Network inference with EM algorithm

Extract the signal from the data using replications

Yao Chen (s1954369)

Supervisor: Dr. Wessel van Wieringen (VU)

MASTER THESIS

Defended on Month Day, 2019

Specialization: Data Science

STATISTICAL SCIENCE
 FOR THE LIFE AND BEHAVIOURAL SCIENCES
Abstract

Real-world data contains both signal and noise. In this study, we developed a method to utilize replications to separate signal and noise. Our proposed method employed the Expectation-maximization algorithm to estimate both signal and noise precision matrices. The estimated precision matrices were used to construct a Gaussian graphical model, which represents the network of variables. In high-dimensional settings, regularization techniques were used to ensure the positive definiteness of the estimated precision matrices.

In the simulation study, we varied the graphical structure, the number of edges and the size of noise to see how the proposed method performs. As the true signal precision matrix is known, the estimates of the proposed method were compared to those by other methods through Kullback-Leibler divergence from the true one and prediction accuracy of edge presence or absence. The results show that for the clique models in our case, our proposed method unpenalized performed best in edge detection while for banded and star models under certain circumstances, the unpenalized estimates by the proposed method came last in edge detection. The distributions using the penalized estimates by our method are best or second best approximations of the true distribution in terms of KL divergence. Our results also show that with increasing samples and replications, the estimates become better in edge detection and approximation to the true distributions.

In the real-world data analysis, we used three pathways of a lung cancer dataset from TCGA project. The results show that there is more overlapping between estimates of the merged data and single-platform data than between estimates of the two platforms in terms of KL divergence and edges in common. We also found that the distribution constructed by the signal-noise estimator by the proposed method is better approximation to that of the new data than signal estimator.
Acknowledgements

I would like to thank my supervisor Wessel van Wieringen for his support and enthusiasm in this project. The inspiring weekly discussions in the past nine months have been very helpful to me. I have learned so much and have a lot of fun. Without his guidance during this journey, this thesis would not have been possible. Thanks to my lovely roommates, Xiaogao, Shashidi, Huzhuzhu for cooking with me. I have enjoyed the delicious food and our little talks over dinner very much. Thanks to my Fangdong for his mental support and proofreading my thesis. Thanks to Yapeng and Wang Zhe for being the best team partners ever. I would like to thank my beloved parents for always believing in and supporting me.
Contents

1 Introduction .................................................................................................................. 3
   1.1 General Introduction ............................................................................................... 3
       1.1.1 Background ..................................................................................................... 3
       1.1.2 High dimensionality of the data .................................................................... 4
   1.2 Preliminaries .......................................................................................................... 4
       1.2.1 Multivariate statistical analysis .................................................................... 4
       1.2.2 Graphical modeling ....................................................................................... 6
   1.3 Motivation .............................................................................................................. 9
   1.4 Structure of the thesis ............................................................................................ 12

2 Methodology .................................................................................................................. 13
   2.1 Problem formulation ............................................................................................... 13
       2.1.1 Marginal likelihood of Y ............................................................................ 14
   2.2 Expectation-maximization algorithm ..................................................................... 16
   2.3 Penalization ........................................................................................................... 18
       2.3.1 Choice of $\lambda_M, \lambda_e$ ........................................................................ 20
   2.4 Measure of the performance .................................................................................. 21
       2.4.1 Kullback-Leibler divergence .................................................................... 21
       2.4.2 Mutual information ..................................................................................... 24
       2.4.3 ROC and AUC ............................................................................................ 26
   2.5 Local False Discovery Rates ................................................................................... 28

3 Results ............................................................................................................................ 29
   3.1 Simulation Study .................................................................................................... 29
       3.1.1 Scenario 1: 2 replications ......................................................................... 31
3.1.2 Scenario 2: 3 replications

3.1.3 Scenario 3: varied replications

3.1.4 Scenario 4: use of a target matrix

3.1.5 Scenario 5: the effect of noise level

3.1.6 Conclusion

3.2 Real-world data

3.2.1 Data description

3.2.2 Results

3.2.3 Conclusion

4 Discussion

4.1 Summary

4.2 Future work

4.3 Conclusion

A Covariance matrix of $Y_i$

A.1 Determinant

A.2 Inverse covariance matrix

B EM algorithm

B.1 Complete-data likelihood

B.2 EM algorithm

B.3 EM algorithm with penalization

C Kullback-Leibler divergence

D Mutual information

E ROC and AUC plots
Chapter 1

Introduction

1.1 General Introduction

1.1.1 Background

Genes interact with each other forming biochemical interactions and resulting in regulatory activities. The interaction is the interplay between multiple genes, which affects the expression of a phenotype. Such gene-gene interaction can be depicted by a regulatory network.

Genetic regulatory networks capture precious biological information (Butte et al., 2000). In reconstructing those networks, the absence or presence of edges depends on whether they are statistically significant in predictions of molecular interactions. Such reconstructed networks are very helpful in narrowing down the potentially large number of interactions between around 20,000 human genes to interactions for which statistical support is available (Emmert-Streib et al., 2014).

Many methods have been developed to reconstruct such networks, including graphical models, Bayesian and weighted networks. The weighted method considers a fully connected network and associates a weight to each edge by estimating gene dependencies directly from the data. Graphical models and Bayesian networks explicitly formulate a model of the data in terms of probabilities. In this study, we used graphical models in reconstructing the networks.

Graphical modeling is one of the most popular tools for genetic regulatory network inference. The graphical models are flexible and powerful in representation and have been proven to perform effectively in large network inference. Such models provide a way to exploit the independence and dependence relations between variables that exist in many real-world phenomena. Gaussian graphical modeling, an undirected probabilistic graphical model, is a very important part of graphical modeling. It is widely used to investigate conditional independence between two variables given other variables. These models are applied in various fields, such as biology, social studies, and economics.

Real-world data is inherently noisy. It contains both signal and noise. Signal is the information of interest. The true network of the signal can be different from that of noise-diluted data. Some edges of the signal can be drowned by the noise, and some edges can be induced by noise too. The problem
becomes how we can represent the network of signal instead of data. Few studies have considered this problem and tried to extract the signal from the data.

Thus in this study, we aim to give a more accurate profile of the network by separating signal from noise in the data. Our study takes noise into account and reconstructs the networks of signal instead of the data. Without further inference for the noise from the observations, signal and noise are inseparable. However, this goal can be achieved with the help of replications. In this study, a new method is developed to extract signal from the data and reconstruct the signal network by using replications.

1.1.2 High dimensionality of the data

Genetic data are usually high-dimensional data which are characterized by a few dozen or many thousands of dimensions. The number of unknown parameters \( p \) is much larger than the number of samples \( n \) \((p \gg n)\). Such high-dimensional data are common in the fields such as medicine or biology. For example, the development of DNA microarray technology enables a large number of measurements at once.

High-dimensionality has challenged traditional statistical theory. Many statistical techniques, which are effective on a low dimensional level, are no longer applicable on the high dimension. The problem is when the dimensionality increases, the available data become very sparse with respect to the fast growing volume of the space (Bellman, 1966). Therefore, in order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality (Bellman, 1966). High-dimensional data are characterized by many attributes, so it is likely that some of the attributes are correlated. Besides, the high dimensionality gives rise to a high degree of irrelevant and redundant features, which may obscure the effect of the relevant ones. Therefore, dimensionality reduction and feature extraction have played pivotal roles in high-dimensional data analysis.

1.2 Preliminaries

In this section, we introduce relevant techniques used in the high-dimensional data analysis and network inference. Section 1.2.1 presents basic concepts in the multivariate statistical analysis, and section 1.2.2 introduces graphical modeling.

1.2.1 Multivariate statistical analysis

Multivariate statistical analysis concerns analyzing data in high dimensions. In the analysis, more than one dependent and more than one independent variable, or both are involved. It examines relationships among multiple variables simultaneously.

Suppose we are given a set of data \( X \) of \( n \) observations and \( p \) dimensions, then for each observation
where \( x_i \) is the observed value, and the random variable \( X \) of \( p \) dimensions can be written as,

\[
X = (X_1, X_2, \ldots, X_p)
\]

In multivariate analysis, most of the existing inference procedures for analyzing vector-valued data have been developed under the assumption of normality (Tong, 2012). For example, in regression analysis, the error is often assumed to follow a normal distribution. This enables us to perform statistical analysis using normal distributions or distributions derived from the normal distribution. The multivariate Gaussian distribution, also called “Multivariate normal distribution”, is the extension of the univariate normal distribution to higher dimensions. The univariate Gaussian distribution with mean \( \mu \) and variance \( \sigma^2 \) has the density:

\[
f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{1}{2\sigma^2}(x-\mu)^2 \right) \tag{1.1}\]

Then the \( p \)-dimensional Gaussian distribution with mean \( \mu \) and covariance \( \Sigma > 0 \) has the density,

\[
f(X; \mu, \Sigma) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (X - \mu)^T \Sigma^{-1} (X - \mu) \right\} \tag{1.2}\]

where

\[
X = (X_1, \ldots, X_p)^T, \quad \mu = (\mu_1, \ldots, \mu_p)^T, \quad \Sigma = \begin{pmatrix}
\sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\
\sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp}
\end{pmatrix}
\]

where \( \sigma_{ij} \) is the covariance between \( X_i \) and \( X_j \). For \( \Sigma \) to be a valid covariance matrix, it has to be symmetric positive semidefinite. For \( \Sigma^{-1} \) to exist, as required in the definition of multivariate Gaussian distribution, \( \Sigma \) has to be positive definite (Do, 2008). In equation (2.2), \( \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \) is a normalization factor just to ensure

\[
\frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \exp \left( -\frac{1}{2} (X - \mu)^T \Sigma^{-1} (X - \mu) \right) \, dX_1 \, dX_2 \cdots dX_p = 1 \tag{1.3}
\]

There are many properties of multivariate Gaussian distributions that have made it so prevalent in the multivariate analysis. The density function is uniquely determined by the mean and covariance matrix. The distributions of linear transformations of multivariate normal variables are again multivariate normal (Tong, 2012). The product of normal distributions is also normal. A marginal distribution of a joint normal distribution is normal. The conditional distribution of some variables given the remainder of the variables in a multivariate normal distribution is a multivariate normal. Even if the true distribution of the original data is not multivariate normal, by the Central Limit Theorem, the distribution of the average can be approximated by one.

As the genetic data we intend to explore is high-dimensional, multivariate analysis was used. Or
more specifically, multivariate analysis assuming multivariate normal distributions was used.

1.2.2 Graphical modeling

Graphical modeling is a means of multivariate analysis that uses graphs to present key properties of models. The following sections will introduce those relevant concepts in graphical modeling for the thesis.

Conditional independence

Independence and conditional independence are familiar concepts in probability theory. In this study we only consider continuous variables, so here we give definition of independence and conditional independence for continuous variables. Let X and Y be continuous random variables. They are independent if the joint probability density of (X,Y) can be expressed as a factorization of the probability density of X and the probability density of Y:

\[ f_{X,Y}(x, y) = f_X(x)f_Y(y) \]

or equivalently:

\[ f_{X|Y}(x|y) = f_X(x) \]

The relation of independence between X and Y is denoted as \( X \perp Y \).

Now consider a further variable Z. X and Y are said to be conditionally independent given Z if

\[ f_{X,Y|Z}(x, y|z) = f_{X|Z}(x|z)f_{Y|Z}(y|z) \]

or equivalently:

\[ f_{X|Y,Z}(x|y, z) = f_{X|Z}(x|z) \]

The conditional independence of X and Y given Z can be written as \( X \perp Y|Z \), which can be interpreted as “if we know Z, information about Y is irrelevant for knowledge of X” (Edwards, 2012).

Undirected graph

A graph is a mathematical structure that is made up of a finite set of vertices and edges. Vertices correspond to variables and edges are connections between variables. By mathematical notation, a graph can be represented by \( G = (V, E) \) and \( V \) is the set of vertices and \( E \) is the set of edges. Since there is no direction associated with the edges, the graph is called undirected. So the undirected graph represent a symmetric pairwise relationship. Fig.1.1 depicts a simple undirected graph of three vertices and two edges. Vertice 1, 2, 3 represent variable 1, 2, 3 respectively. In this example, variable 2 is conditionally independent from variable 3 given variable 1, that is, \( 2 \perp 3|1 \).

Two vertices \( X, Y \in V \) are adjacent, if there is an edge between them. For example, in Fig.1.1, 1 and 2 are adjacent, but 2 and 3 are not. A graph is complete if every pair of vertices is adjacent. The
graph in Fig.1.1 is not complete with one edge missing between 2 and 3.

Now look at a multivariate example. Let’s consider a dataset of $p$ variables and $n$ samples. The $p$ variables $X_1, X_2, ..., X_p$ are random variables with a joint probability density function $f(X_1, X_2, ..., X_p)$.

Graphical modelling focuses on models for which the conditional independence relations take the form $X_i \perp X_j | (\text{some other variables})$ (Edwards, 2012). In this study, our special interest lies in the conditional independence relations between $X_i$ and $X_j$ given the remaining variables. The set of such relations is portrayed by a graph. For all pairs $(X_i, X_j)$, if $X_i \perp X_j | (\text{the rest})$ holds, then the edge between $X_i$ and $X_j$ is absent; otherwise, the edge is present between them. Thus, if two variables are not adjacent in the graph, then they are conditionally independent given the rest of the variables. This is known as the pairwise Markov property for undirected graphs (Edwards, 2012). This property translates a graph-theoretic property, separation, to a statistical property, conditional independence (Edwards, 2012).

**Precision matrix**

One of the most important concepts in graphical models is the precision matrix. It is used to reconstruct the interaction network. The edge between two variables is absent if those two variables are conditionally independent given others and the corresponding element in the precision matrix will be 0.

The precision matrix is also called the concentration matrix. It is the inverse covariance matrix, $\Omega = \Sigma^{-1}$, which can be written as,

$$
\omega = \begin{bmatrix}
\omega_{11} & \cdots & \omega_{1p} \\
\vdots & \ddots & \vdots \\
\omega_{p1} & \cdots & \omega_{pp}
\end{bmatrix}
$$
Consider the conditional normal distribution of \((X_1, X_2)\) given the remaining variables \((X_3, \ldots, X_p)\), which is a bivariate normal distribution and can be written as:

\[
X_1, X_2 | X_3, \ldots, X_p \sim \mathcal{N}(\mu, \Sigma)
\]

\[
\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{21} & \omega_{22} \end{bmatrix}^{-1} = \frac{1}{\omega_{11}\omega_{22} - (\omega_{12})^2} \begin{bmatrix} \omega_{11} & -\omega_{12} \\ -\omega_{21} & \omega_{22} \end{bmatrix}
\]

It is easy to see the correlation coefficient of \(X_1\) and \(X_2\) in this conditional bivariate normal distribution as

\[
\rho(X_1, X_2 | x_3, \ldots, x_p) = \frac{-\omega_{12}}{(\omega_{11}\omega_{22})^{1/2}},
\]

This is called a partial correlation coefficient. Each off-diagonal element of the precision matrix (scaled to have a unit diagonal) is the negative of the partial correlation between the two corresponding variables, conditioned on the rest of the variables (Whittaker, 2009). A partial correlation coefficient is the correlation of two variables conditioned on other variables.

The partial correlation coefficient between two variables \(X_a\) and \(X_b\) after removing the effect of other variables \(X_c\) is defined as:

\[
\rho(X_a, X_b | V\setminus\{X_a, X_b\}) = \frac{\text{Cov}(X_a, X_b | V\setminus\{X_a, X_b\})}{\sqrt{\text{Var}(X_a | V\setminus\{X_a, X_b\}) \text{Var}(X_b | V\setminus\{X_a, X_b\})}}
\]

(1.5)

\(\rho(X_a, X_b | V\setminus\{X_a, X_b\})\) quantifies the amount of information in \(X_a\) associated with \(X_b\) after removal of all information on those two variables contained in the remaining variables. If \(\rho(X_a, X_b | V\setminus\{X_a, X_b\}) = 0\), the partial correlation between \(X_a\) and \(X_b\) given other variables is 0. That means \(X_b\) adds nothing to other variables in explaining variation in \(X_a\). \(X_a\) and \(X_b\) are independent from each other when conditioning on the remainder of the variables. In combination with Markov property, we can see that:

\[
\omega_{ij} = 0 \iff X_i \perp \!
\!
\!\perp X_j | V\setminus\{X_i, X_j\}
\iff \text{no edge between } X_i \text{ and } X_j
\]

(1.6)

When the partial correlation is not zero, then:

\[
\omega_{ij} \neq 0 \iff X_i \not\!
\!
\!\not\!
\not\!
\not\!
\perp X_j | V\setminus\{X_i, X_j\}
\iff \text{an edge between } X_i \text{ and } X_j
\]

(1.7)

**Gaussian graphical modeling**

A Gaussian graphical model for a multivariate Gaussian distribution is represented by an undirected graph \(G = (V, E)\), where \(V\) is a set of vertices and \(E\) is a set of edges (Lauritzen, 1995). The vertices in this study are variables and the edges are relationships between pairs of variables: any two variables are either connected with an edge or not.

Gaussian graphical models are defined by setting specified elements of the precision matrix to zero,
hence the corresponding partial correlation coefficients would be zero and the two variables would be conditionally independent given other variables. Consider a 4-dimensional dataset and a model setting with zero partial correlation between variable 1 and 4. Then the corresponding precision matrix would be:

\[
\begin{bmatrix}
\omega_{11} & \omega_{12} & \omega_{13} & 0 \\
\omega_{21} & \omega_{22} & \omega_{23} & \omega_{24} \\
\omega_{31} & \omega_{32} & \omega_{33} & \omega_{34} \\
0 & \omega_{42} & \omega_{43} & \omega_{44}
\end{bmatrix}
\]

which can be translated into the graph in fig.1.2. There is a one-to-one correspondence between models and graphs.

Gaussian graphical models capture the statistical relationships between the variables of interest in the form of a graph, and are widely applied in many fields. Many studies have developed parameter estimation methods in such models and explored the statistical properties of precision matrices. Meinshausen et al. (2006) performed neighbourhood selection for each node and estimated a sparse graph by fitting a lasso model. J. Friedman et al. (2007) sped up the variable selection process. Dobra et al. (2004) presented a new Bayesian framework for estimating Gaussian graphical models. Yuan & Lin (2007) proposed a new penalized -likelihood method that can select models and estimate parameters simultaneously.

### 1.3 Motivation

The real-world data is inherently noisy and the structure of signal may be obscured by noise. Analyzing the network by ignoring the noise can be quite deceptive. For instance, let’s take a look at a simple network of three nodes.
Suppose the noise of these three variables is independent from each other and from the signal. Then the precision matrix of noise should be diagonal. We now consider the following set-ups of precision matrices for the signal. Let \( c \) be an element of the precision matrix:

1. **Random model** with \( c_{1,2} = 1, c_{1,3} = -0.5, c_{2,1} = 0.1, c_{2,3} = 0.03 \).
2. **Banded model** with \( c_{1,1} = 1, c_{1,2} = c_{2,1} = 0.5 \), and \( c_{2,3} = 0.25 \).
3. **Star model** with every node connected to the first node, with \( c_{1,j+1} = c_{j+1,1} = \frac{1}{(j+1)} \), for \( j = 1,2 \) and \( c_{ij} = 0 \) otherwise.

The undirected graphs of the above three models are illustrated in Fig. 1.3a, 1.4a, 1.5a, and the corresponding networks of the data are shown in Fig. 1.3b, 1.4b, 1.5b. For model 1, as Fig. 1.3a shows, the true network of the signal is fully connected, but for the noise-diluted data \( y \), the edge between 2 and 3 is missing. The underlying structure of model 2 is shown in Fig. 1.4a and there is no edge between 1 and 3, but in the network of data \( y \), the edge between those two nodes is found. For model 3, no edge is supposed between node 2 and node 3, but the network of the data is again fully connected.

The network of the data can be very different from the network of the signal. So it is necessary to take into account the noise in the data.

In this paper, we proposed a method to estimate both signal and noise precision matrices. The performance of the proposed method will be assessed under different simulation scenarios by the divergence from the true distribution and prediction accuracy measured by ROC plots. Thus, the first research question is, will replications improve the estimation? The second research question is, will our method outperform other simpler methods, such as averaging the data, under certain simulation scenarios in terms of divergence and edge detection, and is there any overall winner? Furthermore, we will analyse a real-world dataset with replications. The third research question is, how much difference is between the estimates using the merged data and the estimates using data without replications?
Figure 1.4: undirected graphs of model 2

(a) Undirected graph of signal for model 2
(b) Undirected graph of data y for model 2

Figure 1.5: undirected graphs of model 3

(a) Undirected graph of signal for model 3
(b) Undirected graph of data y for model 3
1.4 Structure of the thesis

The thesis is organized as follows. Chapter 2 describes the methodology, including the formulation of the problem and the proposed methods to estimate the underlying network. Chapter 3 outlines the results of several simulation studies and one real-world data study. In the simulation study, the proposed method is applied on the synthetic data of different set-ups and the performance of this method is compared to that of other methods. Chapter 4 presents a discussion of the results, including the advantages and disadvantages of using this method, limitations, practical implications and possible future research.
Chapter 2

Methodology

2.1 Problem formulation

Signal and noise are not separable when each sample is measured only once. Suppose signal and noise co-exist in our genetic data and they are independent, let $M_i, i = 1, \ldots, n$, be the true signal, where $n$ is the number of samples. The unobservable signals $M_i$ are expected to be highly variable due to the inherent biological variability. The model is given by

$$y_i = M_i + \varepsilon_i$$

$$M_i \sim N(0_p, \Omega_M^{-1})$$

$$\varepsilon_i \sim N(0_p, \Omega_\varepsilon^{-1})$$

where $y_i$ is the observation, $\varepsilon_i$ is the noise which is a consequence of acquiring and pre-processing data from the environment, $\Omega_M$ and $\Omega_\varepsilon$ are precision matrices of signal and noise. This model is clearly unidentifiable because it is impossible to estimate signal and noise precision matrices from the observation.

However, replications make the estimation possible. Suppose each sample is measured several times. Sample $i$ is independent from the rest of the samples, but there is dependency between the replications of this sample. So our data are clustered with samples as units. For each sample, the signal is shared by all replicates and the noise is the measurement error that should be independent from each other. The signal and noise come from different channels and there is no correlation between those two. Our model can be formulated as follows:

$$y_{ij} = M_i + \varepsilon_{ij}$$

$$M_i \sim N(0_p, \Omega_M^{-1})$$

$$\varepsilon_{ij} \sim N(0_p, \Omega_\varepsilon^{-1})$$

$y_{ij}$ = observation of j-th replication of sample i, $i=1,\ldots,n$; $j=1,\ldots,k_i$
\( M_i = \text{signal of } i\text{-th sample} \)

\( \varepsilon_{ij} = \text{the error of } j\text{-th replication of sample } i \)

\( n = \text{number of samples} \)

\( k_i = \text{number of replications} \)

\( k = \text{total number of observations} \)

\[
\Omega_{M}^{-1} := \begin{bmatrix}
\sigma_{M11} & \cdots & \sigma_{M1p} \\
\vdots & \ddots & \vdots \\
\sigma_{Mpi} & \cdots & \sigma_{Mpp}
\end{bmatrix} \in \mathbb{R}^{p \times p}, \quad \Omega_{\varepsilon}^{-1} := \begin{bmatrix}
\sigma_{\varepsilon11} & \cdots & \sigma_{\varepsilon1p} \\
\vdots & \ddots & \vdots \\
\sigma_{\varepsilonpi} & \cdots & \sigma_{\varepsilonpp}
\end{bmatrix} \in \mathbb{R}^{p \times p}
\]

where \( p \) is the number of variables, \( \Omega_{M} \) is the precision matrix for the signal, \( \Omega_{\varepsilon} \) is the precision matrix for the noise.

We also assume the independence of signal and noise, the independence of samples and the dependency of repeated measurements, which can be notated as,

\[
\text{Cov}(M_i, \varepsilon_{ij}) = 0_{p \times p}
\]

\[
\text{Cov}(Y_{i1}, Y_{i2}) = 0_{p \times p}
\]

\[
\text{Cov}(y_{ij1}, y_{ij2}) = \Omega_{M}^{-1}
\]

\( y_{ij} \) follows a multivariate normal distribution.

\[
y_{ij} \sim \mathcal{N}(0_p, \Omega_{M}^{-1} + \Omega_{\varepsilon}^{-1})
\]

As there is no correlation between signal and noise, the covariance matrix of \( y_{ij} \) should be the simple combination of \( \Omega_{M}^{-1}, \Omega_{\varepsilon}^{-1} \). The conditional distribution of \( y_{ij} \) on \( M_i \) is also a multivariate normal distribution.

\[
y_{ij|M_i} \sim \mathcal{N}(M_i, \Omega_{\varepsilon}^{-1})
\]

The precision matrix of data \( Y, (\Omega_{M}^{-1} + \Omega_{\varepsilon}^{-1})^{-1} \), is not equal to the precision matrix of the signal \( M, \Omega_{M} \). So, the problem boils down to how we can separate the signal from the noise and estimate the signal precision matrix from the data.

### 2.1.1 Marginal likelihood of \( Y \)

We can achieve the goal of finding the parameters by maximizing the likelihood of the data. Because there is a dependency structure in our data, it is difficult to calculate the likelihood observation by observation, but samples are independently distributed and follow a multivariate normal distribution. So we first calculate the likelihood for each sample. Suppose there are \( k_i \) replications for sample \( i \).
\[ y_{i1} = M_i + \varepsilon_{i1} \]
\[ y_{i2} = M_i + \varepsilon_{i2} \]
\[
\vdots
\]
\[ y_{ik_i} = M_i + \varepsilon_{ik_i} \]
\[ M_i \sim \mathcal{N}(0_p, \Omega_M^{-1}) \]
\[ \varepsilon_{ij} \sim \mathcal{N}(0_p, \Omega_\varepsilon^{-1}) \]

Then \( y_{i1}, y_{i2}, \ldots, y_{ik_i} \) follow a multivariate normal distribution.

\[
\begin{bmatrix}
  y_{i1} \\
  \vdots \\
  y_{ik_i}
\end{bmatrix}
\sim
\mathcal{N}
\left(
\begin{bmatrix}
  0_p \\
  \vdots \\
  0_p
\end{bmatrix},
\begin{bmatrix}
  \Omega_M^{-1} + \Omega_\varepsilon^{-1} & \ldots & \Omega_\varepsilon^{-1} \\
  \ldots & \ldots & \ldots \\
  \Omega_M^{-1} & \ldots & \Omega_M^{-1} + \Omega_\varepsilon^{-1}
\end{bmatrix}
\right)
\]

The likelihood of one sample can be formulated as follow.

\[ f(Y_i) = f(y_{i1}, y_{i2}, \ldots, y_{ik_i}) = \frac{1}{(2\pi)^{kp/2} |\Omega_i^{-1}|^{1/2}} \exp\left\{-\frac{1}{2} Y_i^t \Omega_i^{-1} Y_i \right\} \]

(2.1)

where

\[ \Omega_i^{-1} := \begin{bmatrix}
  \Omega_M^{-1} + \Omega_\varepsilon^{-1} & \ldots & \Omega_\varepsilon^{-1} \\
  \ldots & \ldots & \ldots \\
  \Omega_M^{-1} & \ldots & \Omega_M^{-1} + \Omega_\varepsilon^{-1}
\end{bmatrix} \in \mathbb{R}^{k_p \times k_p}, \quad Y_i = \begin{bmatrix}
  y_{i1} \\
  \vdots \\
  y_{ik_i}
\end{bmatrix} \in \mathbb{R}^{k_p \times 1} \]

The derivations of \(|\Omega_i^{-1}|\) and \(\Omega_i\) are in appendix A.1 and A.2 respectively.

As samples are independent of each other, by equation (2.1), the likelihood of all samples can be factorized as follows:

\[ f(Y_1, \ldots, Y_n) = f(Y_1) \cdots f(Y_n) \]
\[ = \prod_{i=1}^n \frac{1}{(2\pi)^{kp/2} |\Omega_i^{-1}|^{1/2}} \exp\left\{-\frac{1}{2} Y_i^t \Omega_i^{-1} Y_i \right\} \]

(2.2)

The log likelihood is:

\[ \log(f(Y)) = \sum_{i=1}^n \log(f(Y_i)) = \frac{1}{2} (-kp\log(2\pi)) - \sum_{i=1}^n \log(|\Omega_i^{-1}|) - \sum_{i=1}^n Y_i^t \Omega_i^{-1} Y_i \]

(2.3)

To estimate the precision matrices of the signal and noise, usually we maximize the likelihood and get the derivatives of the above equation. However, differentiating equation (2.3) with respect to the precision matrices of the signal and noise would be difficult. In the next section, we introduce the Expectation-maximization algorithm to find the maximum likelihood parameters of models in cases the equations cannot be solved directly.
2.2 Expectation-maximization algorithm

The EM algorithm is generally used to find the maximum-likelihood estimate of a distribution from incomplete data (Dempster et al., 1977; Ghahramani & Jordan, 1995; Jordan & Jacobs, 1994). There are two main applications of this algorithm. First, the data indeed have missing values. Second, the optimization of the likelihood function is often analytically intractable, but the likelihood function can be simplified if we assume those hidden variables to exist. So there are no actual missing values in our study, but those hidden variables are treated as “missing values”. The latter applies here. The signal is hidden from us and only the representation of the signal, observations, is known. While the likelihood function of the observations cannot be solved directly, the complete-data likelihood of both observations and the signal is easy to solve.

The EM algorithm consists of an expectation step and a maximization step in each iteration, that is why it is called EM algorithm. The expectation step is with respect to the latent variables. The latent variables are estimated given the current parameters and the observations. Then in the maximization step, the parameters are updated. These two steps are iterated until convergence (Dempster et al., 1977).

In our case, \( Y \) is a set of observed data, \( M \) a set of unobserved latent data, \( \Omega^{-1}_M, \Omega^{-1}_\varepsilon \) are unknown parameters we intend to estimate. The maximum likelihood estimate of the unknown parameters is obtained by maximizing the marginal likelihood of \( Y \). However, the equation (2.3) is of a complicated form and analytically intractable to optimize. But the likelihood function can be simplified by assuming the existence of \( M \). We can define the complete-data likelihood function, \( L(\Omega^{-1}_M, \Omega^{-1}_\varepsilon; Y, M) = f(Y, M|\Omega^{-1}_M, \Omega^{-1}_\varepsilon) \).

Considering the conditional independence between replications, the complete-data likelihood of \( Y \) and \( M \), \( f(Y, M) \), can be factorized into \( f(Y|M) \) and \( f(M) \). \( Y \) conditioned on \( M \) is independently distributed between replications, thus the complete-data likelihood can be formulated as follows:

\[
f(Y, M) = f(Y|M)f(M) = \prod_{i=1}^{n} f(Y_i|M_i)f(M_i)
= \prod_{i=1}^{n} f(y_{i1}, y_{i2}, ..., y_{ik_i}|M_i)f(M_i)
= \prod_{i=1}^{n} \left( \prod_{j=1}^{k_i} f(y_{ij}|M_i) \right)f(M_i)
= \prod_{i=1}^{n} \prod_{j=1}^{k_i} f(y_{ij}|M_i) \prod_{i=1}^{n} f(M_i)
= \prod_{i=1}^{n} \prod_{j=1}^{k_i} \frac{1}{(2\pi)^{\frac{D}{2}}|\Omega^{-1}_{\varepsilon}|^\frac{D}{2}} e^{\exp\left\{ -\frac{1}{2} (y_{ij} - M_i)^T \Omega^{-1}_\varepsilon (y_{ij} - M_i) \right\}} \prod_{i=1}^{n} \frac{1}{(2\pi)^{\frac{D}{2}}|\Omega^{-1}_M|^\frac{D}{2}} e^{\exp\left\{ -\frac{1}{2} (M_i^T \Omega^{-1}_M M_i) \right\}}
\]
Then the new step:

The estimated mean and variance of the parameters.

We would like to find \( \Omega_M^{-1} \) and the current parameters (Borman, 2004). This can be expressed in two steps. Let \( \theta^{(t)} \) be the estimate of the parameters at the \( t \)-th iteration.

**E step:** Define \( Q(\theta|\theta^{(t)}) \) as the expectation of the log complete-data likelihood function of \( \theta \), with respect to the current conditional distribution of \( M \) given the observed data \( Y \) and the current estimates of the parameters.

\[
Q(\theta|\theta^{(t)}) = E_M[Y, \theta'] [\log L(\theta; Y, M)] = \\
\frac{1}{2} [-k \log(2\pi) - k \log(|\Omega^{-1}|) - \sum_{i=1}^{k} \sum_{j=1}^{k} (y_{ij} - M_i) \Omega^{-1} (y_{ij} - M_i)]
\]

\[
+ (y_{ij} - E(M_i))(y_{ij} - E(M_i))^T
\]

\[
- n \log(2\pi) - n \log(|\Omega^{-1}|) - n \text{tr}(\Omega_M(Cov(M_i) + E(M_i)E(M_i)^T))
\]

The estimated mean and variance of \( M \) given \( Y \) and the current estimates of parameters are as follows:

\[
E(M_i|Y_i) = \Omega_M^{-1} \Omega^{-1}(y_{i1} + \ldots + y_{ik_i})
\]

\[
Cov(M_i|Y_i) = \Omega_M^{-1} - k_i \Omega_M^{-1}(k_i \Omega_M^{-1} + \Omega^{-1})^{-1} \Omega_M^{-1}
\]

**M step:** Find parameters \( \theta^{(t+1)} \) that maximize the expectation of log likelihood function of \( \theta \).

\[
\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)})
\]

Then the new \( \Omega_M^{-1} \) is:

\[
\Omega_M^{-1(t+1)} = \sum_{i=1}^{n} \frac{[Cov(M_i|Y_i) + E(M_i|Y_i)E(M_i|Y_i)^T]}{n}
\]

Then the new \( \Omega^{-1} \) is:

\[
\Omega^{-1(t+1)} = \sum_{i=1}^{n} \sum_{j=1}^{k_i} \frac{[Cov(M_i|Y_i) + (y_{ij} - E(M_i|Y_i))(y_{ij} - E(M_i|Y_i))^T]}{k}
\]
The derivations of the equations and the proof of the increment of the likelihood are in appendix B.1 and B.2.

The EM algorithm in our case can be illustrated as in Algorithm 1.

Algorithm 1 EM algorithm

1: inputs: $Y, \epsilon$
2: initialize: \( \hat{\Omega}_M^{-1}(0) \) and \( \hat{\Omega}_\xi^{-1}(0) \)
3: while \(|(l^{(t+1)}(Y) - l^{(t)}(Y))/l^{(t)}(Y)| > \epsilon \) do \( \triangleright \) the absolute value of the relative change in the marginal likelihood of $Y$
4: E step: estimate the mean and variance of $M_i, i = 1, \ldots, n$
5: \( E^{(t+1)}(M_i|Y_i) \leftarrow \text{Solve (2.7) for } E(M_i|Y_i) \)
6: \( \text{Cov}^{(t+1)}(M_i|Y_i) \leftarrow \text{Solve (2.8) for } \text{Cov}(M_i|Y_i) \)
7: M step: update the parameters
8: \( \hat{\Omega}_M^{-1(t+1)} \leftarrow \text{Solve (2.10) for } \Omega_M^{-1} \)
9: \( \hat{\Omega}_\xi^{-1(t+1)} \leftarrow \text{Solve (2.11) for } \Omega_\xi^{-1} \)
10: end while
11: outputs: \( \hat{\Omega}_M, \hat{\Omega}_\xi \)

2.3 Penalization

The sample covariance matrix is usually a speedy estimator of the true one in low-dimensional settings (Anderson, 1984), but is no longer a good estimate in the settings with small samples and high dimensions (Marchenko & Pastur, 1967). When in high-dimensional settings, the eigenvalues of the sample covariance matrix may differ greatly from those of the true one. Besides, when $n$ is smaller than $p$, the estimator has no longer full rank, which leads to many undesirable consequences. $S$ is not positive definite anymore. The corresponding multivariate normal probability density function is ill-defined. The covariance matrix is not invertible as it becomes singular, which means it becomes ill-conditioned, i.e. the ratio of its maximum and minimum singular value is too large. If the sample covariance matrix cannot be inverted, then we cannot use the inverse to approximate $\Omega$, which is necessary for graphical modeling. So the problem becomes how we can obtain a reliable and well-defined estimate of the true covariance matrix in high-dimensional settings.

In recent literature, regularization techniques have been employed to improve the estimation. $\ell_1$ regularization adds an $\ell_1$ penalty, which constrains the absolute value of the magnitude of elements of the precision matrix. $\ell_1$ can yield sparse models with many zero elements. Lasso regression uses this method. Meinshausen and Bühlmann first applied a lasso regression on each variable. Their method was shown to consistently estimate the active elements of the precision matrix (Meinshausen et al., 2006). Yuan & Lin (2007), Banerjee et al. (2008) and J. Friedman et al. (2007) proposed new approaches based on the Maximum Likelihood Estimator with a lasso penalty on the precision matrix. $\ell_2$ adds
an $\ell_2$ penalty, which constrains the square of the magnitude of elements of the precision matrix. The elements are not shrunk to exactly zero, thus the model will not be sparse. Ridge regression uses this method. Ledoit & Wolf (2004) proposed a linear combination of $S$ and a positive definite target matrix. Schäfer & Strimmer (2004) uses an ad-hoc fix of the singular $S$. In some situations where sparsity is preferred, usually $\ell_1$ penalization is chosen. However, sometimes, a more accurate representation of the precision matrix is desired, then $\ell_2$ penalization is preferred.

In this study, we used a ridge regularization proposed by Van Wieringen & Peeters (2016). The alternative ridge estimate can be formulated as follow.

\[
L(\Omega; S) \propto \ln |\Omega| - \text{tr}(S\Omega) - \frac{\lambda}{2} \text{tr} \left[ (\Omega - T)^T(\Omega - T) \right]
\]  

(2.12)

where $S$ is the sample covariance matrix, $\Omega$ is the population precision matrix and $T$ is a target matrix. Their method is shown to outperform the corresponding graphical lasso estimators in terms of loss.

In our case, both covariance matrices of signal and noise should be estimated and penalized to ensure positive definiteness. Thus the penalized marginal likelihood of data $Y$ can be formulated as follows:

\[
\log(f(Y)) = \frac{1}{2}(-k\log(2\pi) + \sum_{i=1}^{n} \log(|\Omega_i|) - \sum_{i=1}^{n} Y_i^T \Omega_i Y_i)
\]

\[ - \frac{\lambda_M n}{2} \text{tr} \left[ (\Omega_M - T_M)^T(\Omega_M - T_M) \right] \]

(2.13)

where $T_M$ is the target matrix for the signal precision matrix and $T_\varepsilon$ is the target matrix for the noise precision matrix. The penalized complete-data log likelihood of $Y$ given $M$, the current parameters, $\lambda_M$ and $\lambda_\varepsilon$, is:

\[
l_p(Y, M, \Omega, \lambda) \propto k\log|\Omega_\varepsilon| - \sum_{i=1}^{n} \sum_{j=1}^{k_i} (y_{ij} - M_i)^T \Omega_\varepsilon (y_{ij} - M_i) - \frac{\lambda_\varepsilon k}{2} \text{tr} \left[ (\Omega_\varepsilon - T_\varepsilon)^T(\Omega_\varepsilon - T_\varepsilon) \right] + n\log|\Omega_M| - \sum_{i=1}^{n} M_i^T \Omega_M M_i - \frac{\lambda_M n}{2} \text{tr} \left[ (\Omega_M - T_M)^T(\Omega_M - T_M) \right]
\]

(2.14)

The penalized EM algorithm can be stated as follows.

**E step:** Define $Q(\theta|\theta^{(t)}, \lambda_M, \lambda_\varepsilon)$ as the expectation of the penalized log likelihood function of $\theta$, with respect to the current conditional distribution of $M$ given the observed data $Y$, the current estimates of the parameters, $\lambda_M$ and $\lambda_\varepsilon$. 

19
\[Q(\theta^{(t)}, \lambda_M, \lambda_\epsilon) = E_{M|Y, \theta^t, \lambda_M, \lambda_\epsilon}[\log L(\theta; Y, M, \lambda_M, \lambda_\epsilon)]\]

\[= \frac{1}{2} \left[-k \log(2\pi) + \log(\|\Omega_\epsilon\|) - \sum_{i=1}^{n} \sum_{j=1}^{k_i} \text{tr}(\Omega_\epsilon(Cov(M_i)) + (y_{ij} - E(M_i))(y_{ij} - E(M_i))^T) - \frac{\lambda_\epsilon}{2} \text{tr}[(\Omega_\epsilon - T_\epsilon)\Omega_\epsilon - T_\epsilon]\right] \quad (2.15)\]

\[-n \log(2\pi) + n \log(\|\Omega_M\|) - \sum_{i=1}^{n} \text{tr}(\Omega_M(Cov(M_i) + E(M_i)E(M_i)^T)) \]

\[-\frac{\lambda_M n}{2} \text{tr}[(\Omega_M - T_M)^T(\Omega_M - T_M)]\]

The estimated mean and variance of \(M\) given \(Y\), the current estimates of parameters, \(\lambda_M\) and \(\lambda_\epsilon\) are as follows:

\[E(M_i|Y_i) = \Omega_M^{-1}(k_i\Omega_M^{-1} + \Omega_\epsilon^{-1})^{-1}(y_{i1} + ... + y_{ik_i}) \quad (2.16)\]

\[Cov(M_i|Y_i) = \Omega_M^{-1} - k_i(\Omega_M^{-1} + \Omega_\epsilon^{-1})^{-1}\Omega_M^{-1} \quad (2.17)\]

So the **E step** remains the same as in the unpenalized EM algorithm.

**M step:** Find parameters \(\theta^{(t+1)}\) that maximize the expectation of the penalized log likelihood function of \(\theta\).

\[\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)}, \lambda_M, \lambda_\epsilon) \quad (2.18)\]

Then the new \(\Omega_M^{-1}\) should be,

\[\Omega_M^{-1(t+1)} = \frac{\left(S_M - \lambda_M T_M\right)^2}{4} + \lambda_M I_p + \frac{1}{2} \left(S_M - \lambda_M T_M\right) \quad (2.19)\]

where \(S_M = \sum_{i=1}^{n}[\text{Cov}(M_iY_i) + E(M_iY_i)E(M_iY_i)^T]\). Then the new \(\Omega_\epsilon^{-1}\) should be,

\[\Omega_\epsilon^{-1(t+1)} = \frac{\left(S_\epsilon - \lambda_\epsilon T_\epsilon\right)^2}{4} + \lambda_\epsilon I_p + \frac{1}{2} \left(S_\epsilon - \lambda_\epsilon T_\epsilon\right) \quad (2.20)\]

where \(S_\epsilon = \sum_{i=1}^{n}\sum_{j=1}^{k_i}[\text{Cov}(M_iY_i) + (y_{ij} - E(M_iY_i))(y_{ij} - E(M_iY_i))^T]\). The derivations of the equations and the proof of the increment of the penalized likelihood are in appendix B.3.

### 2.3.1 Choice of \(\lambda_M, \lambda_\epsilon\)

The tuning parameter controls the amount of regularization, so it is crucial to choose good tuning parameters. They are usually chosen based on prediction accuracy on the new data.

20
Cross-validation is an intuitive and simple way to estimate prediction error. The data is first divided into $K$ folds of roughly equal size. Then, every time, we leave one fold out as test data and the other $K-1$ folds as training data. The training data is used to train a model and the test data, which does not appear in the training data, is used to evaluate the performance of the trained model. The procedure iterates over $K$ folds, meaning each fold will be used as test data once. So in the end, we have $K$ models and $K$ prediction error rates. The $K$ prediction error rates are then averaged and used as a measurement of the model performance.

Considering the dependency structure in our data, we split it into $K$ folds over samples. Our data can be partitioned into different samples. Instead of randomly dividing the data by observations, we use samples as units. So, the replications of a sample will all be in the same fold. Replications of the same sample should have more in common than replications of different samples. If we sample based on observations, it is unlikely all replications of the same samples will be in the same folds. Then some of the replications of the same sample are used to train a model and the rest are used to test it. Given the dependency between replications, the test data has some overlapping with the training data and the trained model has in a way seen part of the test data. Thus the prediction error is likely to be underestimated.

In this study, we used a 10-fold cross-validation to find a good value of such tuning parameters. The data was first divided over samples into 10 folds of roughly equal size. Each time, 9 out of 10 folds are used as the training data and the remaining one fold as the test data. We then set a range for $\lambda_M, \lambda_\varepsilon$ from a very small value to a large one. For different combinations of $\lambda_M, \lambda_\varepsilon$, we estimated the desired parameters $\Omega^{-1}_M, \Omega_\varepsilon^{-1}$ and evaluated the model performance based on the new data likelihood. The combination of $\lambda_M, \lambda_\varepsilon$ that maximizes the new data likelihood was chosen. To speed up the process, we performed an “automatic” search for the optimal penalty parameter by employing a simplex search method to the calculation of a cross-validated negative log-likelihood score. The automatic search is implemented with R function “optim”.

The penalized EM algorithm in our case is given in Algorithm 2. The penalization is implemented with R function “ridgeP” in package “rags2ridges”.

### 2.4 Measure of the performance

#### 2.4.1 Kullback-Leibler divergence

Kullback-Leibler divergence, short for KL divergence, is a way of comparing two probability distributions. In probability and statistics, a complex distribution is often replaced with a simpler approximating distribution. The KL divergence measures the information we have lost when approximating the true distribution (Kullback & Leibler, 1951). It originates from information theory. The basic idea behind information theory is that the more we know about a topic, the less new information we can get about it. If an event is very likely to happen, then when it happens, little new information would be provided. On the contrary, if an event is very unlikely to happen, then it would be much more informative when it happens.

One of the most important concepts in information theory is “entropy”, often denoted as $H$. The
Algorithm 2 EM algorithm with penalization

1: inputs: Outer loop: $\mathbf{Y}, \epsilon, \{\lambda_M\}, \{\lambda_\epsilon\}, K$ folds
2: initialize: $\hat{\Omega}_M^{-1(0)}$ and $\hat{\Omega}_\epsilon^{-1(0)}$
3: cross-validation: divide the dataset into $K$ folds
4: for $k$ in $K$ folds do
5: for $\lambda_M \in \{\lambda_M\}, \lambda_\epsilon \in \{\lambda_\epsilon\}$ do
6: inputs: Inner loop: $\mathbf{Y}_k, \epsilon, \lambda_M, \lambda_\epsilon$
7: while $|l_{l(t+1)}(\mathbf{Y}_{-k}) - l_{l(t)}(\mathbf{Y}_{-k})|/l_{l(t)}(\mathbf{Y}_{-k})| > \epsilon$ do ▶ the absolute value of the relative change in the penalized marginal likelihood of $\mathbf{Y}_{-k}$
8: E step: estimate the mean and variance of $\mathbf{M}_i, i = 1, ..., n$
9: $E^{(t+1)}(\mathbf{M}_i|\mathbf{Y}_i) \leftarrow$ Solve (2.16) for $E(\mathbf{M}_i|\mathbf{Y}_i)$
10: $Cov^{(t+1)}(\mathbf{M}_i|\mathbf{Y}_i) \leftarrow$ Solve (2.17) for $Cov(\mathbf{M}_i|\mathbf{Y}_i)$
11: M step: update the parameters
12: $\hat{\Omega}_M^{-1(t+1)} \leftarrow$ Solve (2.19) for $\Omega_M^{-1}$
13: $\hat{\Omega}_\epsilon^{-1(t+1)} \leftarrow$ Solve (2.20) for $\Omega_\epsilon^{-1}$
14: end while
15: outputs: $\hat{\Omega}_M, \hat{\Omega}_\epsilon$
16: end for
17: outputs: $\lambda_M, \lambda_\epsilon, l(\mathbf{Y}_k, \mathbf{M}, \theta, \lambda_M, \lambda_\epsilon)$
18: end for
19: outputs: $\lambda_M, \lambda_\epsilon, \bar{l}(\mathbf{Y}, \mathbf{M}, \theta, \lambda_M, \lambda_\epsilon)$ ▶ $\bar{l}$ is the average of $K$ penalized marginal likelihoods of the test data
20: inputs: $\lambda_M, \lambda_\epsilon$ that have the maximum average marginal likelihood of the test data
21: outputs: $\hat{\Omega}_M, \hat{\Omega}_\epsilon$
The definition of “entropy” for a probability distribution of continuous variables is,

\[ H = - \int_{-\infty}^{\infty} p(x) \cdot \log p(x) \, dx \]

(2.21)

Then “entropy” for a multivariate normal distribution is,

\[ H = \frac{1}{2} (\log |\Sigma| + p) \]

(2.22)

where \( p \) is the number of variables.

The word “entropy” means disorder or uncertainty. In information theory, it is a measure of unpredictability of the state, or as the equation (2.21) shows, of the average information content. Entropy is zero when one outcome is bound to happen. The entropy only encapsulates the information about the underlying probability distribution, so the meaning of the events themselves does not matter. In other words, entropy quantifies the information in our data.

KL divergence is the relative entropy. Cover & Thomas (2012) note that the KL distance is “... a measure of the inefficiency of assuming that the distribution is \( g \) when the true distribution is \( f \).” It is a slight modification of the formula for entropy. Apart from the true distribution \( p \), an approximating distribution \( q \) is included.

\[
D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \left( \frac{p(x)}{q(x)} \right) \, dx
\]

(2.23)

The right-hand side is the expected value of log difference in the true distribution and the approximated one for the continuous case. It can be seen as an average of \( \log \left( \frac{p(x)}{q(x)} \right) \) with respect to \( P \). The measure is not symmetric. That means the divergence relative to the true one between the two distributions is different from that with respect to the approximated one.

When dealing with multivariate Gaussian distributions, the KL divergence can be computed by (2.25), where the \( \Sigma_0 \) is the reference distribution, \( \Sigma_1 \) the approximating one, \( \mu_0, \mu_1 \) are the means of the two distributions, \( p \) the dimensions (Duchi, 2007).

\[
D_{KL}(\mathcal{N}_0 \parallel \mathcal{N}_1) = \frac{1}{2} \left( \text{tr} \left( \Sigma_1^{-1} \Sigma_0 \right) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) - p + \ln \left( \frac{|\Sigma_1|}{|\Sigma_0|} \right) \right)
\]

(2.24)

KL divergence is always non-negative. It is zero if and only if the two distributions are identical. The large value indicates small overlapping, while the small value implies large overlapping between those two distributions. In our study, we intend to evaluate the performance of our algorithm by calculating the KL divergence between the true distribution of the signal and our estimated one. The smaller the KL divergence is, the closer those two distributions are. As the mean of the distribution of the signal is assumed to be zero in this study, then the simplified formula is:
The smaller the KL divergence is compared to other methods, the closer to the true distribution the approximating distribution is.

### 2.4.2 Mutual information

The mutual information measures the mutual dependence between two random variables. The mutual information quantifies “the amount of the information” shared by two random variables $X$ and $Y$. More specifically, it measures how much information is obtained about one variable through observing the other. In other words, it determines how much uncertainty is reduced about one variable given the other variable.

It has its origins in information theory and is linked to the fundamental concept entropy. The relationships of entropy and mutual information between two random variables are illustrated in fig.2.1. The circle on the left is the entropy of variable $X$, $H(X)$, the circle on the right is the entropy of variable $Y$, $H(Y)$, and the area in the middle shared by the two variables is the “mutual information”, $I(X;Y)$. The area left of $X$ after removing the shared part is the entropy of $X$ conditioned on $Y$, $H(X|Y)$, which can be interpreted as “the uncertainty remaining from predicting $X$ given $Y$”. Vice versa for $H(Y|X)$. The entropy of the union of $X$ and $Y$ is $H(X,Y)$. If $X$ and $Y$ are independent, knowing one variable would not help with knowing the other, then there is no shared information, that means the mutual information is zero. If $X$ solely determines $Y$, then knowing $X$ provides all information for $Y$ within this observation of system. The mutual information is non-negative, symmetric, additive and subtractive (Cover & Thomas, 2012). It can be equivalently expressed as:

$$I(X;Y) = I(Y;X) = H(X) + H(Y) - H(X,Y) = H(X,Y) - H(X|Y) - H(Y|X) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$ (2.26)

In the case of two continuous random variables $X$ and $Y$, the mutual information can be defined as:

$$I(X;Y) = \int \int_{x,y} p(x,y) \log \left( \frac{p(x,y)}{p(x)p(y)} \right) dx dy$$ (2.27)

where $p(x,y)$ is the joint probability density function of $X$ and $Y$, $p(x)$ and $p(y)$ are the marginal probability density function of $X$ and $Y$ respectively (Cover & Thomas, 2012).

The application of mutual information is not limited to the real-valued variables. As the equation (2.27) shows, it determines the similarity between the joint distribution $p(x,y)$ and the products of factored marginal distribution $p(x)p(y)$.
The mutual information has a link to the KL-divergence and can be expressed as:

\[ I(X;Y) = D_{KL}(p(x,y)||p(x)p(y)) \]  

(2.28)

which can be interpreted as KL divergence between the joint distribution and the products of the two marginal distributions.

In our study, we measure the mutual information between our data \( Y \) and the signal \( M \). The estimated mutual information will be compared to the true one to see how much information is recovered by our estimates. As both our signal and noise follow multivariate Gaussian distributions and we assume the independence of signal and noise, our additive Gaussian channel can be illustrated in fig.2.2. \( Y \) consists of signal and noise, which come from different channels. For a random draw of \( Y \) from this multivariate normal distribution, the mutual information between it and \( M \) is,

\[ I(M;Y) = H(Y) - H(Y|M) \]

\[ = 0.5 \times (\log |\Sigma_y| - \log |\Sigma_e|) \]

\[ = 0.5 \times (\log |\Sigma_M + \Sigma_e| - \log |\Sigma_e|) \]  

(2.29)

where \( \Sigma_M, \Sigma_e \) are the covariance matrices for the signal and the noise respectively. The proof of the above formula is in appendix D.

Since the mutual information has no upper bound, it is difficult to make comparisons and draw conclusions. Thus a normalized version ranging from 0 to 1 is desirable. There are many forms of normalization, here we use one given by Strehl & Ghosh (2002), which uses the geometric mean and is calculated analogous to the Pearson correlation coefficient.
The normalized mutual information has a value between 0 and 1. When it is 0, there is no shared information between $X$ and $Y$. When it is 1, $X$ is deterministic for $Y$. The modified equation to our study is:

$$NMI(M, Y) = \frac{I(M, Y)}{\sqrt{H(M) H(Y)}}$$

(2.31)

The derivation of equation (3.31) is in appendix D too.

### 2.4.3 ROC and AUC

One of the most important applications of precision matrix is to construct networks. As mentioned in Chapter 1, section 1.2.2, the absence of an edge in the network is indicated by a zero element in the corresponding precision matrix, equivalently, a non-zero element represents the presence of an edge. Then the detection of edges can be viewed as a classification problem. The edges can be either present or absent, which is indicated as “1” or “0” classes respectively.

ROC is used in this study to measure the classification performance of our estimates. The ROC graph plots sensitivity against (1- specificity) at various threshold settings. The sensitivity is the true positive rate (TPR), and specificity is the true negative rate (TNR). The contingency table of conditions and outcomes is illustrated in table 2.1. The true positive number is the number of tests that turn out to be positive when the true condition is positive. If the condition is positive, but is identified as negative, then it is called “false negative”, which is the “Type II error”. When the condition is negative and the outcome matches the condition, then it is called “true negative”. If it does not match, it is the “false positive”, which is the “Type I error”. The number of positive conditions is $TP + FN$, the number of
Table 2.1: Confusion matrix

<table>
<thead>
<tr>
<th>Condition positive</th>
<th>Condition negative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test outcome positive</td>
<td>True positive (TP)</td>
<td>False positive (FP)</td>
</tr>
<tr>
<td>Test outcome negative</td>
<td>False negative (FN)</td>
<td>True Negative (TN)</td>
</tr>
<tr>
<td>Total</td>
<td>TP+FN</td>
<td>FP+TN</td>
</tr>
</tbody>
</table>

negative conditions $FP + TN$, the number of positive outcomes $TP + FP$ and the number of negative outcomes $FN + TN$. The total number of conditions or tests is $TP + FP + FN + TN$.

The true positive rate (sensitivity), the false positive rate (1 - specificity), the true negative rate (specificity) and the false negative rate (1 - sensitivity) can be calculated as in equation (2.32), (2.33), (2.34) and (2.35) respectively.

$$TPR(sensitivity) = \frac{TP}{TP + FN}$$ (2.32)

$$FPR = \frac{FP}{FP + TN}$$ (2.33)

$$TNR(specificity) = \frac{TN}{FP + TN} = 1 - FPR$$ (2.34)

$$FNR = \frac{FN}{TP + FN} = 1 - TPR$$ (2.35)

In our study, we used the value of the first $l$th partial correlation coefficient as the threshold, $l = 1, \ldots, p(p - 1)/2$, where $p$ is the dimension. That means, at the $l$th threshold, only $l$ edges with the largest absolute values of partial correlation coefficients are selected. This is realized by keeping only top $l$ partial correlation coefficients with highest absolute values and setting other elements in the precision matrix to zero. The outcomes then are compared to the true conditions and the corresponding true positive rate and false positive rate are calculated.

The Area Under the ROC Curve is called “AUC”. It is equal to the probability that a classifier will assign a higher score to a randomly chosen positive instance than a randomly chosen negative one (Fawcett, 2006). If the area under the ROC curve of A is greater than that of B, then on average, classifier A performs better than B. A high-AUC classifier may be worse than a low-AUC classifier in classification in a specific region of ROC space. The AUC is a portion of the area of the unit square and its value is always between 0 and 1. The random guess produces the diagonal line between (0,0) and (1,1) of the unit square, which has an area of 0.5. So if our classifier has an area larger than 0.5, then the classifier performs better than a random guess. If the AUC is 0.5, then the model has no class separation capacity. If our classifier has an area less than 0.5, then it performs even worse than a random guess. AUC can be approximated by using trapezoidal rules. It is chosen as the performance measurement because it summarises the overall model performance over all possible thresholds so that the supposedly subjectivity in the threshold selection process, when continuous probabilities are converted to binary variables, is avoided.
2.5 Local False Discovery Rates

For high-dimensional data, the statistical significance for many features are assessed simultaneously. But the more inferences we make, the more likely there will be erroneous inferences. This is called the multiple testing problem. To compensate for the number of inferences we make, a stricter significance threshold is required.

In the real-world data analysis part, we used local false discovery rates (fdr) in network inference. It is an implementation of an empirical Bayesian method for large-scale testing situations. The advantage of using ‘fdr’ is that it allows the possibility of local inference (Efron, 2005). In other words, it estimates the probability of null hypothesis being true conditioning on an actual value,

\[
P(H_0 | z_i = z) = \frac{\pi_0 f_0(z_i)}{f(z)}
\]

where \(i = 1, ..., N\), \(N\) is the number of hypothesis testing, \(N\) must be large for local fdr calculations, at least in the hundreds, \(H_{0i}\) is the null hypothesis and \(z_i\) the z-values.

By Bayes rule, we can obtain an expression for equation (2.36),

\[
\text{fdr}(z_i) = P(H_{0i} | z_i = z) = \frac{\pi_0 f_0(z_i)}{f(z)}
\]

where \(f(z)\) is the marginal density of z-values,

\[
f(z) = \pi_0 f_0(z) + \pi_1 f_1(z)
\]

\(\pi_0\) is the prior probability of null models in \(N\) cases, \(\pi_1\) is the prior probability of non-null models, \(f_0(z)\) is the null density, and \(f_1(z)\) is the non-null density. So fdr is computed at the density level.

A conventional threshold in application is,

\[
\frac{\text{fdr}(z_i)}{fdr(z)} \leq 0.20
\]

We also used this threshold in our analysis for the real-world data. If this cutoff threshold is used, then the posterior odds ratio becomes,

\[
P(H_{0i} | z_i = z) / P(H_{1i} | z_i = z) = (1 - \text{fdr}(z)) / \text{fdr}(z)
\]

\[
= \frac{\pi_1 f_1(z)}{\pi_0 f_0(z)}
\]

\[
\geq 0.8 / 0.2 = 4
\]

Local FDR is readily interpretable as it is an empirical Bayesian posterior probability and not some variant of a corrected p-value. However, the calculations are made on the density level, it is difficult to estimate. Strimmer (2008) proposed a semi-parametric method to estimate fdr. Their algorithm is implemented in R package “fdrtool”. We also used this package in our analysis of real-world data.
Chapter 3

Results

The previous chapters introduce the background and the methodology. This chapter mainly presents the results of simulation studies and real-life data analysis.

While the proposed method utilizes replications to estimate both signal and noise parameters, simpler methods, such as averaging the data, can be used. In the simulation studies and real-life data analysis, the performance of the proposed method is compared to that of the following methods.

method 1. **Direct estimation:** Though there is dependency in the data with replications, the precision matrix of observations can be directly estimated by ignoring such dependency structure. By this method, the signal is supposedly weakened or even obscured by the noise. We expect the estimates by this method not to be very close to the true ones.

method 2. **Averaging:** The proposed method estimates both signal precision and noise precision matrix. But the latter is of no use in our study. Instead of estimating it, we can try to get rid of it or at least reduce the effect of it. The averaged data over samples is a simple and intuitive choice. In this way, we expect the noise is largely averaged out and the estimate of this method should be better than that of direct estimation.

method 3. **Individual-sample estimation:** While direction estimation and averaging are attempted in the above methods, random draw of one replication of each sample offers another approach. By this method, the sample size is reduced. We expect it to perform worst. The estimate of this method is taken as a reference.

3.1 Simulation Study

In this section, the proposed method and the above approaches are tested on the synthetic data in a few scenarios. For this simulation study, the true networks of signal are known to us.

The simulation study aims to address the following questions:
In this study, replications are used to extract the signal from the data. Do the replications improve the estimation?

Can the pattern of the signal be recovered by the proposed method?

How would the sparsity of the network affect the performance of our method?

How would the level of noise in the data affect the performance of our method?

How is the performance of the estimate by the proposed method compared to those simpler methods?

How would the performance of the proposed method change with varying samples and replications?

We test 6 set-ups listed as follows:

Set-up 1. $\Omega_M$ (the precision matrix of signal) has a clique structure consisting of three roughly equal-sized blocks along the diagonal, with each diagonal element equal to 1 and off-diagonal element 0.5.

Set-up 2. $\Omega_M$ has a clique model consisting of four roughly equal-sized blocks along the diagonal, with each diagonal element equal to 1 and off-diagonal element 0.5.

Set-up 3. $\Omega_M$ has a clique model consisting of five roughly equal-sized blocks along the diagonal, with each diagonal element equal to 1 and off-diagonal element 0.5.

Set-up 4. $\Omega_M$ has a banded structure representing a conditional independence graph with $c_{ii} = 1, c_{i,i-1} = c_{i-1,i} = 0.5, c_{i,i-2} = c_{i-2,i} = 0.4$, and $c_{i,i-3} = c_{i-3,i} = 0.3$.

Set-up 5. $\Omega_M$ has a banded structure representing a conditional independence graph with $c_{ii} = 1, c_{i,i-1} = c_{i-1,i} = 0.5, c_{i,i-2} = c_{i-2,i} = 0.4, c_{i,i-3} = c_{i-3,i} = 0.3, c_{i,i-4} = c_{i-4,i} = 0.2$, and $c_{i,i-5} = c_{i-5,i} = 0.1$.

Set-up 6. $\Omega_M$ has a star structure representing a conditional independence graph with $c_{ii} = 1, c_{1,j+1} = c_{j+1,1} = 0.1$, for $j = 1, 2, \ldots, p-1$ and $c_{ij} = 0$ otherwise. The first node is a hub and other nodes are connected and only connected to it.

In set-up 1, 2, 3, $\Omega_M$ has clique structures but with different sparsities. In set-up 4 and 5, $\Omega_M$ has banded structures with different numbers of edges. Unlike same partial correlations in the same cluster of the clique structures, banded structures have weaker partial correlations for elements further away from the diagonal. The graph of $\Omega_M$ in set-up 6 is shaped like a star with same strength for each edge connected to the first node.

For each set-up, we simulate data with sample size $n = 25$ and dimension $p = 5$, $n = 100$ and $p = 10$, or $n = 100, 300$ and $p = 50$, the number of replications $k$ varies from 2 to 4. We also evaluate the performance of the proposed method and other methods in the high-dimensional settings with $n = 25, 50$ and $p = 50$.

The above 6 set-ups are tested in 5 scenarios. In scenario 1 to 3, the number of replications is varied to see how this affects the estimation. In scenario 1 and 2, the number of replications is 2 and
3 respectively, so we have balanced design of equal number of replications for each sample. In the third scenario, we break up the balanced design and each sample can have 2 to 4 replications. In high-dimensional settings, the estimates are penalized. In scenario 4, prior knowledge is incorporated and a target matrix is given for our estimate. In the scenario 1-4, the precision matrix of noise is assumed to be diagonal with unit elements. In scenario 5, the diagonal elements of noise precision matrix is re-scaled to control the level of noise.

3.1.1 Scenario 1: 2 replications

In this scenario, we sample the data with 2 replications and test the proposed method on such data in all set-ups. The performance of our method is compared to that of other methods in terms of KL divergence and ROC. Besides, the graphs of the estimates by the proposed method for set-up 1, 4, 6 are compared with the actual patterns of precision matrices in those set-ups.

For set-up 1, the unpenalized estimate by the proposed method seems to be a good estimate and a better one than the penalized counterpart. The graph of our unpenalized estimate of the signal precision matrix shows a similar pattern to that of the actual precision matrix in set-up 1, as indicated in fig.3.1. It is clear to see 3 clusters along the diagonal for the unpenalized estimate, while this pattern is obscured after penalization.

The results of KL divergence deliver a different message that the penalized estimate by the proposed method is better than the unpenalized counterpart or estimates by other methods in set-up 1. As table 3.1 shows, in all cases we have tested our penalized estimate achieved the lowest KL divergence. The penalized estimates by averaging achieves the second lowest KL divergence among all the penalized estimates. It is also worth noticing that the mean and standard deviation of KL divergence of our unpenalized estimate is very high When $n = 100, p = 50$. This may be due to the ill-conditionedness of those estimates.

We also find that KL divergence between the true distribution and the approximating distributions constructed with the estimates is negatively related to the sample size. Table 3.1 shows as $n$ increases from 25 to 50 with fixed $p = 50$ in high-dimensional settings, the KL divergence of all penalized estimates drops. As $n$ increases from 100 to 300 in non-high-dimensional settings, the KL divergence also falls. This is what we expected, as with more samples, the estimation is more accurate.

ROC plots are used to measure the classification ability of those methods. The absence or presence of edges are two classes, and the classification accuracy is calculated by mapping actual instances to those two classes under top $l$ edge selection criterion as mentioned in Chapter 2, section 2.4.3. Only the ROCs of $p = 50$ scenarios are studied, as on lower dimensions, the plots have many fluctuations and are thus not smooth.

The proposed method without penalization performs the best in classifying edges with a high AUC of 0.976, and averaging estimation, direct estimation, individual-sample estimation (the reference method) without penalization comes second, third and fourth respectively, as shown in fig.3.2. With penalization, however, the performance of the proposed method is slightly weaker than averaging and direct estimation. Individual-sample estimation, which utilizes only one replication of each sample thus estimates with fewer samples, comes last as we expected. In the high-dimensional setting of $n = p$, AUCs
Figure 3.1: heatmap of set-up 1. The upper panels depict the patterns of the estimated precision matrices by the proposed method when \( n = 300, p = 50 \). The left-hand panel depicts results for the unpenalized estimate while the right-hand panel shows results for penalized one. The lower panel depicts the true pattern in set-up 1. The axes on the left and at the bottom are the names of variables.

Table 3.1: KL divergence of set-up 1. It records the mean and standard deviation of KL divergence between approximating distributions using different estimates and the reference distribution with the true precision matrix in set-up 1 after 20 runs.

<table>
<thead>
<tr>
<th>n</th>
<th>p</th>
<th>( \Omega_M )</th>
<th>( \Omega_{M,\text{pen}} )</th>
<th>( \Omega_Y )</th>
<th>( \Omega_{Y,\text{pen}} )</th>
<th>( \Omega_{\text{ind}} )</th>
<th>( \Omega_{\text{ind,pen}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>5</td>
<td>5.76(8.57)</td>
<td>0.43(0.12)</td>
<td>0.55(0.08)</td>
<td>0.75(0.11)</td>
<td>0.61(0.11)</td>
<td>0.61(0.12)</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>11.90(9.42)</td>
<td>0.76(0.11)</td>
<td>1.07(0.10)</td>
<td>1.39(0.10)</td>
<td>0.85(0.17)</td>
<td>1.03(0.12)</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>/</td>
<td>6.73(0.12)</td>
<td>/</td>
<td>7.93(0.15)</td>
<td>/</td>
<td>7.81(0.26)</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>/</td>
<td>5.80(0.16)</td>
<td>/</td>
<td>7.35(0.18)</td>
<td>/</td>
<td>6.99(0.30)</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>581.58(113.30)</td>
<td>4.79(0.10)</td>
<td>7.18(0.27)</td>
<td>6.66(0.16)</td>
<td>14.55(1.34)</td>
<td>6.01(0.18)</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>12.31(2.33)</td>
<td>3.43(0.05)</td>
<td>4.36(0.07)</td>
<td>5.43(0.08)</td>
<td>3.54(0.11)</td>
<td>4.39(0.07)</td>
</tr>
</tbody>
</table>

\( \hat{\Omega}_M \) and \( \hat{\Omega}_{M,\text{pen}} \) are the unpenalized and penalized estimates by the proposed method. \( \hat{\Omega}_Y \) and \( \hat{\Omega}_{Y,\text{pen}} \) are the counterparts by direct estimation. \( \hat{\Omega}_{\text{ind}} \) and \( \hat{\Omega}_{\text{ind,pen}} \) are the estimates by averaging method. In high-dimensional settings \( (n \leq p) \), only the penalized estimates are compared, because the unpenalized estimates no longer exist.
Figure 3.2: ROC of set-up 1. The left-hand panel is the ROC plot of all methods when $n = 300, p = 50$ while the right-hand panel depicts curves of estimates with $n = 50, p = 50$. We have both penalized and unpenalized estimates for all methods in non-high-dimensional setting and only penalized one in high-dimensional setting. “Signal” and “Signal penalized” stand for the unpenalized and penalized estimates by the proposed method respectively, “Y” and “Y penalized” are the estimates by direct estimation, “Averaged Y” and “Averaged Y Penalized” by averaging, “Individual Y” and “Individual Y Penalized” by individual-sample estimation.

of all methods with penalization is close to 0.5, indicating the classification abilities of those methods are no better than random guess. This is probably due to a small sample size with respect to a large number of edges (one-third of the edges are present).

The network in set-up 1 is quite dense with one-third of the edges present while in real life, the genetic networks are very often sparsely connected, so how would those methods react to sparser networks? To evaluate the performance, we test those methods in set-up 2 and 3, where networks have fewer number of edges present. As the ROC plots of set-up 2 and 3 shown in fig.18, 2 in Appendix E, all the classification accuracies for those two sparser networks improve while the rank of performance remains the same. However, the gap in AUCs of our unpenalized estimate (the best) and the unpenalized estimate by averaging (the second best) is closing, meaning the advantage of using our unpenalized method over the averaging approach is diminishing with decreasing number of edges.

The precision matrices in set-up 4, 5 have different structures than in set-up 1, 2 and 3. In set-up 1, 2 and 3, the precision matrices have a clique structure, which is shaped like clusters and elements in the same cluster have same partial correlation coefficients. The precision matrices in set-up 4 and 5 have a banded structure with three and 5 bands along the diagonal respectively. Elements in the same band have same partial correlation coefficients and those partial correlation values are smaller for bands further away from the diagonal.

For set-up 4, the penalized estimate recovers the sparse structure well, as indicated in 3.3. The graph of our unpenalized estimate shows the bands along the diagonal, but also reveals other patterns away from the diagonal, which indicates fake conditional dependencies. Compared to the unpenalized estimate, the pattern our penalized estimate has presented matches well with the true one. Only the area close to the diagonal is dark, so the sparse network in this set-up is correctly reconstructed.
Figure 3.3: heatmap of set-up 4. The upper panels depict the patterns of the estimated precision matrices by the proposed method when $n = 300, p = 50$. The left-hand panel depicts results for the unpenalized estimate while the right-hand panel depicts results for penalized one. The lower panel depicts the true pattern of model 4. The axes on the left and at the bottom are the names of variables.
Figure 3.4: ROC of model 4. The left-hand panel is the ROC plot of all methods when \( n = 300, p = 50 \) while the right-hand panel depicts curves of estimates with \( n = 50, p = 50 \).

The results of KL divergence in set-up 4 are similar to those in set-up 1. As shown in table 3.2, the penalized estimate by our method again achieves the lowest or second lowest KL divergence in all cases. Our proposed method without penalization has a large averaged KL divergence with a high standard deviation, which indicates the instability of the fit.

Table 3.2: KL divergence of set-up 4. It records the mean and standard deviation of KL divergence between approximating distributions using different estimates and the reference distribution with the true precision matrix in set-up 4 after 20 runs.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( p )</th>
<th>( \Omega_M )</th>
<th>( \Omega_{M,\text{pen}} )</th>
<th>( \Omega_Y )</th>
<th>( \Omega_{Y,\text{pen}} )</th>
<th>( \Omega_{\text{ind}} )</th>
<th>( \Omega_{\text{ind,pen}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>5</td>
<td>3.98(5.73)</td>
<td>0.42(0.09)</td>
<td>0.52(0.10)</td>
<td>0.70(0.13)</td>
<td>0.41(0.10)</td>
<td>0.58(0.12)</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>10.39(9.62)</td>
<td>0.85(0.13)</td>
<td>1.10(0.09)</td>
<td>1.40(0.09)</td>
<td>0.91(0.17)</td>
<td>1.09(0.11)</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>/</td>
<td>10.32(0.28)</td>
<td>/</td>
<td>11.36(0.22)</td>
<td>/</td>
<td>11.34(0.29)</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>/</td>
<td>8.49(0.24)</td>
<td>/</td>
<td>9.90(0.15)</td>
<td>/</td>
<td>9.56(0.25)</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>679.78(103.52)</td>
<td>6.40(0.18)</td>
<td>7.63(0.27)</td>
<td>8.47(0.15)</td>
<td>13.51(1.41)</td>
<td>7.61(0.24)</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>20.98(4.40)</td>
<td>3.56(0.10)</td>
<td>5.21(0.08)</td>
<td>6.56(0.10)</td>
<td>3.55(0.10)</td>
<td>4.84(0.13)</td>
</tr>
</tbody>
</table>

* the notation is the same as in table 3.1.

In this set-up with a sparse network, all methods achieve high AUCs, indicating good performance in classification, as shown in fig.3.4. The unpenalized estimate by the proposed method comes last with the lowest AUC, the penalized and unpenalized estimates of the individual-sample estimation come second to last. The ROC plots of other methods almost overlap. In contrast to ROC in set-up 1, our penalized estimate in set-up 4 outperforms its unpenalized counterpart. This result also echoes with the heatmaps in fig.3.3. In the high-dimensional setting with \( n = 50, p = 50 \), the ROC plot shows a much higher overall prediction accuracy for this sparser network (with 144 edges) than more dense networks in set-up 1, 2 and 3. Our method performs best and the individual-sample estimation comes last.

In set-up 5, two more bands are added along the diagonal and thus the number of edges is increased to see how that would affect the performance of our method in a banded structure. Overall, the prediction accuracy lowers compared to the sparser network in set-up 4, as indicated in fig.3.5. The unpenalized estimate by the proposed method is one of the best in terms of AUC, close to that of
averaging estimation (0.823 vs. 0.829). It performs better than the penalized counterpart (0.823 vs. 0.784). As we increase the number of edges in this banded structure, the performance of our unpenalized estimate rises from the bottom to the top, like in a dense network with a clique structure in set-up 1, 2, 3. It seems that the proposed method without penalization works better for dense networks.

In set-up 6 with an even sparser network, our unpenalized estimate outperforms the penalized one. As indicated in fig.3.6, the true pattern seems to be lost in the graph of the unpenalized estimate by the proposed method while the penalized method does better than the unpenalized one in recovering this sparse pattern. However, due to the weak correlation between the first node and the rest (0.1 only), the recovered pattern is not very clear.

In terms of KL divergence, the penalized estimate by the proposed method is the best in all cases with a small standard deviation, as shown in table 3.3. This result matches the results of KL divergence in set-up 1 and 4.

Table 3.3: KL divergence of Set-up 6

<table>
<thead>
<tr>
<th>n</th>
<th>p</th>
<th>$\Omega_M$</th>
<th>$\Omega_{M,pen}$</th>
<th>$\Omega_Y$</th>
<th>$\Omega_{Y,pen}$</th>
<th>$\Omega_{Y,ind}$</th>
<th>$\Omega_{Y,ind,pen}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>5</td>
<td>6.13(7.68)</td>
<td>0.26(0.10)</td>
<td>0.50(0.12)</td>
<td>0.72(0.16)</td>
<td>0.38(0.12)</td>
<td>0.50(0.16)</td>
</tr>
<tr>
<td>50</td>
<td>10</td>
<td>14.12(12.15)</td>
<td>0.43(0.08)</td>
<td>1.00(0.11)</td>
<td>1.36(0.12)</td>
<td>0.75(0.14)</td>
<td>0.87(0.14)</td>
</tr>
<tr>
<td>25</td>
<td>50</td>
<td>/</td>
<td>4.93(0.20)</td>
<td>/</td>
<td>8.86(0.32)</td>
<td>/</td>
<td>7.17(0.43)</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>/</td>
<td>3.96(0.26)</td>
<td>/</td>
<td>8.40(0.30)</td>
<td>/</td>
<td>6.43(0.44)</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>956.72(91.66)</td>
<td>2.91(0.11)</td>
<td>6.04(0.18)</td>
<td>7.73(0.21)</td>
<td>11.78(0.88)</td>
<td>5.56(0.26)</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>23.22(3.62)</td>
<td>1.63(0.05)</td>
<td>4.91(0.08)</td>
<td>6.54(0.09)</td>
<td>2.99(0.07)</td>
<td>4.03(0.10)</td>
</tr>
</tbody>
</table>

For this very sparse network, again the AUC of our penalized estimate is higher than the unpenalized one, which performs worst, the same as in set-up 4 (in fig.3.7). In fact, all the penalized estimates perform better than their penalized counterparts in edge classification. Averaging method with penalization is slightly better than the proposed method or direct-sample estimation with penalization. In the high-dimensional setting of $n = 50, p = 50$, the ROC curves have many fluctuations and all lines almost overlap for this really sparse network (with 49 edges).
Figure 3.6: heatmap of set-up 6. The upper panels depict the patterns of the estimated precision matrices by the proposed method when $n = 300, p = 50$. The left-hand panel depicts results for the unpenalized estimate while the right-hand panel depicts results for penalized one. The lower panel depicts the true pattern of model 6. The axes on the left and at the bottom are the names of variables.
Figure 3.7: ROC of Set-up 6. The left-hand panel is the ROC plot of all methods when $n = 300, p = 50$ while the right-hand panel depicts curves of estimates with $n = 50, p = 50$.

In this scenario, we have tested our method in many set-ups on the simulated data with 2 replications. The results show the penalized estimate is better than other estimates in most cases in terms of KL divergence. As KL divergence measures how much information we lose when we choose an approximation, there is less information lost when we choose the distribution constructed with our penalized estimate than with other estimates to approximate the true distribution. As for the ability to reconstruct the network, our method shows no advantage over averaging approach. In set-up 1 with a dense clique structure, our unpenalized estimate achieves the highest AUC and reconstructs the dense network very well. However, as we decrease the number of edges, this advantage diminishes, though the unpenalized estimate remains the best. In set-up 4, 6 with a sparse structure, our penalized estimate does well in classifying edges and the unpenalized one has the lowest AUC. But as we increase the number of edges in set-up 5, the unpenalized estimate rises to be best or second best classifier. So our method without penalization may work better for dense networks.

3.1.2 Scenario 2: 3 replications

In this scenario, we test our method on the simulated data with 3 replications in set-up 1, 4, 6 and also vary replications and sample sizes to see how AUC and mutual information change. We expect an increase in AUC as replications and sample sizes increase. As for mutual information between the representation $Y$ and the signal $M$ of the estimated distributions, it should become closer to the true level with more replications and samples.

Fig.3.8 shows how AUC changes when replications are increased from 2 to 3 in set-up 1. Overall, as we have expected, with more samples and replications, the AUC goes up, but the rank of performance is unaffected. The results are quite similar for set-up 4 and set-up 6 (see appendix E for more details).

Fig.3.9 shows how mutual information changes with samples and replications. The orange line stands for the true normalized mutual information between $Y$ and $M$. With the increasing number of samples and replications, the mutual information of all methods is approaching the true level. The
Figure 3.8: AUC of set-up 1 with 2 and 3 replications. Different colours stand for different methods. The solid lines are the results of 2 replications while the dashed lines represent the results of 3 replications.
proposed method with penalization has the mutual information closest to the true value. The averaging method, however, has very high mutual information. Similar phenomena were found in set-up 4 and 6 (see appendix D).

In conclusion, as the number of replications and sample sizes increase, the AUCs of all estimates improve and mutual information is closer to the true level. Though our method shows no advantage over averaging approach in terms of AUCs, the penalized estimate is better than other estimates regarding mutual information.

3.1.3 Scenario 3: varied replications

In the previous scenarios, we have balanced designs with equal number of replications for each sample. However, in real life, the number of replications can differ by sample. How would the proposed method behave when it is no longer balanced? We vary the number of replications from 2 to 4 by drawing unequal number of replications for samples and test this scenario for set-up 1, 4 and 6 with \( n = 300, 50, p = 50 \).

With more replications than in scenario 1, the overall classification accuracy for set-up 1, 4, 6 improves while the rank of performance remains almost the same, as indicated in fig.3.10, and 3, 3.7 in E. For set-up 1, our unpenalized estimate is still the best and individual-sample estimate, which uses only one replication in the estimation, performs worst. For set-up 4, the rank of performance slightly changes. With more replications, our unpenalized estimate is no longer the worst and it outperforms the unpenalized individual-sample estimate.

3.1.4 Scenario 4: use of a target matrix

In this scenario, we propose to use the penalized estimate by averaging method, which has been seen to have very competitive performance in the above scenarios, as a target matrix for our method. Then our penalized estimator has the desirable property of shrinking to this target matrix when the penalty term \( \lambda_M \) is very large. While \( \lambda_M \) is very small, it is close to the unpenalized equivalent. A Bayesian interpretation of using a target matrix is that we specify a prior. The penalized estimator is the compromise between the prior and the data. When a large \( \lambda_M \) is selected, the prior information weighs more in the estimation. When a small penalty term is chosen, the data dominate. We test set-up 1, 4, 6 on simulated data with 2 replications.

The ROC curves of the penalized estimator by averaging method and the penalized estimate by the proposed method overlap, which indicates the dominance of the prior information in the estimation process, as given in fig.3.11. This also matches the results of estimation without the target matrix. It is shown that in fig.3.2 in scenario 1, the penalized estimate by averaging method is better than that of the proposed method. The same phenomena happen to set-up 4 (see appendix E. For set-up 6, there are fewer edges and the plot is not very smooth. Our penalized estimate is better than that by averaging method (c.f. appendix E also).
Figure 3.9: Mutual information in set-up 1 with varying samples. Only the results of the proposed method and averaging method were compared, because for mutual information, both signal and noise parameters are required.
Figure 3.10: ROC of set-up 1 with 2-4 replications. The design is no longer balanced and the number of replications varies from 2 to 4 over samples. The left-hand panel is the ROC plot of all methods when \( n = 300, p = 50 \) while the right-hand panel depicts curves of estimates with \( n = 50, p = 50 \).

Figure 3.11: ROC of set-up 1 with a target. The proposed method uses the penalized estimate by averaging as a target matrix. Other methods have no target. The left-hand panel is the ROC plot of all methods when \( n = 300, p = 50 \) while the right-hand panel depicts curves of estimates with \( n = 50, p = 50 \).
3.1.5 Scenario 5: the effect of noise level

In this scenario the level of noise in the data is changed to see how that would affect the performance of our method. In the above scenarios we assume a unit diagonal precision matrix for the noise, which is the same size as the diagonal elements of signal precision matrix. In this scenario we instead enlarge or reduce the diagonal elements of noise precision matrix to control the noise level. We test set-up 1, 4, 6 on simulated data with 2 replications.

The rank of performance does not change when the noise is enlarged or reduced to the specified level, as indicated in fig. 3.12. When the diagonal elements of noise precision matrix in set-up 1 are halved, AUCs all improve as noise is lowered but the rank is unaffected. Our unpenalized estimate outperforms other estimates, the same as in scenario 1. Fig. 3.13 shows when the diagonal elements of noise precision matrix are doubled in set-up 1, AUCs all drop but our unpenalized estimate is still the best, the same as in the reduced noise level situation. Similar phenomena are found in set-up 4, 6.

3.1.6 Conclusion

In the simulation study, we test our method on the synthetic data and compare the results to other simpler approaches. The results show that in terms of KL divergence and mutual information, our penalized estimate is best in most cases. As for edge classification ability, our method shows no advantage over the averaging approach. In set-up 1, 2, 3, our unpenalized estimate performs best in classifying edges in non-high-dimensional settings, but with the decreasing number of edges, this advantage diminishes. In high-dimensional settings, the performance of all methods is weak with AUCs close to 0.5. In set-up 4,6, our unpenalized estimate performs worst and our penalized estimate is among the best and performs slightly weaker than the penalized estimate by averaging method.

We also vary the number of replication and samples, change the noise level in the data and use
Figure 3.13: ROC of set-up 1 with larger noise. The diagonal elements of the noise precision matrix are doubled to 2. The left-hand panel is the ROC plot of all methods when $n = 300, p = 50$ while the right-hand panel depicts curves of estimates with $n = 50, p = 50$.

a target matrix to see how those factors affect the performance of our method compared to other methods. We find that with an increase in replications and samples, the overall classification accuracy of all methods goes up. With less noise, the performance also improves, as we expected. When a target matrix is used in our estimation, the ROC curve of the penalized estimate by our method can overlap with that of the target matrix as the target matrix dominates in the estimation.

3.2 Real-world data

In real-world data analysis, we test the proposed method on the same population samples from different platforms. Those platforms all measure gene expression levels of same subjects by using different techniques, in this way we have replications. Those replications are measurements of the same samples, so similar networks would be expected. The networks are constructed per platform or by using data from both platforms to see whether different platforms will give similar results and what effect replications have on the estimation.

3.2.1 Data description

The real-world data concerns lung squamous cell carcinoma which is publicly available from the Cancer Genome Atlas (TCGA) project. Samples included in this study were diagnosed as lung squamous cell carcinoma by a pathologist at the contributing center and passed specific quality control parameters for tumor cellularity and RNA quality (Network et al., 2012).

The dataset comprises expression levels of 14504 genes from two platforms. So we have two replications for each sample. The first part of the data consists of 151 samples measured by RNA sequencing and the second part is composed of 151 samples measured by RNA micro-array. They were preprocessed separately as described in Liu et al. (2009) so that the expression levels can be modeled with
a Gaussian distribution and the data of the two parts are comparable. Then those two parts were
merged at the gene level and for common patients using the OMICSBind function from TCGA2STAT
R package. Three pathways were tested, they are p53, erbb and apoptosis, which have 62, 83 and 79
genes respectively.

The chosen pathways, p53, erbb, and apoptosis are defined by KEGG (Kanehisa & Goto, 2000). The
p53 gene is a tumor suppressor gene. Alterations in the p53 pathway are found in most human
cancers (Vogelstein et al., 2010). The erbb protein family is a family of four structurally related receptor
tyrosine kinases (Bublil & Yarden, 2007). Excessive erbb signaling is associated with the development
of a wide variety of types of solid tumors (Cho & Leahy, 2002). Apoptosis is a form of regulated cell
death. Excessive apoptosis causes atrophy, whereas an insufficient level can lead to cancer (Green,
2011).

3.2.2 Results

Since the true conditional dependencies of those pathways are mostly unknown, we explore the differ-
ences and similarities in the results by data from platform 1, 2 only or the joint data. Considering the
ill-conditionedness of our unpenalized estimate, only the penalized ones are compared. Besides, to assess
the effect of noise, an estimator \( \hat{\Omega}_{M}^{+} \) which is equal to \( (\hat{\Omega}_{M}^{-1} + \hat{\Omega}_{e}^{-1})^{-1} \) (the estimated data precision
matrix by our method), is constructed. The networks with other estimates are also constructed and the
similarities and differences between those estimates are evaluated in terms of pairwise KL divergence
and the number of common edges they have found.

In the network inference, both local FDR with 0.2 as FDR cut-off and top 100 edges with the
highest absolute values of partial correlations are used as edge selection criterion. Then we compare
the networks pair by pair and counted the number of edges commonly found in both networks.

The pairwise KL divergence between distributions constructed by using those estimates is also investi-
gated. As KL divergence involves a reference matrix and an approximating one, it is not symmetric. In
order to make the comparison easier, we use the symmetrised KL divergence also proposed by Kullback
& Leibler (1951),

\[
D_{KL}(P\|Q) + D_{KL}(Q\|P)
\]

The add-up value is a symmetric measurement and more like a distance.

The results of joint data are compared to the single-platform data and it is shown that overall, there
is more overlapping between estimates of one platform and the joint data than between estimates of
those two platforms. As table 3.4 shows, more or equal number of edges are found between the estimates
of the joint data and platform 1 or 2 than between 2 platforms themselves for all three pathways. This
phenomenon is more pronounced in table 3.5. For example, there are only 52 common edges (about 50%
of all edges) between two platforms for pathway p53, but the number of common edges found between
platform 1 and other methods using replications is 74, 76, 76 and 72. The corresponding number for
platform 2 is 70, 73, 69 and 72. The symmetrised KL divergence also shows the distance between the
two platforms is larger than that between one platform and the joint data for three pathways in most
cases, as indicated in table 3.6. For example, for p53, the symmetrised KL divergence between \( \hat{\Omega}_1 \) and
\( \hat{\Omega}_2 \) is 7.58, higher than KL divergence between other estimates and \( \hat{\Omega}_1 \) or \( \hat{\Omega}_2 \). The same is for pathway
apoptosis and erbb. The only exception is the KL divergence between $\hat{\Omega}_1$ and $\hat{\Omega}_M$ for apoptosis. One possible reason is that $\hat{\Omega}_M$ is not a simple combination of $\hat{\Omega}_1$ and $\hat{\Omega}_2$ and some samples from platform 2 could be more emphasized.

To see the effect of data noise on network inference, the common edges between $\hat{\Omega}_M + \epsilon$ and $\hat{\Omega}_M$ are explored. By top 100 edges selection criterion, both networks have 100 edges. The number of common edges is 81, 71 and 75 for pathway p53, apoptosis and erbb respectively. That means 19, 29, 25 edges out of 100 edges (about 20%, 30%, 25%) originally found in the network of $\hat{\Omega}_M$ by the proposed method are obscured by noise.

Table 3.4: The number of common edges by lfd for pathway p53, apoptosis, erbb

<table>
<thead>
<tr>
<th>pathway</th>
<th>$\hat{\Omega}_1$</th>
<th>$\hat{\Omega}_2$</th>
<th>$\hat{\Omega}_M$</th>
<th>$\hat{\Omega}_Y$</th>
<th>$\hat{\Omega}_Y^*$</th>
<th>$\hat{\Omega}_{M+\epsilon}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P53</td>
<td>$\Omega_1$</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>apoptosis</td>
<td>$\Omega_1$</td>
<td>30</td>
<td>6</td>
<td>14</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>erbb</td>
<td>$\Omega_1$</td>
<td>21</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

* $\hat{\Omega}_1$, $\hat{\Omega}_2$, $\hat{\Omega}_M$, $\hat{\Omega}_Y$, $\hat{\Omega}_Y^*$ are the estimated precision matrices for platform 1, 2, the proposed method, direct estimation and averaging method respectively. $\hat{\Omega}_{M+\epsilon}$ is equal to $(\hat{\Omega}_M^{-1} + \hat{\Omega}_Y^{-1})^{-1}$ (standardized), the estimated data precision matrix by our method. All estimates are penalized.

Table 3.5: The number of common edges by top 100 edges for pathway p53, apoptosis, erbb

<table>
<thead>
<tr>
<th>pathway</th>
<th>$\hat{\Omega}_1$</th>
<th>$\hat{\Omega}_2$</th>
<th>$\hat{\Omega}_M$</th>
<th>$\hat{\Omega}_Y$</th>
<th>$\hat{\Omega}_Y^*$</th>
<th>$\hat{\Omega}_{M+\epsilon}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P53</td>
<td>$\Omega_1$</td>
<td>100</td>
<td>52</td>
<td>74</td>
<td>76</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>52</td>
<td>100</td>
<td>70</td>
<td>73</td>
<td>69</td>
</tr>
<tr>
<td>apoptosis</td>
<td>$\Omega_1$</td>
<td>100</td>
<td>41</td>
<td>69</td>
<td>67</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>41</td>
<td>100</td>
<td>55</td>
<td>60</td>
<td>58</td>
</tr>
<tr>
<td>erbb</td>
<td>$\Omega_1$</td>
<td>100</td>
<td>40</td>
<td>66</td>
<td>68</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>40</td>
<td>100</td>
<td>59</td>
<td>62</td>
<td>60</td>
</tr>
</tbody>
</table>

Table 3.6: Symmetrized pairwise KL divergence of 6 estimates for pathway p53, apoptosis, erbb

<table>
<thead>
<tr>
<th>pathway</th>
<th>$\hat{\Omega}_1$</th>
<th>$\hat{\Omega}_2$</th>
<th>$\hat{\Omega}_M$</th>
<th>$\hat{\Omega}_Y$</th>
<th>$\hat{\Omega}_Y^*$</th>
<th>$\hat{\Omega}_{M+\epsilon}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P53</td>
<td>$\Omega_1$</td>
<td>0</td>
<td>7.58</td>
<td>4.38</td>
<td>2.03</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>7.58</td>
<td>0</td>
<td>5.87</td>
<td>1.83</td>
<td>2.63</td>
</tr>
<tr>
<td>apoptosis</td>
<td>$\Omega_1$</td>
<td>0</td>
<td>14.95</td>
<td>18.19</td>
<td>6.43</td>
<td>6.43</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>14.95</td>
<td>0</td>
<td>13.81</td>
<td>4.11</td>
<td>5.79</td>
</tr>
<tr>
<td>erbb</td>
<td>$\Omega_1$</td>
<td>0</td>
<td>16.49</td>
<td>6.83</td>
<td>4.62</td>
<td>4.62</td>
</tr>
<tr>
<td></td>
<td>$\Omega_2$</td>
<td>16.49</td>
<td>0</td>
<td>10.59</td>
<td>3.86</td>
<td>6.28</td>
</tr>
</tbody>
</table>

To evaluate the prediction ability of our estimates, we divide our data into two parts by randomly drawing 100 samples out of 151. The 100 samples are first used to construct the estimates by direct estimation (per platform), the proposed method and averaging approach (joint data). The remaining 51 samples are treated as new data and the new estimates are the penalized precision matrices by direct estimation. Those 51 samples from one platform are supposedly independent, thus the new estimates
are assumed to have a construction of \((\Omega_M^{-1} + \Omega_z^{-1})^{-1}\). KL divergence is used as the evaluation criterion and the estimates which minimize the relative entropy between the estimated distribution and new data distributions are considered the best estimates. The three pathways are all analysed in the same way, but each time different samples are used as new data.

The results of KL divergence between the estimated distribution and new data distribution are in table 3.7. It is found that estimates of new data from platform 1 and 2 are closest to each other. For example, for pathway p53, the KL divergence between the reference distribution constructed with \(\hat{\Omega}_{1,\text{new}}\) and the approximating distribution \(\hat{\Omega}_{2,\text{new}}\) is 3.45, much lower than that of other estimates. And the KL divergence between the reference distribution constructed with \(\hat{\Omega}_{2,\text{new}}\) and the approximating distribution \(\hat{\Omega}_{1,\text{new}}\) is 4.34, the lowest. This is within our expectations as the new data of platform 2 are actually replications of the new data of platform 1. They are measurements of the same samples and thus should be very similar. \(\hat{\Omega}_{M+\varepsilon,\text{train}}\) tends to be a good approximation to the new estimates. For example, for pathway p53 and apoptosis, the KL divergence of this estimate is second or third lowest, only higher than estimates of the same samples or of new data from the same platform. For pathway erbb, \(\hat{\Omega}_{M+\varepsilon,\text{train}}\) has higher KL divergence than \(\hat{\Omega}_{Y,\text{train}}\). \(\hat{\Omega}_{Y,\text{train}}\) has second lowest KL divergence. One possible reason is that the proportion of noise in this pathway is quite low and estimation by ignoring the noise is good enough.

One possible explanation is that the new data are independent samples which contain both signal and noise while the estimates by the proposed method or averaging method only concerns signal. The combined estimate \(\hat{\Omega}_{M+\varepsilon,\text{train}}\) considers both signal and noise, thus is closer to the distributions of the new data.

Table 3.7: KL divergence for pathway p53, apoptosis, erbb. The distribution of the new data is used as a reference (estimates in the column) and the distributions of the estimates as approximating ones (estimates in the row). The lower the KL divergence is, the better approximation to the distribution of new data.

<table>
<thead>
<tr>
<th>pathway</th>
<th>(\hat{\Omega}_{1,\text{train}})</th>
<th>(\hat{\Omega}_{1,\text{new}})</th>
<th>(\hat{\Omega}_{2,\text{train}})</th>
<th>(\hat{\Omega}_{2,\text{new}})</th>
<th>(\hat{\Omega}_{M,\text{train}})</th>
<th>(\hat{\Omega}_{Y,\text{train}})</th>
<th>(\hat{\Omega}_{Y,\text{train}})</th>
<th>(\hat{\Omega}_{M+\varepsilon,\text{train}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>p53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\hat{\Omega}_{1,\text{new}})</td>
<td>8.91</td>
<td>0</td>
<td>9.69</td>
<td>3.45</td>
<td>10.60</td>
<td>10.94</td>
<td>9.22</td>
<td>8.87</td>
</tr>
<tr>
<td>(\hat{\Omega}_{2,\text{new}})</td>
<td>10.54</td>
<td>4.34</td>
<td>8.51</td>
<td>0</td>
<td>11.49</td>
<td>11.42</td>
<td>9.73</td>
<td>8.67</td>
</tr>
<tr>
<td>apoptosis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\hat{\Omega}_{1,\text{new}})</td>
<td>11.28</td>
<td>0</td>
<td>18.52</td>
<td>7.19</td>
<td>32.24</td>
<td>9.36</td>
<td>18.19</td>
<td>15.55</td>
</tr>
<tr>
<td>(\hat{\Omega}_{2,\text{new}})</td>
<td>16.00</td>
<td>8.10</td>
<td>16.20</td>
<td>0</td>
<td>35.15</td>
<td>10.31</td>
<td>20.81</td>
<td>15.65</td>
</tr>
<tr>
<td>erbb</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\hat{\Omega}_{1,\text{new}})</td>
<td>19.83</td>
<td>0</td>
<td>21.13</td>
<td>8.58</td>
<td>34.34</td>
<td>23.54</td>
<td>22.96</td>
<td>17.38</td>
</tr>
<tr>
<td>(\hat{\Omega}_{2,\text{new}})</td>
<td>24.64</td>
<td>10.15</td>
<td>18.23</td>
<td>0</td>
<td>37.81</td>
<td>24.23</td>
<td>24.90</td>
<td>17.08</td>
</tr>
</tbody>
</table>

\(\hat{\Omega}_{1,\text{train}}, \hat{\Omega}_{2,\text{train}}\) are constructed with direct estimation of 100 samples from platform 1 and platform 2 respectively. \(\hat{\Omega}_{M,\text{train}}, \hat{\Omega}_{Y,\text{train}}, \hat{\Omega}_{Y,\text{train}}, \hat{\Omega}_{M+\varepsilon,\text{train}}\) are constructed with the proposed method, direct estimation and averaging approach respectively by using 100 samples from both platforms. \(\hat{\Omega}_{1,\text{new}}, \hat{\Omega}_{2,\text{new}}\) are direct estimates (penalized precision matrices) of the new data from platform 1 and 2 respectively. All estimates are penalized.
3.2.3 Conclusion

The results for those three pathways are quite similar. There is more overlapping between single-platform and the joint data than between platforms themselves in terms of the number of common edges. The symmetrised KL divergence also shows that the distance between distributions of estimates constructed from using replications, \( \hat{\Omega}_M, \hat{\Omega}_Y, \hat{\Omega}_Y, \hat{\Omega}_{M+\varepsilon} \), tend to give better approximation to the estimates of single-platform data, \( \hat{\Omega}_1, \hat{\Omega}_2 \). The proposed method, however, shows no advantage over other methods in terms of KL divergence. It may have something to do with the number of parameters our method has to estimate. The proposed method estimates much more parameters \((p(p+1)/2)\) on a high dimension than those simpler methods at the cost of accuracy. For prediction ability, \( \hat{\Omega}_{M+\varepsilon,\text{train}} \) tends to do better in approximating the estimates of the new data than the signal estimate \( \hat{\Omega}_{M,\text{train}} \) by the proposed method or \( \hat{\Omega}_{Y,\text{train}} \) by averaging method. So we think considering noise is necessary.
Chapter 4

Discussion

4.1 Summary

In this study, we propose a method to separate signal in the data from noise. The performance of this method is evaluated and compared to that of other methods.

In the simulation study, we investigate how the sample size, replication number, sparsity of the network, use of a target matrix and noise level will affect the performance of the proposed method compared to other simpler methods. The effect of those factors is explored with different set-ups in varied scenarios.

In scenario 1, the proposed method is tested on the simulated data with 2 replications in set-ups with different network structures. KL divergence shows that the proposed method with penalization gives a stable estimate and is better than other methods in most cases. However, in terms of edge classification, the proposed method shows no advantage over averaging method. In set-up 1, 2, 3, our method without penalization does better in classifying edges than other methods. However, as the number of edges decreases, the benefits of using our unpenalized estimate diminish. In set-up 4, our method without penalization performs in most cases worst. However, our penalized estimate performs as well as the penalized counterpart by averaging method. The network of set-up 4 is relatively sparser than that of set-up 1, 2, 3 and the penalized estimates of all methods are better than their unpenalized counterparts. When the number of bands is increased from 3 to 5 in set-up 5, thus a denser network, the AUC of our unpenalized estimate is no longer the lowest. In set-up 6, due to the sparsity of edges, the ROC curves are not smooth. There are many fluctuations because one mistake can cause a big change in the plot. Our unpenalized estimate gives the worst AUC, the same as in set-up 4. One possible reason is that our method is more robust with dense networks and a bit “overoptimistic” about finding edges.

We notice that in set-up 1, the unpenalized estimates of all methods outperform the penalized counterparts in terms of AUC while in set-up 4 and 6, the opposite is observed. This inconsistency may be due to the sparsity of the network. As shown in fig.19 in appendix E, after penalization, most of the partial correlations are concentrated around zero. In set-up 1 where the network is densely
connected, there are many non-zero partial correlations (393 out of 1225) in the true precision matrix. The unpenalized estimates have more spread than the penalized estimates in fig.19a. That means many elements are not too close to zero in the unpenalized estimates, which matches with the true condition. The range of penalized estimates, however, is very small. Most of the elements are very close to zero. In set-up 4 where the network is sparsely connected, the penalized estimates outperform the unpenalized ones (more obvious when n=100). This also matches the results in fig.19b. Most of the partial correlations of penalized estimates center around zero, but there are still many outliers. In this case, the unpenalized estimates also work well and the range of partial correlations is small. In set-up 6 where the network is even sparser, the penalized estimates are also better than the unpenalized estimates. The penalized estimates are less dispersed than unpenalized estimates, but there are still some partial correlations away from zero. This also matches the true condition.

Comparing the spreads of the penalized estimates in set-up 1 with those in set-up 4, it looks that partial correlations are more dispersed in set-up 4 than in set-up 1. One possible reason is that penalization overshrinks the partial correlations, even the large entries when the network is dense. With more edges, the variables may be more correlated. Thus the penalization heavily shrinks the partial correlations, perhaps shrinks too much.

In scenario 2, the relationship between AUC, mutual information and the number of replications, sample size was explored. When the number of replication is increased from 2 to 3 and sample size from 100 to 300, the AUC of all methods improves as expected. And the mutual information is found to be positively correlated with the number of replications and samples. We also notice that the distribution constructed with the penalized estimate by the proposed method has the mutual information closest to the true one.

In scenario 3, 4, 5, we look into the effects of balanced design, use of a target matrix and the noise level respectively. The results show that in terms of edge classification, the rank of the performance of all methods remains the same as in scenario 1 for set-up 1, 4 and 6.

In general, both KL divergence and mutual information show that the proposed method with penalization is better than other methods in most cases. However, in terms of edge classification, our method shows no advantage over averaging. One possible reason is that our method is estimating two parameters: the precision matrices of signal and noise while the other methods are estimating one: the precision matrix of the data or the averaged data. With the increasing dimensions, the number of elements to be estimated by the proposed method can be much larger than other methods \((p(p+1)/2)\). The large number of elements may compromise the accuracy of the estimators.

KL divergence and mutual information show that the penalized estimates by the proposed method approximated the true distribution best while the ROC curves did not support this conclusion. The discrepancy may be due to the different purposes ROC curves and KL divergence serve. ROC curves mainly aim to measure the ability of a model to detect edges while KL divergence and mutual information measure how well our estimate approximates the true distribution. For example, suppose \(\Omega_M\) is the true precision matrix, \(\Omega_\varepsilon\) the diagonal matrix with unit elements, then the undirected graph of \((\Omega_M^{-1} + \Omega_\varepsilon^{-1})^{-1}\) may be the same as that of \(\Omega_M\). However, adding a diagonal noise precision matrix may cause a bigger increase in KL divergence and mutual information than setting a non-zero element of \(\Omega_M\) to 0, by which the network would be changed. There are also studies showing that the edge
discrepancy increases but the KL divergence remains bounded above by a constant (Jog & Loh, 2015; Santhanam & Wainwright, 2012).

The mutual information between the representation $Y$ and signal $M$ in the distribution constructed with estimates given by averaging method is found to be much higher than the true level. It is may be due to the small number of replications. The averaging method takes the mean of replications over samples in order to get rid of noise, but with only two or three replications, the noise is unlikely to be averaged out. It is possible that the noise level is underestimated. The calculation of mutual information for a multivariate normal distribution involves the determinant of both signal and noise covariance matrices. The determinant of noise covariance matrix given by averaging method can be much smaller than the true one.

In the real-world data analysis, we compare the similarities and differences across platforms. The results show that the estimates given by the joint data and a single platform have more in common than estimates of platform 1 and 2 in terms of both KL divergence and number of common edges. We also randomly divide the samples into two parts: one part to construct estimates and another to test them on. The results show estimates from the same sample-population (replications) are closer than estimates from different samples in terms of KL divergence, indicating the signal is not drowned by the noise. Using the combined signal-noise estimator does a better job in approximating the distribution of the new data than using the signal estimator, which verifies the necessity of considering noise.

4.2 Future work

In the simulation study we assume a diagonal noise precision matrix, but this is a strong assumption. There may be some random connection between variables of the noise. It would be interesting to know how the proposed method would behave when the noise has some random structure, for example, uniformly distributed. So in the future research, we can experiment with different types of noise precision matrices.

For real-life datasets, we assume the same noise precision matrix for both platforms, but the proportion of noise in the data can be different across platforms. Ideker et al. (2000) shows that larger intensity measurements may have a proportionately larger error over repeated samples. They propose an “error model”, which uses different noise variance for repeated measurements of the same sample in proportion to the mean intensities. The real-world data in the study also come from two platforms using different measurement techniques. It is likely that the size of the noise is different. In the future study, different noise precision matrices can be assumed for different platforms.

ROC curves are used to measure the performance of all methods in classifying presence-or-absence of edges, but the imbalance of two classes in the data may result in an overoptimistic view of a method’s performance (Davis & Goadrich, 2006). In the ROC analysis, the sensitivity and specificity are com-
puted as,

$$TPR(sensitivity) = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

$$TNR(specificity) = \frac{TN}{FP + TN} = 1 - FPR$$

$$FNR = \frac{FN}{TP + FN} = 1 - TPR$$

when the number of negative examples is much larger than the number of positive examples, a large change in the number of false positives can lead to a small change in the false positive rate. So the false positive rate increases slowly, because the true negatives (in the denominator of FPR) would probably be very high and make this metric very small. As the false positive rate increases slowly, the specificity (1 - FPR) decreases slowly. And the number of false negatives is small (in the denominator of TPR), sensitivity tends to increase faster. Then the slope, sensitivity against specificity, would probably be large. This may explain why in set-up 2, 3, when the number of edges decreases (so the proportion of “presence” against “absence” lowers), the overall AUCs improve. In set-up 4, when the network is sparser, AUCs are even higher for all methods but the proposed method without penalization. In the future research, instead of ROC curves, an alternative measure such as Precision-Recall curves, which compares false positives to true positives rather than true negatives and thus would not be affected by a large number of negatives, should be used (Saito & Rehmsmeier, 2015).

We also find discrepancy in the results of KL divergence and ROC curves. It would be interesting to explore the relationship between KL divergence and network. We could do simulations and experiment with the number of the edges to see how KL divergence changes along with it.

Our method with penalization is computationally intensive for high-dimensional data analysis. Because our method estimates both signal and noise precision matrices, the EM algorithm is slow to converge. And in the penalized EM, cross-validation is employed to select optimal $\lambda_M, \lambda_c$. For the three pathways, it takes a few hours before the optimal penalty terms are found. In the future study, an alternative method to cross-validation may be used to reduce computation time.

Cross-validation has been shown to not consistently select models (Shao, 1993) and perform poorly with high-dimensional data, sometimes even dramatically (Meinshausen & Bühlmann, 2010). A different selection criterion can be used, such as AIC or BIC.

The proposed method does not show advantages over the averaging method in edge detection. It may be due to the much larger number of parameters the proposed method has to estimate simultaneously than averaging. It would be interesting to know how the edge classification accuracy of our method is compromised by the number of parameters.

### 4.3 Conclusion

In real-life data, there will be signal and noise, so in this study, we propose a method to separate signal from noise by utilizing replications. We find methods utilizing replications (the proposed method and averaging method) do better than methods without using replications (direct estimation and individual-
sample estimation) in terms of KL divergence. In particular, the averaging method achieves higher AUCs than direct estimation or individual-sample estimation in all scenarios we have explored. Moreover, the proposed method estimates both signal and noise precision matrices and averaging method tries to average out the noise. So we think it is necessary to consider the noise in the data. Though our method performs no better than averaging method in detecting edges, the penalized estimate can give a stable and good fit in terms of KL divergence and mutual information. So, replications do help improve the estimation. The effects of sample size and the number of replications are also explored. As we expected, with the increasing samples and replications, the KL divergence falls and the mutual information becomes closer to the true level.
References


Appendix A

Covariance matrix of \( Y_i \)

The covariance matrix \( \Sigma_i \in \mathbb{R}^{k_i \times k_i} \) is composed of \( \Sigma_M + \Sigma_\epsilon \) on the diagonal, and \( \Sigma_M \) off the diagonal. For computation’s sake, we derived the simplified versions of the determinant and the inverse (if exists) of this covariance matrix of \( Y_i \). The process is shown as follows.

A.1 Determinant

The determinant of the covariance matrix of \( Y_i \), where \( r_j \) is a row and \( c_j \) a column of the matrix, \( k_i \) is the number of replications for sample \( i \).

Thus the log version is:

\[
\log|\Sigma_i| = (k_i - 1)\log|\Sigma_\epsilon| + \log|k_i\Sigma_M + \Sigma_\epsilon| \quad \text{(A.1)}
\]
A.2 Inverse covariance matrix

The inverse of covariance matrix of $Y_i$,

$$
\begin{bmatrix}
I_{p \times p} & 0_{p \times p} & \cdots & 0_{p \times p} \\
0_{p \times p} & I_{p \times p} & \cdots & 0_{p \times p} \\
\vdots & \vdots & \ddots & \vdots \\
0_{p \times p} & 0_{p \times p} & \cdots & I_{p \times p}
\end{bmatrix} - r_j^{-1} \Sigma_i^{-1} \begin{bmatrix}
I_{p \times p} & 0_{p \times p} & \cdots & 0_{p \times p} \\
0_{p \times p} & I_{p \times p} & \cdots & 0_{p \times p} \\
\vdots & \vdots & \ddots & \vdots \\
0_{p \times p} & 0_{p \times p} & \cdots & I_{p \times p}
\end{bmatrix} r_j - r_k^{-1} \Sigma_i^{-1} \begin{bmatrix}
I_{p \times p} & 0_{p \times p} & \cdots & 0_{p \times p} \\
0_{p \times p} & I_{p \times p} & \cdots & 0_{p \times p} \\
\vdots & \vdots & \ddots & \vdots \\
0_{p \times p} & 0_{p \times p} & \cdots & I_{p \times p}
\end{bmatrix}
$$

\[ (k-1)\Sigma_i^{-1} \Sigma_M + I_{p \times p} \]
\[
\begin{array}{cccccccc}
(k_i - 1)\Sigma_e^{-1} & \Sigma_e^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} & (k_i, \Sigma_M + \Sigma_e)^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\Sigma_e^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\end{array}
\]

So the simplified inverse covariance matrix is,

\[
\begin{pmatrix}
(k_i - 1)\Sigma_e^{-1} \Sigma_M + I_p \Sigma_e^{-1} \\
-\Sigma_e^{-1} \Sigma_M(k_i, \Sigma_M + \Sigma_e)^{-1} & (k_i, \Sigma_M + \Sigma_e)^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} \Sigma_M(k_i, \Sigma_M + \Sigma_e)^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-\Sigma_e^{-1} \Sigma_M(k_i, \Sigma_M + \Sigma_e)^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} \Sigma_M(k_i, \Sigma_M + \Sigma_e)^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} \Sigma_M(k_i, \Sigma_M + \Sigma_e)^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
-\Sigma_e^{-1} \Sigma_M(k_i, \Sigma_M + \Sigma_e)^{-1} & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\end{pmatrix}
\]

where \(j = 1, ..., k_i - 1\), the diagonal element is \([(k_i - 1)\Sigma_e^{-1} \Sigma_M I_p \Sigma_e^{-1}](k_i, \Sigma_M + \Sigma_e)^{-1}\) and the off-diagonal element is \(-\Sigma_e^{-1} \Sigma_M(k_i, \Sigma_M + \Sigma_e)^{-1}\).

Note:

\[
[(k_i - 1)\Sigma_e^{-1} \Sigma_M I_p \Sigma_e^{-1}](k_i, \Sigma_M + \Sigma_e)^{-1} - I_p \Sigma_p = [(k_i - 1)\Sigma_e^{-1} \Sigma_M I_p \Sigma_e^{-1}](k_i, \Sigma_M + \Sigma_e)^{-1} - \Sigma_e^{-1} (k_i, \Sigma_M + \Sigma_e)(k_i, \Sigma_M + \Sigma_e)^{-1} \Sigma_e
\]

\[
= [(k_i - 1)\Sigma_e^{-1} \Sigma_M I_p \Sigma_e^{-1} - \Sigma_e^{-1} \Sigma_M I_p \Sigma_e^{-1}](k_i, \Sigma_M + \Sigma_e)^{-1} \Sigma_e
\]

\[
= (\Sigma_e^{-1} \Sigma_M)(k_i, \Sigma_M + \Sigma_e)^{-1} \Sigma_e
\]
Appendix B

EM algorithm

B.1 Complete-data likelihood

We used EM algorithm in our study because the likelihood function of data $Y$ is intractable. We can solve this problem by assuming the existence of latent variables. The log likelihood for the complete data is:

$$
\log(f(Y, M)) = \frac{1}{2} \left[ -k \log(2\pi) - k \log(|\Sigma|) - \sum_{i=1}^{k} \sum_{j=1}^{h_i} (y_{ij} - M_i \Sigma^{-1} M_i \Sigma^{-1}) \right] - n \log(2\pi) - n \log(|\Sigma|) - \sum_{i=1}^{n} (M_i \Sigma^{-1} M_i) 
$$

(B.1)

Then, since the latent variables are unknown to us, we calculated the expected value of the complete-
data likelihood of $Y$ and $M$ given the observed data $Y$, which is given as follows,

$$E_{M|Y}(\log(f(Y, M))) = \frac{1}{2}[-k\log(2\pi) - \log(\Omega^{-1}_\varepsilon)] - \sum_{i=1}^{n} \sum_{j=1}^{k_i} \text{tr}(\Omega_{\varepsilon} E((y_{ij} - M_i)(y_{ij} - M_i)^t))$$

$$- np\log(2\pi) - n\log(|\Omega_M^{-1}|) - \sum_{i=1}^{n} \text{tr}(\Omega_M E(M_i M_i^t))$$

$$= \frac{1}{2}[-k\log(2\pi) - \log(\Omega^{-1}_\varepsilon)] - \sum_{i=1}^{n} \sum_{j=1}^{k_i} \text{tr}(\Omega_{\varepsilon} (\text{Cov}(y_{ij} - M_i))$$

$$+ E(y_{ij} - M_i)E(y_{ij} - M_i)^t)$$

$$- np\log(2\pi) - n\log(|\Omega_M^{-1}|) - \sum_{i=1}^{n} \text{tr}(\Omega_M (\text{Cov}(M_i) + E(M_i)E(M_i)^t))$$

$$= \frac{1}{2}[-k\log(2\pi) - \log(\Omega^{-1}_\varepsilon)] - \sum_{i=1}^{n} \sum_{j=1}^{k_i} \text{tr}(\Omega_{\varepsilon} (\text{Cov}(M_i))$$

$$+ (y_{ij} - E(M_i))(y_{ij} - E(M_i))^t)$$

$$- np\log(2\pi) - n\log(|\Omega_M^{-1}|) - \sum_{i=1}^{n} \text{tr}(\Omega_M (\text{Cov}(M_i) + E(M_i)E(M_i)^t))$$

(B.2)

Note: by moments estimation:

$$E((M_i|Y_i)(M_i|Y_i)^t) = \text{Cov}(M_i|Y_i) + E(M_i|Y_i)E((M_i|Y_i)^t)$$

As the expectation of the complete-data likelihood is calculated with respect to $M$ conditioned on $Y$, $Y$ is constant here. We have,

$$\text{Cov}(y_{ij} - M_i) = \text{Cov}(M_i)$$

### B.2 EM algorithm

**Parameters to estimate: $\Sigma_M, \Sigma_\varepsilon$**

**E step:**

for each sample $i$:

The variance and covariance of replications are $\Sigma_M + \Sigma_\varepsilon$ and $\Sigma_M$ respectively, so the distribution of sample $i$ is:

$$\begin{bmatrix} y_{i1} \\ \vdots \\ y_{ik_i} \end{bmatrix} \sim \mathcal{N}
\left( \begin{bmatrix} 0_p \\ \vdots \\ 0_p \end{bmatrix}, \begin{bmatrix} \Omega^{-1}_M + \Omega^{-1}_\varepsilon & \ldots & \Omega^{-1}_M \\ \ldots & \ldots & \ldots \\ \Omega^{-1}_M & \ldots & \Omega^{-1}_M + \Omega^{-1}_\varepsilon \end{bmatrix} \right)$$

$$Y_i \sim N(0_{k_i p}, \Sigma_i)$$
\[ \Sigma_i := \begin{bmatrix} \Omega_M^{-1} + \Omega_\epsilon^{-1} & \cdots & \Omega_M^{-1} \\ \cdots & \cdots & \cdots \\ \Omega_M^{-1} & \cdots & \Omega_M^{-1} + \Omega_\epsilon^{-1} \end{bmatrix} \in \mathbb{R}^{k_i \times k_i p} \]

Then the joint distribution of \( Y_i \) and \( M_i \) is:

\[
\begin{bmatrix} Y_i \\ M_i \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} 0_{k_i P} \\ 0_p \end{bmatrix}, \begin{bmatrix} \Sigma_i & \Sigma_i \\ \Sigma_i & \Sigma_M \end{bmatrix} \right) \]

\( M_i | Y_i \sim \mathcal{N}[\mu_{M_i|Y_i}, \Sigma_{M_i|Y_i}] \)

By Theorem 6.5 of Bickel & Doksum (2015), we have:

\[
\mu_{M_i|Y_i} = \mu_{M_i} + \Sigma_{M,Y_i} \Sigma_{Y_i}^{-1} (Y_i - \mu_{M_i}) = \Sigma_M (k_i \Sigma_M + \Sigma_\epsilon)^{-1} (y_{i1} + \ldots + y_{ik_i}) \tag{B.3}
\]

\[
\Sigma_{M_i|Y_i} = \Sigma_M - \Sigma_{M,Y_i} \Sigma_{Y_i}^{-1} \Sigma_{Y,M_i} = \Sigma_M - \Sigma_M (\Sigma_i)^{-1} \begin{bmatrix} \Sigma_M \\ \vdots \\ \Sigma_M \end{bmatrix} \tag{B.4}
\]

**M step:**

in M step, \( M_i, i = 1, \ldots, n \) is fixed. We find the parameters that maximize the expectation of the complete-data likelihood given in equation (B.2).

\[
\Sigma_M, \Sigma_\epsilon = \arg \max_{\Sigma_M, \Sigma_\epsilon} l(Y, M)
\]

Compute the derivative of \( \Sigma_\epsilon^{-1} \)

\[
\frac{\partial E_{M|Y}(l(Y, M))}{\partial \Sigma_\epsilon^{-1}} = k \Sigma_\epsilon - \sum_{i=1}^n \sum_{j=1}^{k_i} [\text{Cov}(M_i) + (y_{ij} - E(M_i))(y_{ij} - E(M_i))^\top] \tag{B.5}
\]

Set (B.5) to 0.

\[
\Sigma_\epsilon^{t+1} = \frac{\sum_{i=1}^n \sum_{j=1}^{k_i} [\text{Cov}(M_i) + (y_{ij} - E(M_i))(y_{ij} - E(M_i))^\top]}{k} \tag{B.6}
\]
Compute the derivative of $\Sigma_M^{-1}$

$$\frac{\partial E_M[Y|\{Y, M\}]}{\partial \Sigma_M^{-1}} = \frac{n\Sigma_M - \sum_i^n[Cov(M_i) + E(M_i)E(M_i)']}n$$  \hspace{1cm} (B.7)

Set (B.7) to 0.

$$\Sigma_M^{t+1} = \frac{\sum_i^n[Cov(M_i) + E(M_i)E(M_i)']}n$$  \hspace{1cm} (B.8)

In our study, the observed data is $Y$ and the latent variable is $M$. We are trying to estimate the parameters of a model for $Y$ and $M$ by maximizing the marginal likelihood of $Y$, but this problem cannot be solved directly. So, instead, we maximized the expectation of the joint likelihood of $Y$ and $M$. By EM algorithm, the marginal likelihood of data $Y$ is non-decreasing in each iteration. Little & Rubin (2014) gave the proof of this property.

We first introduce Jensen’s inequality which will be used in the proof later. Jensen’ inequality states: if $X$ is a random variable and $\varphi$ is a concave function, then

$$\varphi(E(X)) \geq E(\varphi(X))$$  \hspace{1cm} (B.9)

$ln(x)$ is strictly concave on $(0, \infty)$, so in this case,

$$ln(E(X)) \geq E(ln(X))$$  \hspace{1cm} (B.10)

The marginal likelihood of data $Y$ can be written as,

$$f(Y|\theta) = \int_M f(Y|M, \theta)f(M|\theta)$$  \hspace{1cm} (B.11)

$$\log f(Y|\theta) = \log \int_M f(Y|M, \theta)f(M|\theta)$$  \hspace{1cm} (B.12)

subtract $\log f(Y|\theta^{(t)})$ from the above equation, where $\theta^{(t)}$ is the estimates from the previous iteration.

$$\log f(Y|\theta) - \log f(Y|\theta^{(t)}) = \log \int_M f(Y|M, \theta)f(M|\theta) - \log f(Y|\theta^{(t)})$$

$$= \log \int_M f(Y|M, \theta)f(M|\theta) \cdot \frac{f(M|M, \theta^{(t)})}{f(M|Y, \theta^{(t)})} - \log f(Y|\theta^{(t)})$$

$$= \log \int_M f(M|Y, \theta^{(t)}) \cdot \frac{f(Y|M, \theta)f(M|\theta)}{f(M|Y, \theta^{(t)})} - \log \{f(Y|\theta^{(t)})\}$$

$$\geq \int_M f(M|Y, \theta^{(t)}) \log \left( \frac{f(Y|M, \theta)f(M|\theta)}{f(M|Y, \theta^{(t)})} \right) - \int_M f(M|Y, \theta^{(t)}) \log f(Y|\theta^{(t)})$$

$$= \int_M f(M|Y, \theta^{(t)}) \log \left( \frac{f(Y|M, \theta)f(M|\theta)}{f(M|Y, \theta^{(t)})} \right) - \int_M f(M|Y, \theta^{(t)}) \log f(Y|\theta^{(t)})$$

(B.13)

because we are integrating over $M$, then $\log f(Y|\theta^{(t)})$ is a constant, and as a probability density function,
the following formula will always stand.

\[
\int_M f(M|Y, \theta^{(t)}) = 1
\]  

(B.14)

Equation (B.13) can be rewritten as,

\[
\log f(Y|\theta) - \log f(Y|\theta^{(t)}) = \int_M f(M|Y, \theta^{(t)}) \log \left( \frac{f(Y|M, \theta)}{f(Y|M, \theta^{(t)})} \right) - \int_M f(M|Y, \theta^{(t)}) \log \left( \frac{f(Y|M, \theta^{(t)})}{f(Y|M, \theta)} \right)
\]

(B.15)

\[
= Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)})
\]

because estimated \( \theta \) should maximize \( Q(\theta|\theta^{(t)}) \), that is,

\[
\theta^{(t+1)} = \arg\max_{\theta} Q(\theta|\theta^{(t)})
\]  

(B.16)

so \( Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)}) \geq 0 \), therefore,

\[
\log f(Y|\theta) - \log f(Y|\theta^{(t)}) \geq 0
\]  

(B.17)

By this way, the marginal likelihood of \( Y \) is non-decreasing. About the convergence property of this algorithm, McLachlan & Krishnan (2007) has discussed it in detail. They showed that it is possible for this algorithm to converge to a local minimum of the cost function or saddle points in unusual cases (McLachlan & Krishnan, 2007).

### B.3 EM algorithm with penalization

When \( n > p \), the covariance matrix can be semi-definite positive. To ensure the precision matrix exists, penalization is required. The penalization method we used is ridge penalty proposed by Van Wieringen & Peeters (2016). This penalty maximizes,

\[
\ln |\Omega| - \text{tr}(\Sigma\Omega) - \frac{\lambda_n}{2} \text{tr} \left[ (\Omega - T)^\dagger(\Omega - T) \right]
\]

(B.18)

In our study, two precision matrices are estimated. So we have two penalty terms for the signal and noise respectively. The penalized log-likelihood we should maximize is as follows:
Compute the derivative of \( \Omega \):

\[
I^p(Y, M, \Omega, \lambda) \propto \text{log}|\Omega| - \sum_{i=1}^{k} \sum_{j=1}^{k_i} (y_{ij} - M_i)^t \Omega(y_{ij} - M_i) - \frac{\lambda \epsilon}{2} \text{tr} ((\Omega - T_\epsilon)^t(\Omega - T_\epsilon))
\]

\[+ \text{nlog}|\Omega_M| - \sum_{i=1}^{n} M_i^t \Omega M_i - \frac{\lambda_M}{2} \text{tr} ((\Omega_M - T_M)^t(\Omega_M - T_M))
\]

\[= \text{log}|\Omega| - \sum_{i=1}^{n} \sum_{j=1}^{k_i} \text{tr}(\Omega(y_{ij} - M_i)(y_{ij} - M_i)^t) - \frac{\lambda \epsilon}{2} \text{tr} ((\Omega - T_\epsilon)^t(\Omega - T_\epsilon))
\]

\[+ \text{nlog}|\Omega_M| - \sum_{i=1}^{n} \text{tr}(\Omega_M(M_i M_i^t)) - \frac{\lambda_M}{2} \text{tr} ((\Omega_M - T_M)^t(\Omega_M - T_M))
\]

(B.19)

For the penalized EM, the \textbf{E step} remains unchanged and the \textbf{M step} goes as follows:

\[E_{M|Y}(I^p(Y, M, \Omega, \lambda)) \propto \text{log}|\Omega| - \sum_{i=1}^{n} \sum_{j=1}^{k_i} \text{tr}(\Omega^{-1}(\text{Cov}(M_i)) + (y_{ij} - E(M_i))(y_{ij} - E(M_i))^t)
\]

\[+ \frac{\lambda \epsilon}{2} \text{tr} ((\Omega - T_\epsilon)^t(\Omega - T_\epsilon))
\]

\[= n\text{log}|\Omega_M| - \sum_{i=1}^{n} \text{tr}(\Sigma^{-1} \text{Cov}(M_i) + E(M_i)E(M_i)^t) - \frac{\lambda_M}{2} \text{tr} ((\Omega_M - T_M)^t(\Omega_M - T_M))
\]

(B.20)

Compute the derivative of \( \Omega \):

\[
\frac{\partial E(I^p)}{\partial \Omega} = k\Omega^{-1} - kS_\epsilon - \lambda \epsilon (\Omega - T_\epsilon)
\]

(B.21)

where \( S_\epsilon = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k_i} (\text{Cov}(M_i) + (y_{ij} - E(M_i))(y_{ij} - E(M_i))^t)}{k} \).

Set (B.21) to 0.

\[
\Omega^{-1} - S_\epsilon - \lambda \epsilon (\Omega - T_\epsilon) = 0
\]

\[
\Omega^{-2} - \frac{S_\epsilon}{2} \Omega^{-1} - \lambda \epsilon I_p + \lambda \epsilon T_\epsilon \Omega^{-1} = 0
\]

\[
\Omega^{-2} - \frac{(S_\epsilon - \lambda \epsilon T_\epsilon)}{2} \Omega^{-1} = \frac{(S_\epsilon - \lambda \epsilon T_\epsilon)^2}{2} + \lambda \epsilon I_p
\]

\[
[S_\epsilon - \lambda \epsilon T_\epsilon] = \frac{(S_\epsilon - \lambda \epsilon T_\epsilon)^2}{2} + \lambda \epsilon I_p
\]

\[
\Omega^{-1} = \frac{(S_\epsilon - \lambda \epsilon T_\epsilon)^2}{4} + \frac{1}{2} (S_\epsilon - \lambda \epsilon T_\epsilon)
\]

(B.22)

Compute the derivative of \( \Omega_M \)

\[
\frac{\partial I^p}{\partial \Omega_M} = n\Omega^{-1} - nS_M - \lambda_M (\Omega_M - \lambda \epsilon T_M)
\]

(B.23)

where \( S_M = \frac{\sum_{i=1}^{n} (\text{Cov}(M_i) + E(M_i)E(M_i)^t)}{n} \).

66
Set (B.23) to 0.

\[ \Omega^{-1}_M = \left(\frac{(S_M - \lambda_M T_M)^2}{4} + \lambda_M I_p\right)^{1/2} + \frac{1}{2} (S_M - \lambda_M T_M) \]  
(B.24)

The penalized EM algorithm is also non-decreasing.

\[
\log f(Y|\theta) - \log f(Y|\theta^{(t)}) \\
= \log \int_M f(Y|M, \theta) \, p(M|\theta) \, f(M|\theta) \, \frac{f(Y|M, \theta^{(t)})}{f(Y|M, \theta)} \, \frac{f(M|Y, \theta^{(t)})}{f(M|Y, \theta)} \, \log f(Y|\theta) \, dM \\
- \log f(Y|\theta^{(t)}) \\
= \log \int_M f(Y|M, \theta) \, f(M|\theta) \, \frac{f(Y|M, \theta^{(t)})}{f(Y|M, \theta)} \, \frac{f(M|Y, \theta^{(t)})}{f(M|Y, \theta)} \, \log f(Y|\theta) \, dM \\
- \log f(Y|\theta^{(t)}) \\
= \log \int_M f(Y|M, \theta) \, f(M|\theta) \, \frac{f(Y|M, \theta^{(t)})}{f(Y|M, \theta)} \, \frac{f(M|Y, \theta^{(t)})}{f(M|Y, \theta)} \, \log f(Y|\theta) \, dM \\
- \log f(Y|\theta^{(t)}) \\
\geq \int_M f(Y|M, \theta) \, f(M|\theta) \, \frac{f(Y|M, \theta^{(t)})}{f(Y|M, \theta)} \, \frac{f(M|Y, \theta^{(t)})}{f(M|Y, \theta)} \, \log f(Y|\theta) \, dM \\
- \log f(Y|\theta^{(t)}) \\
= \int_M f(Y|M, \theta) \, f(M|\theta) \, \frac{f(Y|M, \theta^{(t)})}{f(Y|M, \theta)} \, \frac{f(M|Y, \theta^{(t)})}{f(M|Y, \theta)} \, \log f(Y|\theta) \, dM \\
- \log f(Y|\theta^{(t)}) \\
= Q(\theta^{(t)}, \lambda_M, \lambda_c) - Q(\theta^{(t)}|\theta^{(t)}, \lambda_M, \lambda_c) \geq 0
\]
Appendix C

Kullback-Leibler divergence

KL divergence measures the difference between one probability distribution and a reference probability distribution. In this study, the divergence between the probability distributions with estimated parameters and the reference probability distribution of the signal will be measured and compared as an evaluation of performance. All the distributions involved are multivariate normal. Duchi (2007) gave the derivation for KL divergence between two multivariate normal distributions.

\[
D(P_1 || P_2) = E_{P_1} \left[ \log P_1 - \log P_2 \right]
\]

\[
= \frac{1}{2} E_{P_1} \left[ -\log |\Sigma_1| - (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) + \log |\Sigma_2| + (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \right]
\]

\[
= \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} + \frac{1}{2} E_{P_1} \left[ - (x - \mu_1)^T \Sigma_1^{-1} (x - \mu_1) + (x - \mu_2)^T \Sigma_2^{-1} (x - \mu_2) \right]
\]

\[
= \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} + \frac{1}{2} E_{P_1} \left[ - \text{tr} \left( \Sigma_1^{-1} (x - \mu_1) (x - \mu_1)^T \right) + \text{tr} \left( \Sigma_2^{-1} (x - \mu_2) (x - \mu_2)^T \right) \right]
\]

\[
= \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} + \frac{1}{2} E_{P_1} \left[ - \text{tr} \left( \Sigma_1^{-1} \Sigma_1 \right) + \text{tr} \left( \Sigma_2^{-1} (x x^T - 2x \mu_2^T - \mu_2 \mu_2^T) \right) \right]
\]

\[
= \frac{1}{2} \log \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2} \log \left( \Sigma_1 + \mu_1 \mu_1^T - 2\mu_2 \mu_1^T + \mu_2 \mu_2^T \right)
\]

\[
= \frac{1}{2} \left( \log \frac{|\Sigma_2|}{|\Sigma_1|} - \log \left( \Sigma_2^{-1} \Sigma_1 \right) + \text{tr} \left( \mu_1^T \Sigma_2^{-1} \mu_1 - 2 \mu_1^T \Sigma_2^{-1} \mu_2 + \mu_2^T \Sigma_2^{-1} \mu_2 \right) \right)
\]

\[
= \frac{1}{2} \left( \log \frac{|\Sigma_2|}{|\Sigma_1|} - \log \left( \Sigma_2^{-1} \Sigma_1 \right) + (\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1) \right)
\]

where \(P_1\) is the reference probability distribution, and \(P_2\) is the approximating probability distribution.

KL divergence is not symmetric. It is obvious to see that,

\[
D(P_1 || P_2) \neq D(P_2 || P_1)
\]
and KL divergence is non-negative, by Gibbs’ inequality,

\[
D(P_1||P_2) = \int P_1 \log \frac{P_1}{P_2}
\]

\[
= - \int P_1 \log \frac{P_2}{P_1}
\]

\[
\geq - \int P_1 \left( \frac{P_2}{P_1} - 1 \right) \quad \triangleright \text{for } x \text{ greater than zero, } \log x \leq x - 1
\]

\[
= - \left( \int P_2 - \int P_1 \right)
\]

\[
= 1 - 1
\]

\[
= 0
\]
Appendix D

Mutual information

Mutual information measures the remaining information of one variable given the other. In our case, $Y$, which contains a certain degree of noise, is the representation of signal $M$. Then what is the level of mutual information between $Y$ and $M$ in the distributions using those estimates compared to the true one? Here we give the derivation of the mutual information between $Y$ and $M$ in a multivariate normal distribution adapted to our case.

\[
I(Y; M) = \int_{m \in M} \int_{y \in Y} f(y, m) \log \frac{f(y, m)}{f(y)f(m)} \, dy \, dm
\]
\[
= \int_{m \in M} \int_{y \in Y} f(y, m) \log \frac{f(y, m)}{f(m)} \, dy \, dm - \int_{m \in M} \int_{y \in Y} f(y, m) \log f(m) \, dy \, dm
\]
\[
= \int_{m \in M} f(m) \int_{y \in Y} f(y|M = m) \log f(y|M = m) \, dy \, dm - \int_{y \in Y} \log f(y) \left( \int_{m \in M} f(y, m) \, dm \right) \, dy
\]
\[
= - \int_{m \in M} f(m) H(Y|M = m) \, dm + H(Y)
\]
\[
= H(Y) - H(Y|M)
\]
\[
= 0.5 \times (E_Y(\log |\Sigma_y|) + y^t \Sigma_y^{-1} y + E_m(E_Y|M=m)(-\log |\Sigma_{\epsilon}| - (y - m)^t \Sigma_{\epsilon}^{-1}(y - m)))
\]
\[
= 0.5 \times ((\log |\Sigma_y| + p) + (\log |\Sigma_{\epsilon}| - p)
\]
\[
= 0.5 \times (\log |\Sigma_y| - \log |\Sigma_{\epsilon}|)
\]

By our assumption, $\Sigma_y = \Sigma_M + \Sigma_{\epsilon}$, so the above formula can be rewritten as,

\[
0.5 \times (\log |\Sigma_y| - \log |\Sigma_{\epsilon}|) = 0.5 \times (\log |\Sigma_M + \Sigma_{\epsilon}| - \log |\Sigma_{\epsilon}|)
\]

To have a more straightforward interpretation of the mutual information, we normalized the value
to make it between 0 and 1.

\[
\frac{I(Y; M)}{\sqrt{H(Y)H(M)}} = \frac{\int_{m \in M} \int_{y \in Y} f(y, m) \log \frac{f(y|m)}{f(y)} dy dm}{\sqrt{\left(\int_{y \in Y} f(y) \log f(y) dy \right)^2 - \int_{m \in M} f(m) \log f(m) dm}}
\]

\[-\int_{y \in Y} f(y) \log f(y) dy = -E_y(\log f(y))
\]

\[= -0.5 \times E_y(-p \log 2\pi - \log |\Sigma_y| - y^T \Sigma_y^{-1} y)\]

\[= 0.5 \times (p \log 2\pi + \log |\Sigma_y| + p)\]

\[= 0.5 \times (p \log 2\pi + \log |\Sigma_M + \Sigma_\varepsilon| + p)\]

\[-\int_{m \in M} f(m) \log f(m) dm = -E_m(\log f(m))
\]

\[= -0.5 \times E_m(-p \log 2\pi - \log |\Sigma_M| - m^T \Sigma_M^{-1} m)\]

\[= 0.5 \times (p \log 2\pi + \log |\Sigma_M| + p)\]
Appendix E

ROC and AUC plots

(a) ROC of set-up 2, n=300, p=50  
(b) ROC of set-up 2, n=50, p=50

Figure 1: ROC of set-up 2
Figure 2: ROC of set-up 3

Figure 3: ROC of set-up 4 with 2-4 replications
Figure 4: ROC of set-up 6 with 2-4 repetitions

(a) ROC of set-up 6, n=300, p=50
(b) ROC of set-up 6, n=50, p=50

Figure 5: ROC of set-up 4 with a target

(a) ROC of set-up 4, n=300, p=50
(b) ROC of set-up 4, n=50, p=50
Figure 6: ROC of set-up 6 with a target
Figure 7: AUC of set-up 4 with 2 and 3 repetitions
Figure 9: Mutual information of set-up 4 with varying samples
Figure 11: AUC of set-up 6 with 2 and 3 replications
Figure 13: Mutual information of set-up 6 with varying samples
Figure 15: ROC of set-up 4, error=0.5

Figure 16: ROC of set-up 4, error=2
Figure 17: ROC of set-up 6, error=0.5

(a) ROC of set-up 6, n=300, p=50

(b) ROC of set-up 6, n=50, p=50

Figure 18: ROC of set-up 6, error=2

(a) ROC of set-up 6, n=300, p=50

(b) ROC of set-up 6, n=50, p=50
Figure 19: partial correlations of set-up 1, 4, 6 in scenario 1 when n=300, p=50. “Signal”, “Signal penalized” stand for unpenalized and penalized partial correlations respectively by the proposed method, “Y” and “Y penalized” stand for unpenalized and penalized estimates by direct estimation, “averaged Y”, ” averaged Y penalized” are estimate by averaging approach, “Individual” and “individual penalized” are estimates by individual-sample estimation.