene expression is, POE provides a unified measure across studies.

Then, the expression of gene j arises from a mixture of three distributions:

$$(x_{ij}|e_{ij} = 1) \sim f_{1,j}(\cdot), \quad (x_{ij}|e_{ij} = 0) \sim f_{0,j}(\cdot), \quad \text{and} (x_{ij}|e_{ij} = -1) \sim f_{-1,j}(\cdot),$$

with mixture probabilities π_j^+, π_j^0, and π_j^{-1} ($\pi_j^+ + \pi_j^0 + \pi_j^{-1} = 1$). $f_{1,j}$, $f_{0,j}$ and $f_{-1,j}$ are assumed to be the following density functions

$$f_{1,j} = \mathrm{U}(\alpha_\mathrm{i} + \mu_\mathrm{j}, \alpha_\mathrm{i} + \mu_\mathrm{j} + \kappa_\mathrm{j}^+),$$

$$f_{0,j} = \mathrm{N}(\alpha_\mathrm{i} + \mu_\mathrm{j}, \sigma_\mathrm{j}^2),$$

$$f_{-1,j} = \mathrm{U}(\alpha_\mathrm{i} + \mu_\mathrm{j} - \kappa_\mathrm{j}^-, \alpha_\mathrm{i} + \mu_\mathrm{j}).$$

The latent probabilities can be derived by Bayes Rule:

$$P_{ij}^+(e_{ij} = 1|x_{ij}) = \frac{\pi_j^+ f_{1,j}(x_{ij})}{\pi_j^+ f_{1,j}(x_{ij}) + \pi_j^- f_{-1,j}(x_{ij}) + (1 - \pi_j^+ - \pi_j^-) f_{0,j}(x_i j)}$$

$$P_{ij}^-(e_{ij} = 1|x_{ij}) = \frac{\pi_j^- f_{-1,j}(x_{ij})}{\pi_j^+ f_{1,j}(x_{ij}) + \pi_j^- f_{-1,j}(x_{ij}) + (1 - \pi_j^+ - \pi_j^-) f_{0,j}(x_i j)}$$

To combine the information from multiple data sets, the parameters (μ_j, σ_j^2, κ_j^+, κ_j^-, π_j^+, π_j^-) in the distributions are set to follow some common prior distributions. The posterior distribution of the parameters are approximated by Metropolitan Hastings (MCMC) algorithm and Choi et. al (2007) proposed a faster EM algorithm.

This method is only applicable to cases when all studies have the same conditions. For the prostate cancer example discussed before, some studies have two conditions and others have three. It is not clear how to apply this method.

Warnat et al. (2005) propose Median Rank Scores (MRS, Toedling and Spang, 2003) and Quantile discretization (QD) to change the microarray expressions into a common scale. It replaces the expression by the median rank scores to make the multiple data sets comparable to each other. One data set is picked as the reference and the median expression for every gene is calculated. Usually the largest data set is selected as the reference set unless the data quality is poor. This method keeps only the expression order for the non-reference data set. Also, it is not clear whether the selection of reference set will lead to different results.

Shabalin et al. (2008) discuss a cross platform normalization based on a block linear model. The expression value of gene g of replicate sample r in study s, is

decomposed by

$$x_{sgr} = A_{\alpha^*(g),\beta_s^*(r),s} \times b_{gs} + c_{gs} + \sigma_{gs}\epsilon_{sgr},$$

where $\alpha^* : \{1,\ldots,G\} \longmapsto \{1,\ldots,K\}$ maps a total of G genes into K clusters; $\beta_s^* : \{1,\ldots,n_s\} \longmapsto \{1,\ldots,L\}$, $s = 1,\ldots,S$ defines groups of sample replicates for study $s(s = 1,\ldots,S)$. A_{ijs} is the block mean; b_{gs} and c_{gs} capture sensitivity and offset effect, respectively, associated with gene and platform; ϵ_{sgr} is the independent standard normal noise term. The estimate of the decomposition equation is done by a two step procedure.

1.2.2 Methods of Combining Individual Results

Choi et al. (2003) propose to combine the effect size of genes from each study and conduct a permutation test to determine the significance level. A hierarchical model is used to quantify the effect size for every gene in the studies

$$y_{gs} = \theta_s + \epsilon_s, \qquad \epsilon_s \sim N(0, \sigma_s^2),$$

$$\theta_s = \mu + \delta_s \qquad \delta_s \sim N(0, \tau^2),$$

where y_{gs} is the effect size of gene g in study s; μ is the overall mean, representing the average differential expression across the datasets for each gene; σ_s^2 measures the within-study variation of the gene expression; τ^2 is the between-study variation.

Given microarrays in each study, y_{gs} and σ_s^2 are calculated respectively as

$$y_{gs} = \frac{\bar{x}_{gt} - \bar{x}_{gn}}{s_p},$$

$$\sigma_s^2 = (n_t^{-1} + n_n^{-1}) + y_{gs}^2(2(n_t + n_n))^{-1},$$

where $\bar{x}_{gt}$ and $\bar{x}_{gn}$ are the sample mean over the tumor and normal group respectively; s_p is the pooled standard deviation estimated.

Under this framework, a fixed-effect model interprets that the variation of effect size comes from sampling error only. And random-effects model assumes that the

effect size follows a distribution whose mean parameter θ_s is randomly drawn from another distribution $N(0, \tau^2)$. The test of $\tau^2 = 0$ is applied to decide which model fits the data sets better. For each gene, a z-score of the average effect size is obtained to evaluate whether the gene is significantly differentially expressed or not. The threshold of z-score is determined by repeatedly calculating the z-scores based on the permutation within each study.

Rhodes et al. 2002 consider combining p-values of the studies. For each study, the gene specific actual t-statistic is calculated. A permutation is conducted to randomly assign the sample labels to the gene expressions, such that the t-statistic can be calculated many times, which is called random t-statistic. p-value is calculated as the fraction of the random t-statistic greater than or equal to the actual t-statistic. Such summary statistics is computed for every gene present in all studies and the corresponding p-value is estimated by a permutation to get the random summary statistics. Although this method integrates the result of individual studies, it is difficult to control false negatives. Due to high variation in the experimental process studies, the obtained p-values are noisy. One single poor p-value may deteriorate the combined statistic. The combined statistic is not a robust summary to the p-values. As a result, the test will lead to a contrary conclusion.

To jointly consider combining individual results while keeping False Discovery Rate (FDR) under control, Pyne et al. 2006 propose to combine only those good p-values which clear their respective experiment-specific false discovery thresholds. As the quality of all experiments is not homogenous, experiments should contribute differently to the integrated statistic. A weighting scheme is presented to combine the individual results. How to select the threshold to screen the good p-values is also discussed.

1.3 Model and Inference

1.3.1 Parametric Empirical Bayes Model for A Single Study

We begin with modeling gene expression data from a single study. Various methods have been developed for such purposes. Interested readers are referred to Parmigiani et al. (2003), Allison et al. (2006) and Do et al. (2006) for recent surveys. Here we adopt a parametric empirical Bayes approach introduced by Newton et al. (2001) and Kendziorski et al. (2003).

Let x_{gcr} be the gene expression measurement taken from the rth replicate under condition c for gene g. Take the data from Dhanasekaran et al. (2001) as an example, three biological conditions ($c = 1, 2$, or 3), namely benign prostate, localized prostate cancer or metastatic prostate cancer; 4,839 genes ($g = 1, 2, \ldots, 4839$) are considered. A total of 14 replicates ($r = 1, 2, \ldots, 14$) are obtained for benign prostate; 14 for localized and 20 for metastatic prostate cancer respectively.

To fix ideas, we focus on two conditions ($c = 1$ or 2) in what follows. Sensible expression patterns concerning the comparison between two conditions for a particular gene include equivalent expression and differential expression. This can be formulated through latent variables μ_{gc} representing a population level of expression for gene g under biological condition c. Equivalent expression means that $\mu_{g1} = \mu_{g2}$ whereas differential expression indicates $\mu_{g1} \neq \mu_{g2}$. Our goal is therefore to infer such expression patterns from $\mathbf{x}_{g1\cdot} = (x_{g11}, x_{g12}, \ldots, x_{g1n_1})$ and $\mathbf{x}_{g2\cdot} = (x_{g21}, x_{g22}, \ldots, x_{g2n_2})$ where n_1 and n_2 are the number of replicates obtained under each condition respectively. It is not hard to see that the marginal distribution of $(\mathbf{x}_{g1\cdot}, \mathbf{x}_{g2\cdot})$

$$
\begin{aligned}
f\left(\mathbf{x}_{g1\cdot}, \mathbf{x}_{g2\cdot}\right) &= P(\mu_{g1} = \mu_{g2}) f\left(\mathbf{x}_{g1\cdot}, \mathbf{x}_{g2\cdot} \middle| \mu_{g1} = \mu_{g2}\right) && (1) \\
&\quad + P(\mu_{g1} \neq \mu_{g2}) f\left(\mathbf{x}_{g1\cdot}, \mathbf{x}_{g2\cdot} \middle| \mu_{g1} \neq \mu_{g2}\right) && (2)
\end{aligned}
$$

where we use f to denote a generic density function, marginal or conditional; and $P(\mu_{g1} = \mu_{g2}) + P(\mu_{g1} \neq \mu_{g2}) = 1$. The two conditional distributions can be modeled

through a two level hierarchical model:

$$f\left(\mathbf{x}_{g1\cdot}, \mathbf{x}_{g2\cdot} \middle| \mu_{g1} = \mu_{g2}\right) = \int \left(\prod_{k=1}^{n_1} f\left(x_{g1k} \middle| \mu_{g1} = \mu; \theta\right)\right) \left(\prod_{k=1}^{n_2} f\left(x_{g2k} \middle| \mu_{g2} = \mu; \theta\right)\right) f(\mu; \tau) d\mu;$$

$$f\left(\mathbf{x}_{g1\cdot}, \mathbf{x}_{g2\cdot} \middle| \mu_{g1} \neq \mu_{g2}\right) = \int \left(\prod_{k=1}^{n_1} f\left(x_{g1k} \middle| \mu_{g1}; \theta\right)\right) \times \times \left(\prod_{k=1}^{n_2} f\left(x_{g2k} \middle| \mu_{g2}; \theta\right)\right) f(\mu_{g1}; \tau) f(\mu_{g2}; \tau) d\mu_{g1} d\mu_{g2},$$

where θ and τ are parameters shared by all genes and determined by the experiment characteristics.

Two particular choices of $f(\cdot|\mu;\theta)$ and $f(\cdot;\tau)$ are advocated, often referred to the lognormal-normal (LNN) model and Gamma-Gamma (GG) model. In the LNN model, $f(\cdot|\mu;\theta)$ is a lognormal distribution, i.e.,

$$f(x|\mu;\theta) = \frac{1}{\sqrt{2\pi\theta}} \exp\left(-\frac{(\ln x - \mu)^2}{2\theta}\right); \tag{3}$$

whereas $f(\cdot;\tau)$ is also a normal distribution with $\tau = (\tau_1, \tau_2)'$ represents the mean and variance parameter respectively. Alternatively for the GG model, $f(\cdot|\mu;\theta)$ is a Gamma distribution, i.e.,

$$f\left(x|\mu;\theta\right) = \frac{\lambda^\theta}{\Gamma(\theta)} x^{\theta-1} \exp\left\{-\lambda x\right\} \tag{4}$$

where the shape parameter is given by $\lambda = \theta/\mu$. $f(\cdot;\tau)$ is chosen such that λ also follows a Gamma distribution

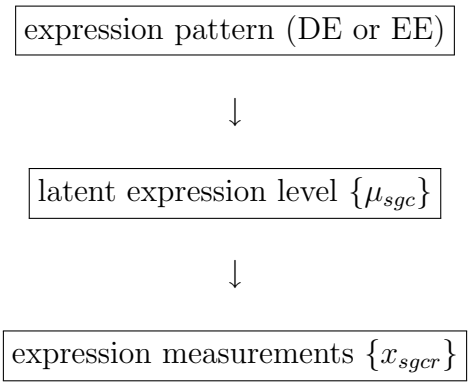
$$f(\lambda;\tau_1,\tau_2) = \frac{{\tau_2}^{\tau_1}}{\Gamma(\tau_1)} \lambda_g^{\tau_1-1} \exp\left\{-\tau_2\lambda\right\}. \tag{5}$$

Closed form expression are available for $f\left(\mathbf{x}_{g1\cdot}, \mathbf{x}_{g2\cdot} \middle| \mu_{g1} = \mu_{g2}\right)$ and $f\left(\mathbf{x}_{g1\cdot}, \mathbf{x}_{g2\cdot} \middle| \mu_{g1} \neq \mu_{g2}\right)$ with both LNN and GG models. The readers are referred to Kendziorski et al. (2003) for further details.

1.3.2 Joint Modeling with Multiple Studies

We now consider multiple studies. For brevity, we shall first assume that in each study, the same set of genes ($g = 1, 2, \ldots, G$) and the same set of conditions ($c = 1, 2, \ldots, C$)

are considered. This assumption will later be relaxed. With slight abuse of notation, let $\mathbf{X}_s := \{x_{sgcr} : g = 1, \ldots, G; c = 1, \ldots, C; r = 1, \ldots, n_{sc}\}$ be the gene expression measurements obtained in the sth study ($s = 1, 2, \ldots, S$) where n_{sc} is the number of replicates under condition c in the study. Clearly $\mathbf{X}_s$ can be modeled using the parametric empirical Bayes model discussed before. The hierarchical modeling can be summarized by the diagram below:



The latent expression levels are determined stochastically by the expression pattern through distribution $f(\mu; \tau)$ whereas the expression measurement by the latent levels through conditional distribution $f(x|\mu; \theta)$. Both parameters θ and τ reflects the variation of an experimental data and therefore should be experiment-dependent, which conveniently addresses the issue that microarrays may come from different platforms. To this end, we shall write θ_s and τ_s in what follows to emphasize the dependence between these parameters and the study. On the other hand, given that the same biological process is studied, a gene's differential expression pattern should remain the same across all studies. Let $\mathbf{x}_{\cdot gc\cdot} = \{\mathbf{x}_{sgcr} : s = 1, \ldots, S; r = 1, \ldots, n_{sc}\}$ be the collection of all expression measurements obtained from all studies on gene g and condition c. Then the conditional distribution of these measurements under the

two differential expression patterns can be given by

$$f\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot} \middle| \mathrm{DE}\right) = \prod_{s=1}^{S} f\left(\mathbf{x}_{sg1\cdot}, \mathbf{x}_{sg2\cdot} \middle| \mathrm{DE}\right); \tag{6}$$

$$f\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot} \middle| \mathrm{EE}\right) = \prod_{s=1}^{S} f\left(\mathbf{x}_{sg1\cdot}, \mathbf{x}_{sg2\cdot} \middle| \mathrm{EE}\right), \tag{7}$$

where the experiment specific conditional distributions are given in the previous subsection.

1.3.3 Empirical Bayes Inference

If the experiment specific parameters θ_s and τ_s, $s = 1, \ldots, S$ are known, inference on a gene's expression pattern can be conducted through their posterior probabilities, i.e.,

$$f\left(\mathrm{DE} \middle| \mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot}\right) = \frac{\pi f\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot} \middle| \mathrm{DE}\right)}{\pi f\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot} \middle| \mathrm{DE}\right) + (1-\pi) f\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot} \middle| \mathrm{EE}\right)} \tag{8}$$

where $\pi = P(\mathrm{DE})$ is the probability that a randomly selected gene is differentially expressed. According to Bayes rule, we classify a gene as differentially expressed if the posterior probability of differential expression is greater than 50% and equivalent expression otherwise. These posterior probabilities provide a natural means of inferring differential expression by integrating multiple studies.

Following Efron et al. (2001) and Newton et al. (2001), parameters $\{\theta_s, \tau_s : s = 1, \ldots, S\}$ as well as π can be estimated in an empirical Bayes fashion. Note that these parameters are shared by all genes. The log-likelihood for all data can then be given by

$$\ell\left(\mathbf{x}_{\cdot\cdot 1\cdot}, \mathbf{x}_{\cdot\cdot 2\cdot}\right) = \sum_{g=1}^{G} \ell\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot}\right)$$

where

$$\ell\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot}\right) = \log\left((1-\pi) f\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot} \middle| \mathrm{EE}\right) + \pi f\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot} \middle| \mathrm{DE}\right)\right),$$

The maximum likelihood estimator of all parameters θ_s and τ_s, $s = 1, \ldots, S$ and π can be efficiently computed using EM algorithm by treating a gene's differential expression pattern (i.e, EE or DE) as missing.

Instead of imposing a hierarchical model specifying the hyper parameters for the study-based parameters, the proposed modeling scheme fits a separate EB model for each study. It allows flexibility of accommodating different chip types. And it also encourages information sharing from every study based on the fact that each gene follows one particular pattern, which is common across all studies.

1.3.4 Gene Set Mismatch and Missing Data

As mentioned in Section 1, one of the most common difficulties associated with joint analysis is the mismatch of gene sets. Due to various limitation of the technology and quality control, valid measurements obtained in one data set may not be available for another data set. In practice, only those genes with valid measurement across all experiments are included in the joint analysis. This can be a significant loss of information as we shall see in the prostate cancer data in Section 4 where 30% to 75% of the data from each experiment are wasted if this approach is taken. In contrast, the problem of gene set mismatch can be conveniently addressed within our framework. Rather than considering only genes that are present in all experiment, we include all genes that appears in at least one experiment. If a particular gene is not present in an experiment, we treat it as missing data. In particular,if one gene is missing in one study, it does not contribute to the likelihood (2) and as a result (6) and (7) only collects contributions from the studies with the gene present.

1.3.5 Multiple Conditions and Condition Mismatch

The proposed framework for joint analysis can be easily extended to handle more than two conditions. Consider, for example, the data taken from Dhanasekaran et al. (2001) where three biological conditions are investigated. For each condition, we

introduce a latent gene expression level, μ_{sgc}, $c = 1, 2$ or 3. When comparing these conditions for gene g, we have the following equality or inequality conditions that may hold:

$$\begin{aligned}
\text{Pattern 1}: &\quad \mu_{sg1} = \mu_{sg2} = \mu_{sg3}, \\
\text{Pattern 2}: &\quad \mu_{sg1} = \mu_{sg2} \neq \mu_{sg3}, \\
\text{Pattern 3}: &\quad \mu_{sg1} \neq \mu_{sg2} = \mu_{sg3}, \\
\text{Pattern 4}: &\quad \mu_{sg1} = \mu_{sg3} \neq \mu_{sg2}, \\
\text{Pattern 5}: &\quad \mu_{sg1} \neq \mu_{sg2} \neq \mu_{sg3}. \qquad (9)
\end{aligned}$$

Similar to before, these latent expression level can be modeled by an experiment-specific distribution $f(\mu; \tau_s)$ where under Pattern 1, all three latent expression levels are obtained as a single sample from $f(\cdot; \tau_s)$; under Pattern 2, $\mu_{sg1} = \mu_{sg2}$ and μ_{sg3} are two independent samples from $f(\cdot; \tau_s)$ and so on. Similar formula as before can therefore be derived for $f(\mathbf{x}_{\cdot gc\cdot}|\text{Pattern } k)$:

$$f\left(\mathbf{x}_{\cdot g1\cdot}, \mathbf{x}_{\cdot g2\cdot}, \mathbf{x}_{\cdot g3\cdot}\middle|\text{Pattern } k\right) = \prod_{s=1}^{S} f\left(\mathbf{x}_{sg1\cdot}, \mathbf{x}_{sg2\cdot}, \mathbf{x}_{\cdot g3\cdot}\middle|\text{Pattern } k\right),$$

where the conditional densities can be computed and the inferences can also be conducted in a similar fashion as before.

A practical challenge that often arises with multiple biological conditions is the possible condition mismatch. Different experiments are designed to address and compare different but overlapping conditions. The overlap in biological conditions makes information sharing possible but the difference in biological conditions makes the information sharing difficult. For example, among the four prostate cancer studies we discussed earlier in the introduction, Dhanasekaran et al. (2001) considered three conditions including benign prostate, localized prostate cancer and metastatic prostate cancer; whereas Luo et al. (2001), only investigated the first two conditions. A common practice is to ignore data obtained under the third condition from Dhanasekaran

et al. (2001) and compare the first condition through a joint analysis. Although a convenient and sensible solution, it is clearly not the most efficient way of using data. In general, following this practice, when including multiple studies, we can only use those conditions that are present in all studies. Furthermore, as we shall demonstrate by simulations in the next section, doing so may result in loss of efficiency as well.

The problem of condition mismatch can be handled conveniently within our proposed framework of joint analysis. For illustration purpose, we assume the one study has three conditions but the other one missed the third condition. As shown in (9), we can see that the joint of the first two "sub-patterns" is exactly EE pattern between condition one and two, and we call them sub-pattern of EE; and similarly the last three is DE sub-pattern for the two conditions. For the study with only first two conditions, it can get the posterior probability of EE and DE as discussed before. Then it estimates the posterior probability of the two EE sub-patterns by equally dividing the EE probability and similarly evenly divides the DE probability to estimate the probability for the three DE sub-patterns.

1.4 Simulation Studies

1.4.1 Benefit of Joint Analysis

To demonstrate the effectiveness of the proposed method, we first conducted several sets of simulation studies. To demonstrate the benefit of joint analysis, we begin with a simple setting: two biological conditions, and no gene set mismatch. A total of $G = 5,000$ genes and $S = 4$ experiments were simulated. For each experiment, $n_{sc} = 3$ replicates were simulated under each condition. The gene expression data were simulated from LNN or GG model. Due to their similarity in performance, we report here only the results from LNN models. The simulation settings for each experiment are similar to those previously employed by Kendziorski et al. (2003) to mimic the real gene expression data and represent different experimental variations in practice.

Denote $\eta = (\tau_1, \log(\tau_2), \log(\theta))$ the parameters associated with the LNN model. The parameters of the four experiments are set at $\eta_1 = (2, 0.5^2, 0.15^2)$, $\eta_2 = (5, 0.6^2, 0.25^2)$, $\eta_3 = (15, 1^2, 0.35^2)$, $\eta_4 = (30, 1.2^2, 0.45^2)$ respectively. τ_2 controls the variation of the latent mean of the gene expression levels. Larger τ_2 corresponds to DE genes better separated between conditions. And this parameter setting has an average effect size of 1.62, 1.8, 2.64 and 3.23 (calculated as the median of the effect sizes of differential expressions) for the four studies respectively. A randomly chosen $\pi = 10\%$ genes are set to be differentially expressed. We compare the performance of the proposed method with two joint analysis methods, Choi et al. (2003) and Choi et al. (2007), and separate analysis based on the area under curves (AUC). We also compare the our results and the separate analysis with the criteria of sensitivity, specificity and FDR.

The performance of the four separate analyses are combined to compare with the joint analysis based on the rule that every DE gene claimed by any one separate analysis is regarded as DE genes. And in the separate analysis, each experiment is analyzed separately using the empirical Bayes approach of Kendziorski et al. (2003), referred to as EBarrays. We also compare sensitivity, specificity and false discovery rate (FDR) with separate analysis. Note that these measures are calculated based on a natural threshold of 0.5 by Bayes rule in the EB framework but they can not be evaluated at the same stage in Choi et al. (2003) and Choi et al. (2007) because they require an arbitrary confidence level α, which is not equivalent to the Bayes rule threshold. The operating characteristics based upon 100 runs are summarized in Table 1.

We observe that joint analysis can significantly improve the performance. Although four separate analyses have high AUC, they have high false discovery rate as reported in Table (1). Among the four experiments, Experiment 4 has the strongest signal to noise ratio, also reflected by its largest effect size of differential expressions.

Table 1: Operating characteristics of joint analysis and separate analysis. The results are summarized from 100 runs. All units are in percentages and $(\cdot)$ represents standard error.

	AUC	Sensitivity	Specificity	FDR
Joint EB	99.97 (0)	99.38 (0.03)	99.99 (0)	0.12 (0.02)
Separate EB	99.74 (0.01)	93.92 (0.13)	99.99 (0)	0.07 (0.01)
Choi et al. (2003)	58.74 (0.14)			
Choi et al. (2007)	57.74 (0.47)			

A possible misconception is that it is fruitless to combine such a good-quality experiment with others with relatively poor quality. Our result clearly suggests otherwise. It indicates that joint analysis can greatly improve even the experiment with the best quality.

To evaluate the robustness of the proposed method, we consider a more complex simulation setup where the experimental data were generated as follows:

Experiment 1: The latent gene expression levels were simulated from an inverse Gamma distribution with shape parameter 2 and location parameter 10. Then the gene expression measurement were simulated from a Gamma distribution with the latent means and shape parameter 20.

Experiment 2: The latent means were simulated so that $A := \log((\mu_{2g1}\mu_{2g2})^{1/2})$ follows a uniform distribution between 5 and 11; and $M = \log(\mu_{2g1}/\mu_{2g2})$ follows a uniform distribution between -1 and 1 for differentially expressed genes and 0 for equivalently expressed genes. Then the observed gene expression measurements were simulated from Gamma distribution with shape parameter 15.

Experiment 3: Similar to Experiment 2 except that now M follows a uniform distribution between -2 and 2 and the expression measurements were simulated with shape parameter 25.

Experiment 4: Data were simulated from a LNN model with parameter $\theta = \exp(0.3)$ and $\tau = (2.3, \exp(1.39))$.

The other setting are similar as before. The effect sizes of the four studies are 2.09, 1.65, 2.71 and 7.69 respectively. To gain further insights, we also consider three different percentages of differential expression: $\pi = 5\%, 10\%$ and 20%. Figure 1 summarizes the results, again averaged over 100 runs. Joint four analysis has much lower false discovery rate (0.2%) than the combined separate analysis (10%). The sensitivity and specificity of joint analysis $(93\%, 99\%)$ are similar to the combined separate analysis $(95\%, 97\%)$. AUC of the proposed method, combined separate analysis, Choi et al. (2003) and Choi et al. (2007) are not sensitive to π and is similar to Table 1. It is evident that the joint analysis significantly outperforms all others.

Table 2: Operating characteristics of joint analysis and separate analysis with differential probability fixed at 0.05, 0.1 and 0.2 respectively. The results are summarized from 100 runs. All units are in percentages and $(\cdot)$ represents standard error.

$\pi = 0.05$	AUC	Sensitivity	Specificity	FDR
Joint EB	99.63 (0.02)	92.33 (0.16)	99.99 (0)	0.25 (0.03)
Separate EB	98.56 (0.04)	94.17 (0.16)	99.46 (0.01)	9.69 (0.2)
Choi et al. (2003)	61.42 (0.19)			
Choi et al. (2007)	57.94 (0.24)			
$\pi = 0.1$	AUC	Sensitivity	Specificity	FDR
Joint EB	99.61 (0.02)	93.05 (0.11)	99.98 (0)	0.24 (0.02)
Separate EB	98.48 (0.03)	95.51 (0.09)	98.72 (0.02)	10.75 (0.15)
Choi et al. (2003)	61.11 (0.14)			
Choi et al. (2007)	58.86 (0.16)			
$\pi = 0.2$	AUC	Sensitivity	Specificity	FDR
Joint EB	99.59 (0.01)	93.96 (0.07)	99.95 (0)	0.2 (0.01)
Separate EB	98.49 (0.02)	96.89 (0.06)	96.74 (0.04)	11.88 (0.12)
Choi et al. (2003)	60.92 (0.12)			
Choi et al. (2007)	60.15 (0.13)			

1.4.2 Gene Set Mismatch and Missing Data

We now consider the problem of gene set mismatch and missing data. To this end, we consider the following simulation scheme with a total of $G = 5,000$ genes at two conditions. The proportion of DE genes is 5%. Similar to before, three replicates were simulated at every condition. Because of the robustness of the method, we focus here

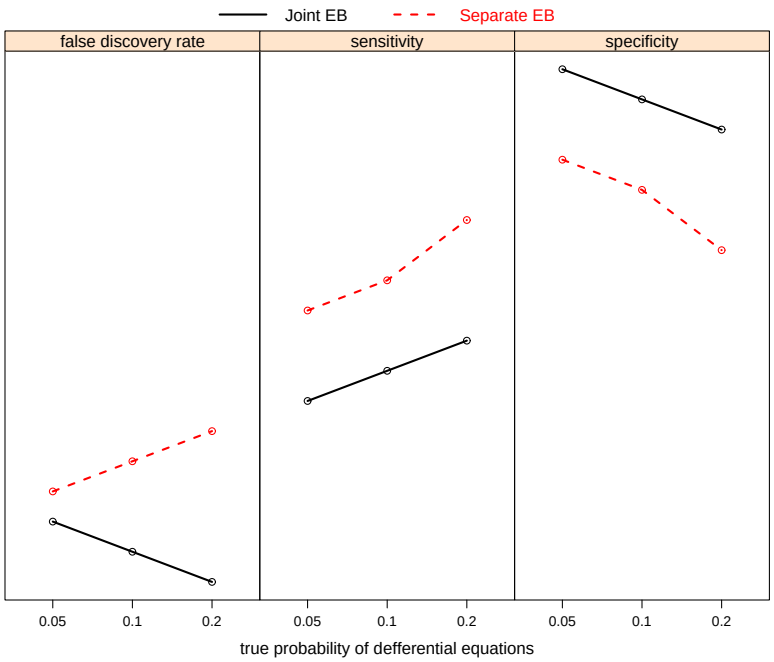


Figure 1: Performance of joint and separate analyses, the y-axis in each sub-figure is false discovery rate, sensitivity and specificity respectively.

only on the LNN model with the parameters given before. The difference is now each experiment only involves a subset of the genes. In particular, Experiment 1 includes $4,500$ randomly selected genes; and each of the remaining three experiments has 80% overlap with the first experiment and the set of overlapping genes is drawn randomly. In addition, Experiments 2 and 3 each has 250 new genes randomly selected from the 500 genes not included in Experiment 1. Experiment 4 covers all 500 genes not available in Experiment 1. As a result, Experiments 2 and 3 each has 3850 genes whereas Experiments comprises of 4100 genes. To mimic missing data for all studies, we randomly incorporate 10% of genes with missing data. In particular, those genes have a mild missing data issue, i.e., they evenly have 1, 2, 3, 4, 5 observations missing and the scheme is done randomly.

As we can see from Table 3 based on 100 simulation runs, when the data quality is okay, the new method beats others over almost all performance characteristics.

Table 3: Performance comparison among joint, combined separate analysis and two meta analyses Choi et al. (2003) and Choi et al. (2007), when there are mild gene set mismatch and missing observations. All units are in percentages and $(\cdot)$ represents the standard error.

	AUC	Sensitivity	Specificity	FDR
Joint EB	98.92 (0.04)	87.42 (0.24)	99.9 (0)	2.01 (0.1)
Separate EB	93.07 (0.12)	55.23 (0.32)	100 (0)	0.02 (0.01)
Choi et al. (2003)	60.77 (0.34)			
Choi et al. (2007)	60.02 (0.36)			

1.4.3 Condition Mismatch

Our final simulation study is designed to illustrate the effect of condition mismatch. We adopt a similar simulation set as before, with 5000 genes and four experiments. There are a total of three biological conditions but one condition is missing at each of the first three experiments. Specifically, the first experiment has three replicates under the first condition, three under the second condition, none under the third condition. The second experiment has three replicates under each of the first and third condition, but none under the second condition. The third experiment features three replicates under each of the second an third condition, and none under the first condition. The last experiment has three replicates under each of the three conditions. As we pointed out earlier, such condition mismatch is a direct consequence of different biological hypothesis of interest. In Experiment 1, our interest is in comparing the first two conditions. The goal is therefore to determine genes that are differentially expressed between these two conditions. Similarly, in Experiment 2, we want to identify genes that are differentially expressed between the first and third condition; and Experiment 3, between the second and third condition. In the last experiment, there are five possible patterns as we discussed before, all patterns except for Pattern 1 can be identified as differential expression.

Given the different hypotheses, the natural question is whether or not a joint analysis of all four experiments can be beneficial. For example, for the "investigators"

of the first experiment, combining with data on the first two conditions from the last experiment might be helpful, but it is not immediately clearly whether or not it helps if we include all four experiments. To illustrate the merits of the proposed joint analysis of all experiments, we apply three different strategies here: separate analysis of the first experiment; joint analysis of the first experiment and the last experiment with data from the third condition discarded; and the proposed method of joint analysis of all four experiments with missing conditions handled as missing data as we discussed before. Since joint analysis of other methods cannot handle condition mismatch and their analysis based on throwing the unmatch condition reduces to the situation evaluated before, we do not compare with them in this setting.

Table 4 summarizes the operating characteristics of all three methods averaged over 100 runs. It is clear that both joint analyses improve upon the separate analysis with the proposed method outperforms the joint analysis with only two experiments. Similar comparisons were conducted from the angles of the "investigators" of Experiments 2 and 3 and the results remain similar. Now consider the last experiment where the goal is identify differentially expressed genes among all three conditions. We compare the joint analysis that uses data from all four experiments and the individual analysis that only uses data from the last experiment. The results are also given in Table 4, which suggests that joint analysis gives superior performance.

1.5 Real Examples

To further illustrate the merits of the proposed method, we now return to the prostate cancer examples discussed before. As mentioned earlier, four public microarray datasets generated independently by Dhanasekaran et al. (2001), Luo et al. (2001), Magee et al. (2001) and Welsh et al. (2001) were collected to determine genes that are differentially expressed between benign prostate and cancer tumors. As stated before, the data were generated with different platforms: Dhanasekaran et al. (2001)

Table 4: Performance comparison of separate analysis and joint analysis with condition mismatch. All unit are in percentages and $(\cdot)$ is standard error.

DE in Conditions 1 and 2					DE in Conditions 1 and 3				
	Sensitivity	Specificity	FDR	AUC		Sensitivity	Specificity	FDR	AUC
Exp 1	63.16	99.54	36.84	90.4	Exp 2	52.37	99.42	47.63	87
	(0.2)	(0.01)	(0.2)	(0.08)		(0.21)	(0.01)	(0.21)	(0.08)
Exp 1 & 4	83.72	99.64	16.28	97.59	Exp 2 & 4	79.28	99.52	20.72	96.75
	(0.14)	(0.01)	(0.14)	(0.04)		(0.15)	(0.01)	(0.15)	(0.04)
All Exp	89.25	99.51	10.75	99.02	All Exp	87.37	99.37	12.63	98.8
	(0.12)	(0.01)	(0.12)	(0.02)		(0.15)	(0.01)	(0.15)	(0.02)
DE in Conditions 2 and 3					DE among three conditions				
	Sensitivity	Specificity	FDR	AUC		Sensitivity	Specificity	FDR	AUC
Exp 3	58.63	99.49	41.37	89.03	Exp 4	64.35	99.32	35.65	90.63
	(0.21)	(0.01)	(0.21)	(0.09)		(0.16)	(0.02)	(0.16)	(0.06)
Exp 3 & 4	81.99	99.58	18.01	97.21					
	(0.14)	(0.01)	(0.14)	(0.04)					
All Exp	88.49	99.45	11.51	98.93	All Exp	92.31	99.72	7.69	99.11
	(0.11)	(0.01)	(0.11)	(0.02)		(0.08)	(0.01)	(0.08)	(0.02)

and Luo et al. (2001) employed spotted cDNA microarrays whereas the other two experiments utilized Affymetrix technology. All four studies include comparisons between locally advanced prostate cancer and benign prostate. Dhanasekaran et al. (2001) and Magee et al. (2001) also included a third biological condition: metastatic prostate cancer. A total of $13,474$ unique genes are present in at least one of the experiment. There is, however, a severe gene set mismatch among the four experiments with less than 10% $(1,322)$ of the genes presented in all four experiments. Among the 1,322 intersection genes, only one gene does not have missing observations. Methods such as Choi et al. (2003) and Choi et al. (2007) applicable to complete data do not work in this case. So we only compare joint and combined separate analyses. Table 5 summarizes some basic information of the data and gives the number of genes overlapped between the four experiments.

Table 5: Basic information of the four prostate cancer datasets. D – data from Dhanasekaran et al. (2001); L – data from Luo et al. (2001); M – data from Magee et al. (2001); and W – data from Welsh et al. (2001).

	Array	Number of Replicates			Pairwise Overlap Genes			
	Type	Benign	Local PCA	Metastatic PCA	D	L	M	W
D	cDNA	14	14	20	4,839	2,642	1,596	2,126
L	cDNA	9	16	0		6,109	2,895	3,574
M	Affy	4	8	3			5,228	4,963
W	Affy	9	23	0				9,071

We ran the joint analysis both with the LNN and GG model and the results are similar. Therefore, we focus here on the results from the LNN model. Similar to the simulation study conducted before, there are two primary hypotheses concerning differential expression. The goal is to identify genes that are differentially expressed between either cancer tumor and benign prostate. In other words, among the five expression patterns given in (9), we are interested in identifying genes in Patterns 2, 3, 4 and 5 as opposed to Pattern 1. Hereafter, we shall refer to genes with Pattern

1 equivalently expressed genes; and the genes with other patterns differentially expressed genes. Similar to earlier studies (see,e.g., Choi et al., 2003), a large number of genes demonstrate significant difference between prostate cancer and benign prostate. To fix idea, we focus on the top one hundred genes identified to follow Patterns 2, 3, 4 or 5 by joint analysis. All of these genes have posterior probabilities of differential expression greater than 99%. Among these genes, 31 genes are not identified by any studies; 69 are identified to be differentially expression with posterior probability at least 95% in at least one of the fours studies when analyzing the four datasets separately; 34 in at least two studies; 7 in three studies; and 0 in all four studies. For the top 100 genes found by joint analysis, venn diagram in Figure 2 shows the availability in each of the individual analysis. In particular, for the 31 genes not identified by all separate studies, they are partly available at least two conditions in each data set (2 genes in Dhanasekaran et al. (2001), 7 genes in Luo et al. (2001), 17 genes in Magee et al. (2001), and 31 genes in Welsh et al. (2001)). The median effect size of the available genes in each study is 1.28, 1.19, 2.77, and 4.55 respectively. The effect size is calculated as the largest effect size of each pair conditions.

Joint analysis reveals significant genes agreed across studies more than would be expected by chance. Also, it improves sensitivity. Genes that are not identified by individual analysis can be discovered by joint analysis. We examine the 31 genes among the one hundred Table 6 is the information of the 31 genes identified by joint analysis but not to be differentially expressed in individual analysis. In particular, Hs.296638, is a known prostate differentiation factor.

A second example we consider here is the four liver cancer datasets from Choi et al. (2003). All data were generated at two biological conditions: normal and tumor tissues. The goal is to identify genes differentially expressed in normal and tumor tissues. The datasets are of relatively poor quality when compared with the prostate datasets and have been used earlier in Choi et al. (2003) primarily to demonstrate

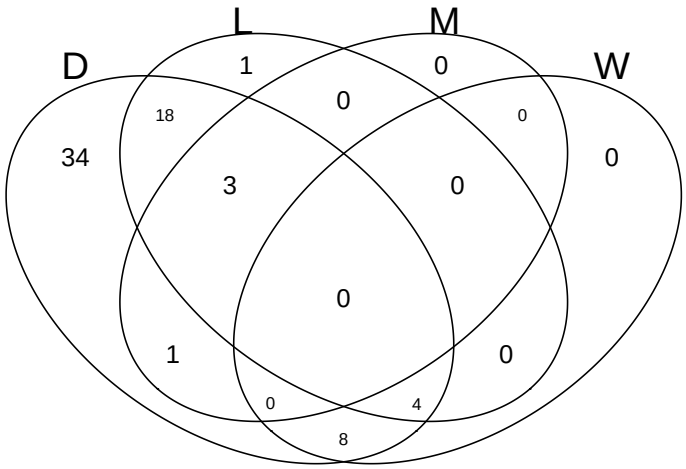


Figure 2: Venn diagram of the prostate cancer: 100 genes are selected as DE genes by joint analysis. Among them, 69 genes are detected as DE genes by any individual analysis and the intersection is the DE genes agreed by individual studies.

Table 6: In prostate cancer, list of 31 genes identified as differential expressed in joint analysis but not identified by any one study.

Unigene	name
Hs.34114	ATPase, Na+/K+ transporting, alpha 2 (+) polypeptide
Hs.296638	prostate differentiation factor
Hs.352107	trefoil factor 3 (intestinal)
Hs.334688	phytanoyl-CoA hydroxylase interacting protein
Hs.162209	claudin 8
Hs.118127	actin, alpha, cardiac muscle
Hs.54435	Human dystrobrevin (DTN) gene
Hs.2388	apolipoprotein F
Hs.56966	KIAA0906 protein
Hs.355723	aldehyde oxidase 1
Hs.34853	inhibitor of DNA binding 4, dominant negative helix-loop-helix protein
Hs.334703	hypothetical protein FLJ14529
Hs.20166	prostate stem cell antigen
Hs.173571	KIAA1053 protein
Hs.372612	for protein disulfide isomerase-related
Hs.49998	LIM domain binding 3
Hs.27695	midline 1 (Opitz/BBB syndrome)
Hs.237506	DnaJ (Hsp40) homolog, subfamily B, member 5
Hs.242271	KIAA0471 gene product
Hs.139336	ATP-binding cassette, sub-family C (CFTR/MRP), member 4
Hs.119498	thyroid hormone receptor interactor 6
Hs.194765	H.sapiens GENX-5624 mRNA, 3' UTR
Hs.348994	nidogen (enactin)
Hs.55279	serine (or cysteine) proteinase inhibitor, clade B (ovalbumin), member 5
Hs.278581	fibroblast growth factor receptor 2
Hs.153322	phospholipase C-like 1
Hs.167531	methylcrotonoyl-Coenzyme A carboxylase 2 (beta)
Hs.103839	erythrocyte membrane protein band 4.1-like 3
Hs.149098	smoothelin
Hs.153179	fatty acid binding protein 5 (psoriasis-associated)
Hs.114346	cytochrome c oxidase subunit VIIa polypeptide 1 (muscle)

the necessity of a joint analysis. Table 7 is some basic information of the data and gives the number of genes overlapped between the four experiments.

Table 7: Basic information of the four liver cancer datasets.

	Number of Replicates		Pairwise Overlap Genes			
	Normal	Tumor	D	L	M	W
D1	16	16	10314	10289	10194	9921
D2	23	23		10311	10202	9906
D3	29	5			10216	9815
D4	12	9				9931

Similarly, the analysis is based on the LNN model. In joint analysis, top 18 genes have posterior probability of differential equation greater than 90%. We evaluate the information of these genes and exclude genes of unknown functions. Then we get 10 genes. In particular, 9 out of 10 are identified by the combined separate analysis and 1 is failed to be selected. Table 8 shows the information of these genes. The median effect size of the 10 genes in every study is 2.09, 1.86, 1.78, and 1.9 respectively and Figure 3 displays the venn diagram.

Note that Choi et al. (2003) analyzed this data by joint analysis but the result there is not comparable to our result at the same stage because it evaluates DE genes by confidence level but our method is by posterior probability. So we can only compare the joint analysis and combined separate analysis under EB framework. Table (8) displays the gene list, where 9 out of 10 genes are identified by only one study. The first gene in Table (8) is not identified by any studies.

1.6 Conclusions

With the explosion of popularity of microarray experiments, it becomes a necessity to develop statistical methods that can effectively integrate data from multiple studies. Meta-analysis of multiple experiments can alleviate the low sample size and high variability problem that is often faced in individual studies. At the same time, meta-analysis also presents an unprecedented opportunity for comparative analyses

Table 8: In liver cancer, list of 10 genes identified as differential expressed in joint analysis but identified by at most one study. The first gene is failed to be selected by all studies.

Tissue	name
21.2.D.1	TATA box binding protein(TBP),mRNA
15.2.F.4	AL564975 cDNA
15.2.H.9	IL3-CT0219-271099-022-C02 cDNA
16.1.C.4	KIAA0107 gene product(KIAA0107),mRNA
19.4.D.12	KIAA0304 gene product(KIAA0304),mRNA
2.2.D.2	thioredoxin reductase 1(TXNRD1),mRNA
20.2.A.9	triosephosphate isomerase 1(TPI1),mRNA
22.3.A.4	ribosomal protein L13a(RPL13A),mRNA
23.3.A.4	CD24 antigen (small cell lung carcinoma cluster 4 antigen) (CD24), mRNA
7.1.E.7	hepatocyte growth factor regulated tyrosine kinase substrate (HGS), mRNA

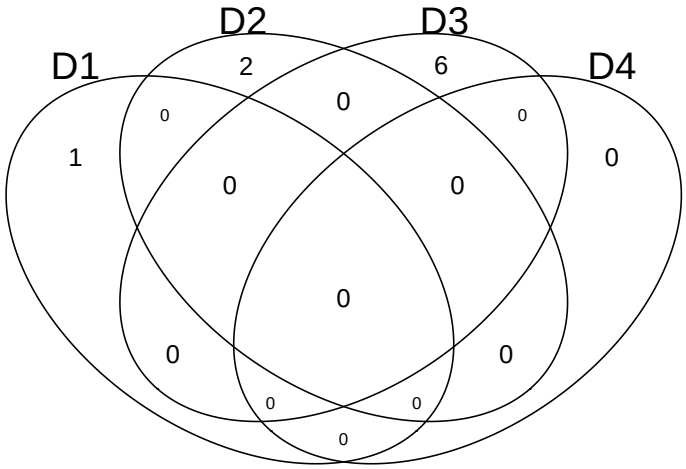


Figure 3: Venn diagram of the liver cancer: 10 genes are selected by joint analysis. In particular, 9 genes are found by any individual analysis and no common genes between individual studies.

of broad scope. In this paper, we propose a model-based joint analysis of gene expression data from multiple studies to determine differentially expressed genes between multiple biological conditions. The proposed method shares information both among genes within one study and across studies without data transformation. The method is flexible to handle various practical complications such as gene set mismatch and condition mismatch. Simulation studies and real data examples show that the accuracy of the statistical inferences can be drastically improved when using the proposed approach to combine multiple studies.

CHAPTER II

HIGH DIMENSIONAL COVARIANCE MATRIX ESTIMATION

2.1 Introduction

Covariance matrix or its inverse matrix is widely used in many statistical tools such as principal component analysis, classification of linear and quadratic discriminant analysis (LDA, QDA), multivariate normal studies, correlation inference in the graphical models. In particular, high dimensional covariance and inverse matrix have many applications in the actual world. For example, in financial data analysis, there are many stocks. To study the stock returns, the inverse matrix is needed to get the optimal portfolio and the covariance matrix is a measure to assess the risk returns. Other examples include gene analysis, image, climate data etc..

The usual sample covariance matrix, is not a good estimator when p is large. When p is moderately large, the sample covariance is ill conditioned and the inverse will amplify the error dramatically. If p is greater than the sample size n, it is not even invertible. Moreover, the eigenvalues of the sample covariance are more dispersed than the true covariance matrix (Ledoit and Wolfe, 2004). All these facts call an urgent demand for alternative estimators of the covariance matrix or its inverse.

There has been an abundance of existing literatures discussing various types of covariance matrix estimator. Some of them work on the estimator of covariance matrix and others study the inverse matrix directly, depending on the application purposes. Loosely speaking, these methods can be broadly divided into two categories. One is the estimators based on the sample covariance estimate. And the other class works on the likelihood function and they are penalised likelihood estimators. In this

chapter, we briefly review the main ideas of those alternative estimators.

2.2 Sample Covariance Based Estimators

Ledoit and Wolfe (2004) propose one estimator as a linear combination of the sample covariance matrix S and the identity matrix $\hat{\Sigma}_{LW} = \rho_1 I + \rho_2 S$. The coefficients are decided by the the optimization problem:

$$\begin{aligned} &\min_{\rho_1,\rho_2} = \mathrm{E}\|\hat{\Sigma}_{\mathrm{LW}} - \Sigma\|^2 \\ &\text{subject to } \hat{\Sigma}_{LW} = \rho_1 I + \rho_2 S; \end{aligned}$$

It turns out ρ_1 and ρ_2 are non-random numbers and the optimal solution is

$$\hat{\Sigma}_{LW} = \frac{b_n^2}{d_n^2} m_n I_n + \frac{a_n^2}{d_n^2} S$$

where

$$m_n = \mathrm{trace(S)/p}, \mathrm{d_n^2} = \|\mathrm{S} - \mathrm{m_n I}\|_{\mathrm{F}}^2,$$
$$b_n^2 = \min\left\{ d_n^2, \frac{1}{n}\sum_{i=1}^{n} \|(y_i - \bar{y})(y_i - \bar{y})'\|_F^2 \right\},$$

$a_n^2 = b_n^2 - d_n^2$, $\|\cdot\|_F$ is the Frobenius norm.

This approach only imposes shrinkage on the eigenvalues, not on eigenvectors. And the eigenvectors of sample covariance are not consistent when p increases (Johnstone and Lu, 2004).

Thresholding (Bickel and Levina 2008a, El Karoui 2008a) of the sample covariance matrix is defined by,

$$T_s(\hat{\Sigma}) = \hat{\sigma}_{ij} I(|\sigma_{ij}| \geq s) \tag{10}$$

Banding the sample covariance matrix (Bickel and Levina 2008b) is based on the fact that components far apart in the ordering have weak correlation. Denote $\hat{\Sigma} = (\hat{\sigma}_{ij})$ as the MLE of the covariance matrix Σ, define

$$B_k(\hat{\Sigma}) = \hat{\sigma}_{ij} I(|i - j| \leq k) \tag{11}$$

as the banded estimate. It is ideal when the truth is $\sigma_{ij} = 0$, if $|i - j| > k$. One example of this situation is that $y^{(1)}, \ldots, y^{(p)}$ are moving average process of order k, i.e., $y^{(t)} = \theta_{t,t-1}\epsilon_1 + \ldots + \theta_{t,t-k}\epsilon_k$ and $\epsilon_1, \ldots, \epsilon_k$ are i.i.d. variables with mean 0.

Banding can not preserve the positive definiteness of the covariance matrix. Furrer and Bengtsson (2007) claim that $\hat{\Sigma} * R$ is a suitable estimator of Σ if $R = [r_{ij}]$ is symmetric and positive definite, where * is Schur (coordinate-wise) matrix multiplication. Banding is simply taking $r_{ij} = I(|i - j| \leq k)$, which is not nonnegative definite. They also present examples of positive definite symmetric R's.

The thresholding and banding parameters can be chosen by cross validation to minimize the empirical risk.

2.3 Penalised Likelihood Estimators

Banerjee et al. (2006), Yuan and Lin (2007) propose a penalised estimator for Gaussian distribution, using different semi-definite programming algorithms. Friedman et al. (2008) propose a faster algorithm. With ℓ_1 penalty on the off-diagonal elements of the inverse covariance matrix, the Lasso-type estimator is

$$-\log|C| + \mathrm{tr(CA)}, \quad \text{subject to} \quad \sum_{\mathrm{i}\neq\mathrm{j}} |\mathrm{C_{ij}}| \leq \mathrm{M} \tag{12}$$

where $A = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})'$.

Lam and Fan (2009) extend the Lasso penalty to nonconvex penalty functions on the covariance matrix, the inverse matrix and the Cholesky factor matrix. They show that for Lasso penalised estimator, only if the number of nonzero elements is $O(p)$, can the estimator achieve sparsistency and optimal convergence rate. Other estimators such as hard thresholding and SCAD (Fan and Li, 2001) do not have this constraint.

When an initial estimate of $\tilde{C}$ is available, Yuan and Lin (2007) present a nonnegative garrote estimator, $C_{ij} = d_{ij}C_{ij}$, where $D = (d_{ij})$ is a symmetric matrix,

determined by

$$-\log|C| + \mathrm{tr}(\mathrm{CA}), \quad \text{subject to} \quad \sum_{\mathrm{i}\neq\mathrm{j}} \mathrm{d_{ij}} \leq \mathrm{M}, \quad \mathrm{d_{ij}} \geq 0.$$

When there is a natural ordering among the variables, for example, longitudinal data, Huang et al. (2006) discuss the regularization of covariance matrix in the context of the modified Cholesky decomposition. The coordinates are assumed to have the following regression relationship,

$$y_k = \sum_{j=1}^{k-1} \phi_{k,j} y_j + \epsilon_k, \tag{13}$$

where $1 < k \leq p$; ϵ_k is uncorrelated and $\mathrm{Cov}(\epsilon) = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_{\mathrm{p}}^2)$. It corresponds to the matrix form

$$\epsilon = Ty, \tag{14}$$

where T is a unit lower triangular matrix with $-\phi_{t,j}$ in the (t, j) position for $2 \leq t \leq p$ and $1 \leq j \leq t-1$. Then the covariance of y has the modified Cholesky decomposition

$$T\Sigma T' = D, \tag{15}$$

where

$$D = \mathrm{Cov}(\epsilon) = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_{\mathrm{p}}^2) \tag{16}$$

. The ℓ_1 or ℓ_2 penalty is put on the Cholesky factors $\phi_{k,j}$

$$\lambda \sum_{j=1}^{k-1} |\phi_{k,j}| \tag{17}$$

The solution can be derived by the procedure in regularized regression. Smith and Kohn (2002) also penalise the Cholesky factors T and their approach is based on a hierarchical prior.

The approach to shrink Cholesky factors is more flexible than banding. However, it is not invariant to the variable permutations. And the sparsity of the T matrix does not necessary lead to the sparsity of the inverse matrix. Wong et al. (2003) use a prior

to allow the the inverse matrix to have zero entries. Rothman et al. (2008) propose an algorithm based the modified Cholesky decomposition to solve (12) to make the estimator invariant to the permutation of the variables. Levina et al. (2008) apply a nested Lasso penalty

$$\lambda \left(|\phi_{k,k-1}| + \frac{\phi_{k,k-2}}{\phi_{k,k-1}} + \frac{\phi_{k,k-3}}{\phi_{k,k-2}} + \ldots + \frac{\phi_{k,1}}{\phi_{k,2}} \right). \tag{18}$$

to preserve sparsity in the estimated inverse matrix. With scaling parameter $1/|\phi_{k,l}|$ in the penalty, we can see that if $\phi_{k,l} = 0$, then its preceding Cholesky factors will be shrinked to be zero, $\phi_{k,t} = 0 (t < l)$.

Other examples of covariance matrix estimate are Wu and Pourahmadi (2003), Fan et al. (2008) of factor analysis, D'Aspremont et al. (2008) of first order condition, Yuan (2008), Deng and Yuan (2009), among others.

CHAPTER III

PARAMETER ESTIMATION IN HIGH DIMENSIONAL T-DISTRIBUTION

3.1 Introduction

The student t-distribution is of increasing interest in mathematical finance, education (Jones, 2002; Nevzorov et al 2003) and other applications. In particular, the tail dependence of t-distribution has been widely established in risk management to model dependent risks (Wang, 1997; Daul et al., 2003; Embrechts et al., 2002; Frey et al., 2001; Schloegl and OKane, 2005). The dependence between financial instruments is usually captured by correlation. For example, in The Capital Asset Pricing Model (CAPM) and the Arbitrage Pricing Theory (APT) (Campbell et al. 1997), such dependence is studied to derive an optimal portfolio selection. Usually, those studies are based on the assumption of multivariate normally distribution. With the increasing complexity of financial products, however, actuarial world has presented different phenomena. For instance, insurance claim data has typical skewness and heavy-tailedness (Embrechts et al. 2002). And Frey et al. (2001) showed that the normal dependence and t-dependence are different although they have the same correlation; The index returns have positive excess kurtosis (Mandelbrot, 1963; Fama, 1965). Ferguson and Platen (2006) suggest that the t-distribution with degrees of freedom $\nu = 4$ is a good model.

Although quite useful, the application in t-distribution is limited by the difficulty in the parameter estimation of high dimensional data. The challenge arises with the estimation of the covariance matrix. Let $y = (y^{(1)}, \ldots, y^{(p)})$ be a random vector from a p-dimensional multivariate t-distribution with ν degrees of freedom, the density

function is

$$f(y|\mu,\Psi) = \frac{\Gamma(\frac{p+\nu}{2})|\Psi|^{-1/2}}{(\pi\nu)^{p/2}\Gamma(\nu/2)[1+\frac{1}{\nu}(y-\mu)'\Psi^{-1}(y-\mu)]^{(p+\nu)/2}}, \tag{19}$$

where μ and Ψ are referred as mean and scale matrix. It is equivalent to the covariance matrix up to a constant ($\Sigma = \frac{\nu}{\nu-2}\Psi$). The covariance matrix has an order of p^2 parameters and it needs to be positive definite. The usual estimator of t-distribution is MLE and it can be efficiently calculated by EM algorithm (Liu and Rubin, 1995). But EM algorithm is notoriously instable with the increase of dimensionality and the positive definiteness of the covariance matrix can not be guaranteed.

These facts prompts us to construct a new estimator for high dimensional t-distribution. In this chapter, we propose on penalised likelihood estimator of the inverse covariance matrix. They are naturally related to the success of parameter estimation in high dimensional multivariate Gaussian distributions (Ledoit and Wolf, 2004; Huang et al., 2006; Yuan and Lin, 2007; Friedman et al. 2008 among others). Yuan and Lin (2007) proposed an ℓ_1 penalised likelihood estimator. The problem is a maxdet algorithm (Vandenberghe et al. 1998) and can be efficiently solved with the interior algorithm. Friedman et al. 2008 presented a faster algorithm to solve the problem. Huang et al. (2006) imposed ℓ_1 and ℓ_2 penalty on the Cholesky factors of the covariance matrix and the covariance matrix selection problem is reduced to the variable selection in regression. Although the regularization estimate for gaussian distribution is receiving most of the attention, few discussion has been done on multivariate t distribution. As we discuss before, the applications of t-distribution can not be replaced by Gaussian. In this chapter, we introduce an ℓ_1 penalised likelihood estimator. The penalty on the off-diagonal encourages sparsity of the inverse covariance matrix. Also, we discuss the alternative estimator based on the modified Cholesky decomposition of the covariance matrix.

This chapter is organized as follows. We derive the algorithm in the next section and show that the parameters can be estimated through EM algorithm efficiently. To

show the merits of the new estimator, in section three, we compare the performance with a good candidate LWE (Ledoit and Wolf, 2004), the popular MLE and sample covariance in simulations. Conclusion will be given in section four.

3.2 Methodology

Given independent samples $y_1, \ldots, y_n$ from t-distribution with ν degrees of freedom, denote $\Theta = (\mu, C)$, where $C^{-1} = \frac{\nu}{\nu-2}\Psi$, the likelihood of Θ is

$$\ell(\Theta; y) = -\sum_{i=1}^{n} \log f(y_i|\Theta), \tag{20}$$

where $f(\cdot)$ is the density function in (19). Sample covariance matrix is one common estimator $S = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})'$. MLE is another alterative solved by EM algorithm. And the inverse covariance matrix can be estimated by the inverse of them. But they are not stable for a moderate size of p and not sparse either.

We propose a penalised likelihood estimator based on a modified Cholesky decomposition of the covariance matrix. As discussed in the previous chapter, penalty is imposed on the Cholesky factors and the penalised likelihood function is

$$\ell(\Theta; y) = -\sum_{i=1}^{n} \log f(y_i|\Theta) + \lambda \sum_{k=2}^{p}\sum_{j=1}^{k-1} |\phi_{kj}|, \tag{21}$$

Formulation (21) does not lead to a close form of C. Similar to the algorithm of MLE, we adapt one property of t-distribution that it is a scaled mixture of normals: if $y|\tau \sim N_p(\mu, \Psi/\tau)$ and $\tau \sim \chi^2_\nu/\nu$, then $y \sim t_p(\mu, \Psi, \nu)$. We augment the data by introducing a hidden variable τ_i to each observation y_i. Given τ_i, the alternative negative likelihood function of (20) can be formulated by the multivariate Gaussian distribution of $(y_i|\tau_i)$, which is

$$\frac{p}{n}\sum_{i=1}^{n} \log \tau_i + \log|\Psi| + \frac{1}{n}\sum_{i=1}^{n} \tau_i (y_i - \mu)'\Psi^{-1}(y_i - \mu), \tag{22}$$

Note that $\Psi = \frac{\nu-2}{\nu}\Sigma$ and $|T| = 1$, (22) is equivalent to

$$\log|D| + \frac{\nu}{n(\nu-2)}\sum_{i=1}^{n} \tau_i (Ty_i)'D^{-1}(Ty_i) + \lambda \sum_{k=2}^{p}\sum_{j=1}^{k-1} |\phi_{kj}|. \tag{23}$$

By (14) and (16), (23) is simplified as

$$\begin{aligned}
&\sum_{k=1}^{p} \log \sigma_k^2 + \frac{\nu}{n(\nu-2)} \sum_{i=1}^{n} \tau_i \sum_{k=1}^{p} \frac{\epsilon_{ik}^2}{\sigma_k^2} + \lambda \sum_{k=2}^{p} \sum_{j=1}^{k-1} |\phi_{kj}| \\
= \quad & \left\{ \log \sigma_1^2 + \frac{\nu}{n(\nu-2)} \sum_{i=1}^{n} \tau_i \frac{\epsilon_{i1}^2}{\sigma_1^2} \right\} + \sum_{k=2}^{p} \left\{ \log \sigma_k^2 + \frac{\nu}{n(\nu-2)} \sum_{i=1}^{n} \tau_i \frac{\epsilon_{ik}^2}{\sigma_k^2} + \lambda \sum_{j=1}^{k-1} |\phi_{kj}| \right\}
\end{aligned} \tag{24}$$

Minimization of (24) is decomposed by p individual problems

$$\min_{\sigma_1^2} \log \sigma_1^2 + \frac{\nu}{n(\nu-2)} \sum_{i=1}^{n} \tau_i \frac{\epsilon_{i1}^2}{\sigma_1^2} \tag{25}$$

and

$$\min_{\sigma_k^2, \phi_{kj}, j=1,\ldots,k-1} \log \sigma_k^2 + \frac{\nu}{n(\nu-2)} \sum_{i=1}^{n} \tau_i \frac{\epsilon_{ik}^2}{\sigma_k^2} + \lambda \sum_{j=1}^{k-1} |\phi_{kj}|, \quad k = 2, \ldots, p \tag{26}$$

It is clear to see that the minimizer of (25) is

$$\sigma_1^2 = \frac{\nu}{n(\nu-2)} \sum_{i=1}^{n} \tau_i \epsilon_{i1}^2 = \frac{\nu}{n(\nu-2)} \sum_{i=1}^{n} \tau_i y_{i1}^2. \tag{27}$$

For each $k, 2 \le k \le p$, the minimization of (26) is determined by the alterative minimization over σ_k^2 and $\phi_{kj}, j = 1, \ldots, k-1$:

Given $\phi_{kj}, j = 1, \ldots, k-1$, the optimal σ_k^2 is given by

$$\sigma_k^2 = \frac{\nu}{n(\nu-2)} \sum_{i=1}^{n} \tau_i (y_{ik} - \sum_{j=1}^{k-1} \phi_{kj} y_{ij})^2; \tag{28}$$

Given $\sigma_k, k = 2, \ldots, p$, the optimal $\phi_{kj}, j = 1, \ldots, k-1$ is determined by the following ℓ_1 regression problem

$$\sum_{i=1}^{n} \frac{\tau_i}{\sigma_t^2} (y_{ik} - \sum_{j=1}^{k-1} \phi_{kj} y_{ij})^2 + \lambda \sum_{j=1}^{k-1} |\phi_{kj}| \tag{29}$$

which can be conveniently computed by the lasso subroutine.

So far we have discussed the algorithm based on known τ_i. When τ_i is missing, EM algorithm (Dempster, Laird, and Rubin, 1977) can be applied to compute the parameters. Since τ_i is hidden and unknown, we replace it with its expectation

$\mathrm{E}[\tau_\mathrm{i}|\mathrm{y_i}, \theta]$. The expectation of τ can be calculated if parameters are given. So EM algorithm works iteratively: in E-step, calculate $\mathrm{E}[\tau_\mathrm{i}|\mathrm{y_i}, \theta]$ given parameters; in M-step, we estimate parameters as discussed before given $\tau = \mathrm{E}[\tau_\mathrm{i}|\mathrm{y_i}, \theta]$.

In particular, $\mathrm{E}[\tau_\mathrm{i}|\mathrm{y_i}, \theta]$ is derived as follows

$$\mathrm{E}[\tau_\mathrm{i}|\mathrm{y_i}, \theta] = \int \tau_\mathrm{i} \mathrm{f}(\tau_\mathrm{i}|\theta, \mathrm{y_i}) \mathrm{d}\tau_\mathrm{i} = \int \tau_\mathrm{i} \frac{\mathrm{f}(\tau_\mathrm{i}, \mathrm{y_i}|\theta)}{\mathrm{f}(\mathrm{y_i}|\theta)} \mathrm{d}\tau_\mathrm{i} = \frac{\int \tau_\mathrm{i} \mathrm{f}(\mathrm{y_i}|\theta, \tau_\mathrm{i}) \mathrm{f}(\tau_\mathrm{i}) \mathrm{d}\tau_\mathrm{i}}{\int \mathrm{f}(\mathrm{y_i}|\theta, \tau_\mathrm{i}) \mathrm{f}(\tau_\mathrm{i}) \mathrm{d}\tau_\mathrm{i}} = \frac{\mathrm{p} + \nu}{\mathrm{d_i} + \nu}, \tag{30}$$

where d_i is defined as

$$d_i = (y_i - \mu)' \Psi^{-1} (y_i - \mu) \tag{31}$$

The denominator in (32) is derived as

$$\begin{aligned}
\int f(y_i|\theta, \tau_i) f(\tau_i) d\tau_i &= \int (2\pi)^{-0.5p} (\tau_i)^{0.5p} |\Psi|^{-0.5} \exp\left\{-0.5\tau_i d_i\right\} \left\{ \frac{2^{-0.5\nu}}{\nu\Gamma(0.5\nu)} \tau_i^{0.5\nu-1} \exp^{(-0.5\tau_i)} \right\} d\tau_i \\
&= (2\pi)^{-0.5p} |\Psi|^{-0.5} \frac{2^{-0.5\nu}}{\nu\Gamma(0.5\nu)} \int (\tau_i)^{0.5(p+\nu)-1} \exp^{\{-0.5(d_i+\nu)\tau_i\}} d\tau_i \\
&= (2\pi)^{-0.5p} |\Psi|^{-0.5} \frac{2^{-0.5\nu}}{\nu\Gamma(0.5\nu)} \int \left(\frac{2z_i}{d_i + \nu} \right)^{0.5(p+\nu)-1} \exp^{-z_i} \frac{2}{d_i + \nu} dz_i \\
&= (2\pi)^{-0.5p} |\Psi|^{-0.5} \left(\frac{2}{d_i + \nu} \right)^{0.5(p+\nu)} \Gamma\left(0.5(p + \nu)\right)
\end{aligned} \tag{32}$$

where the third equal sign is based on the transformation $z_i = 0.5(d_i + 1)\tau_i$.

Similarly, the numerator is

$$\int f(y_i|\theta, \tau_i) f(\tau_i) d\tau_i = (2\pi)^{-0.5p} |\Psi|^{-0.5} \left(\frac{2z_i}{d_i + \nu} \right)^{0.5(p+\nu)+1} \Gamma\left(0.5(p + \nu) + 1\right). \tag{33}$$

To sum up, the algorithm goes as follows:

Initialize $\theta^{(t)} = \theta^{(0)}$, begin iteration;

E step: Impute τ_i by (30);

M step: For $k = 1, \ldots, p$, update σ_1^2 by (27); given ϕ_{kj}, update σ_k^2 by (28), given σ_k^2, update ϕ_{kj} by solving (29).

Iteration continues until converges.

Similar to the argument in Dempster et al. (1977), the objective functions of the two estimators decrease in every iteration and the algorithm converges. The algorithm discussed so far is based on the fact that the tuning parameter λ is given. For unknown λ, we can use cross validation(CV) and BIC criterion to do a grid search.

3.3 Simulation

In this section, we investigate the performance of the proposed penalized likelihood estimate (PLE) with finite samples. Three covariance models are considered.

Model 1 : AR(1) model, $\sigma_{ij} = 0.4^{|i-j|}$.

Model 2 : AR(2) model, $T(i+1,i) = -0.4, T(i+2,i) = 0.4, D = \text{diag(p}, \ldots, 1)$.

Model 3 : For the lower triangular matrix T, for any row i, randomly draw two entries with probability 0.5 such that the entry is -0.4, $D = \text{diag}1, \ldots, \text{p}$.

For each model, $n = 50$ observations are simulated from a $p = 20, 30, 50$ dimensional t distribution of with $\mu = 0$ and degrees of freedom $\nu = 3, \nu = 5$ and $\nu = 8$ respectively. The performance of the proposed penalised likelihood estimator is compared with three other popular estimates, sample covariance matrix (Sample), maximum likelihood estimate (MLE), and Ledoit and Wolfe (2004;LWE) under three loss criterion.

$$\text{SL} = \|\hat{\Sigma}^{-1} - \Sigma^{-1}\| \tag{34}$$

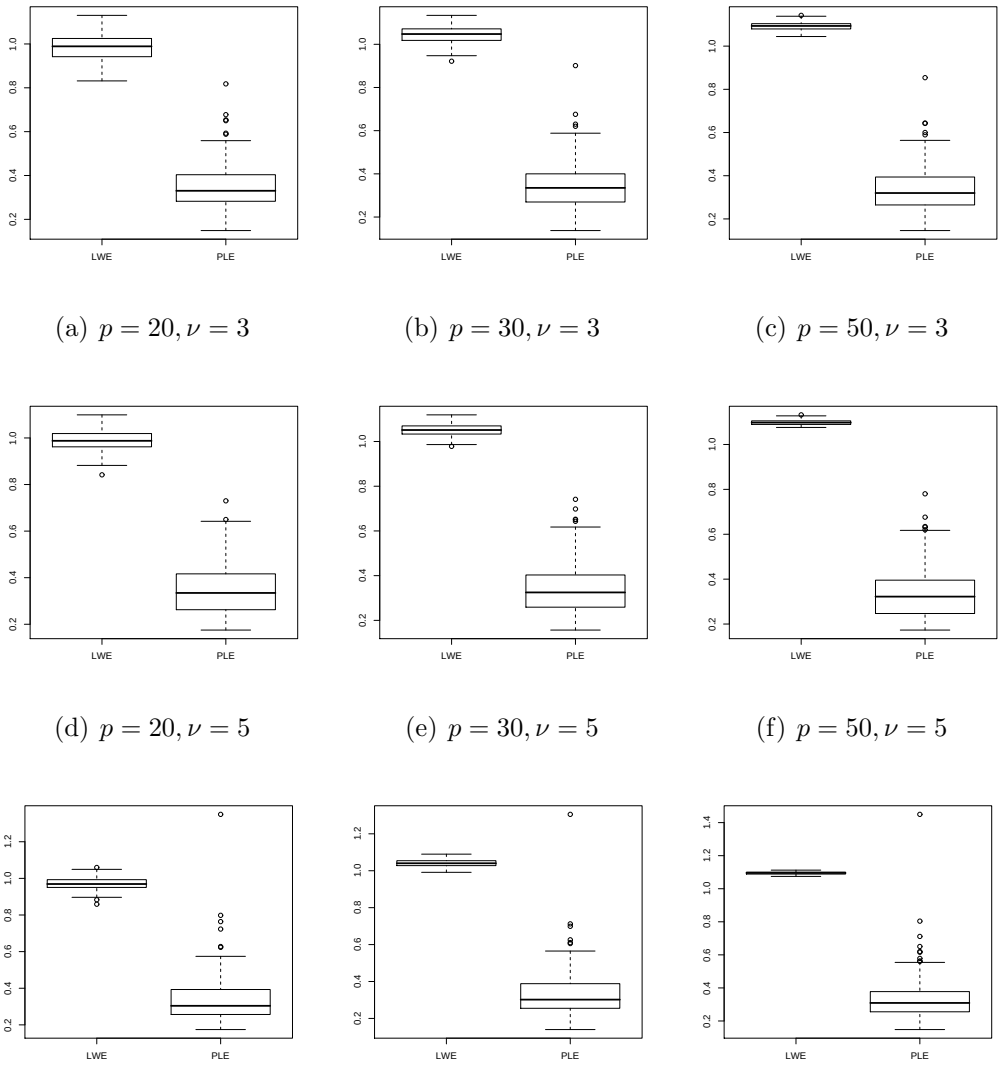
where $\|A\|$ is the largest singular value of matrix A; the Frobenius norm of the difference

$$\text{FL} = \|\hat{\Sigma}_k^{-1} - \Sigma_k^{-1}\|_{\text{F}} = \sqrt{\sum_{i,j}(\hat{\Sigma}^{-1}(i,j) - \Sigma^{-1}(i,j))^2}; \tag{35}$$

and Kullback-Leibler(KL) loss

$$\text{KL} = \text{tr}\left(\Sigma\hat{\Sigma}^{-1}\right) - \log\left|\Sigma\hat{\Sigma}^{-1}\right| - p. \tag{36}$$

The experiment is repeated 100 times and the tuning parameter is selected by BIC and Cross Validation (CV). All the simulation results are listed in Table (9)- Table (11). Figure 4 - 12 display the boxplot comparison of SL, FL, KL loss of LWE and the proposed method with CV tuning.



(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 4: Boxplot of 100 runs: y-axis is the Largest Singular Value Loss (SL) of Model I with Cross Validation Tuning.

Table 9: Simulation Table for Model I: SL is the absolute error in the largest singular value of the inverse matrix; FL is the Frobenius norm of the error; KL is the Kullback-Leibler(KL) Loss. The reported number is the average over 100 runs. $(\cdot)$ is the standard error.

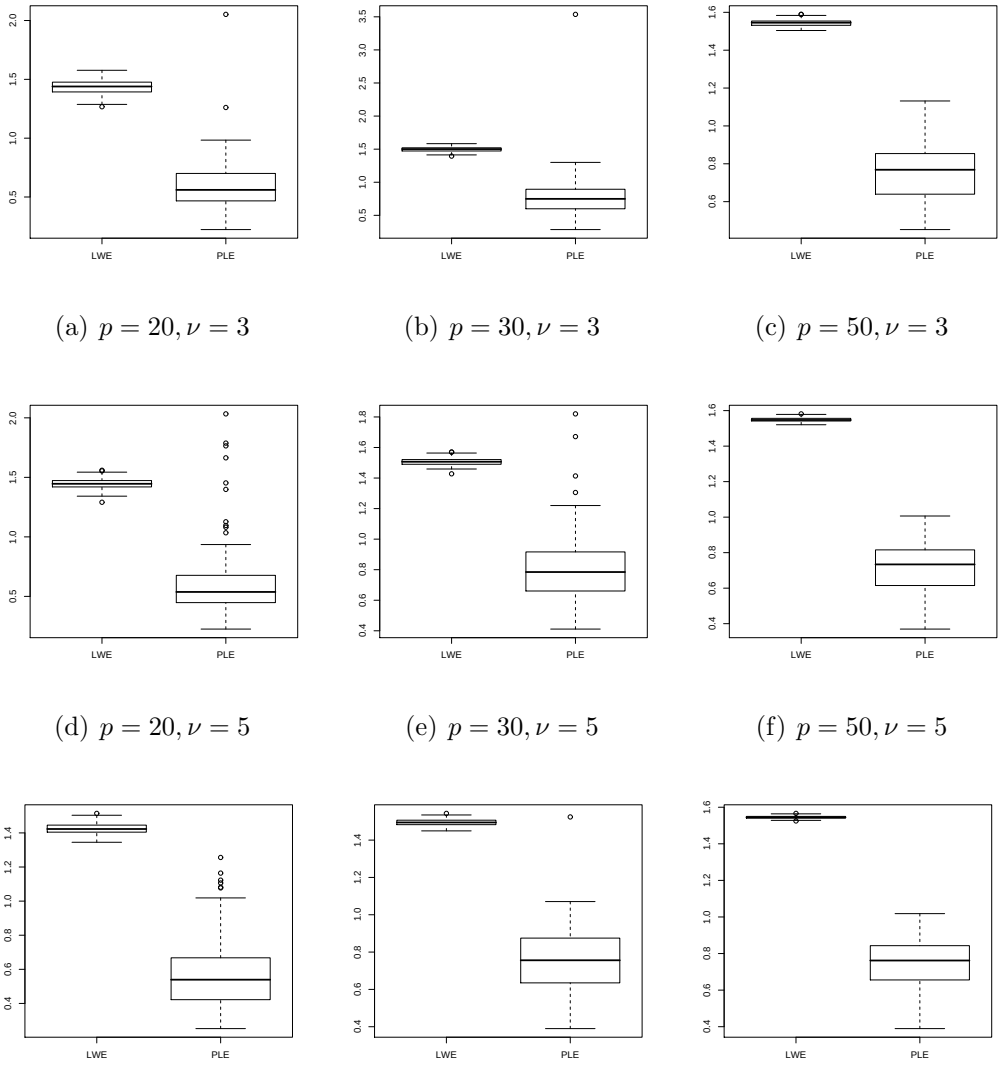
			$\nu = 3$			$\nu = 5$			$\nu = 8$		
		p	SL	FL	KL	SL	FL	KL	SL	FL	KL
BIC	PLE	20	0.4	0.57	3.39	0.4	0.55	3.4	0.39	0.55	3.39
			(0.009)	(0.004)	(0.045)	(0.01)	(0.008)	(0.025)	(0.014)	(0.012)	(0.027)
CV	PLE		0.36	0.49	2.38	0.36	0.48	2.36	0.35	0.47	2.28
			(0.012)	(0.01)	(0.047)	(0.012)	(0.01)	(0.05)	(0.016)	(0.014)	(0.047)
	LWE		1.03	1.19	5.55	0.99	1.13	5.09	0.97	1.1	4.39
			(0.001)	(0.001)	(0.005)	(0.005)	(0.007)	(0.102)	(0.004)	(0.005)	(0.053)
	MLE		1.28	1.71	9.69	1.52	1.84	10.79	1.53	1.86	10.85
			(0.013)	(0.002)	(0.159)	(0.062)	(0.061)	(0.254)	(0.07)	(0.068)	(0.246)
	Sample		3.66	4.86	30.36	2.26	2.76	17.84	1.87	2.27	13.93
			(0.062)	(0.04)	(0.569)	(0.086)	(0.083)	(0.401)	(0.084)	(0.082)	(0.323)
BIC	PLE	30	0.39	0.56	5.38	0.39	0.55	5.3	0.39	0.55	5.3
			(0.01)	(0.008)	(0.038)	(0.01)	(0.008)	(0.035)	(0.013)	(0.011)	(0.031)
CV	PLE		0.35	0.48	3.6	0.35	0.47	3.6	0.34	0.47	3.51
			(0.012)	(0.01)	(0.067)	(0.012)	(0.01)	(0.07)	(0.015)	(0.013)	(0.062)
	LWE		1.04	1.23	11.87	1.05	1.24	8.94	1.04	1.22	7.9
			(0.004)	(0.007)	(0.339)	(0.003)	(0.005)	(0.137)	(0.002)	(0.003)	(0.06)
	MLE		3.34	4.05	40.72	3.4	4.11	41.41	3.25	3.98	41.83
			(0.158)	(0.152)	(0.884)	(0.167)	(0.16)	(0.875)	(0.135)	(0.13)	(0.858)
	Sample		9.34	11.35	130.93	5.27	6.38	68.5	4.07	5	54.16
			(0.453)	(0.434)	(2.888)	(0.239)	(0.228)	(1.368)	(0.166)	(0.158)	(1.051)
BIC	PLE	50	0.38	0.56	9.2	0.39	0.55	9.18	0.39	0.56	9.17
			(0.01)	(0.008)	(0.045)	(0.009)	(0.007)	(0.047)	(0.013)	(0.011)	(0.039)
CV	PLE		0.35	0.48	6.38	0.35	0.47	6.56	0.35	0.48	6.48
			(0.012)	(0.01)	(0.094)	(0.012)	(0.01)	(0.111)	(0.016)	(0.014)	(0.108)
	LWE		1.09	1.32	22.01	1.1	1.33	17.64	1.09	1.32	16.11
			(0.002)	(0.004)	(0.56)	(0.001)	(0.003)	(0.202)	(0.001)	(0.002)	(0.071)

Table 10: Simulation Table for Model II: SL is the absolute error in the largest singular value of the inverse matrix; FL is the Frobenius norm of the error; KL is the Kullback-Leibler(KL) Loss. The reported number is the average over 100 runs. $(\cdot)$ is the standard error.

			$\nu = 3$			$\nu = 5$			$\nu = 8$		
		p	SL	FL	KL	SL	FL	KL	SL	FL	KL
BIC	PLE	20	1.2	1.36	6.72	1.2	1.36	6.7	1.19	1.35	6.68
			(0.007)	(0.006)	(0.036)	(0.006)	(0.006)	(0.032)	(0.006)	(0.006)	(0.03)
CV	PLE		0.6	0.77	3.83	0.63	0.8	3.95	0.58	0.77	3.99
			(0.023)	(0.021)	(0.09)	(0.033)	(0.029)	(0.099)	(0.022)	(0.02)	(0.099)
	LWE		1.44	1.6	8.24	1.44	1.61	6.21	1.42	1.57	5.27
			(0.006)	(0.01)	(0.255)	(0.005)	(0.007)	(0.14)	(0.003)	(0.005)	(0.066)
	MLE		1.84	2.18	10.9	1.88	2.23	10.79	1.79	2.15	10.85
			(0.087)	(0.085)	(0.283)	(0.084)	(0.083)	(0.254)	(0.07)	(0.067)	(0.246)
	Sample		5.08	6.09	36.59	2.86	3.39	17.84	2.21	2.65	13.93
			(0.212)	(0.208)	(0.914)	(0.115)	(0.114)	(0.401)	(0.085)	(0.082)	(0.323)
BIC	PLE	30	1.21	1.38	9.75	1.2	1.37	9.75	1.19	1.36	9.7
			(0.007)	(0.006)	(0.04)	(0.007)	(0.006)	(0.044)	(0.006)	(0.005)	(0.034)
CV	PLE		0.77	0.96	7.2	0.8	0.98	7.29	0.76	0.95	7.23
			(0.035)	(0.033)	(0.122)	(0.023)	(0.021)	(0.106)	(0.018)	(0.015)	(0.081)
	LWE		1.5	1.7	13.8	1.51	1.71	10.82	1.49	1.69	9.52
			(0.004)	(0.006)	(0.356)	(0.003)	(0.005)	(0.18)	(0.002)	(0.003)	(0.076)
	MLE		3.95	4.69	40.72	4.01	4.77	41.41	4.2	4.98	41.83
			(0.158)	(0.155)	(0.884)	(0.166)	(0.16)	(0.875)	(0.172)	(0.166)	(0.858)
	Sample		11.18	13.35	130.93	6.33	7.53	68.5	5.3	6.3	54.16
			(0.443)	(0.427)	(2.888)	(0.263)	(0.252)	(1.368)	(0.204)	(0.193)	(1.051)
BIC	PLE	50	1.21	1.39	15.58	1.21	1.39	15.63	1.2	1.38	15.6
			(0.006)	(0.006)	(0.042)	(0.006)	(0.006)	(0.05)	(0.006)	(0.006)	(0.046)
CV	PLE		0.75	0.96	12.09	0.72	0.93	12.21	0.75	0.96	12.24
			(0.015)	(0.012)	(0.108)	(0.014)	(0.012)	(0.122)	(0.012)	(0.01)	(0.126)
	LWE		1.54	1.79	25.69	1.55	1.8	20.95	1.55	1.79	19.19
			(0.002)	(0.004)	(0.563)	(0.001)	(0.003)	(0.24)	(0.001)	(0.002)	(0.091)

Table 11: Simulation Table for Model III: SL is the absolute error in the largest singular value of the inverse matrix; FL is the Frobenius norm of the error; KL is the Kullback-Leibler(KL) Loss. The reported number is the average over 100 runs. $(\cdot)$ is the standard error.

			$\nu = 3$			$\nu = 5$			$\nu = 8$		
		p	SL	FL	KL	SL	FL	KL	SL	FL	KL
BIC	PLE	20	0.48	0.62	3.45	0.47	0.61	3.36	0.48	0.61	3.34
			(0.011)	(0.009)	(0.038)	(0.01)	(0.008)	(0.026)	(0.012)	(0.01)	(0.024)
CV	PLE		0.43	0.56	2.68	0.43	0.55	2.6	0.43	0.55	2.55
			(0.012)	(0.01)	(0.052)	(0.01)	(0.008)	(0.045)	(0.014)	(0.012)	(0.047)
	LWE		1.15	1.3	7.56	1.16	1.31	5.52	1.15	1.28	4.77
			(0.006)	(0.009)	(0.255)	(0.004)	(0.006)	(0.1)	(0.003)	(0.005)	(0.051)
	MLE		1.74	2.05	10.9	1.69	2.02	10.79	1.64	1.96	10.85
			(0.087)	(0.085)	(0.283)	(0.083)	(0.079)	(0.254)	(0.072)	(0.07)	(0.246)
	Sample		4.69	5.6	36.59	2.54	3.05	17.84	2	2.4	13.93
			(0.229)	(0.22)	(0.914)	(0.117)	(0.111)	(0.401)	(0.086)	(0.083)	(0.323)
BIC	PLE	30	0.49	0.64	5.39	0.48	0.63	5.29	0.49	0.63	5.28
			(0.011)	(0.009)	(0.045)	(0.01)	(0.008)	(0.036)	(0.011)	(0.01)	(0.032)
CV	PLE		0.43	0.57	4.16	0.43	0.56	4.18	0.43	0.56	4.15
			(0.012)	(0.01)	(0.071)	(0.01)	(0.008)	(0.066)	(0.013)	(0.011)	(0.068)
	LWE		1.21	1.41	12.59	1.22	1.42	9.63	1.21	1.4	8.62
			(0.004)	(0.006)	(0.353)	(0.002)	(0.004)	(0.135)	(0.002)	(0.003)	(0.061)
	MLE		3.7	4.43	40.72	3.81	4.53	41.41	3.58	4.32	41.83
			(0.191)	(0.183)	(0.884)	(0.222)	(0.211)	(0.875)	(0.149)	(0.143)	(0.858)
	Sample		10.47	12.5	130.93	5.82	6.97	68.5	4.49	5.44	54.16
			(0.532)	(0.505)	(2.888)	(0.31)	(0.293)	(1.368)	(0.183)	(0.175)	(1.051)
BIC	PLE	50	0.49	0.65	9.5	0.48	0.64	9.45	0.49	0.64	9.47
			(0.01)	(0.009)	(0.048)	(0.009)	(0.008)	(0.049)	(0.011)	(0.01)	(0.046)
CV	PLE		0.49	0.65	9.5	0.42	0.56	7.74	0.43	0.57	7.88
			(0.01)	(0.009)	(0.048)	(0.01)	(0.008)	(0.113)	(0.014)	(0.012)	(0.118)
	LWE		1.26	1.5	23.21	1.27	1.51	18.83	1.26	1.5	17.34
			(0.002)	(0.004)	(0.552)	(0.001)	(0.003)	(0.205)	(0.001)	(0.002)	(0.073)

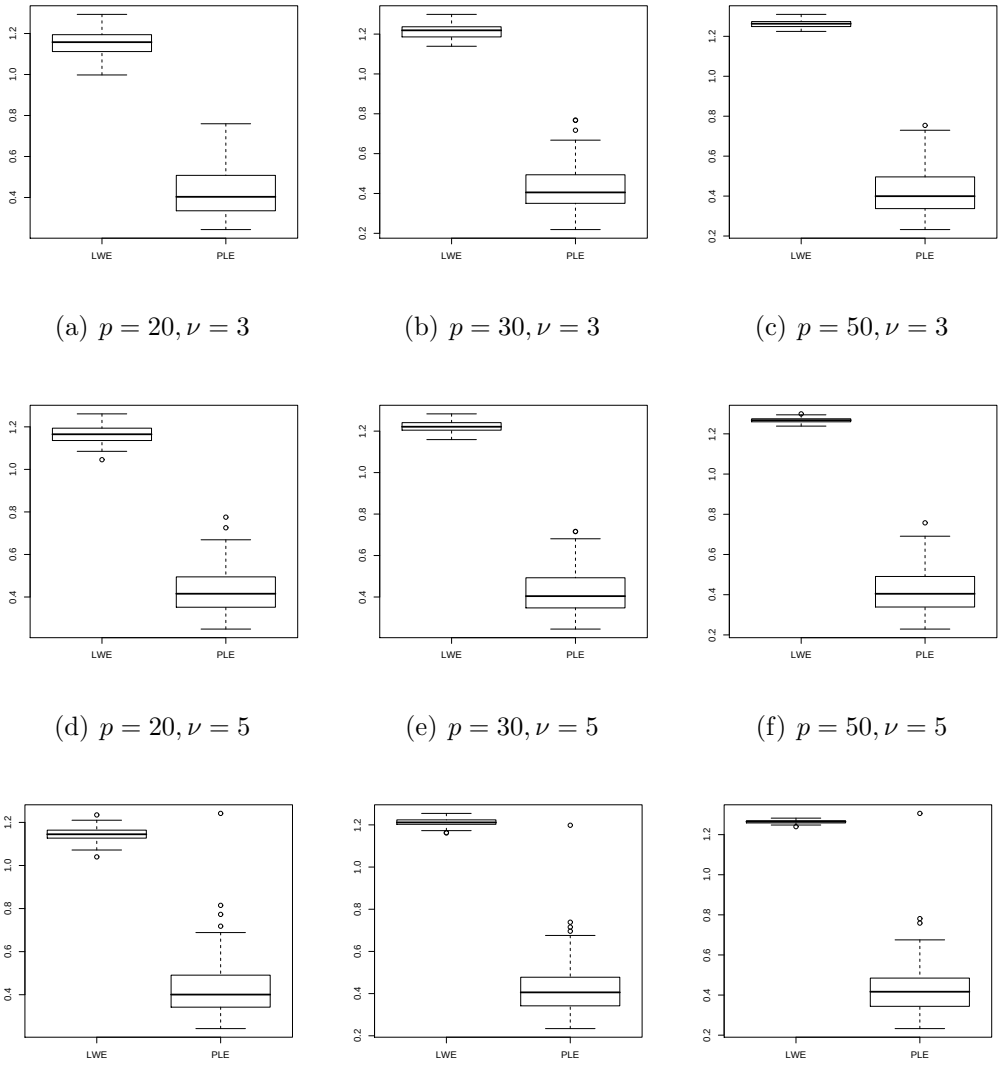


(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 5: Boxplot of 100 runs: y-axis is the Largest Singular Value Loss (SL) of Model II with Cross Validation Tuning.

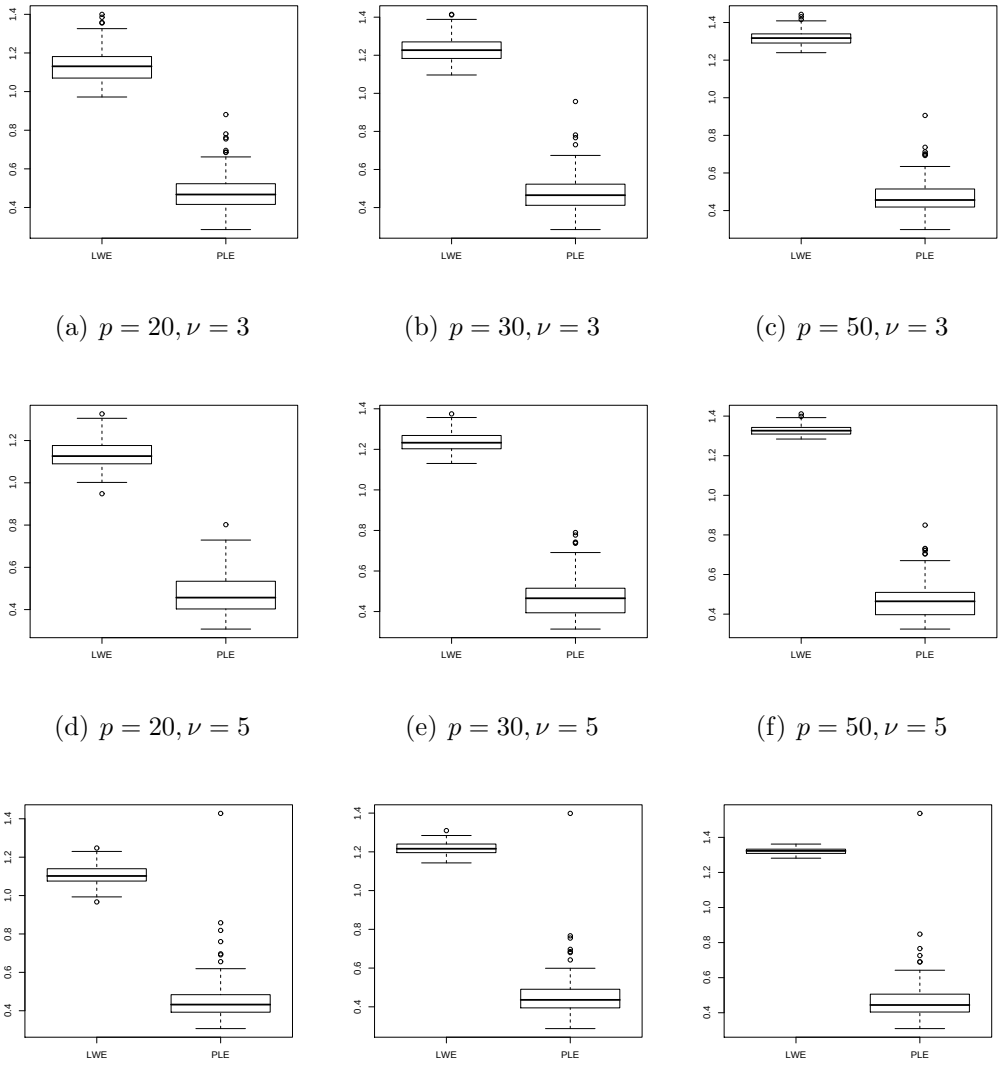


(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 6: Boxplot of 100 runs: y-axis is the Largest Singular Value Loss (SL) of Model III with Cross Validation Tuning.

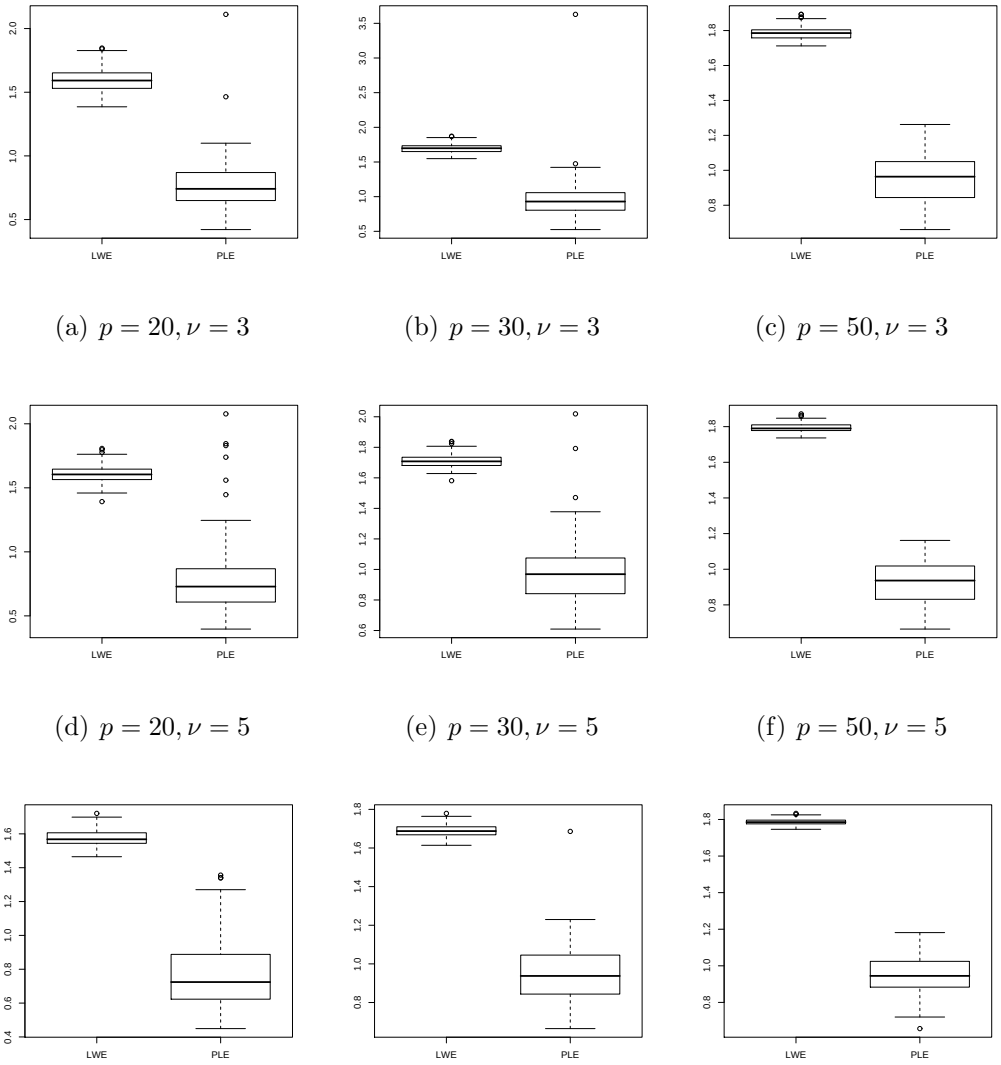


(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 7: Boxplot of 100 runs: y-axis is the Frobenius Norm Loss (FL) of Model I with Cross Validation Tuning.

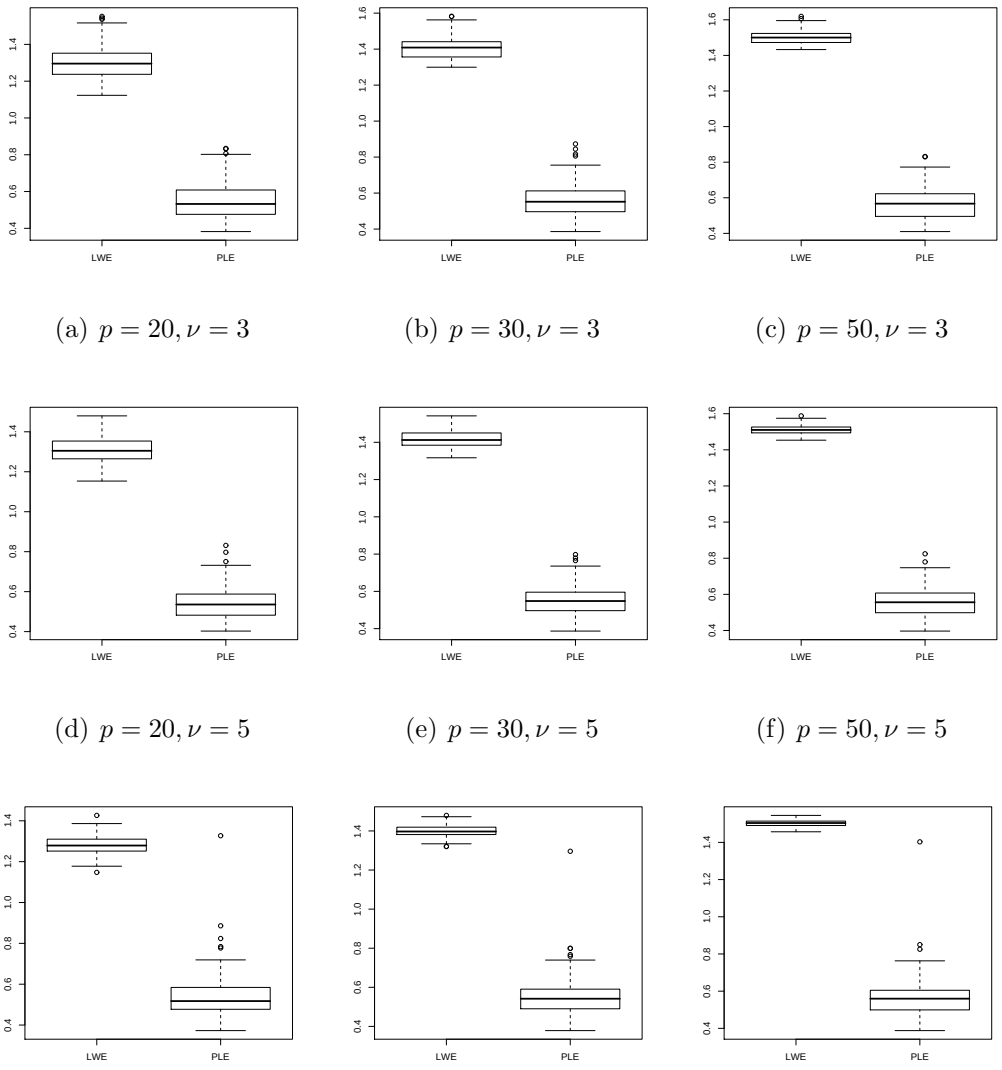


(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 8: Boxplot of 100 runs: y-axis is the Frobenius Norm Loss (FL) of Model II with Cross Validation Tuning.

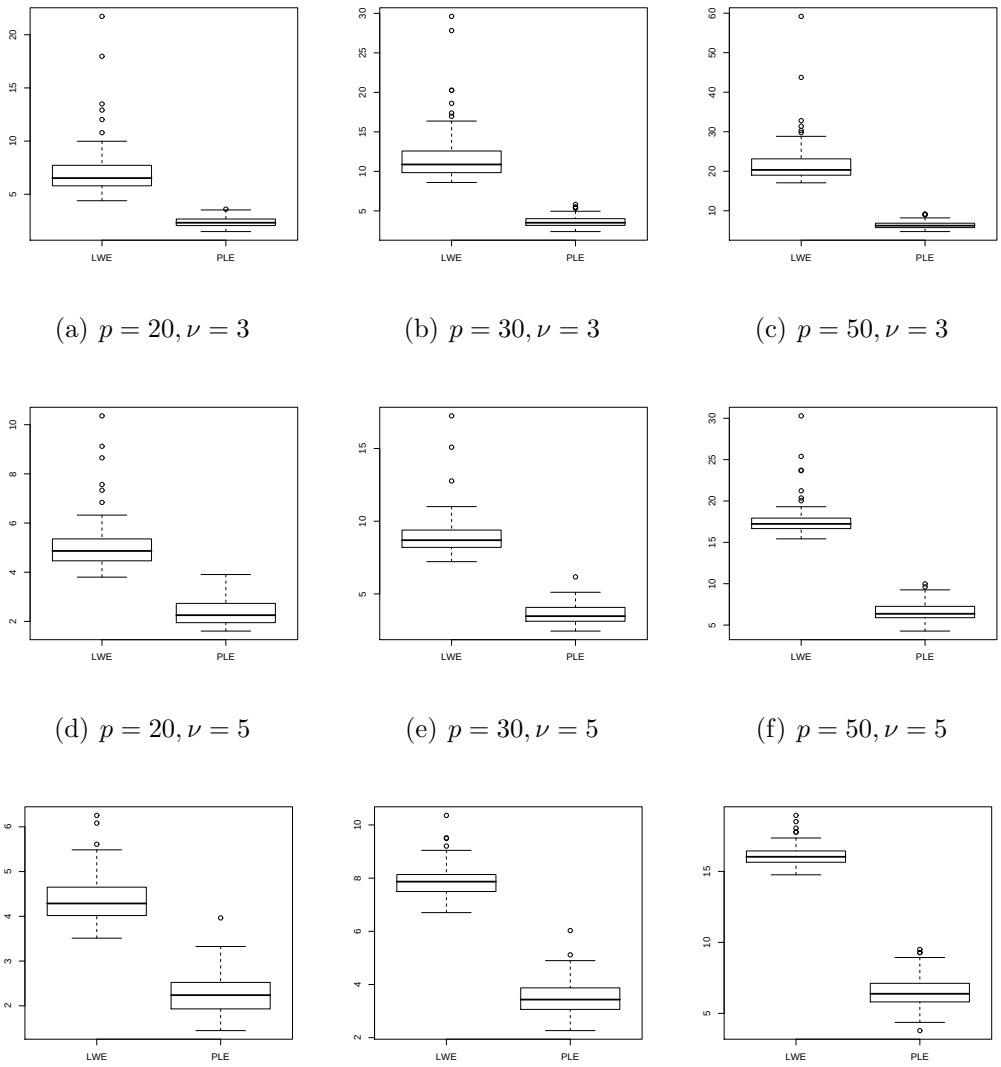


(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 9: Boxplot of 100 runs: y-axis is the Frobenius Norm Loss (FL) of Model III with Cross Validation Tuning.

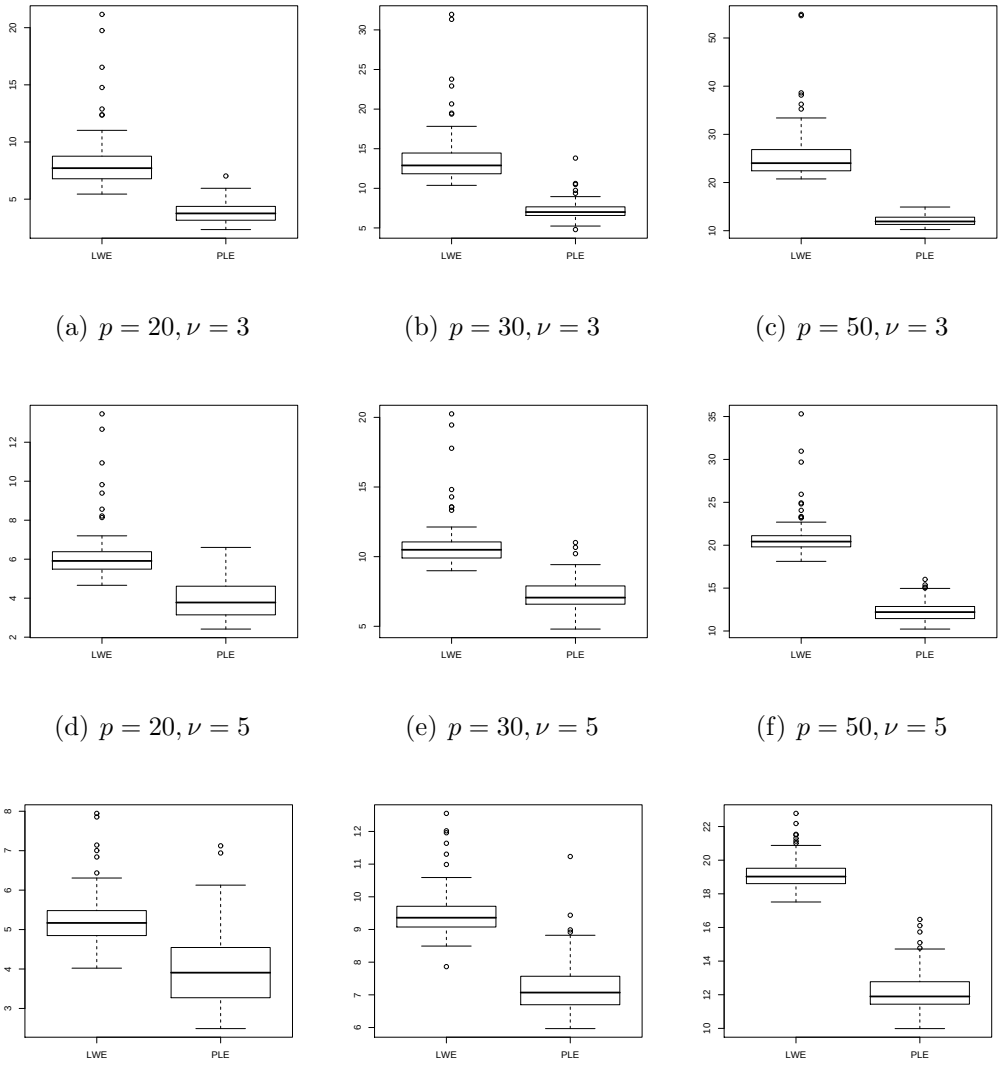


(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 10: Boxplot of 100 runs: y-axis is the Kullback-Leibler (KL) Loss of Model I with Cross Validation Tuning.

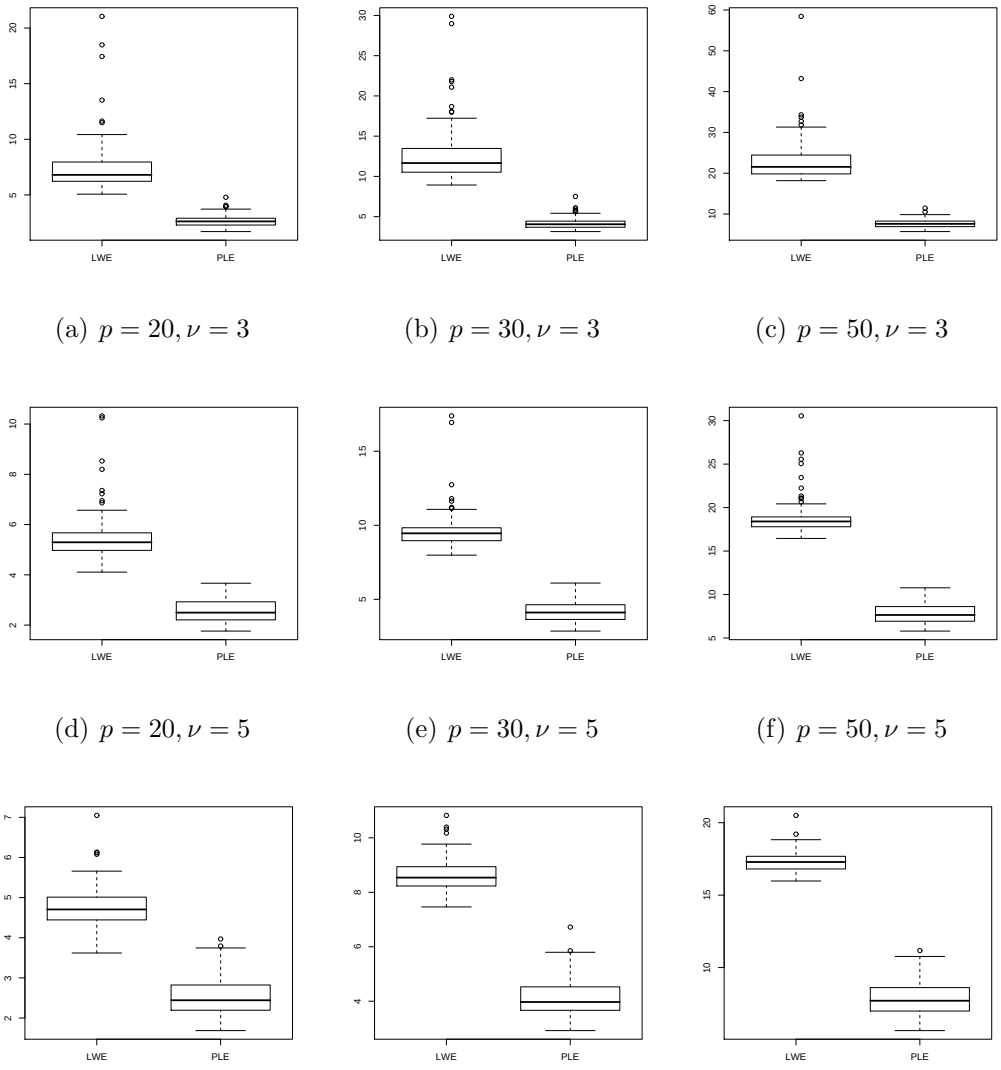


(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 11: Boxplot of 100 runs: y-axis is the Kullback-Leibler (KL) Loss of Model II with Cross Validation Tuning.



(a) $p = 20, \nu = 3$ (b) $p = 30, \nu = 3$ (c) $p = 50, \nu = 3$

(d) $p = 20, \nu = 5$ (e) $p = 30, \nu = 5$ (f) $p = 50, \nu = 5$

(g) $p = 20, \nu = 8$ (h) $p = 30, \nu = 8$ (i) $p = 50, \nu = 8$

Figure 12: Boxplot of 100 runs: y-axis is the Kullback-Leibler (KL) Loss of Model III with Cross Validation Tuning.

3.4 Conclusion

In this chapter, we proposed a penalised likelihood estimator for multivariate t-distribution. It has been shown that EM algorithm can efficiently compute the estimators. Numerical studies demonstrated that the new estimators perform well compared with three other alternative methods.

CHAPTER IV

REGULARIZED PARAMETER ESTIMATION IN HIGH DIMENSIONAL GAUSSIAN MIXTURE MODELS

4.1 Introduction

In finite Gaussian mixture models, a p-dimensional random vector $X = (X^{(1)}, \ldots, X^{(p)})$ is assumed to come from a mixture distribution

$$\pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2) + \ldots + \pi_M \mathcal{N}(\mu_M, \Sigma_M) \tag{37}$$

where $\mathcal{N}(\mu, \Sigma)$ is a multivariate normal distribution with mean vector μ and covariance matrix Σ, and π_ks are nonnegative proportions such that $\pi_1 + \ldots + \pi_M = 1$. Gaussian mixture models are among the most popular statistical modeling tools and are routinely used for density estimation, clustering, discriminant analysis among others (see, e.g., Fraley and Raftery, 2002; McLachlan and Peel, 2000).

Despite its great flexibility, the practical use of Gaussian mixture models in modeling high dimensional data is often hampered by the difficulty in parameter estimation. The number of parameters required to specify a covariance matrix quickly grows with the dimensionality. The problem is exacerbated in mixture models where multiple covariance matrices are to be estimated. Without any parameter restriction, each cluster must have at least $(p+1)$ observations to ensure the existence of the maximum likelihood estimate (Symons 1981). As a result, it is well known that the usual MLE can be notoriously unstable if well-defined at all when the data is of moderate or high dimensionality when compared with the sample size. To address this issue, a variety of parameter reduction techniques have been developed. In particular, Banfield and Raftery (1993) suggest to reparametrize Σ_k through its eigenvalue decomposition and assume through this parametrization that some parameters are shared across clusters.

Extensive studies within the same framework can also be found in Celeux and Govaert (1995). It has been demonstrated in both articles and studies later on that through parameter sharing across clusters, the problem of estimating a large number of parameters can be alleviated for data of moderate dimensions. The challenge, however, persists for high dimensional data, as the number of parameters remains of the order of p^2 even if a common covariance matrix is assumed for all clusters. In this paper, we propose a new technique to specifically address this challenge. Built upon recent advances in estimating covariance matrices of high dimensional multivariate Gaussian distributions, we propose a penalized likelihood estimate for high dimensional Gaussian mixture models. The ℓ_1 type of penalty we employ encourages sparsity of the inverse covariance matrices and therefore can help reduce the effective dimensionality of the problem. We show that the proposed estimate can be conveniently computed using an EM algorithm. Moreover, a BIC type of criterion is introduced to select the tuning parameter as well as the number of clusters. Numerical experiments, both simulated and real data examples, are also presented to demonstrate the merits of the proposed method.

Our method could prove useful for a variety of statistical problems. For illustration purpose, we consider in particular model-based clustering (Fraley and Raftery, 2002) and mixture discriminant analysis (Hastie and Tibshirani, 1996), two notable methods that take advantage of the flexibility of finite Gaussian mixture models in clustering and classification respectively. We demonstrate that with the proposed ℓ_1 penalized estimator, both approaches can be substantially improved when dealing with high dimensional problems.

Our investigation here is naturally related to recent studies on parameter estimation in high dimensional multivariate Gaussian distribution which can also be viewed as a special case of finite Gaussian mixture models (37) with $M = 1$. A number

of methods have been introduced in the past several years to estimate the covariances matrix with high dimensional data (Ledoit and Wolf, 2004; Huang et al., 2006; Yuan and Lin, 2007; Banerjee, El Ghaoui and d'Aspremont, 2008; Bickel and Levina, 2008a,b; Rothman et al., 2008; d'Aspremont, Banerjee and El Ghaoui, 2008; El Karoui, 2008; Fan, Fan and Lv, 2008; Friedman, Hastie and Tibshirani, 2008; Lam and Fan, 2008; Levina, Rothman and Zhu, 2008; Rothman et al., 2008; Yuan, 2008; Deng and Yuan, 2009; and Rothman, Levina and Zhu, 2009 among others). A common strategy there is to work with the sample covariance matrix which is readily computable regardless of the dimensionality (see, e.g., Bickel and Levina, 2008). For the more general finite Gaussian mixture model, we no longer have the luxury of such an initial estimate and regularization as employed here becomes critical. For our purpose, we adopt the idea of penalized likelihood estimate from Yuan and Lin (2007) and apply an ℓ_1 type penalty on the off-diagonal entries of the inverse covariance matrices.

The rest of the paper is organized as follows. In the next two sections, we introduce the proposed penalized likelihood estimator and discuss how it can be efficiently computed in practice. Section 4 presents numerical studies to demonstrate the practical merits of the proposed method. Applications of the new method to model-based clustering and mixture discriminant analysis are discussed in Sections 5. We conclude with some comments and discussions in Section 6.

4.2 Methodology

To fix ideas, we start with the case when the number of clusters, M, is known apriori. In this case, the log-likelihood for a sample $X_1, \ldots, X_n$ of n independent copies of X is given by

$$\mathrm{L}(\text{data}|\Theta) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{M} \pi_k \phi(X_i|\mu_k, \Sigma_k) \right) \tag{38}$$

where $\Theta = \{(\pi_k, \mu_k, \Sigma_k) : k = 1, \ldots, M\}$ is the collection of all unknown parameters, and $\phi(\cdot|\mu, \Sigma)$ is the density function of a multivariate Gaussian distribution with mean vector μ_k and covariance matrix Σ_k.

The usual maximum likelihood estimator can be computed by maximizing $\mathrm{L}(\text{data}|\Theta)$ with respect to Θ. Without any constraints on the parameter, Θ includes a total of $Mp(p+1)/2$ free parameters, which can be prohibitive from both statistical and computational points of view when p is moderate or large when compared with the sample size n. To address this problem, we suggest to exploit potential sparsity in the covariance matrix. Sparsity can be found in multiple ways for covariance matrix estimation. In particular, we consider sparsity on the entries of the inverse covariance matrix. In the case of multivariate Gaussian distribution, the inverse covariance matrix collects all the partial correlations and a zero entry of the inverse covariance matrix corresponds to conditional independence between the corresponding variables given the remaining ones. Such relationship naturally connects with the so-called Gaussian graphical models (Whittaker, 1990; Lauritzen, 1996) and makes this type of sparsity particularly suitable for a lot of applications. Similar interpretation can also be given to the Gaussian mixture models where each cluster can be viewed as instances of a particular Gaussian graphical model. For such purpose, we suggest to use the following penalized likelihood estimate for Gaussian mixture models:

$$\hat{\Theta} := \mathrm{argmin}_{\mu_k, \Sigma_k \succ 0} \left\{ -\sum_{i=1}^{n} \log \left(\sum_{k=1}^{M} \pi_k \phi(X_i|\mu_k, \Sigma_k) \right) + \lambda \sum_{k=1}^{M} \|\Sigma_k^{-1}\|_{\ell_1} \right\}, \qquad (39)$$

where $\Sigma \succ 0$ indicates that Σ is a symmetric and positive definite matrix, $\lambda \geq 0$ is a tuning parameter, and $\|A\|_{\ell_1} = \sum_{i \neq j} |a_{ij}|$. Obviously, when $M = 1$ the estimate defined above reduces to the so-called graphLasso estimate of Yuan and Lin (2007). The single tuning parameter λ puts equal penalty on the inverse matrix, which is not unreasonable for clusters have comparable scales. Multiple tuning parameters allow different penalties but it dramatically increases the computational burden. A more general adaptive Lasso (Zou 2006) penalty can be extended to replace the Lasso

penalty in (39), which has not been studied in this paper yet.

Thus far, we have treated the number of clusters M and the tuning parameter λ as fixed. In practice, their choice is critical in determining the performance of our method. A commonly used strategy to choose these parameters is the multi-fold cross validation(CV). In CV, the data are first split into training and testing sets. For each pair of tuning parameters (M, λ), we compute the penalized likelihood estimate on the training data and then evaluate its performance on the testing data. Such split, estimation and evaluation are repeated many times to obtain a score for each pair of tuning parameters. The pair associated with the optimal score is then used for computing the final estimate based on all data. Despite its general applicability and competitive performance, a major drawback of CV is the intensive computation it requires. To overcome this problem, we suggest here a BIC type of criterion as an alternative to the CV score.

Following Yuan and Lin (2007), the degrees of freedom for each estimated covariance matrix using the ℓ_1 type of regularization can be approximated by the number of nonzero entries in the upper half of the inverse covariance matrix. Therefore, the total number of degrees of freedom can be approximated by

$$\mathrm{df}(M, \lambda) = \sum_{k=1}^{M} \left(p + \sum_{i \le j} I((\hat{C}_k)_{ij} \neq 0) \right), \tag{40}$$

where p represents the degrees of freedom associated with the unknown mean and $\hat{\Sigma}_k$ is the penalized likelihood estimate associated with tuning parameters (M, λ). Now for each pair of (M, λ), the corresponding BIC score function is defined as

$$\mathrm{BIC}(M, \lambda) = -\mathrm{L}(X|\hat{\Theta}(M, \lambda)) + \log(n)\mathrm{df}(M, \lambda). \tag{41}$$

Let $(\hat{M}, \hat{\lambda})$ be the pair with the smallest BIC score, we shall let $\hat{\Theta}(\hat{M}, \hat{\lambda})$ be our final estimate.

4.3 Computation

Direct computation of $\hat{\Theta}$ as defined by (39) can be quite complicated because the objective function is non-convex and the optimization problem is of rather high dimensionality. Fortunately, we show here that it can be efficiently done using an EM algorithm (Dempster et al., 1977). To this end, we consider the following "missing data" formulation. Let τ be a random variable indicating which cluster X comes from such that

$$X|\tau = k \sim \mathcal{N}(\mu_k, \Sigma_k) \tag{42}$$

and

$$P(\tau = k) = \pi_k, \qquad k = 1, \ldots, M. \tag{43}$$

If we can observe the "complete data" (X_i, τ_i), $i = 1, \ldots, n$, we can follow the same strategy as before and estimate Θ by the ℓ_1 penalized log-likelihood can be given by

$$\begin{aligned} \mathrm{PL}(\Theta) &= \sum_{i=1}^{n} \log\left(f(X_i|\tau_i)P(\tau_i)\right) + \lambda \sum_{k=1}^{M} \|\Sigma_k^{-1}\|_{\ell_1} \\ &= \sum_{i=1}^{n} \log\left(\phi(X_i|\mu_{\tau_i}, \Sigma_{\tau_i})\pi_{\tau_i}\right) + \lambda \sum_{k=1}^{M} \|\Sigma_k^{-1}\|_{\ell_1}. \end{aligned} \tag{44}$$

Now that we can only observe X_is, we may treat τ_is as missing data and the following EM algorithm can therefore be employed. We proceed in an iterative fashion. Each iteration consists of two steps, often referred to as the E step and M step respectively. Let $\Theta^{(t)}$ be the estimate of Θ at the tth iteration. In the E step, we compute the conditional expectation of τ_i given X_i and the current estimate of Θ. In particular, from Bayes rule,

$$\gamma_{ik}^{(t)} := P(\tau_i = k|X_i; \Theta^{(t)}) = \frac{\pi_k^{(t)}\phi(X_i|\mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{l=1}^{M} \pi_l^{(t)}\phi(X_i|\mu_l^{(t)}, \Sigma_l^{(t)})} \tag{45}$$

This leads to the construction of the so-called Q function

$$\begin{aligned} Q(\Theta, \Theta^{(t)}) &= \sum_{k=1}^{M} \left\{ \sum_{i=1}^{n} \log(\pi_k)\gamma_{ik}^{(t)} + \sum_{i=1}^{n} \log(\phi(X_i|\mu_k^{(t)}, \Sigma_k^{(t)}))\gamma_{ik}^{(t)} + \lambda\|\Sigma_k^{-1}\|_{\ell_1} \right\} \\ &=: \sum_{k=1}^{M} Q_k(\Theta_k, \Theta_k^{(t)}), \end{aligned}$$

where $\Theta_k = \{\pi_k, \mu_k, \Sigma_k\}$ and $\Theta^{(t)}$ is defined in a similar manner.

In the so-called M step, we update the estimate of Θ by maximizing the Q function, which can be done by maximizing Q_k with respect to Θ_k separately. More specifically, the updated value of Θ_k can be given by

$$\pi_k^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ik}^{(t)}, \tag{46}$$

and

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} \gamma_{ik}^{(t)} X_i}{\sum_{i=1}^{n} \gamma_{ik}^{(t)}}. \tag{47}$$

Moreover,

$$\Sigma_k^{(t+1)} = \text{argmin}_{\Sigma_k} \left\{ \log|\Sigma_k| + \text{tr}(\Sigma_k^{-1} A_k^{(t)}) + \lambda\|\Sigma_k^{-1}\|_{\ell_1} \right\} \tag{48}$$

where

$$A_k^{(t)} = \sum_{i=1}^{n} \left(\frac{\gamma_{ik}^{(t)}}{\sum_{j=1}^{n} \gamma_{jk}^{(t)}} \right) (X_i - \mu_k^{(t+1)})(X_i - \mu_k^{(t+1)})'.$$

The optimization problem of (48) are in a similar form as the graphLasso of Yuan and Lin (2007) and can be computed efficiently using a newly developed algorithm by Friedman et al. (2008).

This algorithm starts with an initial value $\Theta^{(0)}$ of the parameter. It can be decided either by the method of Banfield and Raftery, or we can randomly draw $1/M$ samples to estimate the mean and covariance for the clusters and set a equal proportion of clusters. To sum up, we have the following algorithm to compute $\hat{\Theta}$ as defined by (39).

Step 1: Initialize $\Theta^{(0)}$.

Step 2: For each iteration, update the estimate for each mixture component individually.

- E step: Calculate the distribution of unknown variables by (45).
- M step: update parameters by (47), (48), and (46).

Step 3: Go back to Step 2 until a certain convergence criterion is met.

Following the same argument as that of Dempster et al. (1977), it is not hard to see that in each iteration, the objective function of (39) decreases. Furthermore, the algorithm converges and converges to its minimizer, i.e., $\hat{\Theta}$.

4.4 Simulation Studies

To assess the finite sample performance of the proposed method, we now conduct several sets of simulation studies.

We begin with the case where the number of clusters is known in advance. In particular, we fix $M = 2$ in the first set of simulations. The sample size is set to be small at 100 whereas the dimension p is set to be 30, 50, 100, or 300. To evaluate the performance under larger sample size settings, we also fix p at 100 and set the sample size to be 200 and 400. The tuning parameter λ is determined either by 5-fold CV or the BIC criterion defined by (41). For simplicity, we fix the mean vector of each mixture component to be 0_p and three different covariance structures are considered.

Model 1 : In this case, the covariance matrix for both clusters follows an AR(1) model:

$$\Sigma_1(i,j) = 0.4^{|i-j|}; \qquad \Sigma_2(i,j) = 0.5 \times 0.8^{|i-j|}. \tag{49}$$

Model 2 : In this model, both covariance matrices are diagonal

$$\Sigma_1(j,j) = \log(j+1); \qquad \Sigma_2(j,j) = \log(p+2-j). \tag{50}$$

Model 3 : In the last model, the two covariance matrices follow AR(1) and AR(2) model respectively:

$$\Sigma_1^{-1}(i,j) = \begin{cases} 1 & \text{if } i = j \\ 0.2 & \text{if } |i-j| = 1 \\ 0 & \text{otherwise} \end{cases} ; \tag{51}$$

and

$$\Sigma_2^{-1}(i,j) = \begin{cases} 2 & \text{if } i = j \\ 0.25 & \text{if } |i-j| = 1 \\ 0.2 & \text{if } |i-j| = 2 \\ 0 & \text{otherwise} \end{cases} . \tag{52}$$

We compare the proposed estimate with the method of Banfield and Raftery (1993) and MLE if applicable. The method of Banfield and Raftery has been implemented in the R package mclust and the MLE can be computed using EM algorithm (see, e.g., McLachlan and Peel, 2000). We examine these estimate through several criteria. The averaged spectral norm of the difference between the estimating inverse covariance matrix and the truth

$$\text{SL} = \frac{1}{M}\sum_{k=1}^{M} \|\hat{\Sigma}_k^{-1} - \Sigma_k^{-1}\| \tag{53}$$

where $\|A\|$ is the largest singular value of matrix A; the averaged Frobenius norm of the difference

$$\text{FL} = \frac{1}{M}\sum_{k=1}^{M} \|\hat{\Sigma}_k^{-1} - \Sigma_k^{-1}\|_{\text{F}} = \frac{1}{M}\sum_{k=1}^{M} \sqrt{\sum_{i,j} (\hat{\Sigma}_k^{-1}(i,j) - \Sigma_k^{-1}(i,j))^2}; \tag{54}$$

and report the average Kullback-Leibler(KL) loss

$$\text{KL} = \frac{1}{M}\sum_{k=1}^{M} \text{KL}(\Sigma_k, \hat{\Sigma}_k), \tag{55}$$

where

$$\text{KL}(\Sigma, \hat{\Sigma}) = \text{tr}\left(\Sigma\hat{\Sigma}^{-1}\right) - \log\left|\Sigma\hat{\Sigma}^{-1}\right| - p. \tag{56}$$

The results, averaged over one hundred runs for each case, are reported in Table 17, 18 and 14 for the the three models respectively. It is clear from these results that the proposed method outperforms the other two methods for all three models. The superiority becomes more evident when the dimension increases. We also note the similar behavior of the penalized likelihood estimates tuned with either CV or BIC. This observation is of great practical importance because BIC is much more efficient to compute than the CV. For this reason, we shall use BIC as the tuning criterion in the rest of the paper unless otherwise indicated.

Table (15) shows how sparse the estimated matrix is in each cluster for all models. We now consider a more complicated setting where the number of clusters also needs to be selected. We consider the true number of clusters, M to be either 2 or 3. The sample size n is fixed at 100 whereas the dimension p is set to be 50. When $M = 2$, we used Model III as our data generating mechanism. When $M = 3$, the last cluster has the same covariance matrix as the first cluster in Model II. The experiment was repeated 100 times for each value of M. The proposed method with BIC as the tuning criterion correctly identifies the number of clusters in all runs. On the other hand, for the one hundred runs with two clusters, Mclust only identifies the correct M 47 out of 100 runs. When $M = 3$ it identified the correct number of clusters 56 out of 100 times. This information is summarized in Table (16). To gain further insight, we give in Figure 4.4 the smallest BIC scores for each value of the number of clusters for one typical simulated dataset with $M = 2$ and $M = 3$ respectively.

4.5 Applications

The proposed method for estimating high dimensional Gaussian mixture models could be useful for a variety of applications. For illustration purpose, we consider here in particular model-based clustering and the mixture discriminant analysis.

Table 12: Simulation results for model I: performance comparison of the proposed penalised likelihood estimator and traditional methods (Mclust, MLE). SL: error of the largest singular value of the inverse matrix; FL: Frobenius norm of the error; KL: Kullback-Leibler(KL) loss; TP: correctly identified nonzeros; TN: correctly identified zeros. Numbers in the table is averaged over one hundred runs and $(\cdot)$ represents the standard errors.

		Penalised Likelyhood											
		SL	FL	KL	SL	FL	KL	SL	FL	KL	SL	FL	KL
n	p		CV			BIC			Mclust			MLE	
100	30	2.53	6.87	5.7	3.27	9.1	9.68	3.26	10.39	17.59	31.14	43.42	40.81
		(0.032)	(0.096)	(0.143)	(0.071)	(0.113)	(0.366)	(0.006)	(0.022)	(0.069)	(1.077)	(1.101)	(1.376)
	50	2.81	10.59	11.44	3.14	11.56	15.7	3.38	13.94	29.25			
		(0.012)	(0.025)	(0.099)	(0.06)	(0.161)	(0.687)	(0.006)	(0.028)	(0.148)			
	100	2.61	13.8	18.44	3.75	18.68	44.3	3.41	20	59.12			
		(0.006)	(0.018)	(0.093)	(0.052)	(0.119)	(0.663)	(0.002)	(0.012)	(0.008)		NA	
	300	2.7	24.27	70.42	3.52	33.13	149.63	3.42	34.86	179.38			
		(0.014)	(0.15)	(1.205)	(0.016)	(0.139)	(2.874)	(0.001)	(0.015)	(0.015)			
200	100	2.02	10.04	9.75	2.77	15.23	20.49	3.42	20.06	59.08			
		(0.005)	(0.019)	(0.043)	(0.007)	(0.029)	(0.139)	(0.002)	(0.01)	(0.004)			
400	100	1.96	9.97	6.38	1.96	9.97	6.38	3.43	20.09	59.07	17.29	42.37	61.3
		(0.005)	(0.017)	(0.028)	(0.005)	(0.017)	(0.028)	(0.001)	(0.009)	(0.004)	(0.134)	(0.19)	(0.216)

Table 13: Simulation results for model II: performance comparison of the proposed penalised likelihood estimator and traditional methods (Mclust, MLE). SL: error of the largest singular value of the inverse matrix; FL: Frobenius norm of the error; KL: Kullback-Leibler(KL) loss; TP: correctly identified nonzeros; TN: correctly identified zeros. Numbers in the table is averaged over one hundred runs and $(\cdot)$ represents the standard errors.

		Penalised Likelyhood											
		SL	FL	KL	SL	FL	KL	SL	FL	KL	SL	FL	KL
n	p		CV			BIC			Mclust			MLE	
100	30	0.53	0.84	1.15	0.53	0.84	1.15	1.05	1.29	1.64	10.72	13.72	52.42
		(0.021)	(0.019)	(0.03)	(0.022)	(0.02)	(0.03)	(0.001)	(0.001)	(0.003)	(0.744)	(0.728)	(1.897)
	50	0.48	0.83	1.62	0.48	0.83	1.6	1.11	1.42	2.44			
		(0.018)	(0.016)	(0.034)	(0.018)	(0.016)	(0.034)	(0.001)	(0.001)	(0.004)			
	100	0.39	0.82	2.57	0.39	0.82	2.57	1.16	1.59	4.01			
		(0.011)	(0.009)	(0.034)	(0.011)	(0.009)	(0.034)	(0)	(0.001)	(0.005)		NA	
	300	0.39	1.02	7.06	0.39	1.02	7.06	1.23	1.83	8.45			
		(0.013)	(0.009)	(0.05)	(0.013)	(0.009)	(0.05)	(0)	(0)	(0.006)			
200	100	0.25	0.55	1.2	0.25	0.55	1.19	1.16	1.61	4.54			
		(0.01)	(0.008)	(0.015)	(0.01)	(0.008)	(0.015)	(0.002)	(0.006)	(0.167)			
400	100	0.17	0.36	0.56	0.17	0.36	0.56	1.17	1.59	3.97	3.46	7.42	74.21
		(0.005)	(0.004)	(0.006)	(0.005)	(0.004)	(0.006)	(0)	(0)	(0.001)	(0.033)	(0.03)	(0.277)

Table 14: Simulation results for model III: performance comparison of the proposed penalised likelihood estimator and traditional methods (Mclust, MLE). SL: error of the largest singular value of the inverse matrix; FL: Frobenius norm of the error; KL: Kullback-Leibler(KL) loss; TP: correctly identified nonzeros; TN: correctly identified zeros. Numbers in the table is averaged over one hundred runs and $(\cdot)$ represents the standard errors.

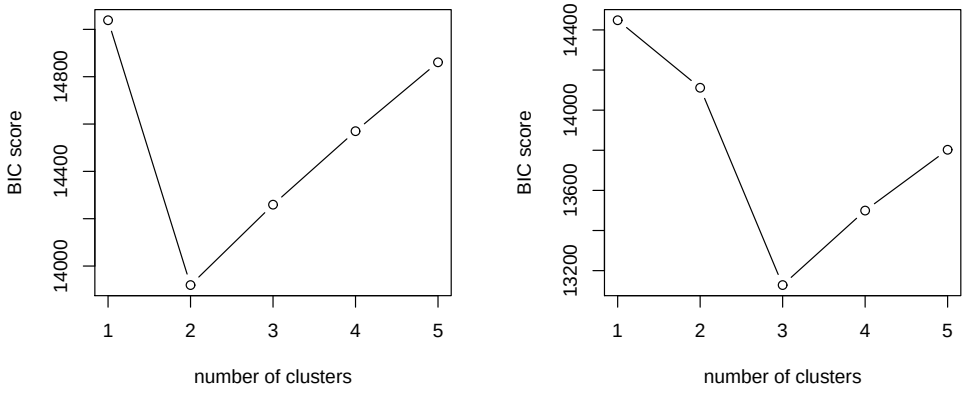
		Penalised Likelyhood											
		SL	FL	KL	SL	FL	KL	SL	FL	KL	SL	FL	KL
n	p		CV			BIC			Mclust			MLE	
100	30	1.01	2.6	1.61	3.27	9.1	9.68	1.21	4.06	5.59	22.68	31.71	40.26
		(0.019)	(0.017)	(0.016)	(0.071)	(0.113)	(0.366)	(0.048)	(0.195)	(0.439)	(0.923)	(0.929)	(1.301)
	50	1.09	3.38	2.73	3.14	11.56	15.7	1.2	5.07	7.47			
		(0.02)	(0.015)	(0.021)	(0.06)	(0.161)	(0.687)	(0.03)	(0.157)	(0.476)			
	100	1.15	4.78	5.49	3.75	18.68	44.3	1.18	7	12.53			
		(0.016)	(0.015)	(0.028)	(0.052)	(0.119)	(0.663)	(0.002)	(0.017)	(0.074)		NA	
	300	1.38	8.34	16.61	3.52	33.13	149.63	1.18	12.17	37.86			
		(0.018)	(0.017)	(0.048)	(0.016)	(0.139)	(2.874)	(0.002)	(0.027)	(0.204)			
200	100	0.88	4.15	4.02	2.77	15.23	20.49	1.19	7.09	12.92			
		(0.005)	(0.007)	(0.02)	(0.007)	(0.029)	(0.139)	(0.002)	(0.012)	(0.054)			
400	100	0.79	3.65	2.86	1.96	9.97	6.38	1.2	7.12	13.07	12.56	31.04	64.5
		(0.002)	(0.009)	(0.023)	(0.005)	(0.017)	(0.028)	(0.002)	(0.009)	(0.04)	(0.09)	(0.122)	(0.22)

Table 15: Averaged percentage of zeros in the estimated inverse matrix: We calculate the averaged sparsity over two clusters and the reported number is the mean over 100 runs; $(\cdot)$ represents the standard errors. Note that (0) is because of rounding, not truly 0.

		model I		model II		model III	
n	p	CV	BIC	CV	BIC	CV	BIC
100	30	65.35	77.07	89.84	90.22	94.98	77.91
		(0.502)	(0.532)	(0.064)	(0)	(0.263)	(0.112)
	50	81.76	85.15	96.14	96.48	97.34	80.7
		(0.138)	(0.489)	(0.085)	(0)	(0.159)	(0.194)
	100	87.75	94.24	99.1	99.12	98.92	88.22
		(0.091)	(0.168)	(0.005)	(0)	(0.008)	(0.049)
	300	91.9	98.37	99.9	99.9	99.65	94.17
		(0.113)	(0.112)	(0)	(0)	(0.004)	(0.027)
200	100	80.57	90.44	99.07	99.12	97.87	99
		(0.088)	(0.061)	(0.022)	(0)	(0.064)	(0)
400	100	87.31	87.31	99.1	99.12	93.83	98.99
		(0.07)	(0.07)	(0.003)	(0)	(0.166)	(0.007)

Table 16: Frequency of correctly identified number of clusters over 100 runs. Sample size n is set at 100 and p 50.

	Number of Clusters	
Methods	2	3
PLE	100	100
Mclust	47	56



(a) $M = 2$ (b) $M = 3$

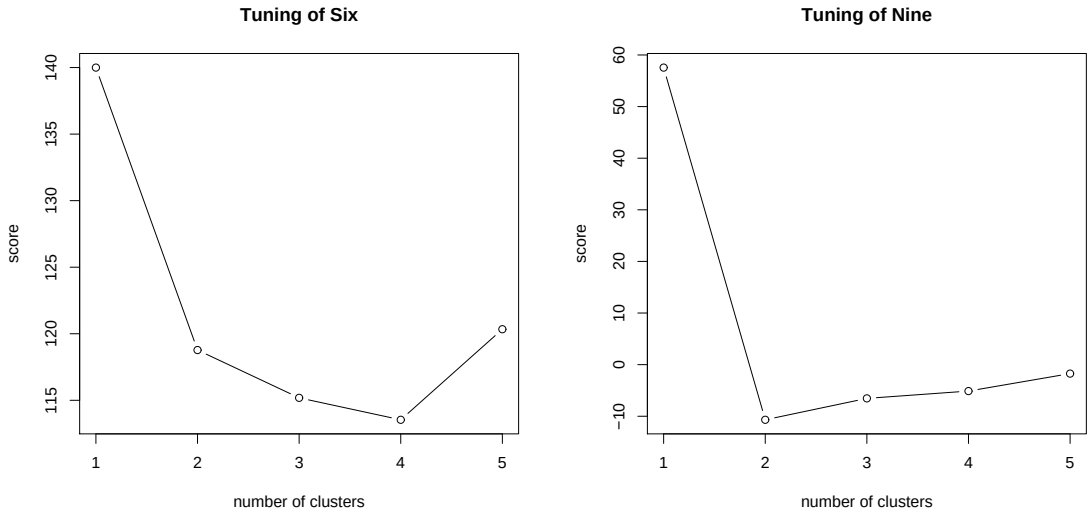
Figure 13: BIC score vs the number of clusters.

4.5.1 Model-based Clustering

As described previously, Gaussian mixture models have been one of the more popular tools for clustering (Fraley and Raftery, 2002). It provides a principled statistical approach to the practical questions arising in clustering. To demonstrate the potential of our method in clustering high dimensional inputs, we apply it to the handwritten digit data (LeCun et al., 1990). The data set consists of scanned digits from handwritten zip code on envelopes and was collected by the United States Postal Service. Every handwritten digit image has been digitalized to a 16×16 image with intensity value of each pixel normalized to range from -1 and 1. To fix idea, we focus on digits 6 and 9. There are a total of 834 images of digit 6 and 821 of digit 9. We fit for each digit a Gaussian mixture model using the proposed method with both the number of clusters and the tuning parameter λ jointly chosen by minimizing the BIC score as discussed earlier. The minimal BIC score associated with each value of the number of clusters is given in Figure 4.5.1, which suggests that there are four clusters for digit 6 whereas only two for digit 9. To gain further insight, we give in Figure 4.5.1 typical examples from each cluster which shows that the clustering based on our method is indeed meaningful.

4.5.2 Mixture Discriminant Analysis

We now turn our attention to classification where the mixture discriminant analysis (MDA) introduced by Hastie and Tibshirani (1996) provides a much more flexible alternative to linear or quadratic discriminant analysis. The basic idea here is to model each class distribution using a Gaussian mixture model and then classify an instance according to Bayes rule. Unlike the usual linear or quadratic discriminant analysis, MDA is able to produce more general nonlinear classification boundaries. The main difficulty of using MDA in classification with high dimensional inputs remains how to fit high dimensional Gaussian mixture models where our method could be a valuable



(a) BIC score for digit 6

(b) BIC score for digit 9

Figure 14: Selecting the number of clusters for handwritten digit data.

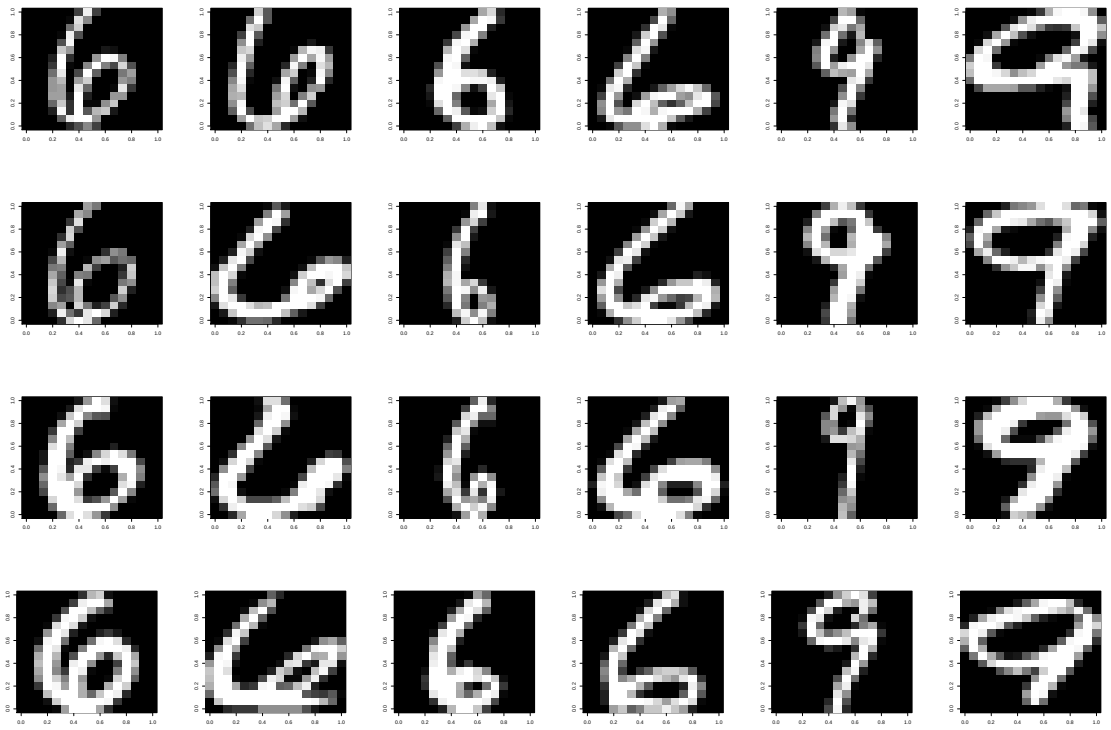


Figure 15: Clustering of digit 6 and 9: images from each column are randomly chosen from a particular cluster, i.e., the first four columns correspond to the four selected clusters of digit 6 and the last two correspond to digit 9.

tool. To demonstrate the merit of such practice, we now apply this strategy to the handwritten digit data. Similar to before, we focus on digits 6 and 9 and investigate automatic classification between these two classes. For evaluation purpose, we randomly select 80% of combined images as the training set and use the remaining 20% as testing set. Gaussian mixture models were fit with tuning parameters determined by BIC for digit 6 and 9 respectively on the training set and the resulting classifier is applied on the testing data to obtain a test error. This procedure was repeated, i.e., splitting the data, fitting the mixture model and evaluating the test error, for one hundred times. With the proposed method, the average test error rate is 0.26% with a standard error 0.029%. Note that direct maximum likelihood estimate as employed in the original mixture discriminant analysis is rather unstable for this example due to its high dimensionality. Generalization with the proposals of Banfied and Raftery (1993) has been investigated by Fraley and Raftery (2002; 2007) and implemented in R. For comparison purpose, we ran similar analysis using this method as well. It yields an error rate of 0.42% with a standard error 0.03%.

For illustration purpose of how sparse the inverse matrix is, we evaluate the percentage of zeros in every cluster. The experiment is based on one run of 80% samples. The four clusters of digit 6 has (68%, 30%, 78%, 66%) zeros respectively and digit 9 has (72%, 33%), which confirm the image display in Figure 4.5.1.

4.6 Discussions

In this paper, we have developed a penalized likelihood estimator for high dimensional Gaussian graphical models. By imposing an ℓ_1 penalty on the inverse covariance matrices, the proposed estimator encourages sparsity and therefore could be useful for high dimensional cases. We show that the estimate can be efficiently computed by an EM algorithm. Simulation studies show that the method is quite promising in extending the scope of Gaussian mixture model in handling high dimensional data.

Its usefulness is further assessed in the context of model based clustering and mixture discriminant analysis.

CHAPTER V

HIGH DIMENSIONAL STRUCTURED GAUSSIAN MIXTURE MODELS

5.1 Introduction

As discussed in the previous chapter, Gaussian mixture models has attracted a lot of attention in a variety of applications with its flexibility. The k-th ($k = 1, \ldots, M$) cluster in the GMMs corresponds to a Gaussian graphical model (GGM), which represents a network of the coordinates in a random vector X. Essentially, each network is an undirected graph $G = (V, E)$, where $V = \{1, 2, \ldots, p\}$ is a vertex set of the variables $(X^{(1)}, \ldots, X^{(p)})$; $E = (e_{ij})_{1 \leq i < j \leq p}$ is an edge set describing the conditional independence among the variables. The absent edge between vertices $X^{(i)}$ and $X^{(j)}$ indicates that they are conditional independent given all other vertices and corresponds to $C_{ijk} = 0$ (Whittaker, 1990; Lauritzen, 1996).

The graphical presentation of every cluster may have the same or similar structure, which is an important feature in many applications. For example, in image processing, the pixels in a grey-scale image exhibit a conditional local correlation. Usually, two pixels with large distance are believed to be non-correlated given all other pixels, while nearby pixels are more likely to be correlated given others. Such conditional correlation is a network in GGMs and it may be shared by similar images. Figure 5.1 displays two images of the handwritten digits of 6, where the large overlap of the pixel areas is a strong evidence of the common structures. A second example is in speech signals, where filter banks discretizes continuous signals into overlap bins from the low frequency to high frequency. Because of the overlap, there is a strong local correlation between the energies in neighboring filter banks. A third example

is genetic pathways. Individual genes are presented as nodes or vertices. An edge between two genes refers to the informational interactions. Such gene network may be shared by different cells.

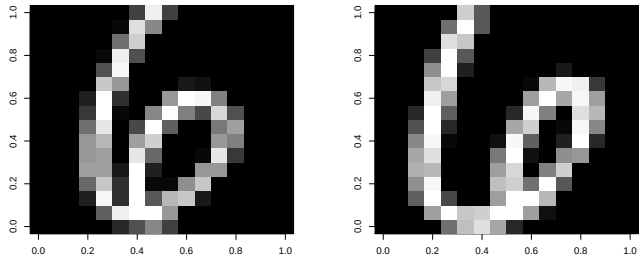


Figure 16: Display of two images of hand written digit 6. The over lap area of the pixels indicate sharing structures of the two images

The estimate of similar graphs among clusters is equivalent to estimate the inverse matrices C_{ijk} with similar structures. To our knowledge, there has been no such discussions in the context of GMMs. Instead, estimation to impose geometric features (shape, volume, orientation) of the clusters, are proposed in Banfield and Raftery (1993). Celeux and Govaert (1995) enumerate extensive models within the framework. Flury et al. (1994) also discuss the similar idea in discriminant analysis. In particular, Celeux and Govaert (1995) consider several forms of parametrization on the covariance matrix Σ_k. One is eigenvalue decomposition

$$\Sigma_k = \lambda_k D_k A_k D_k^{'}$$

where $\lambda_k = |\Sigma_k|^{1/p}$ controls the volume of the kth cluster; D_k is the eigenvector matrix of Σ_k representing the orientation; $A_k(|A_k| = 1)$ is a diagonal matrix with the normalized eigenvalues in a descending order, determining its shape. The second way is

$$\Sigma_k = \lambda_k B_k,$$

where $|B_k| = 1$. And the last one assumes spherical shapes,

$$\Sigma_k = \lambda_k I.$$

However, in high dimensional cases, such parametrization can not remedy the challenge of parameter estimation. For moderate restrictions allowing some of the quantities to vary or be equal among clusters, the parameter size is still in the order of p^2 or p, which is a concern for large p as discussed in the previous chapter. For extreme restriction of the third case, although the parameter size is reduced a lot, no flexibilities persists in the clusters, which deteriorates the accuracy of the estimators. We have proposed a Lasso-type estimator proposed in the previous chapter to impose sparsity on the concentration matrix. It has been shown to work well in the high dimensional settings but it does not encourage structure sharing of the clusters.

This necessity prompts us to consider new estimators. The philosophy of the proposed structured estimators is related to the parameter estimation in regression (Yuan and Lin 2006; Huang et al. 2009; Zhou and Zhu 2010), and multivariate graphical models (Guo et al. 2009). Variables within the same group are either zero or nonzero in the discussion of Yuan and Lin (2006). Huang et al. (2009) and Zhou and Zhu (2010) propose that even if the group is identified as significant, variables in the same group can also be regularized to be zero. Guo et al. (2009) consider a joint estimator with similar structures of the inverse matrices for independent Gaussian samples. For our purpose, we propose two estimators, a hierarchical Lasso estimator and a group Lasso estimator with different penalties imposed on the inverse matrix. This research is still ongoing and currently we focus more on the introduction of the methodology and we show part of the simulations in section three. Future work and conclusions are presented in the end.

5.2 Methodology

Suppose $Y_1, \ldots, Y_n$ are n independent and identically distributed samples from a p-dimensional Gaussian mixture distribution with M mixtures. The k-th ($k = 1, \ldots, M$) mixture follows a Gaussian distribution with mean μ_k and inverse covariance matrix C_k. The log-likelihood is given by

$$\mathrm{L}(\text{data}|\Theta) = \sum_{m=1}^{n} \log \left(\sum_{k=1}^{M} \pi_k \phi(Y_m|\mu_k, C_k) \right) \tag{57}$$

where $\phi(\cdot|\mu, C)$ is the Gaussian density function.

Denote $\Theta = \{(\pi_k, \mu_k, C_k) : k = 1, \ldots, M\}$ as the set of all unknown parameters. The usual maximum likelihood is computed by maximizing $\mathrm{L}(\text{data}|\Theta)$ with respect to Θ, which is of size $Mp(p+1)/2$ in optimization. Compared with the sample size n, when p is moderate or large compared, the computation of the parameter estimate is infeasible.

In the previous chapter, we have shown the benefit of a sparse estimator with an ℓ_1 penalty. In that approach, the maximizing of (57) is decomposed by individually maximizing the likelihood of each mixture. In such a way, the parameters in every mixture are estimated separately. We call this approach as separate Lasso estimator and use it as a benchmark to compare with the new estimator.

5.2.1 Hierarchical Lasso Estimator

When the mixture graphical models may share common structures and keep their unique structures at the same time, the separate estimator is not efficient to capture the underlying structure. With the success of hierarchical estimator in variable selection of regression (Zhou and Zhu, 2010), we propose the following hierarchical Lasso

estimator for the inverse matrices

$$\begin{aligned}
\min_{\mu_k, C_k \succ 0, 1\le k\le M} \quad & -\sum_{m=1}^{n} \log\left(\sum_{k=1}^{M} \pi_k \phi(Y_m|\mu_k, C_k)\right) + \lambda_1 \sum_{i\neq j} \theta_{ij} + \lambda_2 \sum_{i\neq j}\sum_{k=1}^{M} |\omega_{ijk}|, \\
\text{subject to} \quad & C_{ijk} = \theta_{ij}\omega_{ijk}, \quad \theta_{ij} > 0, \quad 1 \le i, j \le p, \\
& \theta_{ij} = \theta_{ji}, \quad \omega_{ijk} = \omega_{jik}, 1 \le i \neq j \le p; 1 \le k \le M, \\
& \theta_{jj} = 1, \omega_{jjk} = c_{jjk}, 1 \le j \le p; 1 \le k \le M, \qquad (58)
\end{aligned}$$

where $C_k \succ 0$ requires that C_k is positive definite, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are tuning parameters. λ_1 controls the sparsity of the common shrinkage factor θ_{ij}'s. If $\theta_{ij} = 0$, then there is no link between node i and j for all precision matrices. Given $\theta_{ij} > 0$, λ_2 governs the sparsity of ω_{ijk} and a zero ω_{ijk} leads to $C_{ijk} = 0$. This double penalties allow the mixtures to have similar and different structures.

In the previous chapter, we have shown that an EM algorithm can conveniently solve the problem by introducing hidden variables τ_m, where τ_m is a random variable indicating the cluster label of Y_m. Then incomplete data is expanded into the "complete data" (Y_m, τ_m), $m = 1, \ldots, n$. Suppose the distribution of τ_m is

$$P(\tau_m = k) = \pi_k, \qquad k = 1, \ldots, M. \qquad (59)$$

Now the complete log-likelihood is

$$\begin{aligned}
\text{PL}(\Theta) &= -\sum_{m=1}^{n} \log\left(f(Y_m|\tau_m)P(\tau_m)\right) + \lambda_1 \sum_{i\neq j} \theta_{ij} + \lambda_2 \sum_{i\neq j}\sum_{k=1}^{M} |\omega_{ijk}| \\
&= -\sum_{m=1}^{n} \log\left(\phi(Y_m|\mu_{\tau_m}, C_{\tau_m})\pi_{\tau_m}\right) + \lambda_1 \sum_{i\neq j} \theta_{ij} + \lambda_2 \sum_{i\neq j}\sum_{k=1}^{M} |\omega_{ijk}|. \qquad (60)
\end{aligned}$$

Here, we do not repeat the constraints in (58) for brevity and we will omit it everywhere necessary in the rest of this chapter. Denote $\Theta^{(t)}$ as the estimate at the t-th iteration, we apply EM algorithm to estimate the parameters. In the E step, we compute the conditional expectation of τ_m given Y_m and $\Theta^{(t)}$, i.e., the so-called Q function. By Bayes rule,

$$\gamma_{mk}^{(t)} := P(\tau_m = k|Y_m; \Theta^{(t)}) = \frac{\pi_k^{(t)} \phi(Y_m|\mu_k^{(t)}, C_k^{(t)})}{\sum_{l=1}^{M} \pi_l^{(t)} \phi(Y_m|\mu_l^{(t)}, C_l^{(t)})} \qquad (61)$$

And the Q function is

$$Q(\Theta, \Theta^{(t)}) = - \sum_{k=1}^{M} \left\{ \sum_{m=1}^{n} \log(\pi_k) \gamma_{mk}^{(t)} + \sum_{m=1}^{n} \log(\phi(Y_m | \mu_k^{(t)}, C_k^{(t)})) \gamma_{mk}^{(t)} \right\} + \lambda_1 \sum_{i \neq j} \theta_{ij} + \lambda_2 \sum_{i \neq j} \sum_{k=1}^{M} |\omega_{ijk}| \quad (62)$$

In M step, we update the estimate of Θ by minimizing the Q function. It can be shown that the update of π_k and μ_k is defined by

$$\pi_k^{(t+1)} = \frac{1}{n} \sum_{m=1}^{n} \gamma_{mk}^{(t)}, \quad (63)$$

and

$$\mu_k^{(t+1)} = \frac{\sum_{m=1}^{n} \gamma_{mk}^{(t)} Y_m}{\sum_{m=1}^{n} \gamma_{mk}^{(t)}}, \quad (64)$$

respectively. And the update of $C_k, k = 1, \ldots, M$ is

$$\min_{C_k \succ 0, k=1,\ldots,M} \sum_{k=1}^{M} \left\{ -\bar{\gamma}_k^{(t)} \log |C_k| + \mathrm{tr}(C_k A_k^{(t)}) \right\} + \lambda_1 \sum_{i \neq j} \theta_{ij} + \lambda_2 \sum_{i \neq j} \sum_{k=1}^{M} |\omega_{ijk}| \quad (65)$$

where

$$\bar{\gamma}_k^{(t)} = \frac{1}{n} \sum_{m=1}^{n} \gamma_{mk}^{(t)},$$

and

$$A_k^{(t)} = \sum_{m=1}^{n} \frac{1}{n} \left\{ \gamma_{mk}^{(t)} (Y_m - \mu_k^{(t+1)})(Y_m - \mu_k^{(t+1)})' \right\}.$$

The double tuning in (65) requires comprehensive computation. Fortunately, it can be shown that (65) can be replaced to a single tuning optimization where $\lambda = \lambda_1 \lambda_2$

$$\begin{aligned} \min_{C_k \succ 0, k=1,\ldots,M} \quad & \sum_{k=1}^{M} \left\{ -\bar{\gamma}_k^{(t)} \log |C_k| + \mathrm{tr}(C_k A_k^{(t)}) \right\} + \sum_{i \neq j} \theta_{ij} + \lambda \sum_{i \neq j} \sum_{k=1}^{M} |\omega_{ijk}| \\ \text{subject to} \quad & C_{ijk} = \theta_{ij} \omega_{ijk}, \quad \theta_{ij} > 0, \quad 1 \leq i, j \leq p, \\ & \theta_{ij} = \theta_{ji}, \quad \omega_{ijk} = \omega_{jik}, 1 \leq i \neq j \leq p; 1 \leq k \leq M \\ & \theta_{jj} = 1, \omega_{jjk} = c_{jjk}, 1 \leq j \leq p; 1 \leq k \leq M, \end{aligned} \quad (66)$$

Moreover, (66) has a more convenient computational form

$$\min_{C_k \succ 0, k=1,\ldots,M} \sum_{k=1}^{M} \left\{ -\bar{\gamma}_k^{(t)} \log |C_k| + \mathrm{tr}(C_k A_k^{(t)}) \right\} + \lambda \sum_{i \neq j} \sqrt{\sum_{k=1}^{M} |C_{ijk}|} \quad (67)$$

However, the update of C_k is complicated due to the nonlinearity of the penalty term. We approximate the penalty by the following local linear approximation (LLA) (Zou and Li, 2008)

$$\sqrt{\sum_{k=1}^{M} |C_{ijk}|} \approx \frac{\sum_{k=1}^{M} |C_{ijk}|}{\sqrt{\sum_{k=1}^{M} |C_{ijk}^{(t)}|}}. \tag{68}$$

Now the update of C_k's can be decomposed into the individual update of each C_k

$$C_k^{(t+1)} = \text{argmin}_{C_k} \left\{ -\log |C_k| + \text{tr}\left(C_k \frac{A_k^{(t)}}{\bar{\gamma}_k^{(t)}} \right) + \lambda \sum_{i<j} \frac{|C_{ijk}|}{\bar{\gamma}_k^{(t)} \sqrt{\sum_{k=1}^{M} |C_{ijk}^{(t)}|}} \right\} \tag{69}$$

Note that (69) is the nonnegative garrote-type estimator in the graphLasso of Yuan and Lin (2007) and can be solved by the algorithm in Friedman et al. (2008).

As a summary, the algorithm to compute $\hat{\Theta}$ defined by (58) is

Step 1: Initialize $\Theta^{(0)}$.

Step 2: For each iteration, update the estimate for each mixture component individually.

- E step: Calculate the distribution of unknown variables by (61).
- M step: update parameters by (64), (69), and (63).

Step 3: Go back to Step 2 until a certain convergence criterion is met.

5.2.2 Group Lasso Estimator

If the clusters share a same structure, a group Lasso estimator of the covariance matrices can be formulated to return an estimator with the same structure for all clusters. The idea of group lasso is discussed in the context of regression (Yuan and Lin, 2006). Similarly, a group lasso estimator of the inverse matrix is defined as

$$\hat{\Theta} := \text{argmin}_{\mu_k, C_k \succ 0} - \sum_{m=1}^{n} \log \left(\sum_{k=1}^{M} \pi_k \phi(Y_m | \mu_k, C_k) \right) + \lambda \sum_{i<j} \left(\sum_{k=1}^{M} M C_{ijk}^2 \right)^{\frac{1}{2}}, \tag{70}$$

Similar to before, the update of π_k and μ_k is defined by (63) and (64). And the update of $C_k, k = 1, \ldots, M$ is determined by

$$\min_{C_k \succ 0, k=1,\ldots,M} \sum_{k=1}^{M} \left\{ -\bar{\gamma}_k^{(t)} \log |C_k| + \mathrm{tr}(C_k A_k^{(t)}) \right\} + \lambda \sum_{i<j} \left(\sum_{k=1}^{M} M C_{ijk}^2 \right)^{\frac{1}{2}}, \tag{71}$$

Clearly, there is no close form estimate and one computational algorithm should show how the constraint is satisfied. Let's consider a subproblem where every C_k is known up to its j-th column/row. By changing row and column, without loss of generality, we can always assume that the unknown column/row is the last column/row. Thus C_k can be represented as follows

$$C_k = \begin{pmatrix} C_{-p,-p,k} & C_{-p,p,k} \\ C_{-p,p,k}^T & C_{p,p,k} \end{pmatrix}$$

where $C_{-p,-p,k}$ is known. Suppose $C_{-p,-p,k}$ is positive definite, then C_k is positive definite if and only if

$$C_{p,p,k} - C_{-p,p,k}^T C_{-p,-p,k}^{-1} C_{-p,p,k} > 0$$

Therefore the positive definiteness of each C_k is guaranteed if we update the jth column/row one a time. And the semidefinite program of (71) reduces to the following optimization problem

$$\begin{aligned} \min_{C_{.p.}} \quad & \sum_{k=1}^{M} \left\{ -\bar{\gamma}_k^{(t)} \log |C_k| + \mathrm{tr}(\mathrm{C_k A_k^{(t)}}) \right\} + \lambda \sum_{i<j} \left(\sum_{k=1}^{M} M C_{ijk}^2 \right)^{\frac{1}{2}}, \\ \text{subject to} \quad & C_{p,p,k} - C_{-p,p,k}^T C_{-p,-p,k}^{-1} C_{-p,p,k} > 0, \text{for any } k = 1, \ldots, M, \end{aligned} \tag{72}$$

where $C_{.p.} = C_{p..}$ is the pth column/row in all C_ks.

Note that

$$|C_k| = |C_{-p,-p,k}|(C_{p,p,k} - C_{-p,p,k}^T C_{-p,-p,k}^{-1} C_{-p,p,k}) \tag{73}$$

$$\mathrm{tr}(\mathrm{C_k A_k^{(t)}}) = \mathrm{tr}(\mathrm{C_{-p,-p,k} A_{-p,-p,k}^{(t)}}) + 2\mathrm{C_{-p,p,k}^T A_{-p,p,k}^{(t)}} + \mathrm{C_{p,p,k} A_{p,p,k}^{(t)}} \tag{74}$$

Plug (73) and (74) into (72) and we denote the derived objective function as h, take a first order condition for $C_{p,p,k}$, we have

$$C_{p,p,k} = \frac{\bar{\gamma}_k}{A_{p,p,k}} + C_{-p,p,k}^T C_{-p,-p,k}^{-1} C_{-p,pk} \tag{75}$$

Taking (75) back into h and remove the constant terms, $C_{-p,p,k}$ minimizes

$$\frac{1}{2}\sum_{k=1}^{M} C_{-p,p,k}^{T}\left(A_{p,p,k}C_{-p,-p,k}^{-1}\right)C_{-p,p,k}+\sum_{k=1}^{M} C_{-p,p,k}^{T}A_{-p,p,k}+\lambda\sum_{i<p}\left(\sum_{k=1}^{M} MC_{i,p,k}^{2}\right)^{\frac{1}{2}} \quad (76)$$

With the nonlinearity of the penalty term, (76) is not straightforward to get $C_{-p,p,k}$. Consider a subproblem of (76), where $C_{-p,p,k}$ is known except its i-th row. Denote $x_i = (C_{i,p,1},\ldots,C_{i,p,M})^T$ and $x_{-i} = (C_{-(i,p),p,1},\ldots,C_{-(i,p),p,M})^T$, where $C_{-(i,p),p,k}$ is the p-th column without the i- and p-th rows. Then $C_{-p,p,k}$s can be represented by a long vector (x_i, x_{-i}) and the first two terms in (76) can be rewritten as the vector form

$$\frac{1}{2}\left[x_i^T, x_{-i}^T\right]\begin{bmatrix}\alpha_{11} & \alpha_{12}\\ \alpha_{12}^T & \alpha_{22}\end{bmatrix}\begin{bmatrix}x_i\\ x_{-i}\end{bmatrix}+\left[x_i^T, x_{-i}^T\right]\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix},$$

Up to a constant not depending on x_i, the subproblem of (76) reduces to

$$\min_{x_i}\frac{1}{2}x_i^T\alpha_{11}x_i+\left(x_{-i}^T\alpha_{12}^T+\beta_1^T\right)x_i+\lambda\sqrt{Mx_i^Tx_i}, \quad (77)$$

A direct consequence of the KarushKuhnTucker (KKT) condition indicates that a necessary and sufficient condition for x_i to be a solution to expression (77) is

$$x_i=\left(\alpha_{11}+\frac{\lambda\sqrt{M}I_M}{\|x_i\|}\right)^{-1}s_i, \quad x_i\neq 0, \quad (78)$$

$$x_i = 0, \quad (79)$$

where α_{11} is an $M\times M$ diagonal matrix whose (k,k) entry is

$$\alpha_{11}(k,k) = A_{p,p,k}[C_{-p,-p,k}^{-1}](i,i). \quad (80)$$

$$\begin{aligned} s_i &= -(\alpha_{12}x_{-i}+\beta_1)\\ &= A_{p,p,k}[C_{-p,-p,k}^{-1}](i,-i)C_{-(i,p),p,k}+A_{i,p,k}. \end{aligned} \quad (81)$$

$\|x_i\|$ is the root of

$$1=\sum_{k=1}^{M}\frac{S_{ik}^2}{(\alpha_{11k}\|X_i\|+\lambda\sqrt{M})^2} \quad (82)$$

when (82) has no valid root, (79) is the solution.

To sum up, the group Lasso estimate can be obtained by the iterative algorithm

Step 1: Initialize $\Theta^{(0)}$.

Step 2: For each iteration,

- E step: Calculate the distribution of unknown variables by (61).
- M step:
  - update the mean parameters by (64);
  - for $j = 1, \ldots, p$, update the j-th column for all C_k's;
    - for $i = 1, \ldots, p-1$, update $C_{i,p,k}$ by (78) or (79).
    - update $C_{p,p,k}$ by (75);

Step 3: Go back to Step 2 until a certain convergence criterion is met.

So far, we have discussed the algorithm based on a fixed tuning parameter λ. The tuning parameter choice is critical to the performance. We adapt the commonly used multi-fold cross validation(CV) and BIC criterion as discussed before.

5.3 Simulation Studies

We now conduct simulation studies to assess the performance of the proposed estimator. We compare the performance of the hierarchical Lasso (HLasso) estimator with separate Lasso (SLasso) estimator and Banfield and Raftery (1993) (Mclust). We treat the number of mixtures as known.

5.3.1 Common Structures

Suppose $M = 2$ is known and the sample size is set to be 100 where the dimension p is set to be 30, 50 and 100. The tuning parameter λ is determined either by BIC criterion defined by (41) or cross validation. For simplicity, we fix the mean vector of each mixture component to be 0_p. First, we consider the case when the inverse matrices of all clusters have common structures.

Model 1 : AR(1) model:

$$\begin{aligned} C_1(i,i) &= 1.36; C_1(i,i+1) = C_1(i+1,i) = -0.6; C_1(i,j) = 0, |j-i| > 1; \\ C_2(i,i) &= 1.81; C_2(i,i+1) = C_2(i+1,i) = -0.9; C_2(i,j) = 0, |j-i| > 1. \end{aligned}$$

Model 2 : AR(2) model:

$$\begin{aligned} C_1(i,i) &= 1.32; C_1(i,i+1) = C_1(i+1,i) = -0.56; \\ C_1(i,i+2) &= C_1(i+2,i) = 0.4; C_1(i,j) = 0, |j-i| > 2; \\ C_2(i,i) &= 4.56; C_2(i,i+1) = C_2(i+1,i) = -2.88; \\ C_2(i,i+2) &= C_2(i+2,i) = 1.6; C_2(i,j) = 0, |j-i| > 2. \end{aligned}$$

We examine these estimates through the criteria defined before as SL, FL and KL. To show that the penalised estimator can capture the structure of the concentration matrix, we also report the percentage of correctly identified nonzeros (TP) and zeros (TN).

The results, averaged over one hundred runs for each case, are reported in Table 17 and 18 respectively. It is clear from these results that the proposed method outperforms the other two methods. The superiority becomes more evident when the dimension increases. We also note the similar behavior of the penalized likelihood estimates tuned with either CV or BIC. This observation is of great practical importance because BIC is much more efficient to compute than the CV. For this reason, we shall use BIC as the tuning criterion in the rest of the paper unless otherwise indicated.

5.3.2 With individual Structures

Now we contaminate the inverse matrices of model I and II by putting one or two individual edges for each of the cluster respectively.

Model I.1 : Model I with 1 individual edge added.

$$C_1(j,j+2) = C_1(j+2,j) \neq 0, C_2(j,j+3) = C_2(j+3,j) \neq 0;$$

Table 17: Simulation results for model I: averaged over one hundred runs, the numbers in parentheses are the standard errors.

		BIC					CV				
p	Method	SL	FL	KL	TP	TN	SL	FL	KL	TP	TN
30	SLasso	3.27	9.1	9.68	0.85	0.84	2.53	6.87	5.7	0.95	0.72
		(0.071)	(0.113)	(0.366)	(0.007)	(0.005)	(0.032)	(0.096)	(0.143)	(0.004)	(0.005)
	HLasso	1.94	3.59	1.6	1	1	1.66	3.42	1.67	0.99	1
		(0.056)	(0.06)	(0.031)	(0)	(0)	(0.041)	(0.058)	(0.043)	(0.002)	(0)
	Mclust	3.26	10.39	17.59	0.34	1					
		(0.006)	(0.022)	(0.069)	(0)	(0)					
50	SLasso	3.14	11.56	15.7	0.85	0.9	2.81	10.59	11.44	0.91	0.86
		(0.06)	(0.161)	(0.687)	(0.01)	(0.005)	(0.012)	(0.025)	(0.099)	(0.003)	(0.001)
	HLasso	1.81	4.12	2.35	1	1	1.55	4.27	2.57	1	1
		(0.04)	(0.041)	(0.031)	(0)	(0)	(0.023)	(0.065)	(0.071)	(0.001)	(0)
	Mclust	3.38	13.94	29.25	0.34	1					
		(0.006)	(0.028)	(0.148)	(0)	(0)					
100	SLasso	3.75	18.68	44.3	0.75	0.96	2.61	13.8	18.44	0.95	0.9
		(0.052)	(0.119)	(0.663)	(0.005)	(0.002)	(0.006)	(0.018)	(0.093)	(0.001)	(0.001)
	HLasso	1.95	5.64	4.54	1	1	1.62	5.79	4.69	1	1
		(0.035)	(0.04)	(0.044)	(0)	(0)	(0.018)	(0.029)	(0.041)	(0)	(0)
	Mclust	3.41	20	59.12	0.34	1					
		(0.002)	(0.012)	(0.008)	(0)	(0)					
300	SLasso	3.65	26.52	90.14	0.73	0.98	2.7	24.27	70.42	0.95	0.93
		(0.032)	(0.059)	(1.08)	(0.009)	(0.001)	(0.014)	(0.15)	(1.205)	(0.003)	(0.001)
	HLasso	1.76	8.22	9.31	1	1	1.82	10.09	14.08	1	1
		(0.016)	(0.034)	(0.061)	(0)	(0)	(0.016)	(0.033)	(0.079)	(0)	(0)
	Mclust	3.42	28.42	119.26	0.33	1					
		(0.002)	(0.015)	(0.013)	(0)	(0)					

Table 18: Simulation results for model II: averaged over one hundred runs, the numbers in parentheses are the standard errors.

		BIC					CV				
p	Method	SL	FL	KL	TP	TN	SL	FL	KL	TP	TN
30	SLasso	7.2	17.49	13.68	0.7	0.82	6.46	15.38	8.93	0.92	0.66
		(0.039)	(0.093)	(0.378)	(0.019)	(0.009)	(0.01)	(0.03)	(0.073)	(0.003)	(0.003)
	HLasso	2.82	4.97	2.33	1	1	2.58	5.17	2.48	1	1
		(0.071)	(0.077)	(0.035)	(0)	(0)	(0.05)	(0.119)	(0.065)	(0.001)	(0)
	Mclust	8.57	19.34	20.71	0.37	0.79					
		(0.244)	(0.144)	(0.171)	(0.032)	(0.041)					
50	SLasso	7.61	24.14	29.03	0.53	0.93	6.51	19.9	15	0.93	0.71
		(0.037)	(0.073)	(0.521)	(0.016)	(0.004)	(0.025)	(0.092)	(0.167)	(0.005)	(0.005)
	HLasso	2.54	5.77	3.49	1	1	2.99	7.79	3.95	1	1
		(0.046)	(0.051)	(0.035)	(0)	(0)	(0.044)	(0.154)	(0.089)	(0.001)	(0)
	Mclust	7.63	24.93	36.15	0.2	1					
		(0.007)	(0.029)	(0.274)	(0)	(0)					
100	SLasso	7.64	34.64	56.28	0.53	0.96	7.25	31.75	36.06	0.83	0.87
		(0.003)	(0.018)	(0.456)	(0.007)	(0.001)	(0.002)	(0.007)	(0.085)	(0.003)	(0.001)
	HLasso	2.69	8.02	7.01	1	1	2.98	10	7.13	1	1
		(0.039)	(0.049)	(0.056)	(0)	(0)	(0.026)	(0.055)	(0.051)	(0)	(0)
	Mclust	7.68	35.73	73.95	0.2	1					
		(0.002)	(0.012)	(0.01)	(0)	(0)					
300	SLasso	7.74	61.15	186.85	0.38	0.99	7.63	59.41	150.77	0.59	0.97
		(0.004)	(0.096)	(3.321)	(0.016)	(0.001)	(0.016)	(0.172)	(0.894)	(0.01)	(0.003)
	HLasso	3.87	25.18	28	1	1	3.87	25.18	28	1	1
		(0.024)	(0.133)	(0.182)	(0)	(0)	(0.024)	(0.133)	(0.182)	(0)	(0)
	Mclust	7.7	62.39	226.26	0.2	1					
		(0.002)	(0.016)	(0.021)	(0)	(0)					

Model I.2 : Model I with 2 individual edges added.

$$C_1(j, j+2) = C_1(j+2, j) \neq 0, C_1(j, j+4) = C_1(j+4, j) \neq 0,$$

$$C_2(j, j+3) = C_2(j+3, j) \neq 0, C_1(j, j+5) = C_1(j+5, j) \neq 0;$$

Model II.1 : Model II with 1 individual edge added.

$$C_1(j, j+3) = C_1(j+3, j) \neq 0, C_2(j, j+4) = C_2(j+4, j) \neq 0;$$

Model II.2 : Model II with 2 individual edges added.

$$C_1(j, j+3) = C_1(j+3, j) \neq 0, C_1(j, j+5) = C_1(j+5, j) \neq 0,$$

$$C_2(j, j+4) = C_2(j+4, j) \neq 0, C_2(j, j+6) = C_2(j+6, j) \neq 0;$$

where the conditional correlation of the added edge is fined selected to ensure the matrix is positive definite.

5.4 Future Work

In this chapter, we have proposed two structured estimators to capture sharing information in clusters of the high dimensional Gaussian mixture models. The performance of the new estimators has been shown superior over separate Lasso estimator when then concentration matrices share common structures and/or uncommon structures. We will continue studying properties of the proposed methods in more applications.

Table 19: Simulation results for model I.1: model I with 1 individual edge for each inverse matrix, averaged over one hundred runs, the numbers in parentheses are the standard errors.

		BIC					CV				
p	Method	SL	FL	KL	TP	TN	SL	FL	KL	TP	TN
30	SLasso	3.27	9.09	9.62	0.63	0.84	2.53	6.87	5.7	0.72	0.73
		(0.071)	(0.113)	(0.362)	(0.008)	(0.005)	(0.032)	(0.096)	(0.143)	(0.006)	(0.005)
	HLasso	1.82	3.47	1.58	0.62	1	1.64	3.4	1.68	0.61	1
		(0.053)	(0.057)	(0.032)	(0.001)	(0)	(0.037)	(0.055)	(0.043)	(0.001)	(0)
	Mclust	3.26	10.39	17.55	0.21	1					
		(0.006)	(0.022)	(0.069)	(0)	(0)					
50	SLasso	3.14	11.56	15.7	0.61	0.9	2.81	10.59	11.45	0.67	0.87
		(0.06)	(0.161)	(0.686)	(0.01)	(0.004)	(0.012)	(0.025)	(0.099)	(0.003)	(0.001)
	HLasso	1.76	4.07	2.34	0.61	1	1.55	4.27	2.57	0.61	1
		(0.038)	(0.039)	(0.031)	(0)	(0)	(0.023)	(0.065)	(0.071)	(0.001)	(0)
	Mclust	3.37	13.93	29.21	0.21	1					
		(0.006)	(0.028)	(0.147)	(0)	(0)					
100	SLasso	3.75	18.69	44.31	0.75	0.96	2.61	13.8	18.44	0.95	0.9
		(0.052)	(0.119)	(0.664)	(0.005)	(0.002)	(0.006)	(0.018)	(0.093)	(0.002)	(0.001)
	HLasso	1.92	5.61	4.52	0.99	1	1.62	5.79	4.69	0.99	1
		(0.035)	(0.04)	(0.044)	(0)	(0)	(0.018)	(0.029)	(0.041)	(0)	(0)
	Mclust	3.41	20	59.07	0.33	1					
		(0.002)	(0.012)	(0.008)	(0)	(0)					

Table 20: Simulation results for model I.2: model I with 2 individual edges for each inverse matrix, averaged over one hundred runs, the numbers in parentheses are the standard errors.

		BIC					CV				
p	Method	SL	FL	KL	TP	TN	SL	FL	KL	TP	TN
30	SLasso	3.28	9.11	9.7	0.51	0.85	2.53	6.87	5.7	0.61	0.72
		(0.071)	(0.114)	(0.368)	(0.007)	(0.005)	(0.032)	(0.096)	(0.142)	(0.006)	(0.005)
	HLasso	1.82	3.48	1.61	0.47	1	1.64	3.4	1.68	0.46	1
		(0.057)	(0.061)	(0.034)	(0.001)	(0)	(0.04)	(0.057)	(0.044)	(0.001)	(0)
	Mclust	3.26	10.38	17.54	0.16	1					
		(0.006)	(0.022)	(0.069)	(0)	(0)					
50	SLasso	3.17	11.63	15.99	0.48	0.9	2.81	10.59	11.45	0.53	0.87
		(0.063)	(0.165)	(0.699)	(0.009)	(0.004)	(0.012)	(0.025)	(0.098)	(0.003)	(0.001)
	HLasso	1.76	4.06	2.34	0.45	1	1.55	4.28	2.57	0.45	1
		(0.038)	(0.037)	(0.031)	(0)	(0)	(0.023)	(0.065)	(0.071)	(0.001)	(0)
	Mclust	3.37	13.93	29.18	0.15	1					
		(0.006)	(0.028)	(0.147)	(0)	(0)					
100	SLasso	3.75	18.69	44.31	0.74	0.96	2.61	13.8	18.44	0.94	0.9
		(0.052)	(0.12)	(0.664)	(0.005)	(0.002)	(0.006)	(0.018)	(0.093)	(0.002)	(0.001)
	HLasso	1.91	5.59	4.52	0.99	1	1.62	5.79	4.69	0.98	1
		(0.034)	(0.039)	(0.044)	(0)	(0)	(0.018)	(0.029)	(0.041)	(0)	(0)
	Mclust	3.41	20	59.04	0.33	1					
		(0.002)	(0.012)	(0.008)	(0)	(0)					

Table 21: Simulation results for model II.1: model II with 1 individual edge for each inverse matrix, averaged over one hundred runs, the numbers in parentheses are the standard errors.

		BIC					CV				
p	Method	SL	FL	KL	TP	TN	SL	FL	KL	TP	TN
30	SLASSO	0.69	1.68	10.75	0.67	0.75	0.62	1.47	8.41	0.78	0.61
		(0.002)	(0.004)	(0.077)	(0.004)	(0.003)	(0.001)	(0.003)	(0.076)	(0.003)	(0.003)
	HLASSO	0.27	0.56	2.82	0.77	0.99	0.31	0.67	3	0.74	1
		(0.005)	(0.01)	(0.043)	(0.003)	(0.001)	(0.005)	(0.011)	(0.063)	(0.002)	(0)
	Mclust	0.82	1.91	19.97	0.27	0.86					
		(0.021)	(0.012)	(0.102)	(0.03)	(0.035)					
50	SLASSO	0.7	2.17	16.51	0.69	0.8	0.7	2.16	16.48	0.7	0.8
		(0)	(0.001)	(0.071)	(0.003)	(0.001)	(0.002)	(0.006)	(0.076)	(0.003)	(0.003)
	HLASSO	0.27	0.62	4.09	0.76	1	0.28	0.72	3.85	0.73	1
		(0.005)	(0.006)	(0.039)	(0.001)	(0)	(0.003)	(0.008)	(0.044)	(0.001)	(0)
	Mclust	0.76	2.48	35.27	0.15	1					
		(0.001)	(0.003)	(0.278)	(0)	(0)					
100	SLASSO	0.7	3.06	34.71	0.7	0.84	0.7	3.06	34.71	0.7	0.84
		(0)	(0.001)	(0.094)	(0.003)	(0.001)	(0)	(0.001)	(0.094)	(0.003)	(0.001)
	HLASSO	0.27	0.81	7.47	0.73	1	0.29	0.96	7.29	0.72	1
		(0.004)	(0.005)	(0.057)	(0.001)	(0)	(0.003)	(0.007)	(0.052)	(0)	(0)
	Mclust	0.77	3.57	72.89	0.15	1					
		(0)	(0.001)	(0.01)	(0)	(0)					

Table 22: Simulation results for model II.2: model II with 2 individual edges for each inverse matrix, averaged over one hundred runs, the numbers in parentheses are the standard errors.

		BIC					CV				
p	Method	SL	FL	KL	TP	TN	SL	FL	KL	TP	TN
30	SLASSO	0.69	1.68	10.63	0.62	0.76	0.61	1.46	8.29	0.72	0.61
		(0.001)	(0.003)	(0.069)	(0.004)	(0.003)	(0.001)	(0.003)	(0.07)	(0.003)	(0.003)
	HLASSO	0.31	0.65	3.06	0.62	0.99	0.32	0.68	3.31	0.61	1
		(0.004)	(0.009)	(0.049)	(0.002)	(0)	(0.006)	(0.016)	(0.121)	(0.003)	(0)
	Mclust	0.98	1.99	20.9	0.5	0.57					
		(0.034)	(0.021)	(0.289)	(0.043)	(0.05)					
50	SLASSO	0.7	2.16	16.39	0.63	0.81	0.7	2.15	16.36	0.63	0.8
		(0)	(0.001)	(0.069)	(0.002)	(0.001)	(0.002)	(0.006)	(0.074)	(0.003)	(0.003)
	HLASSO	0.28	0.7	3.98	0.59	1	0.28	0.7	3.92	0.59	1
		(0.003)	(0.008)	(0.049)	(0.001)	(0)	(0.003)	(0.007)	(0.043)	(0.001)	(0)
	Mclust	0.76	2.49	38.52	0.12	1					
		(0.001)	(0.005)	(0.745)	(0)	(0)					
100	SLASSO	0.7	3.06	34.74	0.62	0.84	0.7	3.06	34.74	0.62	0.84
		(0)	(0.001)	(0.094)	(0.002)	(0.001)	(0)	(0.001)	(0.094)	(0.002)	(0.001)
	HLASSO	0.28	0.84	8.39	0.59	1	0.28	0.94	7.28	0.57	1
		(0.004)	(0.005)	(0.062)	(0.001)	(0)	(0.003)	(0.007)	(0.051)	(0)	(0)
	Mclust	0.77	3.58	75.64	0.12	1					
		(0)	(0.001)	(0.011)	(0)	(0)					

REFERENCES

[1] Allison, D. B., Cui, X., Page, G. P., and Sabripour, M. (2006), Microarray Data Analysis: From Disarray to Consolidation & Consensus, *Nature Reviews Genetics*, **7**, 55-65.

[2] Banerjee, O., Ghaoui, L. E., D'Aspr´émont, A., and Natsoulis, G. (2006), Convex optimization techniques for fitting sparse Gaussian graphical models, *Proceedings of the 23 rd International Conference on Machine Learning*, Pittsburgh, PA.

[3] Banerjee, O., Ghaoui, L. E., and D'Aspr´émont, A. (2008), Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data, *Journal of Maching Learning Research*, **9**, 485-516.

[4] Banfield, J. D., and Raftery, A. E. (1993), Model-Based Gaussian and Non-Gaussian Clustering, *Biometrics*, **49**, 803821.

[5] Bellman, R.E. (1961), Adaptive Control Processes, *Princeton University Press*, Princeton, NJ.

[6] Bickel, P.J. and Levina, E. (2008a), Covariance Regularization by Thresholding, *Annals of Statistics*, **34(6)**, 2577-2604.

[7] Bickel, P.J. and Levina, E. (2008b), Regularized estimation of large covariance matrices, *Annals of Statistics*, **36(1)**, 199-227.

[8] Breiman, L. (2001), Random Forests, *Machine Learning*, **45(1)**, 5-32.

[9] Campbell, J. Y., Lo, A. W., and MacKinlay, A. (1997), The Econometrics of Financial Markets, *Princeton University Press, Princeton*.

[10] Celeux, G. and Govaert, G. (1995), Gaussian parsimonious clustering models, *Pattern Recognition*, **28(5)**, 781-793.

[11] Choi, J. K., Yu, U., Kim, S., and Yoo, O. J. (2003), Combining multiple microarray studies and modeling interstudy variation, *Bioinformatics*, **19**, i84-i90.

[12] Choi, H., Shen, R., Chinnaiyan, A.M., and Ghosh, D. (2007), A Latent Variable Approach for Meta-Analysis of Gene Expression Data from Multiple Microarray Experiments, *BMC Bioinformatics*, 8:364.

[13] D'Aspremont, A., Banerjee, O. and El Ghaoui L. (2008), First-order methods for sparse covariance selection, *SIAM Journal on Matrix Analysis and its Applications*, **30(1)**, 56-66.

[14] Daul, S., DeGiorgi, E., Lindskog, F., and McNeil, A.J. (2003), The grouped t-copula with an application to credit risk, *RISK*, 16, 73.

[15] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977), Maximum likelihood from incomplete data via the EM algorithm (with discussiion), *Journal of the Royal Statistical Society Series B*, **39**, 1-38.

[16] Deng, X. and Yuan, M. (2007), Large Gaussian Covariance Matrix Estimation with Markov Structures, *Journal of Computational and Graphical Statistics*, to appear.

[17] Dhanasekaran, S. M., Barrette, T. R., Ghosh, D., Shah, R., Varambally, S., Kurachi, K., Pientas, K. J., Rubin, M. A., and Chinnaiyan, A. M. (2001), Delineation of prognostic biomarkers in prostate cancer, *Nature*, **412**, 822-826.

[18] Do, K. A., Müller, P., and Vannucci, M., Bayesian Inference for Gene Expression and Proteomics, *Cambridge University Press*, 2006.

[19] Donoho, D.L. (2000), High-dimensiaonal data analysis: the curse and blessings of dimensionality, Department of Statistics, Standford University.

[20] Efron, B., Tibshirani, R., Storey, J. D., and Tusher, V. (2001), Empirical Bayes analysis of a microarray experiment, *Journal of the American Statistical Association*, **96(456)**, 1151-1160.

[21] El Karoui, N. (2008a), Operator norm consistent estimation of large-dimensional sparse covariance matrices, *Annals of Statistics*, **36(6)**, 2717-2756.

[22] El Karoui, N. (2008b), Spectrum estimation for large dimensional covariance matrices using random matrix theory, *Annals of Statistics*, **36(6)**, 2757-2790.

[23] Embrechts, P., McNeil, A., and Straumann, D. (2002), Correlation and dependence in risk management: properties and pitfalls, in Risk Management: Value at Risk and Beyond, ed. M.A.H. Dempster, Cambridge University press.

[24] Fama, E.F. (1965), The behavior of stock prices, *Journal of Business*, **37**, 34-105.

[25] 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.

[26] Fan, J., Feng, Y. and Wu, Y. (2008), Network exploration via the adaptive LASSO and SCAD penalties, Manuscript.

[27] Fergusson, K., Platen, E. (2006), On the Distributional Characterization of daily Log-returns of a World Stock Index, *Applied Mathematical Finance*, **13(1)**, 19-38.

[28] Flury, B. W., Schmid, M. J. and Narayanan, A. (1994), Error rates in quadratic discrimination with constraints on the covariance matrices, *Journal of Classification* **11**, 101-120.

[29] Fraley, C., and Raftery, A.E. (2002), Model-based clustering, discriminant analysis and density estimation, *Journal of the American Statistical Association*, **97**, 611631.

[30] Fraley, C., and Raftery, A.E. (2007), Bayesian regularization for Normal mixture estimation and model-based Clustering, *Journal of Classification*, **24**, 155-181.

[31] Frey, R., McNeil, A.J., Nyfeler, M.A. (2001), Copulas and credit models, RISK Magazine, October 2001, p. 111.

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

[33] Friedman, H. P., and Rubin, J. (1967), On some invariant criteria for grouping data, *Journal of the American Statistical Association*, **62**, 11591178.

[34] Furrer, R. and Bengtsson, T. (2007), Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants, *Journal of Multivariate Analysis*, **98**, 227255.

[35] Garrett-Mayer, E., Parmigiani, G., Zhong, X., Cope, L., and Gabrielson, E. (2007), Cross study validation and combined analysis of gene expression microarray data, *Biostatistics*, **9(2)**, 333-354.

[36] Ghosh, D., Barette, T. R., Rhodes, D., and Chinnaiyan, A. M. (2003), Statistical issues and methods for meta-analysis of microarray data: a case study in prostate cancer, *Functional Integrative Genomics*, **3**, 180-188.

[37] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., and Lander, E.S. (1999), Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science, **286**, 531537.

[38] Guo, J., Levina, L., Michailidis, G. and Zhu, J. (2009), Joint estimation of multiple graphical models, *Biometrika*, to appear.

[39] Hastie, T., and Tibshirani, R. (1996), Discriminant Analysis by Gaussian Mixtures, *Journal of the Royal Statistical Society Ser. B*, **58**, 155176.

[40] Hong, F. and Breitling R. (2008), A comparison of meta-analysis methods fro detecting differentially expressed genes in microarray experiments, *Bioinformatics*, **24(3)**, 374-382.

[41] Huang, J., Liu, N., Pourahmadi, M. and Liu, L. (2006), Covariance matrix selection and estimation via penalised normal likelihood, *Biometrika*, **93(1)**, 85-98.

[42] Huang, J., Ma, S., Xie, H., and Zhang, C. (2009), A Group Bridge Approach for Variable Selection, *Biometrika*, **96**, 339355.

[43] Jiang, H., Deng, Y., Chen, H.-S., Tao, L., Sha, Q., Chen, J., Tsai, C.-J., and Zhang, S. (2004), Joint analysis of two microarray gene expression data sets to select lung adenocarinoma marker genes, *BMC Bioinformatics*, **5**, 81.

[44] Jones, M.C. (2002), Students simplest distribution, *The Statistician (Journal of the Royal Statistical Society Series D)*, **51(1)**, 41-49.

[45] Johnstone, I.M., Lu, A.Y., Sparse Principal Components Analysis, Unpublished manuscript.

[46] Kendziorski, C. M., Newton, M. A., Lan, H., and Gould, M. N. (2003), On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles, *Statistics in Medicine*, **22**, 3899-3914.

[47] Kuo, W. P., Jenssen, T.-K., Butte, A. J., Ohno-Machado L., and Kohane, I. S. (2002), Analysis of matched mRNA measurements from two different microarray technologies, *Bioinformatics*, **18**, 405-412.

[48] Lam, C. and Fan, J. (2009), Sparsistency and rates of convergence in large covariance matrices estimation, *The Annals of Statistics*, **37 (6B)**, 42544278.

[49] Lauritzen, S. L. (1996). Graphical Models. *Oxford: Clarendon Press*.

[50] LeCun,Y., Boser, B., Denker, J.S., Henderson,D., Howard, R.E., Hubbard, W., and Jackel, L.D. (1990), Handwritten digit recognition with a back propagation network, in *Advances in Neural Information Processing Systems*, Vol.2, ed. D. Touretzky, Denver, CO: Morgan Kaufman.

[51] Ledoit, O. and Wolf, M. (2004), A well-conditioned estimator for large-dimensional covariance matrices, *Journal of Multivariate Analysis*, 88, 365-411.

[52] Lee, M.-L. T., Kuo, F. C., Whitmore, G. A., and Sklar, J. (2000), Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations, *Proceedings of the National Academy of Sciences of the United States of America*, **97**, 9834-9839.

[53] Levina, E., Rothman, A. and Zhu, J. (2008), Sparse estimation of large covariance matrices via a nested LASSO penalty, *The Annals of Applied Statistics*, 2(1), 245-263.

[54] Lipshutz, R. J., Fodor, S. P., Gingeras, T. R., and Lockhart, D. J. (1999), High densitiy synthetic oligonucleotide arrays, *Nature Genetics* , **21**, 20-24.

[55] Liu, C. and Rubin, D. B. (1995), ML estimation of the t distribution using EM and its extensions, ECM and ECME. Statistica Sinica, 5, 19-39.

[56] Luo, J., Duggan, D. J., Chen, Y., Sauvageot, J., Ewing, C. M., Bittner, M. L., Trent, J. M., and Isaacs, W. B. (2001), Human prostate cancer and benign prostatic hyperplasia: molecular dissection by gene expression profiling, *Cancer Research*, **61**, 4683-4688.

[57] Magee, J. A., Araki, T., Patil, S., Ehrig, T., True, L., Humphrey, P. A., Catalona, W. J., Watson, M. A., and Milbrandt, J. (2001), Expression profiling reveals hepsin overexpression in prostate cancer, *Cancer Research*, **61**, 5692-5696.

[58] Mah, N., Thelin, A., Lu, T., Nikolaus, S., Kühbacher, T., Gurbuz, Y., Eickhoff, H., Klöppel, G., Lehrach, H., Mellgård, B., Costello, C.M., and Schreiber, S. (2004), A comparison of oligonucleotide and cDNA-based microarray systems, *Physiological Genomics*, **16**, 361-370.

[59] Mandelbrot, B. (1963), The variation of certain speculative prices, *Journal of Business*, **36**, 394-419.

[60] McLachlan, G.J. and Peel, D. (2000), *Finite Mixture Models*, Wiley.

[61] Meinshausen, N. and Buhlmann, P. (2006), High-dimensional graphs with the Lasso, *Annals of Statistics*, **34**, 143662.

[62] Mukherjee, S., Tamayo, P., Rogers, S., Rifkin R., Engle, A., Campbell, C., Golub, T. R., and Mesirov, J. P. (2003), Estimating dataset size requirements for classifying DNA microarray data, *Journal of Computational Biology*, **10(2)**, 119-142.

[63] Nevzorov, V.B., Balakrishnan, N., Ahsanullah, M. (2003), Simple characterizations of Student's t_2-distribution, *The Statistician (Journal of the Royal Statistical Society Series D)*, **52(3)**, 395-400.

[64] Newton, M. A., Kendziorski, C. M., Richmond, C. S., Blattner, R. R., and Tsui, K. W. (2001), On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data, *Journal of Computational Biology*, **8(1)**, 37-52.

[65] Parmigiani G., Garrett-Mayer, E., Anbazhagan, R., Gabrielson, E. (2002), A statistical framework for expression-based molecular classification in cancer, *Journal of Royal Statistical Society Series B* **64**, 717-736.

[66] Parmigiani, G., Garett-Mayer, E., Irizarry, R., and Zeger, S. (2003), The analysis of gene expression data: methods and software, *Springer*.

[67] Parmigiani, G., Garrett-Mayer, E., Anbazhagan, R., and Gabrielson, E. (2004), A cross-study comparison of gene expression studies for the molecular classification of lung cancer, *Clinical Cancer Research*, **10**, 2922-2927.

[68] Pourahmadi, M. (1999), Joint mean-covariance models with applications to longitudinal data: unconstrained parameterisation, *Biometrika*, **86(3)**, 677-690.

[69] Pourahmadi, M. (2000), Maximum likelihood estimation of generalised linear models for multivariate normal covariance matrix, *Biometrika*, **87**, 425 - 435.

[70] Pyne, S., Futcher, B., and Skiena, S. (2006), Meta-analysis based on control of false discovery rate: combining yeast ChIP-chip datasets, *Bioinformatics*, **22(20)**, 2516-2522.

[71] Rhodes, D. R., Barrette, T. R., Rubin, M. A., Ghosh, D., and Chinnaiyan, A. M. (2002), Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer, *Cancer Research*, **62**, 4427-4433.

[72] Rhodes, D. R., Yu, J., Shanker, K., Deshpande, N., Varambally, R., Ghosh, D., Barrette, T. R., Pandey, A., and Chinnaiyan, A. M. (2004), A large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression, *Proceedings of the National Academy of Sciences of the United States of America*, **101(25)**, 9309-9314.

[73] Rothman, A.J., Bickel P.J., Levina,E., and Zhu,J.(2008), "Sparse permutation invariant covariance estimation", *Electronical Journal of Statistics,* **2**, 494-515.

[74] Rothman, A.J., Levina, E. and Zhu, J. (2009), Generalized Thresholding of Large Covariance Matrices, *Journal of American Statistical Association (Theory and Methods)*, **104(485)**, 177-186.

[75] Schena, M. (2000), Microarray biochip technology. Eaton, Sunnyvale, Calif.

[76] Schloegl, L, OKane, D. (2005), A note on the large homogeneous portfolio approximation with the Student-t copula, *Finance and Stochastics*, **9(4)**, p.577.

[77] Shabalin, A. A., Tjelmeland, H., Fan, C., Perou, C. M., and Nobel, A. B. (2008), Merging two gene-expression studies via cross-platform normalizaiton, *Bioinformatics*, **24(9)**, 1154-1160.

[78] Shen, R., Ghosh, D., and Chinnaiyan, A. M. (2004), Prognostic meta-signature of breast cancer developed by two-stage mixture modeling of microarray data, *BMC Genomics*, **5**, 94.

[79] Smith, M. and Kohn, R. (2002), Parsimonious covariance matrix estimation for longitudinal data, *Journal of American Statistical Association*, **97**, 11411153.

[80] Symons, M.J. (1981), Clustering criteria and multivariate Normal mixtures, *Biometrics*, **37(1)**, 35-43.

[81] Tibshirani, R. (1996), Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society Series B*, **58(1)**, 267-288.

[82] Toedling J, Spang R, (2003), Assessment of five microarray experiments on gene expression profiling of breast cancer, Poster Presentation RECOMB, [http://citeseer.ist.psu.edu/611350.html].

[83] van't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002), Gene expression profiling predicts clinical outcome of breast cancer, *Nature*, **415**, 530-536.

[84] Wang, S. (1997), Aggregation of correlated Risk Portfolios: Models and Algorithms, Preprint Casualty Actuarial Society (CAS).

[85] Wang, Y., Klijn, J., Zhang, Y., Sieuwerts, A., Look, M., Yang, F., Talantov, D., Timmermans, M., Gelder, Meijer-van M., and Yu, J. (2005), Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer, *Lancet*, **365**, 671-679.

[86] Warnat, P., Eils R., and Brors, B. (2005), Cross-platform analysis of cancer microarray data improves gene expression based classificationof phenotypes, *BMC Bioinformatics*, **6**, 265.

[87] Welsh, J. B., Sapinoso, L. M., Su, A. I., Kern, S. G., Wang-Rodriguez, J., Moskaluk, C. A., Frierson, H. F., Jr., and Hampton, G. M. (2001), Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer, *Cancer Research*, **61**, 5974-5978.

[88] Whittaker, J. (1990). GraphicalModels in Applied Multivariate Statistics. *Chichester: John Wiley and Sons.*

[89] Wong, F., Carter, C. and Kohn, R. (2003), Efficient estimation of covariance selection models, *Biometrika*, **90**, 809830.

[90] Wu, W. and Pourahmadi M., Nonparametric estimation of large covariance matrices of longitudinal data, *Biometrika*, **90**, 831 - 844.

[91] Yuan, M. (2008), Efficient Computation of the l1 Regularized Solution Path in Gaussian Graphical Models, Journal of Computational and Graphical Statistics, 17, 809-826.

[92] Yuan, M. and Lin, Y. (2006), Model selection and estimation in regression with grouped variables, *Journal of Royal Statistical Society*, Series B, **68**, Part 1, pp. 4967.

[93] Yuan, M. and Lin, Y. (2007), Model selection and estimation in the Gaussian graphical model, *Biometrica*, **94(1)**,19-35.

[94] Zhou, N. and Zhu, J. (2010), Group Variable Selection via a Hierarchical Lasso and Its Oracle Property, Technical report, Department of Statistics, University of Michigan.

[95] Zou, H. (2006), The Adaptive Lasso and its Oracle Properties, *Journal of the American Statistical Association*, **101(476)**, 1418-1429.