COMPLEX MODELLING OF MULTI-OUTCOME DATA WITH APPLICATIONS TO CANCER BIOLOGY

A THESIS SUBMITTED TO

THE UNIVERSITY OF KENT AT CANTERBURY

IN THE SUBJECT OF STATISTICS

FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY BY RESEARCH.

By

Elaheh Oftadeh

December 2017

To my beloved parents

Acknowledgments

I would like to give a special thanks to my supervisor, Prof. Jian Zhang, for all the valuable advice and constructive suggestions that he has offered throughout my Ph.D. studies. I profited immensely from his knowledge, guidance and patience, without which this work could not have been completed. I would also like to thank my second supervisor Dr. Cristiano Villa for his kind encouragements.

I am also grateful to the excellent and supportive lecturers in the statistics group in SMSAS from whom I learned so much during my studies.

Many thanks to Claire Carter who was always so supportive and helpful.

Last but not least, I would like to give a special thanks to my parents, to whom this thesis is dedicated, for their unconditional love, support and encouragements along the way.

Abstract

In applied scientific areas such as economics, finance, biology, and medicine, it is often required to find the relationship between a set of independent variables (predictors) and a set of response variables (i.e., outcomes of an experiment). If we model individual outcomes separately, we potentially miss information of the correlation among outcomes. Therefore, it is desirable to model these outcomes simultaneously by multivariate linear regressions. With the advent of high-throughput technology, there is an enormous amount of high dimensional multivariate regression data being generated at an extraordinary speed. However, only a small proportion of them are informative. This has imposed a challenge on modern statistics because of this high dimensionality. In this work, we propose methods and algorithms for modelling high-dimensional multivariate regression data. The contributions of this thesis are as follows.

Firstly, we propose two variable screening techniques to reduce the high dimension of predictors. One is a beamforming-based screening method which is based on a statistic called SNR. The second approach is a mixture-based screening where the screening is conducted through the so-called likelihood fusion.

Secondly, we propose a variable selection method called principal variable analysis (PVA). In PVA we take into account the correlation between response variables in the process of variable selection. We compare PVA with some of well-known variable selection methods by simulation studies, showing that PVA can substantially enhance the selection accuracy.

Thirdly, we develop a method for clustering and variable selection simultaneously, by using the likelihood fusion. We show the feature of the proposed method by simulation studies.

Fourthly, we study a Bayesian clustering problem through the mixture of normal distributions where we propose mixing-proportion dependent priors for component parameters.

Finally, we apply the proposed methods to cancer drug data. This data contain expression levels of 13321 genes across 42 cell lines and the responses of these cell lines to 131 drugs, recorded as fifty percent inhibitory concentration (IC50) values. We identify 37 genes which are important for predicting IC50 values. We found that although the expressions of these genes are weakly correlated, they are highly correlated in terms of their regression coefficients. We also identify a regression coefficient-based network between genes. We also show that 34 out of 37 selected genes have played certain roles in at least one type of cancer. Moreover, by applying the likelihood fusion model to real data we classify the drugs into five groups.

Contents

List of Figures

List of Tables

Chapter 1

Motivation and Contribution

1.1 Biological background and motivation

The research carried out in this thesis is motivated by a cancer drug study. In the following, we first explain some of the concepts in cancer biology. Then, we raise some issues that one may face in statistical modelling of cancer drug data.

1.1.1 The nature of cancer

Cancer occurs when a cell grows uncontrollably as a result of mutation or changes in gene expressions. *Gene expression* is a process through which the coded information in a cell is transferred to protein (Lee, 2007). Since cancer is mainly caused by changes in genes, it is genetically unique. As a result, even the same type of cancer may still be different across individuals. Also, the response of patients with the same type of cancer to a treatment may vary. So there is no single treatment for cancer (Almeida and Barry, 2011). The diversity in cancer disease and vast variety in responses of patients to different drugs have turned the personalised treatment to a revolutionary matter in modern oncology. The crucial task in personalised treatment is matching up the right drug with the right patient (Yang et al., 2013).

The recent numerous research verifies that there is a strong link between the mutations in genomic features of the cancer cell lines and the responses to the treatment (Yang et al., 2013). Mostly, the sensitivity or resistance of a cancer cell to a drug depends on multiple genomic features of each cell line. However, identifying mutations involved in drug resistance is still challenging. As a result, curing cancer is complex (Almeida and Barry, 2011).

The aim of advanced oncology is to develop therapies, such as targeted therapies, to fight cancer cells with more precision. Targeted therapy is one of the most effective therapies over the past ten years. Targeted cancer therapies are drugs designed to interfere with specific molecules necessary for tumor growth and progression. Since through such treatments, cancer causing genes are attacked more specifically, the amount of damage that may affect normal cells is minimised considerably (Kidd et al., 2015). Identifying these cancer causing genes has become the foundation of targeted therapies.

1.1.2 Biomarker network identification

In order to determine which patients should or should not receive a certain treatment, biomarkers must be tested and validated in clinical studies. A biomarker is a biological process that we can measure to quantify the body's response to a particular medication. Biomarkers are critical components in cancer treatment by providing information about the type of medicine patients should receive (Vogel et al., 2010). Expressions of mutated genes are potential biomarkers that measure the effectiveness of a given treatment. Genes usually function in concert rather than alone. Therefore, gene expression profiles are helpful tools in discovering the functional cooperativity between genes (Yang et al., 2013), (Garnett et al., 2012).

An interaction network is a more precise way to represent the information about genes and how closely they are connected (Bebek, 2012). Knowing the interacting components can help with identifying molecular targets for specific drugs. Such knowledge coupled with understanding the network behaviour can lead to designing controlled systems with potential for producing disease-specific cures and personalised care solutions (Kidd et al., 2015).

1.2 Cancer drug data

The cancer drug data considered in this thesis consist of two parts. The first part is log-expression levels of 13321 genes across 42 cell lines and the second part contains the responses of cell lines to different drugs, recorded as fifty percent inhibitory concentration (IC50) values. According to encyclopaedia of cancer (Schwab, 2008), IC50 is a concentration of drug that reduces a biochemical activity such as cell multiplication to 50 percent of its normal value in the absence of the inhibitor.

The data were extracted in 2013 from the website: Genomics of Drug Sensitivity in Cancer (http://www.cancerrxgene.org). These data first studied by Garnett et al. (2012). In their work, in order to identify the genomic features associated with drug response, two analytical approaches are considered: A multivariate analysis of variance (MANOVA) and elastic net. They regard gene expressions, mutation status, tissue type and copy number as features and used elastic net modelling to find out how these features are linked to drug responses. In their modelling they consider each drug separately and perform the analysis in a univariate multiple regression framework.

Cancer drugs exert their function through binding to one or more protein targets. Early "one gene, one drug, one cancer" paradigm considers the role of individual genes and their changes in drug-perturbed states, which largely ignore a target's cellular and physiological context (Wang et al., 2014). Moreover, many recent researches verify that drug combination therapy is more effective in killing tumor cells and the drug resistance is reduced dramatically when drugs are used simultaneously (Carter et al., 2016).

Motivated by these facts, we use a multivariate multiple linear regression model to fit the cancer drug data, where we consider all drugs simultaneously.

Here, multivariate refers to the cases that we have several response variables in the model. In the data, log-expression levels of 13321 genes across 42 cell lines form the design matrix $\mathbf{X}_{42\times 13321}$ where each column vector $\mathbf{x}_k$ is regarded as observations on the kth predictor. Also the IC50 values of 131 drugs across 42 cell lines form observations on 131 response variables in the form of matrix $\mathbf{Y}_{42\times 131}$.

1.3 Challenges of cancer drug data analysis and contribution of the thesis

The main interest in the analysis of cancer drug data is discovering the association between genes and drugs. As pointed out earlier, one appropriate model that follows the recent "multiple genes, multiple drugs" paradigm is a multivariate multiple linear regression. More precisely, through this model IC50 values of different cancer drugs can be regressed against the gene expression levels. However, most of the time it is not possible to fit this model to the cancer data directly since the gene expression data are collected in large scales. The number of genes in a single cell can be more than tens of thousands, and each gene expression adds a dimension to the data. As a result, high dimensionality has imposed a hurdle in statistical modelling of such data. This makes it impossible to estimate all the parameters in the model without imposing some constraints on the range of the parameters.

Here, we make a sparsity assumption that only a fraction of these genes is significantly associated with the disease. We tackle this challenge and reduce the dimension of gene expressions by proposing a variable screening method in Chapter 3. This approach is a beamforming-based variable screening for multivariate regressions where we wish to relate several response variables against a common set of predictors. We also propose a mixture-based screening approach in Chapter 5 to resolve the high dimensionality problem.

Another challenge in the analysis of cancer data is the high correlation between the genes. Many genes are strongly associated because they share similarities in their expressions. Thus, an expression of a single gene may be interfered by other genes. On the other hand, analysis of a single gene may produce a biased result. To resolve this issue, in Chapter 4 we further improve the proposed screening procedure. To lessen the interference of other genes, we propose a selection process called PVA through which the interferences from other genes are prevented. Therefore, by applying PVA, we can more accurately select those genes that are associated with IC50 values of drugs.

Besides high correlations and large dimensions of gene expressions in the cancer data, the heterogeneity among drugs is another aspect that needs to be considered in the modelling. The drugs can be classified into groups based on, for example, the type of disease that they can cure or the type of genes that they can target. Therefore, we need to design an appropriate model to handle high dimensionality and group structures in the data. One type of models which is widely used in modelling of heterogeneous data is a mixture model. Therefore, to accommodate the group structure in our modelling, in Chapter 5 we propose a mixture-based model to fit the cancer drug data. This model is called likelihood fusion and is applied to screen the gene expressions and classify the drugs simultaneously.

1.4 Organisation of the thesis

Our proposed methods to address the aforementioned challenges form the following chapters.
In Chapter 2 we give some preliminary and background literature on concepts to which we refer in the following chapters.

In Chapter 3 we introduce a beamforming-based variable screening method based on a new statistical filter called SNR. This new screening approach is based on projections of the multivariate response variable into the predictor space.

In Chapter 4 we propose an iterative variable selection procedure for multivariate regressions with high dimensional and correlated predictors, called principal variable analysis (PVA).

In in Chapter 5 we propose a mixture-based model called likelihood fusion and we introduce a two-stage procedure based on the proposed model to perform marginal variable screening and regression classification simultaneously.

In Chapter 6 we study the Bayesian clustering problem through finite mixture of normal distributions. We propose a different prior for the component means which depends on the component variances and mixing proportions.

Chapter 2

Introduction

In this chapter, we review some of the literature on concepts to which we refer in the following chapters. In the first section, we review some of the variable selection methods for both univariate and multivariate regressions. Particularly, we introduce some of the well-known penalisation methods to which we compare our proposed variable selection method.

After giving a brief introduction about these methods we discuss the benefits of fitting a multivariate regression model over the univariate regression model when we have a multivariate response variable. Multivariate regression model refers to a model with a multivariate response variable where certain number of observations are recorded on several response variables rather than just one response variable. Then through simulation studies we show that the outcomes obtained from fitting a multivariate regression model to data are more reliable than fitting separate univariate regression models. This is followed by introducing finite mixture models and finite regression models. We also discuss the Expectation-Maximisation (EM) algorithm that is applied to estimate the model parameters in finite mixture models which is related to the material of Chapter 5. In the last part we introduce Bayesian mixture models.

2.1 Variable selection methods for univariate linear regressions

Univariate linear regression model

In a univariate linear regression model, we are interested in explaining the linear relationship between a response (dependent) variable Y and a set of predictors (independent) variables $X_1, \cdots, X_p$. Let y_i and $x_{ik}, k = 1, \cdots, p$ denote the i-th observations on the response variable and predictors respectively. Each y_i can be specified by the following linear equation

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + e_i, \quad i = 1, \cdots, n \tag{2.1.1}$$

where e_i is the Gaussian error term. Suppose we have already centralized y_is. Then we can omit the intercept in the above equation. Thus, the above model can be written in matrix form as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \tag{2.1.2}$$

where each column of the $n \times p$ matrix $\mathbf{X}$, denotes n observations on predictors $X_k, k = 1, \cdots, p$ and the vector $\mathbf{y}$ contains n observations on the response variable. The vector $\epsilon = (e_1, \cdots, e_n)$ contains the error terms and $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)^T$ is the vector of unknown regression coefficients. We wish to estimate $\boldsymbol{\beta}$ using the observations $(\mathbf{X}, \mathbf{y})$. The commonly used approach is the *Least Squares Estimation.* The absolute value of estimated coefficients quantify the relative contribution of each regressor to the response variable. In other words, large estimates indicate greater influence on the response variable. Total sum of squares TSS $= \sum(y_i - \bar{y})^2$ is the total variability in the response variable Y before performing the regression and residual sum of squares (RSS) reflects the unexplained variability after performing regression and is obtained by RSS $= e_1^2 + \cdots + e_n^2$ (James et al., 2013).

In the linear regression we assume that there is a linear relationship between y_i and $x_{i1}, \cdots, x_{ip}$. It is unlikely that any real-life problem truly has such a simple linear relationship. So performing linear regression will undoubtedly result in some *bias*. Ideally, a good model is the one with low bias and low variance. In those cases where the actual relationship between the response and the predictors is approximately linear, the least squares estimates will have a low bias. If the number of observations, n, is larger than p, the number of independent variables, then least squares estimates also tend to have low variance. However, if n is not much larger than p, there can be much variability in the least square fit, resulting in overfitting (James et al., 2013). This means the model is very accurate for training data, but it has poor accuracy on previously unseen data not used in model training (Miller, 1984). One way to tackle this issue is to restrict the model flexibility by imposing some constraint on regression coefficients. Later in this section, we introduce penalisation methods which are designed for this purpose.

Variable selection

The advance of high-throughput technology has produced high dimensional data. This has imposed a challenge on modern statistics. This challenge is reflected in regression modelling as having $p \gg n$. As a result, the least square coefficient estimate is not unique. In practice, it is often the case that only a fraction of the predictors are informative and substantially associated with the response variable (Bühlmann and Van De Geer, 2011). As an example, consider cancer biology where the study of gene expressions is of high importance. Gene expression data invariably contain tens of thousands of genes, while only a few dozen may be actually responsible for the disease. It is only these genes which are potential targets of drugs. Therefore, it makes sense to assume that only a small proportion of the predictors has non-zero coefficients. Indeed, detecting a small, but informative, subset from such large data is one of the most challenging aspects of modern statistics (Li and Xu, 2008).

Identifying the few significant predictors among a large set of possible variables is referred to as *variable selection*. It is known as an effective way of reducing model complexity while balancing model bias and model variance. Fitting the model to the smaller set of important predictors will improve both prediction accuracy and model interpretability (James et al., 2013). Variable selection methods for regression models can be divided into three broad classes of classical *subset selection*, *penalisation* and *dimensionality reduction* methods.

2.1.1 Subset selection methods

In this section, some of the methods through which a subset of predictors is selected are discussed. These include the best subset and stepwise selection procedures. Miller (1984) suggests that by using only some of the predictors, a more accurate prediction and estimation is attained. Moreover, eliminating the uninformative variables enables us to describe the data parsimoniously and the obtained regression coefficient estimates have small standard errors particularly when some of the predictors are highly correlated.

Best subset selection

This method is performed through a two-stage process. In the first stage, all possible subsets of predictors are obtained. Then a model is fitted to each of these subsets separately. This gives us a set of 2^p different model as candidates. In the second stage of this method, the optimal model is selected according to some selection criteria such as AIC, BIC, Mallow's C_p, R^2 which are defined as follows.

Let m denote the number of predictors in the fitted model and $\hat{\sigma}^2$ is an estimate of the variance of the error term in model (2.1.1). The Akaike information criterion (AIC) is based on the maximum likelihood. Since in the model (2.1.1) errors are Gaussian, the maximum likelihood and the least squares are the same thing. Therefore, AIC is calculated by AIC$= \frac{1}{n\hat{\sigma}^2}(\text{RSS}+2m\hat{\sigma}^2)$ where RSS denotes the residual sum of square and n is the sample size. Bayesian information criteria (BIC) is derived by BIC $= \frac{1}{n}(\text{RSS} + \log(n)m\hat{\sigma}^2)$. The statistic Mallow's C_p is obtained by $C_p = \frac{1}{n}(\text{RSS} + 2m\hat{\sigma}^2)$. The adjusted R^2 statistic is calculated by $1 - \frac{\text{RSS}/(n-m-1)}{\text{TSS}/(n-1)}$. This method is summarised in the following algorithm.

Algorithm 2.1 Best subset selection

1. Start with a model with no predictor which is called *null model*, M_0.
2. For $k = 1, 2, \cdots, p$, fit all $\binom{p}{k}$ models with exactly m predictors and select the best model M_m with smallest RSS among models with m predictors.
3. Select the overall best model among $M_0, \cdots, M_p$ using some selection criteria.

Although best subset selection is a simple and easy to apply approach, it suffers from computational limitations. As the number of possible models grows significantly by increasing the number of predictors, this method becomes computationally infeasible. In addition, due to the large 2^p dimensional search space, high variance of the coefficient estimates is expected. In the following section we introduce two computationally efficient alternatives to best subset selection (James et al., 2013).

Forward and backward selections

Forward selection

Similar to best subset selection approach, forward stepwise selection algorithm begins with the null model. In stepwise forward selection, candidate models are constructed by sequentially adding one predictor at a time, until all of the predictors are in the model. At each step, while the predictor is added to the model, the p-value corresponding to this predictor is calculated. Then the predictor with the lowest p-value less than the critical value is selected and added to the model. The procedure is repeated until no new predictors can be added.

Backward selection

Unlike forward selection, the backward stepwise algorithm starts with the model that all predictors are included. Then the predictor with the highest p-value greater than the critical value is removed. The new model with one less predictor is fitted again, and the remaining least significant predictors will be removed similarly until all non-significant predictors are removed.

2.1.2 Penalisation methods

Although subset selection methods simplify the model and reduce the variance, they may be unstable. In other words, small changes in data could result in drastic changes in regression equations. As a result of the instability in subset selection methods, the prediction error is strongly affected by slight variations in the data. Besides, these methods cannot handle high dimensional data due to computational deficiency (Breiman, 1995). Since in subset selection variables are either selected or discarded, subset selection is a discrete process. As a result this method often suffers from high variability. To tackle this problem shrinkage or penalisation methods are proposed. These methods are continuous and reduce the variance by putting constraints on coefficients estimates (Hastie et al., 2009). The literature on penalisation methods is very rich and considering all these methods is beyond the scope this thesis, therefore in the following sections some of the recent famous methods to which later we compare our proposed method, are introduced.

Ridge

Ridge regression (Hoerl and Kennard, 1970) is an improvement to the ordinary least square (OLS) where model fitting is performed by minimising the residual sum of squares while limiting the ℓ_2-norm of coefficients. Consider the regression model 2.1.1 introduced earlier then the optimisation problem in ridge regression will have the form

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2,$$

where $\|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{p} \beta_j^2$ and $\lambda \geq 0$ is a tuning parameter and it controls the amount of shrinkage. As $\lambda \to \infty$ the amount of shrinkage increases which results in variance reduction and therefore a better prediction accuracy. Although ridge regression is a stable method, it shrinks small coefficients towards zero but not set them to zero hence all predictors are retained in the model. Therefore variable selection cannot be performed through the ridge regression. A nice feature of the ridge penalisation is the ability of this method to shrink correlated variables towards each other. This property is referred to as grouping effect. A new technique which is introduced in the next section was proposed with the aim of improving the ridge regression.

Lasso

Lasso which was proposed by Tibshirani (1996) is an alternative to the ridge regression which imposes the ℓ_1-norm penalty on coefficients. So the residual sum of squares (RSS) will be minimised as follows

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1,$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$. Because of the nice geometric feature of the lasso constraint this method has the property to set some of the coefficients equal to zero. Unlike the strictly convex ℓ_2-norm in ridge regression, the ℓ_1-norm of coefficients hits the RSS contours which are defined by

$$\sum_{i=1}^{n}\left(y_i - \sum_{j}\beta_j x_{ij}\right)^2,$$

on the axes, so the corresponding coefficient will be set to zero.

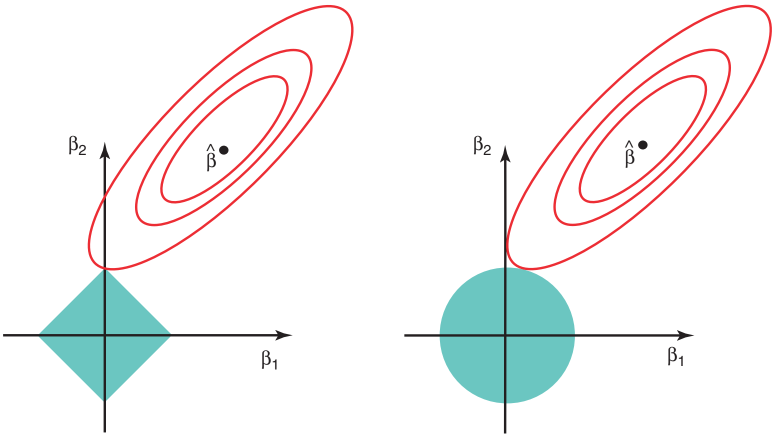


Figure 2.1.1: RSS contours shown in red ellipses and green areas show penalty functions for the lasso (left) with the constraint region, $|\beta_1| + |\beta_2| \leq \lambda$ and ridge regression (right) with the constraint region, $\beta_1^2 + \beta_2^2 \leq \lambda$ (James et al., 2013).

This will lead to a sparse model which is more interpretable (Tibshirani, 1996). However, lasso does not have the grouping feature of the ridge regression. As a result, in the presence of correlated variables, lasso tends to select one from the grouped correlated variables and discards others. This can occur in biological data analysis. For example, gene expressions are highly correlated when genes belong to the same pathway. This cannot be explored by lasso because it lacks the grouping effect property. Another shortcoming of the lasso is that when $p \gg n$ it can select at most n predictors before it saturates, also lasso may not be an ideal approach where the aim is building a predictive model (Zou and Hastie, 2005) .

Elastic net

Zou and Hastie (2005) proposed a new penalty called elastic net. This penalty is a convex combination of ridge and lasso penalty and as a result, it possesses the nice features of both the ridge and the lasso, while it improves the prediction accuracy of the lasso. Elastic net solves the following optimisation problem

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1. \tag{2.1.3}$$

Let $\alpha = \frac{\lambda_2}{(\lambda_1 + \lambda_2)}$ then (2.1.3) can be equivalently written as

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \quad s.t \quad \alpha \|\boldsymbol{\beta}\|_2^2 + (1-\alpha)\|\boldsymbol{\beta}\|_1 \leq t \quad \text{for some } t.$$

Resulted estimates from elastic net regression can be regarded as the weighted average of lasso and ridge solutions. This method also does the variable selection and shrinkage at the same time and is capable of selecting the grouped variables.

Group lasso

An extension of lasso was introduced by Yuan and Lin (2006) where the selection is performed at the group level. Unlike the elastic net, in this method, the covariates are partitioned into non-overlap groups prior to penalisation. In other words, the solution will be non-zero groups of coefficient estimates instead of individual estimates. When the covariates are assumed to come from m non-overlap groups, this method solves

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \sum_{l=1}^{m} X^{(l)} \boldsymbol{\beta}^{(l)}\|_2^2 + \lambda \sum_{l} \sqrt{p_l} \|\boldsymbol{\beta}^{(l)}\|_2$$

where $X^{(l)}$ is the submatrix of $\mathbf{X}$ columns of which are predictors in the l-th group with corresponding coefficients $\boldsymbol{\beta}^{(l)}$ and p_l is the length of $\boldsymbol{\beta}^{(l)}$.

2.1.3 Dimension reduction methods

Dimension reduction is an alternative approach to exclude irrelevant information and noisy features in the data. Such approaches reduce the dimension of data by projecting data onto a lower dimensional space, while the informative and interesting structure in the data is preserved.

Definition 1. *A linear projection $\mathbb{R}^p \to \mathbb{R}^k$ is a linear map $\mathbf{A}$ or $k \times p$ matrix of rank k:*

$$\mathbf{w} = \mathbf{A}\mathbf{x}, \ \ \mathbf{x} \in \mathbb{R}^p, \ \ \mathbf{w} \in \mathbb{R}^k \tag{2.1.4}$$

The projection is orthogonal if the row vectors of $\mathbf{A}$ are orthogonal to each other and have length one. If $k = 1$, then $\mathbf{A}$ reduces to a row vector $\mathbf{a}^T$ which is called direction vector. A direction vector is a vector of norm one (Rao and Rao, 1998).

Projection Pursuit introduced by Friedman and Tukey (1974) is a dimension reduction approach that pursues interesting low dimensional orthogonal projections of data. Koch (2013) describes the projection pursuit as the search for projections worth pursuing. This algorithm associates an index to each projection to measure the interestingness of that projection.

Definition 2. *Let $\boldsymbol{x}$ be a p-dimensional random vector, and let $\mathbf{a} \in \mathbb{R}^p$ be a direction vector. A projection index $\mathcal{Q}$ is a function which assigns a real number to pairs $(\boldsymbol{x}, \boldsymbol{a})$ (Koch, 2013).*

Through projection pursuit data are projected onto a lower dimensional space, then the low dimensional projections are described by the projection index. This index is then maximised to obtain interesting projections. Here, we introduce a special case of projection pursuit methods where the projection index is the variation in the data. In other words, the variation is the index which needs to be maximised. This technique is called *Principal Component Analysis (PCA).* Principal components capture directions with the highest variation in the data. Principal components are calculated as follows. Let

$\mathbf{x} \sim (\mu, \Sigma)$ be a d-dimensional random vector, and let $\mathbf{a} \in \mathbb{R}^d$ be a direction vector. The projection index for $\mathbf{x}$ and $\mathbf{a}$ is the variance of projected data. Hence,

$$\mathcal{Q}(\mathbf{x}, \mathbf{a}) = \mathrm{Var}(\mathbf{a}^T \mathbf{x}),$$

Since Σ denotes the covariance matrix of $\mathbf{x}$, to find the first principal component the following optimisation problem is solved

$$\max_{\mathbf{a}_1} \ \mathrm{Var}(\mathbf{a}_1^T \mathbf{x}) = \max_{\mathbf{a}_1} \ \mathbf{a}_1^T \Sigma \mathbf{a}_1 \quad \text{s.t} \quad \mathbf{a}_1^T \mathbf{a}_1 = 1. \tag{2.1.5}$$

Implementing the method of Lagrange multiplier and differentiating with respect to $\mathbf{a}_1$ gives $(\Sigma - \lambda_1 \mathbf{I}_d)\mathbf{a}_1 = \mathbf{0}$, where $\mathbf{I}_d$ is a $d \times d$ identity matrix. Thus, λ_1 is the eigenvalue of Σ and $\mathbf{a}_1$ is the corresponding eigenvector. Since $\mathbf{a}_1^T \Sigma \mathbf{a}_1 = \mathbf{a}_1^T \lambda_1 \mathbf{a}_1 = \lambda_1$ is to be maximised, λ_1 should be as large as possible. So $\mathbf{a}_1$ is the eigenvector corresponding to the largest eigenvalue. Thus, the maximiser of this projection index over direction vectors $\mathbf{a}_1$ is the eigenvector of Σ with the eigenvalue of

$$\lambda_1 = \max_{\{\mathbf{a}:\|\mathbf{a}\|=1\}} \ \mathcal{Q}(\mathbf{x}, \mathbf{a}).$$

The second principal component is derived by solving the optimisation problem (2.1.5) with the additional normalisation constraint $\mathbf{a}_2^T \mathbf{a}_1 = \mathbf{0}$ to guarantee that these principal components are uncorrelated. Consequently, the second principal component is derived by constructing the following Lagrangian function

$$\mathbf{a}_2^T \Sigma \mathbf{a}_2 - \lambda_2(\mathbf{a}_2^T \mathbf{a}_2 - 1) - \lambda_3(\mathbf{a}_2^T \mathbf{a}_1),$$

differentiating the above function with respect to $\mathbf{a}_2$ and setting the equation equal to zero gives $(\Sigma - \lambda_2 \mathbf{I}_d)\mathbf{a}_2 = \mathbf{0}$. Similarly, λ_2 is an eigenvalue of Σ with the corresponding eigenvector $\mathbf{a}_2$. Also λ_2 is the second largest eigenvalue of Σ. Identically, the m-th principal component of $\mathbf{x}$ is $\mathbf{a}_m^T \mathbf{x}$ where $\mathbf{a}_m$ is the eigenvector corresponding to the m-th largest eigenvalue (Jolliffe, 1986). It is common to find the first few principal components to reduce the dimension of data. Indeed, PCA represents the data in a new orthogonal coordinate system which optimally accounts for the variation in the data. The proportion of the variance that each eigenvector represents can be calculated by dividing the eigenvalue corresponding to that eigenvector by the sum of all eigenvalues.

Dimension reduction for regression model (2.1.1) is also performed by finding the first q principal components. These projections are in the lower dimensional space spanned by the first q eigenvectors of the predictors. Consider the p-dimensional vector $\mathbf{x}_i^T$ which is the i-th row of the design matrix $\mathbf{X}_{n\times p}$ in univariate regression model (2.1.2). We drop the index i in notations for the rest of this section and use $\mathbf{x}$ instead. Suppose $R(\mathbf{x})$ is a function of dimension less than p such that it carries all the information that $\mathbf{x}$ has about the response variable Y. Hence, $E(Y|\mathbf{x}) = E(Y|R(\mathbf{x}))$. Cook (2007) defines the dimension reduction as follows.

Definition 3. *The action of replacing $\boldsymbol{x}$ with a lower dimensional $R(\boldsymbol{x})$ provided that it captures all the information that $\boldsymbol{x}$ contains about Y so that $E(Y|\boldsymbol{x}) = E(Y|R(\boldsymbol{x}))$ is called sufficient dimension reduction.*

Dimension reduction is applied to the regression model (2.1.1) in two steps. On the first step which is the reduction step, $\mathbf{x}$ is reduced linearly to $\mathbf{G}^T\mathbf{x}$ using some methodology that produces $\mathbf{G} \in \mathbb{R}^{p\times q}, q \leq p$. The second step is estimating the mean function $E(Y|\mathbf{G}^T\mathbf{x})$ for reduced predictors. In the following we show that this sufficient reduction is performed through principal components.

Suppose Y is the $n \times 1$ vector of centred response and $\mathbf{X}_{n\times p}$ be the centered design matrix with rows $(\mathbf{x}_i - \bar{\mathbf{x}})^T, i = 1, \cdots, n$, where $\bar{\mathbf{x}} = \sum_{i=1}^{n} \mathbf{x}/n$ is the sample mean. Let $\hat{\mathbf{\Sigma}} = \mathbf{x}^T\mathbf{x}/n$ denotes the sample covariance and $\hat{\mathbf{S}} = \mathbf{X}^T Y/n$. If we denote the OLS estimator by $\hat{\boldsymbol{\beta}}_{ols} = \hat{\mathbf{\Sigma}}^{-1}\hat{\mathbf{S}}$, then (Cook and Forzani, 2009)

$$\hat{\boldsymbol{\beta}}_{\mathbf{G}} = \mathbf{P}_{\mathbf{G}(\hat{\mathbf{\Sigma}})}\hat{\boldsymbol{\beta}}_{ols} = \mathbf{G}(\mathbf{G}^T\hat{\mathbf{\Sigma}}\mathbf{G})^{-1}\mathbf{G}^T\hat{\mathbf{S}} \tag{2.1.6}$$

This estimator is the projection of $\mathbf{P}_{\mathbf{G}(\hat{\mathbf{\Sigma}})}$ of $\hat{\boldsymbol{\beta}_{ols}}$ onto span($\mathbf{G}$) in the $\hat{\mathbf{\Sigma}}$ inner product. If $\mathbf{G} = \mathbf{I}_p$ then $\hat{\boldsymbol{\beta}}_{\mathbf{G}} = \hat{\boldsymbol{\beta}_{ols}}$. If the columns of $\mathbf{G}$ are chosen to be the first q eigenvectors of $\hat{\boldsymbol{\Sigma}}$ then $\mathbf{G}^T\mathbf{x}$ includes the first q principal components and $\hat{\boldsymbol{\beta}}_{\mathbf{G}}$ is the principal component regression estimator (Cook, 2007).

2.2 Extensions to multivariate linear regressions

Multivariate linear regression model

In the multivariate regression setting, we model several response variables by using the same set of covariates. This model is widely used in applied areas such as economics and biology where finding the linear relationship between a set of predictors and several response variables is of interest. Consider the dataset $(\mathbf{Y}, \mathbf{X})$ where $\mathbf{Y} = \mathbf{Y}_{n\times J} = (y_{ij}) = (\mathbf{y}_1\mathbf{y}_2\cdots\mathbf{y}_J)$ and $\mathbf{X} = \mathbf{X}_{n\times p} = (x_{ik}) = (\mathbf{x}_1\cdots\mathbf{x}_p)$, and $\mathbf{y}_j$ and $\mathbf{x}_k$ are vectors of n observations made on the j-th response variable and the k-th predictor. Then, we can formulate a multivariate multiple regression model as follows:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}, \tag{2.2.1}$$

where $\mathbf{B} = \mathbf{B}_{p\times J} = (\mathbf{b}_1\mathbf{b}_2\cdots\mathbf{b}_J)$ and $\mathbf{E} = \mathbf{E}_{n\times J} = (\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_2\cdots\boldsymbol{\varepsilon}_J)$ with $\mathbf{b}_j$ and $\boldsymbol{\varepsilon}_j$ respectively denote the values of the regression coefficients and the error terms related to the j-th response variable.

Such models can be fitted utilizing two different approaches as follows. Since the multivariate regression model (2.2.1) can be written as J separate univariate regression models

$$\mathbf{y}_j = \mathbf{X}\mathbf{b}_j + \boldsymbol{\varepsilon}_j \;\; 1 \leq j \leq J, \tag{2.2.2}$$

we can fit each univariate regression separately to the data, or alternatively, these univariate equations can be estimated jointly by fitting the single multivariate model (2.2.1) to the data. The key feature about considering model

(2.2.1) is that through this model, the correlation structure between and within the columns of the response variable $\mathbf{Y}_{n\times J}$ is taken into account. Although when $p < n$, the least square solution, $\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ gives the same coefficients as fitting J single multiple regression models separately, this solution is not efficient when $p > n$ or when we deal with high dimensional data with $p \gg n$. By fitting separate regression models to $\mathbf{y}_j$'s, the correlations among $\mathbf{y}_j$'s are ignored. Accordingly, such modelings are likely to suffer from overfitting and high variability (Peng et al., 2010). Breiman and Friedman (1997) show that considering all response variables simultaneously can improve prediction accuracy especially when the responses are correlated.

2.2.1 Multivariate subset selection methods

Subset selection for multivariate regression model (2.2.1) can be performed by building J separate models for each response variable. For the same rationales pointed out earlier, this approach is also not efficient for subset selection. The reason is, in many applications we require to find the best subset for all response variables simultaneously. Moreover, implementing subset selection methods using model (2.2.1) is computationally more efficient. Subset selection methods for multivariate regressions follow the same procedure as univariate models. However, the selection criteria is formulated differently under multivariate regression models. In the work done by Al-Subaihi et al. (2002), these selection criteria are extended to the multivariate case. Suppose k specifies the number of predictors in the model and $\mathbf{J}$ is a $J \times J$ unit matrix. The adjusted $\mathbf{R}^2$ is defined by $\mathbf{AR}^2 = 1 - \frac{(n-1)(1-\mathbf{R}^2)}{n-k}$ where,

$$\mathbf{R}^2 = |[\mathbf{Y}^T(\mathbf{I} - \frac{1}{n}\mathbf{J})\mathbf{Y}]^{-1}[\mathbf{Y}^T(\mathbf{X}_k(\mathbf{X}_k^T\mathbf{X}_k)^{-1}\mathbf{X}_k^T - \frac{1}{n}\mathbf{J})\mathbf{Y}]|, \qquad (2.2.3)$$

where, $\mathbf{X}_k$ is the sub-matrix of $\mathbf{X}$ containing vector $\mathbf{1}$ and the columns corresponding to the k predictors in the model. **AIC** and **BIC** and Mallow's $\mathbf{C}_p$

are defined as

$$\begin{aligned}
\mathbf{AIC} &= \ln|\mathbf{RSS}| + \frac{2kJ + J(J+1)}{n} \\
\mathbf{BIC} &= \ln|\mathbf{RSS}^2| + \frac{\ln(n)k}{n} \\
\mathbf{C}_p &= (n-p)\mathbf{TSS}^{-1}\mathbf{RSS} + (2k-n)\mathbf{I}
\end{aligned}$$

where,

$$\begin{aligned}
\mathbf{TSS} &= \mathbf{Y}^T[\mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T]\mathbf{Y}. \\
\mathbf{RSS} &= \mathbf{Y}^T[\mathbf{I} - \mathbf{X}_k(\mathbf{X}_k^T\mathbf{X}_k)^{-1}\mathbf{X}_k^T]\mathbf{Y}.
\end{aligned}$$

2.2.2 Multivariate penalisation methods

Through the following sections, we introduce the multivariate regression model and discuss the multivariate form of the penalisation methods which were mentioned for univariate regressions.

Multivariate lasso, group lasso and sparse group lasso

The idea of the lasso (Tibshirani, 1996) introduced earlier for univariate regressions, is generalised to the multivariate regressions by Peng et al. (2010) and Vincent and Hansen (2014). The group-lasso of Yuan and Lin (2006) and the elastic net (Zou and Hastie, 2005) penalisation methods are extended to the multivariate regression models by Simon et al. (2013). In the work done by Simon et al. (2013), the imposed penalty is an ℓ_2-norm penalty whereby variables are selected at the group level. However, this penalty does not encourage the sparsity within selected groups. On the other hand, in this approach, the dimensionality of the response variable is not taken into account. Therefore, selected non-zero groups are either all zero or non-zero for all of the response variables. In other words, if a predictor has a non-zero coefficient estimate, this predictor is associated with all of the response variables. Although in regression models with multivariate response variable it is likely that the response variables are correlated, and as a result, they may be associated with the same set of predictors, in some cases these predictors may affect some of the responses rather than all of them. This shortcoming is addressed in approaches proposed by Vincent and Hansen (2014) and Peng et al. (2010) where a penalty which is a combination of ℓ_1 and ℓ_2 norm is applied. The following optimisation problem is solved in the Peng et al. (2010) method:

$$\hat{\mathbf{B}} = \underset{\mathbf{B}\in\mathbb{R}^{p\times J}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{Y}-\mathbf{X}\mathbf{B}\|_F^2 + \lambda_1 \sum_{l=1}^{p} \| C_l \mathbf{B}_{l.} \|_1 + \lambda_2 \sum_{l=1}^{p} \| C_l \mathbf{B}_{l.} \|_2,$$

where C_l is the l-th row of $\mathbf{C} = (c_{ij})_{p\times J}$ which is a pre-specified $p \times J$ matrix of 0 and 1 indicating the coefficients on which penalisation is imposed. $\mathbf{B}_{l.}$ is the l-th row of $\mathbf{B}$ and $\| \,.\, \|_F$ denotes the Frobenius norm of matrices. The indicator matrix $\mathbf{C}$ is pre-specified based on prior knowledge: if we know that predictor $\mathbf{x}_l$ affects the response $\mathbf{y}_j$ then the corresponding regression coefficient β_{lj} will not be penalised and c_{lj} is set to zero. When there is no such prior information, $\mathbf{C} = (c_{ij})_{p\times J}$ can be set to a constant matrix $\mathbf{C} = (c_{ij}) \equiv 1$. The ℓ_1 norm controls the overall sparsity of the coefficient matrix $\mathbf{B}$ and the ℓ_2 norm imposes a group penalty on rows of the coefficient matrix. The result of the procedure is called multivariate lasso when $\lambda_2 = 0$ and multivariate group lasso when $\lambda_1 = 0$. In addition,this penalty puts constraints on the total number of predictors entering the model. This is achieved by treating the coefficients corresponding to the same predictor (one row of $\mathbf{B}$) as a group and then penalising its ℓ_2 norm which is equivalent to sparse group lasso. The sparse group penalty simultaneously selects the important groups while selecting some predictors within the selected group. This allows the situation that a predictor can be associated with some of response variables but not with all of them.

In our simulation studies we use *lsgl* R-package (Vincent and Hansen, 2014) to perform multivariate lasso and multivariate sparse group lasso. In this package the following optimisation problem is solved:

$$\hat{\mathbf{B}} = \underset{\mathbf{B} \in \mathbb{R}^{p \times J}}{\text{argmin}} \frac{1}{n} \|\mathbf{Y} - \mathbf{XB}\|_F^2 + \lambda \left\{ (1-\alpha) \sum_{l=1}^{p} \| \mathbf{B}_{l.} \|_2 + \alpha \sum_{j=1}^{J} \sum_{i=1}^{p} |b_{ij}| \right\},$$

where $\mathbf{B}_{l.}$ is a J-vector or the l-th row of $p \times J$ coefficient matrix $\mathbf{B} = (b_{ij})$. This penalty is equivalent to: multivariate lasso penalty when $\alpha = 1$, multivariate group lasso penalty when $\alpha = 0$ and multivariate sparse group lasso penalty for $\alpha \in [0, 1]$.

Multivariate elastic net

Multivariate elastic net proposed by Simon et al. (2013) for multivariate regressions implies a penalty which is a convex combination of the ridge and the group-lasso penalty. The following optimisation equation is solved by multivariate elastic net

$$\hat{\mathbf{B}} = \underset{\mathbf{B} \in \mathbb{R}^{p \times J}}{\text{argmin}} \frac{1}{2n} \|\mathbf{Y} - \mathbf{XB}\|_F^2 + \lambda \left\{ \frac{(1-\alpha)}{2} \| \mathbf{B} \|_F^2 + \alpha \sum_{l=1}^{p} \| \mathbf{B}_{l.} \|_2 \right\},$$

for $\alpha \in [0, 1]$ and $\mathbf{B}_{l.}$ is a J-vector or the l-th row of $p \times J$ coefficient matrix $\mathbf{B}$. As special cases $\alpha = 1$ corresponds to multivariate group lasso and $\alpha = 0$ corresponds to ridge penalty. Setting $\alpha = 0.5$ will give equal weight to both penalties and corresponds to multivariate elastic net.

2.2.3 Multivariate dimension reduction methods

Likewise the univariate regression models, dimension reduction methods for multivariate regression models also specify a set of orthogonal linear combinations of predictors. Then this lower dimensional set is regarded as a new set of predictors which are regressed against the responses. Considering the data set $(\mathbf{Y}, \mathbf{X})$ introduced above, dimension reduction for the multivariate regression model (2.2.1) is performed through principal component regression.

The corresponding principal component estimator is defined as follows. Let $\mathbf{T}_d = \mathbf{X}\mathbf{A}_d$ denote the $n \times d$ orthogonal matrix which contains d orthogonal linear combinations of predictors. The columns of the $p \times d$ matrix $\mathbf{A}_d$ are direction vectors. Then, the regression coefficient matrix obtained by the reduced set of predictors is of the form

$$\hat{\mathbf{B}}_d = \mathbf{A}_d(\mathbf{T}_d^T\mathbf{T}_d)^{-1}\mathbf{T}_d^T\mathbf{Y},$$

now if the columns of $\mathbf{A}_d$ are chosen to be the first d eigenvectors of $\mathbf{X}^T\mathbf{X}$, then the linear transformation matrix $\mathbf{T}_d$ consists of the first d principal components and $\hat{\mathbf{B}}_d$ is the principal component estimator (Abraham and Merola, 2005). Later, Yuan et al. (2007) introduce a different dimension reduction approach which is also based on the linearly transformed predictors. In this method, response variables are regressed against lower dimensional predictors called factors. In principal components regression, the factors are chosen to be the principal components of the predictors. Factor regression model is of the form

$$\mathbf{Y} = \mathbf{F}\boldsymbol{\Omega} + \mathbf{E}, \tag{2.2.4}$$

where $\mathbf{F} = \mathbf{X}\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}$ is a $p \times r$ matrix for some $r \leq \min(p, J)$ and $\boldsymbol{\Omega}$ is an $r \times J$ matrix. The columns of $\mathbf{F}$ are referred to as factors. Note that model (2.2.4) is just a different representation of the model (2.2.1) where the coefficient matrix is replaced by $\mathbf{B} = \boldsymbol{\Gamma}\boldsymbol{\Omega}$. To fit the above model first the factors or $\boldsymbol{\Gamma}$ is estimated and then $\boldsymbol{\Omega}$ is estimated by least squares. Thus, to estimate the coefficient matrix Yuan et al. (2007) suggests a novel penalty wherein the sum of the singular values or the Ky Fan norm (Fan and Hoffman, 1955) of the coefficient matrix is constrained. Since this penalty encourages sparsity among singular values, besides shrinkage, dimension reduction is also performed. In this proposed method, the choice of the number of factors, determining them and estimating the factor loadings $\boldsymbol{\Omega}$ are performed at the same time. Suppose that the singular value decomposition of $\mathbf{B}$ is factorised as $\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ where $\mathbf{V}$ is a $J \times J$ orthonormal matrix and $\mathbf{D}_{p\times J} = (d)_{ij}$ is a diagonal matrix with the i-th largest singular values on diagonal i.e. $d_{ii} = \sigma_i(B)$ where $\sigma_i(.)$ denotes the i-th largest singular value of a matrix. So the estimate of $\mathbf{B}$ is obtained by solving the optimisation problem of

$$\min \left[\operatorname{tr}\left\{(\mathbf{Y} - \mathbf{XB})(\mathbf{Y} - \mathbf{XB})^T\right\}\right] \quad \text{subject to} \quad \sum_{i=1}^{\min(p,q)} \sigma_i(\mathbf{B}) \leq t,$$

where $t \geq 1$ and the constraint $\sum_{i=1}^{\min(p,q)} \sigma_i(\mathbf{B})$ is called the Ky Fan (p or q)-norm of $\mathbf{B}$.

Despite all developments in variable selection methods for multivariate regression, some of these methods cannot handle high dimensional data. Moreover, the correlation between responses is not taken into account for variable selection in these methods. To address these issues, in Chapter 4, we extend the idea of principal component analysis and propose a different variable selection method called *principal variable analysis (PVA)*. This method attempts to capture the maximum variation in the data with a small number of principal variables.

2.3 Criterion for performance evaluation

In this section, we introduce the concept of sensitivity and specificity, two quantities that we use through the thesis to examine and compare the selection accuracy of different variable selection approaches.

Sensitivity and specificity are statistical measures of the performance of a binary classification test. *Specificity*, or true negative rate, measures the proportion of negatives in a binary classification test which are correctly identified. *Sensitivity*, or true positive rate, measures the proportion of true positives correctly detected by a binary classification test (Asche, 2015). A test that has high values of both sensitivity and specificity is considered as a good test. In variable selection framework, these notions are defined as follows. Sensitivity is the survival rate of true active or non-zero predictors and specificity refers to the survival rate of true non-active or zero predictors in selection procedures. In simulation settings, we consider a predictor as true active if its corresponding coefficient is set to a not zero value and true non-active if the corresponding coefficient is set to zero. Let T and T^c denote sets of true active and true non-active predictors, respectively. Also assume that $\hat{T}$ and $\hat{T}^c$ are their estimators obtained after carrying out a variable selection approach. The symbol $|.|$ denotes the size of a set. Accordingly, sensitivity and the specificity are defined as

$$\text{SEN} = \frac{|\hat{T} \cap T|}{|T|}, \quad \text{SPE} = \frac{|\hat{T}^c \cap T^c|}{|T^c|}.$$

$|\hat{T}| \leq n$ where n is the sample size and $T \cup T^c = \hat{T} \cup \hat{T}^c = \{1, 2, ..., p\}$ where p is the number of predictors. Also the following inequality holds for specificity values that reads

$$\text{SPE} = \frac{|\hat{T}^c - \hat{T}^c \cap T|}{|T^c|} \geq \frac{p - n - |T|}{p - |T|}.$$

So the specificity is close to 1 when $p \gg |T| + n$. This holds for most of our simulations, for example for $n = 42, p = 2000, |T| = 37$ we have SPE ≥ 0.978.

2.4 Simulation studies

In this section, our main focus is on the performance of penalisation-based variable selection methods which are introduced in previous sections. Our simulations serve two purposes:

(a) To investigate whether multivariate regression offers any improvement over separate univariate regression.

(b) To compare the performance of the introduced multivariate penalisation methods in variable selection in terms of sensitivity and specificity.

All simulations were programmed and conducted in R software. In our simulation studies, the *glmnet* R-package (Friedman et al., 2010) was used to apply univariate and multivariate penalisation methods with elastic net and lasso penalties. Note that the multivariate lasso in glmnet imposes a group lasso penalty and not a lasso penalty. Therefore, we used the *lsgl* R-package (Vincent and Hansen, 2014) to apply the multivariate lasso and multivariate sparse group lasso penalties.

2.4.1 Data generation

In all simulations the design matrix $\mathbf{X}_{n\times p}$ was generated by sampling n number of p-vectors $\mathbf{x}_i,\ i = 1, \cdots, n$ from a multivariate normal $N_p(\mathbf{0}, \mathbf{\Sigma}_{p\times p})$ where $\mathbf{\Sigma}$ is the covariance matrix of the gene expressions in our real data.

In order to monitor how correlations among response variables influence the selection accuracy, we considered two different correlation structures in simulating the coefficient matrix $\mathbf{B}$ as follows. Let matrix $(\mathbf{IC})_{n\times J}$ where $J = 131$, denote the IC50 values of 131 drugs across n cell lines in real data, and matrix $\mathbf{R}_{IC} = (r_{ij})$ be the correlation matrix. Suppose that $J = J_1 + J_2$ and those columns of $\mathbf{IC}$ with correlation $r_{.j} < 0.3$ form the sub-matrix $IC^l_{n\times J_1}$ and columns with $r_{.j} \geq 0.5$ form the sub-matrix $IC^h_{n\times J_2}$ with the corresponding covariance matrix of $(\Omega_l)_{J_1\times J_1}$ and $(\Omega_h)_{J_2\times J_2}$ respectively. We generated the coefficient matrix $\mathbf{B}$ under two different scenarios.

Scenario 1 (Strongly correlated coefficient matrix): In this scenario, the coefficient matrix $\mathbf{B}^h_{p\times J_2}$ was generated by sampling p number of J_2-vectors from $N_{J_2}(\mathbf{0}, \Omega_h)$.

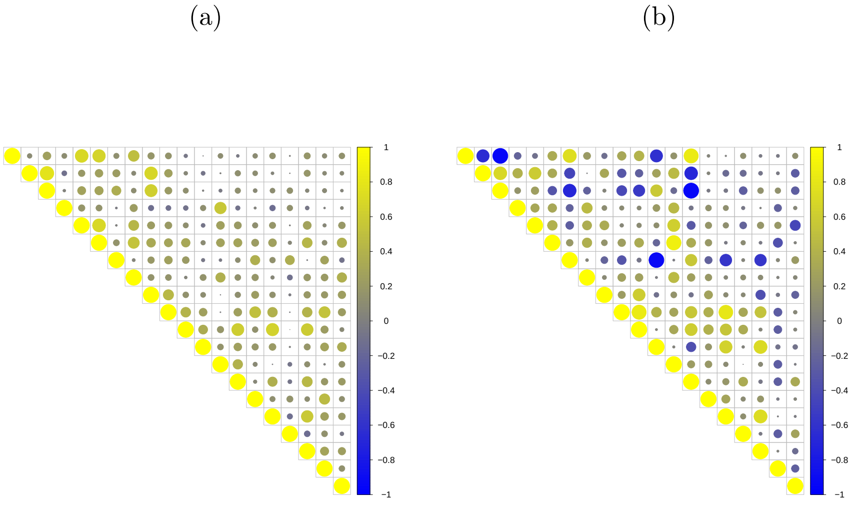


Figure 2.4.1: The correlation structure of (a) simulated coefficient matrix and (b) simulated response variable under scenario 1 with strong correlation structure.

Scenario 2 (Weakly correlated coefficient matrix): In this scenario, the coefficient matrix $\mathbf{B}^l_{p\times J_1}$ was generated by sampling p number of J-vectors from $N_{J_1}(\mathbf{0}, \Omega_l)$. The correlation structure in $\mathbf{B}^l$ and $\mathbf{B}^h$ are represented in Figure 2.4.1a and Figure 2.4.2a.

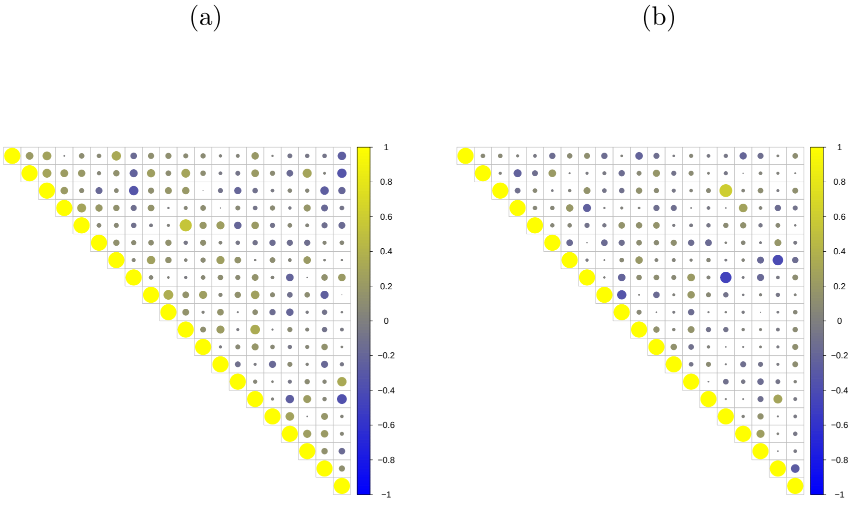


Figure 2.4.2: The correlation structure of (a) simulated coefficient matrix and (b) simulated response variable under scenario 2 with low correlation structure.

According to our model assumption the coefficient matrix is sparse hence, in scenario 1 the non-zero elements were placed in columns of $\mathbf{B}^h_{p\times J_2}$ with $r_{ij} \geq 0.5$ and in scenario 2 we placed the non-zero elements in those columns of $\mathbf{B}^l_{p\times J_1}$ with low correlations $r_{ij} < 0.3$. The error matrix, $\mathbf{E}_{n\times J}$, was generated by sampling J times from a multivariate normal distribution $N_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$, where $\sigma^2 = 0.1$. Finally, the simulated multivariate response variable satisfies the following multivariate regression model

$$\mathbf{Y}_{n\times J} = \mathbf{X}_{n\times p}\mathbf{B}_{p\times J} + \mathbf{E}_{n\times J}, \tag{2.4.1}$$

where $\mathbf{B}_{p\times J} \in \{\mathbf{B}^l_{p\times J_1}, \mathbf{B}^h_{p\times J_2}\}$. For all penalisation methods prior to model fitting, predictors and response variables are standardised. The correlation structure in simulated response variable resulted from scenario 2 with weakly correlated coefficient matrix and scenario 1 with strongly correlated coefficient matrix are shown in Figure 2.4.2b and Figure 2.4.1b.

In all simulations and for all penalisation methods, the regularisation path corresponding to each penalty was computed at a grid of values for the regularisation parameter λ. Therefore, different values of λ yield a different number of selected variables which leads to different values of specificity. To make a fair comparison, we fixed the specificity of each method at the same level and then compared the sensitivities. Suppose we want to fix the specificity on 97%, since it is not possible to have the same number of selected variables for each λ, we searched for a particular value of λ such that the specificity of which satisfied the condition $|\text{SPE} - 97\%| \leq 0.01$. Thus, the specificity values did not differ much and were approximately the same and we could compare the sensitivity values.

2.4.2 Simulation results

(a) Comparing multivariate variable selection with multiple separate univariate variable selection: In this section, we examine the variable selection performance of two different approaches to the problem of multivariate regression modelling. We utilised the single multivariate regression model (2.2.1) and imposed the multivariate lasso and multivariate elastic net penalty to estimate the coefficient matrix $\mathbf{B}$. Taking a different path, we also considered each column $\mathbf{y}_j; j = 1, \cdots, J$ of the response variable $\mathbf{Y}_{n \times J}$ together with the predictor matrix $\mathbf{X}_{n \times p}$ and fit J separate univariate regression where the univariate lasso and univariate elastic net penalties were applied. We then compared the sensitivity values obtained from each approach under scenario 1 and scenario 2 with low and high correlation structure among response variables. Although using multivariate regression might give the same result as using separate univariate regressions, through simulations we present some cases where applying multivariate regression model gives better results. To this aim, we ran 50 simulations with combination of $(n, p, J, |T|) = (88, 2000, 20, 70)$ and compared sensitivities while fixing specificities at the level 97%. Results are presented in Figure 2.4.3 and Figure 2.4.4.

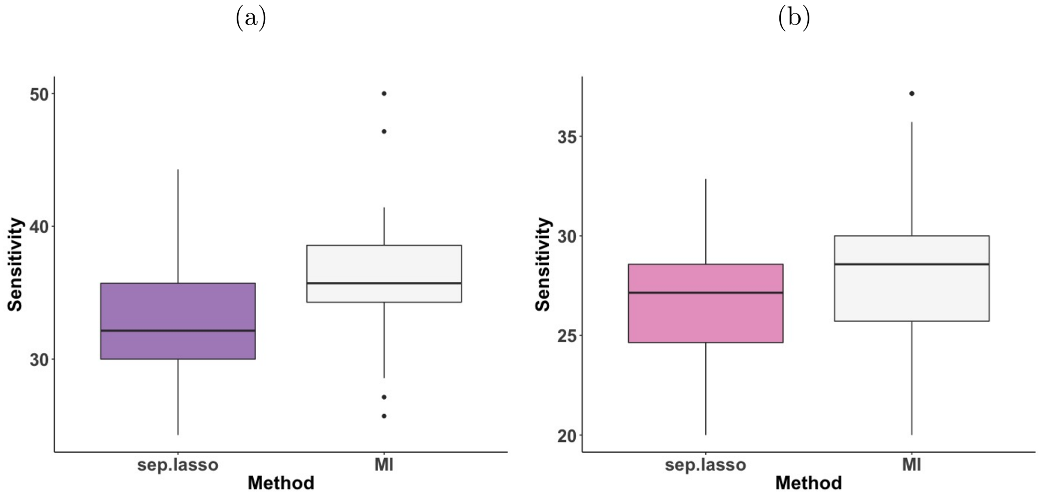


Figure 2.4.3: Sensitivity box plots obtained from 50 simulations with $p = 2000, n = 88, J = 20, |T| = 70$, where specificities are fixed. Results are corresponding to variable selection through lasso penalisation when multivariate (Ml) and univariate (sep.lasso) models were applied with (a) high correlation and (b) low correlation among among response variables.

As we expected, when we consider all the response variables simultaneously through the multivariate regression model (2.2.1) the variable selection performance of lasso was improved. According to sensitivity values, under both scenarios, the selection accuracy of the multivariate regression model is higher than separate univariate regressions.

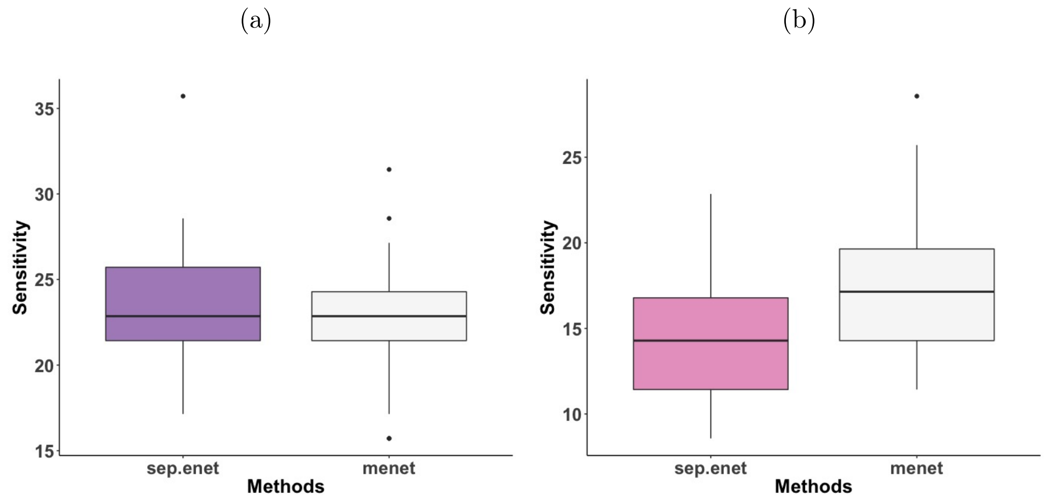


Figure 2.4.4: Sensitivity box plots obtained from 50 simulations with $p = 2000, n = 88, J = 20, |T| = 70$, where specificities are fixed. Results are corresponding to variable selection through elastic net penalisation when the multivariate elastic net (menet) and the univariate elastic net (sep.enet) penalties were applied with (a) high correlation and (b) low correlation among response variables.

Results presented in Figure 2.4.4 show that when response variables are highly correlated, the selection accuracy of elastic net is almost the same for both univariate and multivariate regression models. Since response variables are highly correlated, detection of important predictors becomes easier. As a result, even if multiple univariate regressions are fitted to data, the selection accuracy of elastic net is almost the same or even better than fitting a multivariate regression model. However, for weakly correlated response variables, the univariate elastic net performs poorer and selection accuracy of the multivariate elastic net is noticeably higher. Therefore, for datasets with weakly correlated response variables, fitting a multivariate regression would be preferable. These results also reveal that when the correlation among the response variables is high, the sensitivity is higher. This means that when the correlation among response variables is high, detection of non-zero predictors is easier for both multivariate and univariate model when elastic net and lasso penalty are imposed to the coefficient estimates.

(b) Comparing penalisation methods for multivariate regressions: Results shown in Figure 2.4.5 evaluate the selection accuracy of introduced penalisation methods for multivariate regression model. In both scenarios with high and low correlations among response variables, the multivariate sparse group lasso (MSGL) outperforms all other penalisation methods. The reason is that this penalty possesses the nice property of within group selection as well as the grouping feature which results in sparsity at group and within group level. Thus, this property allows for some zero coefficients inside the groups which leads to a more accurate selection.

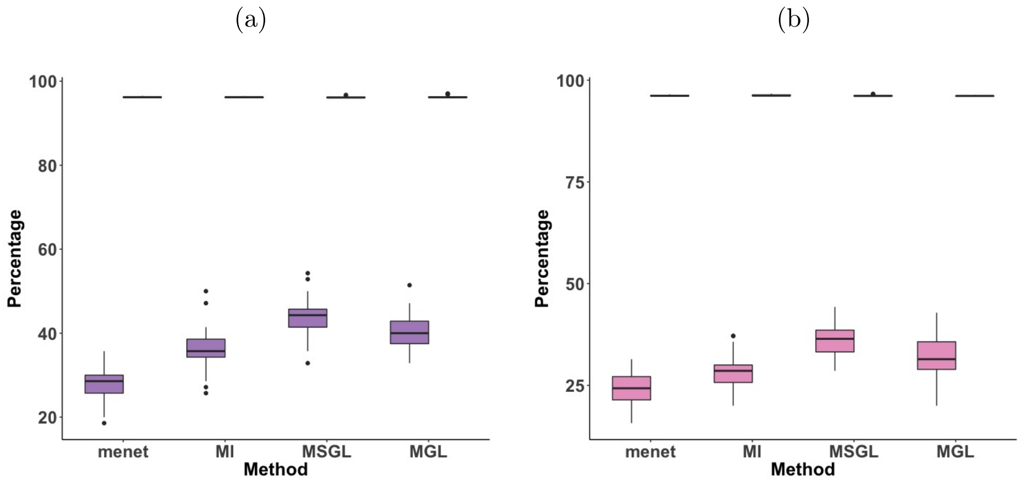


Figure 2.4.5: Box plots of sensitivity values where the specificity is fixed. Results obtained from 50 simulations where $p = 2000, n = 88, J = 20, |T| = 70$ with (a) high correlations, (b) low correlations among response variables. From the left, methods are multivariate elastic net (menet) multivariate lasso (ML), multivariate sparse group lasso (MSGL), multivariate group lasso (MGL).

2.5 Finite mixture models

In applied statistical modelings, data under investigation often have an unobservable group structure. So it is reasonable to partition data into groups. For example, in medicine, it is often desired to categorise diseases that have the same treatment. Also in cancer biology, grouping those mutations that cause the same type of cancer plays a significant role in enhancing the treatment.

In this situations, we may need finite mixture models. These models are of great interest in many areas of science where one wants to uncover the latent group structure in the data. Flexibility and the ability to capture unobserved heterogeneity in data are crucial aspects of mixture models that mark them as one of the appropriate methods for statistical modelling.

A finite mixture model is a weighted sum or a convex combination of a finite number of densities. These densities may have different sets of parameters. In finite mixture models we assume that observations come from these densities with certain probabilities, respectively.

Let $\mathbf{x}_1, \cdots, \mathbf{x}_n$ be a random sample of random variable X where each $\mathbf{x}_i$ is a p-dimensional vector. We suppose that there is a group structure in these data but there is no information available about the group index of each $\mathbf{x}_i$. If we assume that there are K different groups in the data then a probability density function of a mixture model with parameter set $\Phi = (\boldsymbol{\theta}, \boldsymbol{\pi})$ is defined by the following combination of K densities

$$f(\mathbf{x}_i; \Phi) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{x}_i; \theta_k), \tag{2.5.1}$$

where $f_k(\mathbf{x}_i|\theta_k)$ is the density corresponding to the component k and $(\boldsymbol{\theta}, \boldsymbol{\pi}) = (\theta_1, \cdots, \theta_K, \pi_1, \cdots, \pi_K)$. Parameters π_ks are mixing proportions with the following property

$$0 \leq \pi_k \leq 1, \ \sum_{k=1}^{K} \pi_k = 1, \quad k = 1, \cdots, K.$$

Note that each $f_k(\mathbf{x}_i; \theta)$ is a density function, hence, the Equation (2.5.1) defines a probability density function (McLachlan and Basford, 1988). In cases where component densities are Gaussian, the model is known as Gaussian mixture models (GMM). Finite mixture of regression model introduced by Quandt (1975) is a widely applied model which is a special case of the GMM. These models are defined as follows.

Assume that we are interested in explaining the relationship between a univariate response variable Y and a p-dimensional vector of predictors $\mathbf{x}$. Also assume that n independent observations on Y and $\mathbf{x}$ are denoted by $y_1, \cdots, y_n$ and $\mathbf{x}_1, \cdots, \mathbf{x}_n$ with $\mathbf{x}_i = (\mathbf{x}_{i1}, \cdots, \mathbf{x}_{ip})$ for $i = 1, \cdots, n$. Suppose the dependency of Y and $\mathbf{x}$ is expressed through the simple regression model

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2).$$

In this framework it is assumed that the vector of regression coefficients $\boldsymbol{\beta}$ is fixed for all observations. This assumption can be considered as a drawback of regression models since it leads to the ignorance of unobserved heterogeneity in data. More precisely, in the existence of a group-structure in data, regression coefficients may not be the same for all response variables. The extent to which explanatory variables are associated with response variable may vary across different observations of response variable and this induces a group-structure in data (Frühwirth-Schnatter, 2006). A more suitable alternative to the linear regression model is a finite mixture of regressions model. These models possess a combination of properties of mixture models as well as regression models.

Consider the pair of observations $(y_i, \mathbf{x}_i)$ on variables $(Y, \mathbf{x})$ introduced above. If the response variable follows a mixture distribution then a mixture of regressions is obtained. Accordingly, within the scope of linear regression models, the linkage between the response variable and predictors can be expressed by finite mixture of regressions. In this framework, the conditional distribution of Y given $\mathbf{x}$ is a mixture model. This implies that each pair of $(y_i, \mathbf{x}_i)$ belongs to one of the K components in the mixture model. Given that observation $(y_i, \mathbf{x}_i)$ comes from component k, the following regression holds

$$y_i = \mathbf{x}_i\boldsymbol{\beta}_k + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma_k^2),$$

Accordingly a mixture of regression models with K components is formulated through the following conditional distribution

$$f(y_i|\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i\boldsymbol{\beta}_k, \sigma_k^2), \tag{2.5.2}$$

where each $f_k(.|\mathbf{x}_i\boldsymbol{\beta}_k, \sigma_k^2)$ is the normal density with mean $\mathbf{x}_i\boldsymbol{\beta}_k$ and variance σ^2. As a result of the group structure in the response variable, the coefficients and the error terms in regression model (2.5.2) are not homogeneous across all observations $i = 1, \cdots, n$. Yet, these parameters are homogeneous within each group or component (Faria and Soromenho, 2010).

In a mixture model, the parameter set $\Phi = (\boldsymbol{\theta}, \boldsymbol{\pi})$ is unknown. A mixture model is specified by estimating the unknown parameters of the model. Hence, the objective in fitting mixture models is to calculate the maximum likelihood estimates of model parameters and use these estimates in calculating the probability of membership for each object. Assuming that $\mathbf{x}_1, \cdots, \mathbf{x}_n$ are independently distributed, the likelihood function corresponding to the mixture model (2.5.1) is given by

$$\begin{aligned} L(\Phi) &= \prod_{i=1}^{n} f(\mathbf{x}_i; \Phi) \\ &= \prod_{i=1}^{n} \left(\sum_{k=1}^{K} \pi_k f_k(\mathbf{x}_i; \theta_k) \right), \end{aligned} \tag{2.5.3}$$

The above likelihood function is now maximised with respect to all model parameters. Taking derivatives of $L(\Phi)$ with respect to all model parameters gives the maximum likelihood estimators. Therefore, we need to find the solution to equation

$$\frac{\partial L(\Phi)}{\partial \Phi} = 0. \tag{2.5.4}$$

It is more straightforward to maximise the logarithm of this likelihood function due to useful mathematical features that logarithm possesses. Since logarithm is a monotonically increasing function the following equality holds

$$\arg\max_{\phi \in \Phi} L(\Phi) = \arg\max_{\phi \in \Phi} l(\Phi).$$

Now the log-likelihood function corresponding to (2.5.3) is of the form

$$\begin{aligned} l(\Phi) &= \log\left[\prod_{i=1}^{n}\left(\sum_{k=1}^{K}\pi_k f_k(\mathbf{x}_i;\theta_k)\right)\right] \\ &= \sum_{i=1}^{n}\log\sum_{k=1}^{K}\pi_k f_k(\mathbf{x}_i;\theta_k). \qquad (2.5.5) \end{aligned}$$

Therefore, the maximum likelihood estimates can be obtained by solving the equation

$$\frac{\partial l(\Phi)}{\partial \Phi} = 0. \qquad (2.5.6)$$

Maximising the above log-likelihood is not feasible due to the appearance of the summation in this function. This summation is not decomposable as we do not have any information about the group from which each observation is drawn. This means the data contain unobservable or missing variables. In such instances, the most common and powerful approach for estimating the maximum likelihood estimates of model parameters is the Expectation-Maximisation algorithm which is introduced in the next part.

2.5.1 Expectation-Maximisation (EM) algorithm

As pointed out earlier, in the mixture models framework it is assumed that the data arise from K different components, however, in reality the component label of each observation is unknown. Therefore the vector of component labels which provides us with some information about the origin group of each observation is regarded as unobserved or latent part of data. To find the maximum likelihood estimates for such incomplete data, Expectation-Maximisation (EM) algorithm (Dempster et al., 1977) is utilised. This iterative algorithm alters between two steps in each iteration. The Expectation step or E-step and the Maximisation step or M-step. The idea is to augment the incomplete data with an auxiliary variable and calculate the expectation of the incomplete log-likelihood.

EM algorithm starts by augmenting the incomplete data with the hidden variable which is the unobserved component indicator. Let X denote the incomplete data and Z specifies the hidden indicator. Let vector $\mathbf{Z} = (\mathbf{z}_1, \cdots, \mathbf{z}_n)$ denote n observations of indicator variable where $\mathbf{z}_i = (z_{i1}, \cdots, z_{ik})$ $i = 1, \cdots, n$. z_{ik} is a Bernoulli random variable which indicates the component origin of each observation

$$z_{ik} = \begin{cases} 1 \;\; \mathbf{x}_i \in C_k \\ 0 \;\; \mathbf{x}_i \notin C_k. \end{cases} \tag{2.5.7}$$

This means that if $\mathbf{x}_i$ comes from component C_k, then $z_{jk} = 1$ and it is zero otherwise. Random variables $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n$ are independently and identically distributed according to multinomial distribution $Mult_K(1, \boldsymbol{\pi})$ consisting of one draw on k categories with corresponding probabilities $\pi_1, \pi_2, \cdots, \pi_K$ (McLachlan and Krishnan, 2007)

$$\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n \overset{i.i.d}{\sim} Mult_K(1, \boldsymbol{\pi}).$$

Having completed the data, the likelihood is constructed as follows

$$L_c(\Phi) = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{z_{ik}} f_k(\mathbf{x}_i|\boldsymbol{\theta}_k)^{z_{ik}},$$

In practice, the above complete-data likelihood is not observable, hence in the E-step of the algorithm, $\log L_c(\Phi)$ is replaced by its expectation. More precisely, in the m-th iteration, the conditional expectation of the complete-data log-likelihood given the observed data and the current estimates of Φ is calculated

$$\begin{aligned} Q(\Phi|\Phi^{(m)}) &= E\left[\log L_c(\Phi)|X, \Phi^{(m)})\right] \qquad (2.5.8) \\ &= E\left(\sum_{i=1}^{n}\sum_{k=1}^{K}\left[z_{ik}\log \pi_k + z_{ik}\log f_k(\mathbf{x}_i, \boldsymbol{\theta}_k)\right]\right) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} E(z_{ik})\log \pi_k + E(z_{ik})\log f_k(\mathbf{x}_i, \boldsymbol{\theta}_k) \end{aligned}$$

In the above expression, the information about the latent part of the data is specified by posterior of latent variables given the observed data and the current estimates of Φ. This posterior is is calculated as

$$\begin{aligned} E(z_{ik}|\mathbf{x}_i, \Phi^{(m)}) &= w_{ik}^{(m)}(\mathbf{x}_i, \Phi^{(m)}) \\ &= \frac{\pi_k^{(m)} f_k(\mathbf{x}_i; \theta_k^{(m)})}{\sum_{t=1}^{K} \pi_t^{(m)} f_t(\mathbf{x}_i; \theta_t^{(m)})}. \end{aligned} \tag{2.5.9}$$

In the M-step of the EM algorithm, the expectation (2.5.8) is maximised with respect to $\Phi^{(m)}$. More precisely, we find $\Phi^{(m+1)}$ such that

$$\Phi^{(m+1)} = \arg\max_{\Phi} Q(\Phi|\Phi^{(m)})$$

or equivalently

$$Q(\Phi|\Phi^{(m)}) \leq Q(\Phi^{(m+1)}|\Phi^{(m)}). \tag{2.5.10}$$

These steps are alternated iteratively until the log-likelihood function converges. In other words for some $\epsilon > 0$

$$\left|L(\Phi^{(m+1)}) - L(\Phi^{(m)})\right| < \epsilon. \tag{2.5.11}$$

Dempster et al. (1977) showed a nice feature of EM algorithm called *monotonicity* of the EM algorithm which means the incomplete likelihood $L(\boldsymbol{\theta})$ is not decreased after each iteration. That is

$$L(\Phi^{(m+1)}) \geq L(\Phi^{(m)}). \tag{2.5.12}$$

Once EM algorithm has converged, parameter estimates obtained in the last iteration $\Phi^{(m+1)}$ give the set of maximum likelihood estimates. These estimates consist of the component parameters and the mixing proportions corresponding to each component. The posterior (2.5.9) also gives the optimal classification of the data. Often we wish to compare the obtained classifications with the true one or with other classifications resulted from other methods. In the next section we introduce one of the commonly used criterion to validate and compare the classification accuracy.

2.5.2 Adjusted Rand index

Rapid technological advancements have led to the emergence of large databases in many areas in science. Cluster analysis is one practical approach to tackle the inevitable hurdles such databases bring about. The aim of cluster analysis is revealing groups in the data. This technique seeks classifying or partitioning data objects in some way that the objects within each group or cluster are similar and share common characteristics whereas, they are different from objects in other groups. As a result of representing the data in terms of a smaller set of groups, describing the data becomes feasible, and the information can be extracted more efficiently (Everitt et al., 2011). Most of the time we are interested in comparing two classifications or partitions of the same data set. For example, we wish to measure the agreement between two partitions resulted from applying two different methods to the data. Here we review one of the widely used methods that we use in this thesis to compare two different classifications.

The Rand index attributed to Rand (1971) is a measure by which a classification is evaluated. Rand index quantifies the degree of agreement between two partitions or classifications based on the class labels of objects. Suppose we have n objects to classify and $P_1 = \{C_1, \cdots, C_r\}$ is a partition that assigns these objects into r classes and $P_2 = \{C_1, \cdots, C_s\}$ assigns them into s classes. Each pair of objects, either have the same class label or a different one. Since the number of classified objects is n, we have the total number of $n(n-1)/2$ pairs to compare. Let a be the number of pairs that the two partitions agree by assigning the elements to the same classes and b be the number of pairs that partitions agree by assigning them to different classes. Considering all pairs, the proportion of agreement between P_1 and P_2 is evaluated by the following

Rand index (RI)

$$\mathrm{RI}(P_1, P_2) = \frac{a+b}{n(n-1)/2} \tag{2.5.13}$$

Since the expectation of Rand index for two random partitions is not a constant, Hubert and Arabie (1985) proposed a normalised Rand index to provide a more appropriate measure. This measure which is called Adjusted Rand Index (ARI) is the corrected Rand index for chance and is defined as follows.

$$\mathrm{ARI} = \frac{\text{Rand index} - \text{Expected value of Rand index}}{\text{Maximum value of Rand index} - \text{Expected value of Rand index}}.$$

The adjusted Rand index assumes the generalised hyper geometric distribution as a model of randomness (Dhaenens and Jourdan, 2016). Let n_{ij} be the number of elements which happen to be in cluster i in partition P_1 and in cluster j in P_2. Then the adjusted Rand index is formulated by

$$\mathrm{ARI}(P_1, P_2) = \frac{R - E[R]}{M[R] - E[R]}, \quad R = \sum_{ij} \binom{2}{n_{ij}}, \tag{2.5.14}$$

where the expected and the maximum value of Rand index are defined as

$$\begin{aligned} E[R] &= \left[\sum_i \binom{2}{n_{i.}} \sum_j \binom{2}{n_{.j}} \right] \bigg/ \binom{n}{2} \\ M[R] &= \frac{\left[\sum_i \binom{2}{n_{i.}} + \sum_j \binom{2}{n_{.j}} \right]}{2}. \end{aligned}$$

In the above expressions, $n_{i.}$ denotes the number of elements in ith cluster of partition P_1 and $n_{.j}$ is the number of elements in cluster j in partition P_2. When two partitions completely agree, the adjusted Rand index is 1 which is the maximum value for this index. Higher values of ARI indicates greater degree of agreement between two partitions (Xu et al., 2005).

2.6 Bayesian inference of mixture models

In this part we review finite mixture models in a Bayesian framework. One appealing aspect of Bayesian approaches to mixture modelling is that they allow for prior information or expert opinion to be combined with the data. Another aspect is that, probability statements can be made about the unknown parameters. In particular, the posterior provide a convenient device to infer the number of components when it is unknown.

Suppose that we have N observations $y_1, \cdots, y_N$, on random variable Y which comes from a population with K groups. Also suppose we do not have any information about the group of origin that each observation comes from. This means we are dealing with a finite mixture model which was introduced earlier. If the parameters of the mixture model is shown by vector $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K, \boldsymbol{\eta})$ then the mixture density function is given by

$$p(y_i|\boldsymbol{\vartheta}) = \sum_{k=1}^{K} \eta_k p(y_i|\boldsymbol{\theta}_k),$$

As it was pointed out earlier in Section 2.5, in order to fit the above mixture model, all parameters in the model need to be estimated from the data. Under a Bayesian paradigm, the unknown parameters are treated as random variables and described with a probability distribution which is called prior. In other words, this is the distribution before having observed the data and is based on previous experiment (McLachlan and Peel, 2004).

In Bayesian inference, the main task is finding the distribution of the parameter after having observed the data which is the posterior distribution of the parameter. Posterior distribution is obtained through Bayes' theorem where the prior information about the parameter of interest is combined with the data

$$p(\boldsymbol{\vartheta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta}), \tag{2.6.1}$$

where $\mathbf{y} = (y_1, \cdots, y_N)$. Therefore, from a Bayesian point of view, all the knowledge contained in the data about the unknown parameters is specified through the posterior distribution of parameters (Dey and Rao, 2005).

Note that similar to what was discussed in frequentist inference of mixture models in Section 2.5, the group labels are the missing part of the data. Following Frühwirth-Schnatter (2006), let $\mathbf{S} = (S_1, \cdots, S_N)$ be the allocation vector which is missing and S_i denote the group label corresponding to the i-th observation. The posterior probability of membership in group k for the i-th observation, $Pr(S_i = k|\mathbf{y}_i, \boldsymbol{\vartheta})$, is obtained by utilising Bayes' rule (Bayes and Price, 1763) as follows

$$Pr(S_i = k|\mathbf{y}_i, \boldsymbol{\vartheta}) = \frac{Pr(Y = y_i|S_i = k, \boldsymbol{\vartheta})Pr(S_i = k|\boldsymbol{\vartheta})}{\sum_{j=1}^{K} Pr(Y = y_i|S_i = j, \boldsymbol{\vartheta})Pr(S_i = j|\boldsymbol{\vartheta})}, \tag{2.6.2}$$

where $Pr(S_i = k|\boldsymbol{\vartheta})$ is the prior probability that the i-th observation is drawn from the k-th group. As $Pr(S_i = k|\boldsymbol{\vartheta}) = \eta_k$, the posterior probability (2.6.2) is equivalently rewritten as

$$Pr(S_i = k|y_i, \boldsymbol{\vartheta}) = \frac{p(y_i|\boldsymbol{\theta}_k)\eta_k}{\sum_{j=1}^{K} p(y_i|\boldsymbol{\theta}_j)\eta_j}. \tag{2.6.3}$$

The denominator does not change as k changes therefore it is common to express Bayes' rule proportionality as follows

$$Pr(S_i = k|y_i, \boldsymbol{\vartheta}) \propto p(y_i|\boldsymbol{\theta}_k)\eta_k \tag{2.6.4}$$

The incomplete-data likelihood corresponding to the mixture model under the assumption that the data are sampled independently is given by

$$p(\mathbf{y}|\mathbf{S}, \boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{S}, \boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K) = \prod_{i=1}^{N} p(y_i|\boldsymbol{\theta}_{S_i}), \tag{2.6.5}$$

Suppose we denote the complete data as $(\mathbf{y}, \mathbf{S})$, then the complete-data likelihood is defined by

$$p(\mathbf{y}, \mathbf{S}|\boldsymbol{\vartheta}) = p(\mathbf{y}|\mathbf{S}, \boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\vartheta}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{\vartheta}, S_i)p(S_i|\boldsymbol{\vartheta}). \tag{2.6.6}$$

Since $p(y_i|S_i = k, \boldsymbol{\vartheta}) = p(y_i|\boldsymbol{\theta}_k)$ and $Pr(S_i = k|\boldsymbol{\vartheta}) = \eta_k$ therefore the complete-data likelihood function can be rewritten as

$$\begin{aligned} p(\mathbf{y}, \mathbf{S}|\boldsymbol{\vartheta}) &= \prod_{i=1}^{N}\prod_{k=1}^{K}(p(y_i|\boldsymbol{\theta}_k)\eta_k)^{I_{S_i=k}} \\ &= \prod_{k=1}^{K}\left(\prod_{i:S_i=k} p(y_i|\boldsymbol{\theta}_k)\right)\left(\prod_{k=1}^{K}\eta_k^{N_k(\mathbf{S})}\right) \end{aligned} \tag{2.6.7}$$

where $N_k(\mathbf{S})$ is the number of observations in group k. We also denote the mean and the variance of the k-th group as

$$\begin{aligned} \bar{y}_k(\mathbf{S}) &= \frac{1}{N_k(\mathbf{S})}\sum_{i:S_i=k} y_i \\ s_{y,k}^2 &= \frac{1}{N_k(\mathbf{S})}\sum_{i:S_i=k}(y_i - \bar{y}_k(\mathbf{S}))^2 \end{aligned} \tag{2.6.8}$$

2.6.1 Mixture of univariate normals

In this section we consider the mixture model when component densities are gaussian and we derive the posterior distributions for parameters corresponding to each component density. Posterior distributions for σ_k^2 and μ_k are calculated given the complete data $(\mathbf{y}, \mathbf{S})$. We first derive the posterior distribution of the mean for each component. In the following we use the same notation as Frühwirth-Schnatter (2006).

Suppose the group label for observation y_i is k and each observation is normally distributed, $y_i \sim N(\mu_k, \sigma_k^2), i = 1, \cdots, N$, and the vector of parameters corresponding to component k, is $\boldsymbol{\theta}_k = (\mu_k, \sigma_k^2)$. Then the following hierarchical model is considered

$$\begin{aligned} y_i &\sim N(\mu_k, \sigma_k^2), \\ \mu_k|\sigma_k^2 &\sim N(\mu_{k0}, \sigma_{k0}^2), \\ \sigma_k^2 &\sim IG(a_0, b_0). \end{aligned}$$

The priors in the above model are priors that are suggested by Frühwirth-Schnatter (2006). For observation y_i in component k the probability density function is of the form

$$p(y_i|\mu_k, \sigma_k^2) = (\frac{1}{2\pi\sigma_k^2})^{1/2} \exp\left\{-\frac{1}{2\sigma_k^2}(y_i - \mu_k)^2\right\},$$

therefore, the complete-data likelihood function is given by

$$p(\mathbf{y}|\mu_k, \sigma^2, \mathbf{S}) = \prod_{k=1}^{K} \prod_{i:S_i=k} (\frac{1}{2\pi\sigma_k^2})^{-N_k(\mathbf{S})/2} \exp\left\{-\frac{1}{2\sigma_k^2} \sum_{i:S_i=k} (y_i - \mu_k)^2\right\}$$

According to the Bayes' rule, the posterior probability of μ_k given the complete data $(\mathbf{S}, \mathbf{y})$ is given by

$$p(\mu_k|\mathbf{y}, \mathbf{S}, \sigma_k^2) \propto p(\mathbf{y}|\mu_k, \sigma^2, \mathbf{S})p(\mu_k),$$

when the variance σ^2 is fixed, the posterior is obtained by

$$\begin{aligned} p(\mu_k|\mathbf{y}, \mathbf{S}, \sigma_k^2) &\propto \prod_{k=1}^{K} \prod_{i:S_i=k} (\frac{1}{\sigma_k^2})^{-N_k(\mathbf{S})/2} \exp\left\{-\frac{1}{2\sigma_k^2} \sum_{i:S_i=k} (y_i - \mu_k)^2\right\} \\ &\times (\frac{1}{\sigma_{k0}^2})^{1/2} \exp\left\{-\frac{1}{2\sigma_{k0}^2}(\mu_k - \mu_{k0})^2\right\}, \end{aligned}$$

by doing simple algebra we get

$$p(\mu_k|\mathbf{y}, \mathbf{S}, \sigma_k^2) \;\; \propto \;\; \exp\left\{-\frac{1}{2}\left(\frac{N_k(\mathbf{S})}{\sigma_k^2} + \frac{1}{\sigma_{k0}^2}\right)\left(\mu_k - \frac{\sum y_i}{\sigma_k^2} + \frac{\mu_{k0}}{\sigma_{k0}^2}\right)^2\right\},$$

therefore the obtained posterior is the kernel of a normal distribution with the variance of $\left(\frac{N_k(\mathbf{S})}{\sigma_k^2} + \frac{1}{\sigma_{k0}^2}\right)^{-1} = \frac{N_k(\mathbf{S})\sigma_{k0}^2 + \sigma_k^2}{\sigma_k^2\sigma_{k0}^2}$ and with the mean $\frac{N_k(\mathbf{S})\bar{y}_k\sigma_{k0}^2 + \sigma_k^2\mu_{k0}}{\sigma_k^2\sigma_{k0}^2}$.

Thus the posterior distribution of μ_k is the following normal distribution

$$\begin{aligned} p(\mu_k|\mathbf{y},\mathbf{S},\sigma_k^2) &\sim \mathcal{N}(b_k(\mathbf{S}),B_k(\mathbf{S})), \qquad (2.6.9)\\ B_k(\mathbf{S})^{-1} &= \sigma_{k0}^{-2}+\sigma_k^{-2}N_k(\mathbf{S})\\ b_k(\mathbf{S}) &= B_k(\mathbf{S})\left(\sigma_k^{-2}N_k(\mathbf{S})\bar{y}_k(\mathbf{S})+\sigma_{k0}^{-2}\mu_{k0}\right). \end{aligned}$$

Now we find the posterior distribution of σ_k^2 which according to Bayes' rule is defined

$$p(\sigma_k|\mathbf{y},\mathbf{S},\mu_k)\propto p(\mathbf{y}|\mu_k,\sigma^2,\mathbf{S})p(\sigma_k^2),$$

hence by fixing μ_k, under the conjugate Inverse Gamma prior introduced above we obtain

$$\begin{aligned} p(\sigma_k|\mathbf{y},\mathbf{S},\mu_k) &\propto \prod_{k=1}^{K}\prod_{i:S_i=k}(\frac{1}{\sigma_k^2})^{-N_k(\mathbf{S})/2}\exp\left\{-\frac{1}{2\sigma_k^2}\sum_{i:S_i=k}(y_i-\mu_k)^2\right\}\\ &\times (\sigma_k^2)^{-a_0-1}\exp\left\{-b_0/\sigma_k^2\right\}, \end{aligned}$$

this posterior is also an Inverse Gamma with

$$\begin{aligned} p(\sigma_k|\mathbf{y},\mathbf{S},\mu_k) &\sim \mathcal{G}^{-1}(c_k(\mathbf{S}),C_k(\mathbf{S})), \qquad (2.6.10)\\ c_k(\mathbf{S}) &= a_0+\frac{1}{2}N_k(\mathbf{S})\\ C_k(\mathbf{S}) &= b_0+\frac{1}{2}\sum_{i:S_i=k}(y_i-\mu_k)^2. \end{aligned}$$

2.6.2 Mixture of multivariate normals

Now suppose we have N multivariate observations that follow a multivariate normal distribution. Let $\mathbf{y}=(\mathbf{y}_1,\cdots,\mathbf{y}_N)$ denote a set of observations where each $\mathbf{y}_i$ is a d-dimensional vector. Also assume $\mathbf{y}_i\sim\mathcal{N}_d(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$. In this section also we follow Frühwirth-Schnatter (2006) and consider the following conjugate priors for the parameters of normal densities

$$\boldsymbol{\mu}_k \sim \mathcal{N}_d(\mathbf{b}_0, \mathbf{B}_0),$$
$$\boldsymbol{\Sigma}_k^{-1} \sim \mathcal{W}_r(c_0, \mathbf{C}_0),$$

where $\mathcal{W}_d(c, \mathbf{C})$ denotes the Wishart distribution. Let $\mathbf{V}_{d\times d}$ is a symmetric, positive definite matrix which follows a Wishart distribution. Then according to McLachlan and Peel (2004), for $c > (d-1)/2$, the density function of $\mathbf{V}$ is given by

$$\mathcal{W}_d(c, \mathbf{C}) = \frac{|\mathbf{C}|^c}{\Gamma_d(c)} |\mathbf{V}|^{c-(d+1)/2} \exp\left\{-\text{tr }(\mathbf{CV})\right\},$$

where

$$\Gamma_d(c) = \pi^{d(d-1)/4} \prod_{j=1}^{d} \Gamma\left(\frac{2c+1-j}{2}\right)$$

The aim is finding the posterior distribution of $\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k$ given the complete data, $\mathbf{S}, \mathbf{y}$, which is derived by applying the Bayes' rule as before and combining the prior information and the information from all observations which belong to group k. Suppose observation $\mathbf{y}_i$ belongs to the k-th component, then the density function for this observation is of the form

$$p(\mathbf{y}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-d/2} |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_k)\right\}.$$

Let $N_k(\mathbf{S})$ denote the number of observations in component k, then, for each component the sample mean $\bar{\mathbf{y}}_k(\mathbf{S})$ is defined as follows

$$\bar{\mathbf{y}}_k(\mathbf{S}) \quad = \quad \frac{1}{N_k(\mathbf{S})} \sum_{i:S_i=k} \mathbf{y}_i$$

The complete-data likelihood function is defined by

$$\begin{aligned} p(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{S}) &= \prod_{k=1}^{K} \prod_{i:S_i=k} p(\mathbf{y}_i|\boldsymbol{\theta}_k) \qquad (2.6.11) \\ &\propto \prod_{k=1}^{K} |\boldsymbol{\Sigma}_k|^{-N_k(\mathbf{S})/2} \exp\left\{-\frac{1}{2} \sum_{i:S_i=k} (\mathbf{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_k)\right\} \end{aligned}$$

We start by finding the posterior of $\boldsymbol{\mu}_k$, when holding the variance covariance matrix $\boldsymbol{\Sigma}_k$ fixed. Choosing the conjugate prior $\boldsymbol{\mu}_k \sim \mathcal{N}_d(\mathbf{b}_0, \mathbf{B}_0)$, the posterior density of $\boldsymbol{\mu}_k$ given $\boldsymbol{\Sigma}_k$ and all observations in the k-th component is given by

$$\begin{aligned} p(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k, \mathbf{S}, \mathbf{y}) &\propto \prod_{k=1}^{K} |\boldsymbol{\Sigma}_k|^{-N_k(\mathbf{S})/2} \exp\left\{-\frac{1}{2} \sum_{i:S_i=k} (\mathbf{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_k)\right\} \\ &\times |\mathbf{B}_0|^{-1/2} \exp\left\{-\frac{1}{2}(\boldsymbol{\mu}_k - \mathbf{b}_0)^T \mathbf{B}_0^{-1} (\boldsymbol{\mu}_k - \mathbf{b}_0)\right\} \\ &\propto \prod_{k=1}^{K} |\boldsymbol{\Sigma}_k|^{-N_k(\mathbf{S})/2} \\ &\times \exp\left\{-\frac{1}{2}\left(\sum_{i:S_i=k} (\mathbf{y}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_k) + (\boldsymbol{\mu}_k - \mathbf{b}_0)^T \mathbf{B}_0^{-1} (\boldsymbol{\mu}_k - \mathbf{b}_0)\right)\right\} \end{aligned}$$

simplifying the bracket inside the exponential function in the last expression, we get

$$\begin{aligned} &\sum_{i:S_i=k} (\mathbf{y}_i^T \boldsymbol{\Sigma}_k^{-1} \mathbf{y}_i) - 2\bar{\mathbf{y}} N_k(\mathbf{S}) \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k + N_k(\mathbf{S}) \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k \\ &+ \boldsymbol{\mu}_k^T \mathbf{B}_0^{-1} \boldsymbol{\mu}_k - 2\boldsymbol{\mu}_k^T \mathbf{B}_0^{-1} \mathbf{b}_0 + \mathbf{b}_0^T \mathbf{B}_0^{-1} \mathbf{b}_0 \\ &= \boldsymbol{\mu}_k^T \left(N_k(\mathbf{S}) \boldsymbol{\Sigma}_k^{-1} + \mathbf{B}_0^{-1}\right) \boldsymbol{\mu}_k - 2\boldsymbol{\mu}_k^T (\boldsymbol{\Sigma}_k^{-1} N_k(\mathbf{S}) \bar{\mathbf{y}} + \mathbf{B}_0^{-1} \mathbf{b}_0) \\ &+ \sum_{i:S_i=k} (\mathbf{y}_i^T \boldsymbol{\Sigma}_k^{-1} \mathbf{y}_i) + \mathbf{b}_0^T \mathbf{B}_0^{-1} \mathbf{b}_0 \end{aligned}$$

rearranging the above expression, we obtain the following expression which is the kernel of a multivariate normal distribution

$$\begin{aligned} &\left[\boldsymbol{\mu}_k - (\boldsymbol{\Sigma}_k^{-1} N_k(\mathbf{S}) \bar{\mathbf{y}} + \mathbf{B}_0^{-1} \mathbf{b}_0) \left(N_k(\mathbf{S}) \boldsymbol{\Sigma}_k^{-1} + \mathbf{B}_0^{-1}\right)^{-1}\right]^T \left(N_k(\mathbf{S}) \boldsymbol{\Sigma}_k^{-1} + \mathbf{B}_0^{-1}\right)^{-1} \\ &\left[\boldsymbol{\mu}_k - (\boldsymbol{\Sigma}_k^{-1} N_k(\mathbf{S}) \bar{\mathbf{y}} + \mathbf{B}_0^{-1} \mathbf{b}_0) \left(N_k(\mathbf{S}) \boldsymbol{\Sigma}_k^{-1} + \mathbf{B}_0^{-1}\right)^{-1}\right] \end{aligned}$$

Therefore the posterior of $\boldsymbol{\mu}_k$ is again a multivariate normal distribution i.e.

$$\begin{aligned}\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k, \mathbf{S}, \mathbf{y} &\sim \mathcal{N}_d(\mathbf{b}_k(\mathbf{S}), \mathbf{B}_k(\mathbf{S})) \qquad (2.6.12)\\ \mathbf{B}_k(\mathbf{S}) &= \left(N_k(\mathbf{S})\boldsymbol{\Sigma}_k^{-1} + \mathbf{B}_0^{-1}\right)^{-1}\\ \mathbf{b}_k(\mathbf{S}) &= \mathbf{B}_k(\mathbf{S})\left(\boldsymbol{\Sigma}_k^{-1}N_k(\mathbf{S})\bar{\mathbf{y}}_k(\mathbf{S}) + \mathbf{B}_0^{-1}\mathbf{b}_0\right).\end{aligned}$$

Suppose that $\boldsymbol{\mu}_k$ is fixed. Considering the conjugate Wishart prior $\boldsymbol{\Sigma}_k^{-1} \sim \mathcal{W}_d(c_0, \mathbf{C}_0)$, we now find the posterior of $\boldsymbol{\Sigma}_k^{-1}$ given $\boldsymbol{\mu}_k$ and all observations in the k-th component as

$$\begin{aligned}p(\boldsymbol{\Sigma}_k^{-1}|\boldsymbol{\mu}_k, \mathbf{S}, \mathbf{y}) &\propto \prod_{k=1}^{K} |\boldsymbol{\Sigma}_k|^{-N_k(\mathbf{S})/2} \exp\left\{-\frac{1}{2}\sum_{i:S_i=k}(\mathbf{y}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_k)\right\}\\ &\times |\boldsymbol{\Sigma}_k|^{c_0-(d+1)/2} \exp\left\{-\text{tr } (\mathbf{C}_0\boldsymbol{\Sigma}_k^{-1})\right\}\\ &\propto \prod_{k=1}^{K} |\boldsymbol{\Sigma}_k|^{-N_k(\mathbf{S})/2+c_0-(d+1)/2}\\ &\times \exp\left\{-\frac{1}{2}\sum_{i:S_i=k}(\mathbf{y}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_k) - \text{tr } (\mathbf{C}_0\boldsymbol{\Sigma}_k^{-1})\right\}\\ &\propto \prod_{k=1}^{K} |\boldsymbol{\Sigma}_k|^{-N_k(\mathbf{S})/2+c_0-(d+1)/2}\\ &\times \exp\left\{-\text{tr } \left(\frac{1}{2}\sum_{i:S_i=k}(\mathbf{y}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_k)\right)\right.\\ &- \left.\text{tr } (\mathbf{C}_0\boldsymbol{\Sigma}_k^{-1})\right\}\\ &\propto \prod_{k=1}^{K} |\boldsymbol{\Sigma}_k|^{-N_k(\mathbf{S})/2+c_0-(d+1)/2}\\ &\times \exp\left\{-\text{tr } \left(\frac{1}{2}\sum_{i:S_i=k}(\mathbf{y}_i - \boldsymbol{\mu}_k)(\mathbf{y}_i - \boldsymbol{\mu}_k)^T + \mathbf{C}_0\right)\boldsymbol{\Sigma}_k^{-1}\right\}.\end{aligned}$$

The latter expression is the kernel of the Wishart distribution. Therefore, the posterior distribution of $\boldsymbol{\Sigma}_k^{-1}$ is the Whishart distribution

$$\begin{aligned}\boldsymbol{\Sigma}_k^{-1}|\boldsymbol{\mu}_k, \mathbf{S}, \mathbf{y} &\sim \mathcal{W}_d(c_k(\mathbf{S}), \mathbf{C}_k(\mathbf{S})) \qquad (2.6.13)\\ c_k(\mathbf{S}) &= c_0 + N_k(\mathbf{S})/2\\ \mathbf{C}_k(\mathbf{S}) &= \mathbf{C}_0 + \frac{1}{2}\sum_{i:S_i=k}(\mathbf{y}_i - \boldsymbol{\mu}_k)(\mathbf{y}_i - \boldsymbol{\mu}_k)^T.\end{aligned}$$

2.6.3 Estimation of mixture of normals through Markov chain Monte Carlo (MCMC) methods

As mentioned earlier, a mixture model can be regarded as an incomplete data problem where the allocation vector $\mathbf{S}$ is the missing part (Dempster et al., 1977). Therefore, Bayesian estimation of a mixture model is performed through data augmentation where augmented parameters $(\mathbf{S}, \boldsymbol{\vartheta})$ are estimated by sampling from the complete-data posterior distribution $p(\mathbf{S}, \boldsymbol{\vartheta}|\mathbf{y})$ which is given by

$$p(\mathbf{S}, \boldsymbol{\vartheta}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{S}, \boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta}). \tag{2.6.14}$$

Sampling from the above posterior is carried out through MCMC methods where $\boldsymbol{\vartheta}$ is sampled given the allocation $\mathbf{S}$ and allocation vector $\mathbf{S}$ is sampled given $\boldsymbol{\vartheta}$ (Frühwirth-Schnatter, 2006). In order to estimate the parameters of a mixture of normals, Diebolt and Robert (1994) used Gibbs sampling. In their work, in implementation of MCMC methods for mixture models, $\boldsymbol{\vartheta}$ is augmented by allocation vector $\mathbf{S}$ using Gibbs sampling. Through this sampling, samples of allocation vector $\mathbf{S}$ and parameter vector $\boldsymbol{\vartheta}$ are alternately generated resulting in a allocation vector chain and a parameter chain (McLachlan and Peel, 2004). Gibbs sampling algorithms for mixture of univariate normals and multivariate normals (Frühwirth-Schnatter, 2006) are given in Algorithm 2.2 and Algorithm 2.3. Before we introduce these algorithms, we need to specify the posterior distribution of weights which is used in these algorithms.

Since the allocation vector $\mathbf{S}$ is multinomially distributed with probability $\boldsymbol{\eta}$, the conjugate prior for the weights is a Dirichlet distribution (Richardson and Green, 1997). This distribution with the concentration parameter e_0 is given by:

$$\begin{aligned} p(\boldsymbol{\eta}|e_0) &= \frac{\Gamma(\sum_{k=1}^{K} e_0)}{\prod_{k=1}^{K}\Gamma(e_0)} \prod_{k=1}^{K} \eta_k^{e_0-1} \\ &= \frac{\Gamma(Ke_0)}{\Gamma(e_0)^K} \prod_{k=1}^{K} \eta_k^{e_0-1}. \end{aligned} \tag{2.6.15}$$

Considering the above prior, the posterior distribution for weights is of the form

$$p(\boldsymbol{\eta}|\mathbf{S}) \propto \prod_{k=1}^{K} \eta_k^{N_k(\mathbf{S})} p(\boldsymbol{\eta}). \tag{2.6.16}$$

Let $\text{Dir}(e_0, \cdots, e_0)$ denote the Dirichlet distribution with concentration parameter e_0, then the above posterior is denoted by $\text{Dir}(e_1(\mathbf{S}), \cdots, e_K(\mathbf{S}))$ where $e_k(\mathbf{S}) = e_0 + N_k(\mathbf{S}),\ k = 1, \cdots, K$. The algorithm of Gibbs sampling for univariate normals is given below.

Algorithm 2.2: Gibbs sampling for univariate normals

Step 1. Parameter simulation conditional on the classification $\mathbf{S}$:

(a) Sample $\boldsymbol{\eta}$ from the posterior $\text{Dir}(e_1(\mathbf{S}), \cdots, e_K(\mathbf{S}))$ in Equation (2.6.16).

(b) Sample σ_k^2 in each group k from posterior $\mathcal{G}^{-1}(c_k(\mathbf{S}), C_k(\mathbf{S})$ in Equation (2.6.10).

(c) Sample μ_k in each group k from posterior $\mathcal{N}(b_k(\mathbf{S}), B_k(\mathbf{S})$ in Equation (2.6.9).

Step 2. Classification of each observation y_i, for $i = 1, \cdots, N$ conditional on knowing $\boldsymbol{\mu}$, $\boldsymbol{\sigma^2}$ and $\boldsymbol{\eta}$ by sampling from Equation (2.6.4):

$$Pr(S_i = k|y_i, \boldsymbol{\vartheta}) \propto p(y_i|\boldsymbol{\theta}_k)\eta_k.$$

Note that Algorithm 2.2 starts with sampling the parameter $\boldsymbol{\vartheta}$ based on allocations $\mathbf{S}^{(0)}$.These steps could be reversed and in that case the algorithm starts with sampling the allocation $\mathbf{S}$ based on a parameter $\boldsymbol{\vartheta}^{(0)}$. This also applies for steps of the following multivariate case.

Algorithm 2.3: Gibbs sampling for multivariate normals

Step 1. Parameter simulation conditional on the classification $\mathbf{S}$:

(a) Sample $\boldsymbol{\eta}$ from the posterior $\mathrm{Dir}(e_1(\mathbf{S}), \cdots, e_K(\mathbf{S}))$ in Equation (2.6.16).

(b) Sample $\boldsymbol{\Sigma}_k^{-1}$ in each group k from posterior $\mathcal{W}_d(c_k(\mathbf{S}), \mathbf{C}_k(\mathbf{S}))$ in Equation (2.6.13).

(c) Sample $\boldsymbol{\mu}_k$ in each group k from posterior $\mathcal{N}_d(\mathbf{b}_k(\mathbf{S}), \mathbf{B}_k(\mathbf{S}))$ in Equation (2.6.12).

Step 2. Classification of each observation $\mathbf{y}_i$, for $i = 1, \cdots, N$ conditional on knowing $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^{\mathbf{2}}$ and $\boldsymbol{\eta}$ by sampling from:

$$Pr(S_i = k|\mathbf{y}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\eta}) \propto p(\mathbf{y}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\eta_k.$$

Chapter 3

Variable Screening for Multivariate Regression Models

3.1 Introduction

In multivariate regression analysis, variable screening in a high dimensional predictor space is challenging. To overcome this challenge, we propose a new screening approach based on projections of the multivariate response variable into the predictor space. Concepts from different fields inspire the proposed method. One is high-dimensional data analysis and the other, called beamforming, is a staple of signal processing. Beamforming refers to the technique of removing unwanted interference from a signal by controlling the direction that the signal flows. In our proposed procedure, variable selection is carried out by calculating an index for each predictor and threshold these indices at an appropriate threshold level. The appealing property about the proposed screening index is that it allows us to take advantage of the covariance structure of the multivariate response variable in the selection process.

This chapter is organised as follows. In Section 3.2 an existing variable screening method for univariate multiple regression is reviewed. Since our main focus in this thesis is on multivariate multiple regression models where we have several response variables, we revise the existing method introduced in

Section 3.3 to become applicable to the multivariate regression models. A brief introduction on beamforming is given in 3.4. In Section 3.4.1 we construct our proposed screening statistic, called *Predictive Information Index.* This statistic is based on the response covariance matrix estimation. The predictive information index is normalised in Section 3.4.2 to define signal-to-noise ratio (SNR) statistic. Since the response covariance matrix estimator can be ill-posed due to high dimensions in the response relative to the sample size, in Section 3.4.3 the response covariance matrix is shrunk in certain ways. In the last two sections, we compare the performance of the aforementioned indices in screening on both simulated and real data. We also compare the Predictive Information Index-based and the SNR-based screening with the likelihood-based screening approach for both simulated data in Section 3.5 and real data in Section 3.6.

3.2 Variable screening

In modelling of large regression data sets, where the number of predictors p far exceeds the number of observations n, identifying important predictors is a crucial yet a complex task. In previous chapter, variable selection was introduced to deal with such high dimensional data and reduce the dimension by selecting important variables. We also discussed a branch of variable selection methods called regularisation methods. These methods perform variable selection and parameter estimation simultaneously by imposing a sparsity-inducing penalty on the residual sum of squares function. Although regularisation methods facilitate the analysis of data with $p \gg n$, they may not be practically efficient in high dimension settings where the number of predictors is as large as a few thousand. For example, in many modern applications, where the data are collected from areas such as genomics, microarrays, finance and brain images, these methods often suffer from computational deficiency. Moreover, conditions that are necessary to hold for selection to be consistent may not hold due to the significant difference between p and n (Wang et al., 2015); (Fan et al., 2009). To address these challenges, *variable screening* can be applied to reduce the dimension. In the variable screening, a measure is defined to evaluate and rank the importance of each variable. This is followed by thresholding the ranked variables to end up with a reasonable size of the variables. In the linear regression context with high dimension in predictors, the importance of predictors is specified according to the influence of the predictors on the response variable. Therefore, predictors with a weak impact on the response variables are removed.

Variable screening procedures fall into two main categories: model-free and model-based procedures. Unlike model-based approaches, in model-free screening procedures, imposing a specific model structure on regression functions is not necessary. The recent literature in screening procedures is very rich and covers a broad variety of models such as linear regression models, generalised linear models, parametric and non-parametric regression models and even nonlinear models. Liu et al. (2015) provide an excellent overview of these feature screening methods for high dimensional data.

Here, our focus is on model based screening procedures for linear regressions. We review a pioneering work in this field called *sure screening*. The concept of sure screening was first introduced by Fan and Lv (2008) where the aim is to reduce the dimension of variables to a moderate size as small as sample size n, while maintaining the informative part of the variables in the model. A variable screening procedure has sure screening property if the survival probability of important variables after screening tends to one. The Sure Independence Screening (SIS) proposed by Fan and Lv (2008), is a selection technique based on the marginal Pearson correlations of predictors with the response variable. Through this technique, the importance of predictors are evaluated based on their marginal correlations with the response variable, and as a result, predictors that have a weak correlation with the response are discarded. Consider the following univariate multiple regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \tag{3.2.1}$$

where each column of the matrix $\mathbf{X}_{n\times p}$ denote n observations on each predictor $\mathbf{x}_k; k = 1, \cdots, p$ and the vector $\mathbf{y}$ contains n observations on the response variable Y and $\epsilon = (e_1, \cdots, e_n)$ is the error term and $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)$ is the regression coefficient vector. If we denote the estimated coefficient vector by $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \cdots, \hat{\beta}_p)$, by applying (SIS) the regression coefficients corresponding to each predictor k are ranked and thresholded. Then, the predictors with the highest regression coefficients are selected. The following reduced model is obtained through the SIS technique. For any given $\gamma \in (0, 1)$,

$$\mathcal{M}_\gamma = \{1 \leq j \leq p : |\hat{\beta}_j| \text{ is among the first } [\gamma n] \text{ largest of all}\}, \qquad (3.2.2)$$

where $\hat{\beta}_j = \mathbf{X}_j^T \mathbf{y}; j = 1, \cdots, p$ and $[\gamma n]$ denotes the integer part of γn. Here, the assumption is that both $\mathbf{X}$ and $\mathbf{y}$ are standardised which implies that $\hat{\beta}_j$ is actually the Pearson correlation between the jth predictor and the response variable. As a result of this screening, the full model $\mathcal{M} = \{1, \cdots, p;\ p \gg n\}$ is shrunk to the submodel $\mathcal{M}_\gamma$ of size $d = [\gamma n] < n$.

Since SIS procedure is based on the marginal correlations, it may not be perfectly efficient in practice. The reason is that, an important predictor which is marginally uncorrelated but jointly correlated with the response is neglected by SIS; whereas, an unimportant predictor that is highly correlated with important predictors are more likely to be selected than other important predictors with weak correlation with the response variable. To address this issue and enhance the screening accuracy, Fan and Lv (2008) proposed an iterative sure independent screening process (ISIS). The first step of this iterative process starts by selecting a subset of important variables using a variable selection method, say Lasso, then a regression model is fitted to this subset and the fitted residuals are obtained. In the next step, these residuals are treated as response variables. Thus, a model is fitted to these residuals and the remainder of unimportant predictors in the previous step. In such an iterative procedure, the unselected important variables in previous steps can survive. Although the aim of screening is to reduce the high dimension of variables to a dimension as small as sample size n, a model with size $d \geq n$

can also be shosen. In fact by choosing large d, the probability that the true model is included in the submodel $\mathcal{M}_\gamma$ is increased. The possible drawback of such choices is the computational cost. Fan and Lv (2008) chose $d = n - 1$ and $d = [n/\log n]$ in implementing SIS. According to their numerical results, choosing a submodel of size $[n/\log n]$ is consistent with the sure screening property of SIS.

SIS is designed in linear regression framework and Pearson correlation captures the linear dependancy, so to extend this correlation to a nonlinear case, Hall and Miller (2009) proposed a generalised correlation that captures both linear and nonlinear correlations.

Despite the fast growing research in variable screening for univariate regression models wherein one response variable is regressed against a set of predictors, less progress has been made in screening methods that are suitable for multivariate regressions where multiple response variables are regressed against a set of predictors. In this thesis, modelling multi-response data is a centre of attention hence, in the next section we revise the SIS procedure to become applicable to the regression models with multivariate (multiple) response variables.

3.3 Likelihood-based variable screening

Consider the data $(\mathbf{Y}, \mathbf{X})$ where $\mathbf{Y} = \mathbf{Y}_{n\times J} = (y_{ij}) = (\mathbf{y}_1\mathbf{y}_2\cdots\mathbf{y}_J)$ and $\mathbf{X} = \mathbf{X}_{n\times p} = (x_{ik}) = (\mathbf{x}_1\cdots\mathbf{x}_p)$, and $\mathbf{y}_j; 1 \leq j \leq J$ and $\mathbf{x}_k; 1 \leq k \leq p$ are the vectors of n observations made on the response variables and the predictors respectively. Since we have several response variables in these data, the univariate regression model (3.2.1) is extended to the following multivariate multiple regression model

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}, \tag{3.3.1}$$

where $\mathbf{B} = \mathbf{B}_{p\times J} = (b_{ij}) = (\mathbf{b}_1\mathbf{b}_2\cdots\mathbf{b}_p)^T$ and $\mathbf{E} = \mathbf{E}_{n\times J} = (\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_2\cdots\boldsymbol{\varepsilon}_J)$ with $\mathbf{b}_j$ and $\boldsymbol{\varepsilon}_j$ respectively denote the values of the regression coefficients and the error terms related to the jth response variable. Also suppose that the errors are Gaussian. We normalise both response variables and predictors. Suppose $\bar{x}_k = 1/n\sum_{i=1}^n x_{ik}$; $k = 1,\cdots,p$, then predictors are centered by subtracting the column means $\bar{x}_1,\cdots,\bar{x}_p$ of the predictor matrix from their corresponding column and standardised by dividing the centered columns by their standard deviations. The response variables $\mathbf{y}_j = (y_{1j},\cdots,y_{nj})^T; j = 1\cdots,J$ are centralised by subtracting the mean $\bar{\mathbf{y}} = \frac{1}{J}\sum_{j=1}^J \mathbf{y}_j$ from the corresponding column and standardised by dividing the centered column by the standard deviation.

In order to conduct screening with the aim of reducing the dimension, we marginally fit a multivariate regression to each of $(\mathbf{Y}_{n\times J}, \mathbf{x}_k), 1 \leq k \leq p$ as follows

$$\mathbf{Y} = \mathbf{x}_k\mathbf{b}_k + \tilde{\mathbf{E}}, \quad k = 1,\cdots,p. \tag{3.3.2}$$

where $\tilde{\mathbf{E}}$ contains the error terms. To obtain the corresponding least square estimates $\hat{\mathbf{b}}_k$ of $\mathbf{b}_k, 1 \leq k \leq p$, we consider the following optimisation

$$\hat{\mathbf{b}}_k = \underset{\mathbf{b}_k\in\mathbb{R}^J}{\operatorname{argmin}} \ \|\mathbf{Y} - \mathbf{x}_k\mathbf{b}_k\|_F^2,$$

where $\|.\|_F$ denotes the Frobenius norm of matrices. Least square estimates minimise the following expression

$$\|\mathbf{Y} - \mathbf{x}_k\mathbf{b}_k\|_F^2 = \text{tr}\ \left[(\mathbf{Y} - \mathbf{x}_k\mathbf{b}_k)^T(\mathbf{Y} - \mathbf{x}_k\mathbf{b}_k)\right],$$

where tr (.) is the trace of a matrix. Differentiating the above equation with respect to the J-dimensional coefficient vector $\mathbf{b}_k$ and setting it equal to zero, we get

$$\begin{aligned}\mathbf{0} &= \frac{\partial}{\partial\mathbf{b}_k}\text{tr}\ \left[(\mathbf{Y} - \mathbf{x}_k\mathbf{b}_k)^T(\mathbf{Y} - \mathbf{x}_k\mathbf{b}_k)\right] \\ &= -2\mathbf{x}_k^T(\mathbf{Y} - \mathbf{x}_k\mathbf{b}_k),\end{aligned}$$

therefore the marginal least square estimate corresponding to a single predictor $\mathbf{x}_k$ is of the form

$$\hat{\mathbf{b}}_k = (\mathbf{x}_k^T \mathbf{x}_k)^{-1} \mathbf{x}_k^T \mathbf{Y}, \quad k = 1, \cdots, p, \tag{3.3.3}$$

which is the k-th row of the estimated coefficient matrix $\hat{\mathbf{B}}_{p \times J}$. To distinguish predictors that have the highest effect on the response variable, we calculate the squared Euclidean norm of estimated coefficient vectors $\|\hat{\mathbf{b}}_1\|_2^2, \cdots, \|\hat{\mathbf{b}}_p\|_2^2$ where

$$\begin{aligned} \|\hat{\mathbf{b}}_k\|_2^2 &= \hat{\mathbf{b}}_k^T \hat{\mathbf{b}}_k \\ &= \mathbf{Y}^T \mathbf{x}_k (\mathbf{x}_k^T \mathbf{x}_k)^{-2} \mathbf{x}_k^T \mathbf{Y} \\ &= (\mathbf{x}_k^T \mathbf{x}_k)^{-2} \mathbf{x}_k^T \mathbf{Y} \mathbf{Y}^T \mathbf{x}_k. \end{aligned} \tag{3.3.4}$$

Since response variables are centralized, the response sample covariance is of the form $\hat{S} = \frac{1}{J} \sum_{j=1}^{J} \mathbf{y}_j \mathbf{y}_j^T = \frac{1}{J} \mathbf{Y} \mathbf{Y}^T$. Substituting $\hat{S}$ in the Equation (3.3.4), the squared Euclidean norm of each coefficient is obtained by

$$\|\hat{\mathbf{b}}_k\|_2^2 = J (\mathbf{x}_k^T \mathbf{x}_k)^{-2} \mathbf{x}_k^T \hat{S} \mathbf{x}_k. \tag{3.3.5}$$

Predictors with larger squared Euclidean norms are selected as important and will be included in the following reduced model

$$\boldsymbol{\mathcal{M}}_\delta = \{1 \leq k \leq p : \|\hat{\mathbf{b}}_k\|_2^2 \text{ is among the first } \delta \text{ largest of all}\}, \tag{3.3.6}$$

where δ is a pre-specified cutoff point. Since the errors are assumed to be Gaussian, the least square estimates and maximum likelihood estimates are equivalent. Therefore, we refer to the above procedure as likelihood-based marginal screening (LMS). In implementing the LMS, if δ is chosen to be large, the probability that the true model is included in the reduced model $\boldsymbol{\mathcal{M}}_\delta$ is high. However, the reduced model will not be parsimonious and the computational cost might be expensive. Zhong and Zhu (2015), Fan and Lv (2008) and Zhu et al. (2011), in different works, empirically show that setting the cutoff point to $[n/\log n]$ in their proposed screening methods gives good results in simulation studies. In another work by Hall and Miller (2009), $p/2$ was chosen as the cutoff point. This choice may contain too many irrelevant or false positives in cases that dimension of variables is high. Since there is no universal method to choose the cutoff point, this value is mostly user specified and varies based on areas that theses methods are applied. For example, consider the analysis of gene expression data in cancer biology. In such datasets, usually the number of variables (genes) exceeds tens of thousands while the sample size is very small. Suppose we apply variable screening on such data to determine the important variables (genes) which play a role in a particular type of cancer. Since only a small fraction of genes may be responsible for the disease (Moosa et al., 2016), choosing $p/2$ as cutoff point may not be a good choice for such data, while $[n/\log n]$ seems to be a more reasonable choice.

Although the above screening is suitable for multivariate regressions, there is a crucial drawback in conducting such screening. When the regression model is built upon more than one response variable, covariance structure of $\mathbf{Y}_{n\times J}$ is simply ignored by applying the above screening approach. To address this issue, in the following sections we introduce our proposed screening procedure that applies to the cases with the multivariate response variable and also takes into account the response covariance structure.

3.4 Beamforming-based variable screening

In the field of signal processing, it is often desired to estimate the signal radiating from a specific location, in the presence of noise and interfering signals. When the disruption caused by interfering signals is strong, the target signal may be masked by the interference. To address this issue *beamforming* which controls the direction of the target signal is utilised and as a result, received signal is improved. In other words, beamforming aims to enhance the signals coming from a particular location while reducing the signals from other directions. This is accomplished through implementing specific filters known as *beamformers* which reject signals from different directions except for the desired location.

Technically, a beamformer is an operator which is used to estimate the strength or power of a signal at a particular location. One of the widely known beamformers is the minimum-variance filter where the cost function is the signal output variance at the desired location. The output power is often contaminated by not only the noise but also some unwanted signals from other locations than the desired one. Therefore, the minimisation of the output power is done subject to a linear constraint. The linear constraint forces the filter to pass the signal from a specified location while the power minimisation prevents interferences caused by signals from other locations (Van Veen et al., 1997), (Sekihara and Nagarajan, 2008).

3.4.1 Predictive information index (PII)

In this section we formulate our proposed variable screening index. The concept of information index is inspired by beamforming technique in the field of signal processing. The proposed index is based on the projections of the response variable into the predictor space and it allows for the response covariance to come to play in the screening process.

We estimate the predictive information index for each predictor vector, $\mathbf{x}_k; 1 \leq k \leq p$ by minimising the sample variance of the projected data points $\mathbf{w}_k^T \mathbf{y}_j; 1 \leq j \leq J$ along a weight vector $\mathbf{w}_k$. Note that this is inline with beamforming through minimum-variance beamformer for which the cost function is the variance of the output power at particular location and the aim is minimising this variance subject to a linear constraint on the specified location. To construct the predictive information index, say for the k-th predictor, we first project each response variable into the k-th predictor space along an n-dimensional weight vector, $\mathbf{w}_{n\times 1}$

$$\mathbf{w}_k^T \mathbf{Y} = \left(\mathbf{w}_k^T \mathbf{y}_1, \cdots, \mathbf{w}_k^T \mathbf{y}_J\right),$$

now we need to find a direction $\mathbf{w}$ such that the projected data onto predictor space along this direction carry some useful information. To this aim, we minimise the sample variance of the projected data with respect to the weight vector $\mathbf{w}$ subject to the constraint $\mathbf{w}^T\mathbf{x}_k = 1$, i.e.

$$\min_{\mathbf{w}_k} \ \mathrm{S}(\mathbf{w}_k^T\mathbf{Y}), \quad s.t \quad \mathbf{w}_k^T\mathbf{x}_k = 1, \tag{3.4.1}$$

where S(.) denotes the sample variance operator. The minimisation problem in the above equation can be rewritten as follows

$$\min_{\mathbf{w}} \ (\mathbf{w}_k^T\mathbf{C}\mathbf{w}_k), \quad s.t \quad \mathbf{w}_k^T\mathbf{x}_k = 1, \tag{3.4.2}$$

In the above equation, $\mathbf{C}$ is replaced by an estimator $\hat{\mathbf{C}}$ of $\mathbf{C}$. This estimator can be, for example, the sample covariance matrix $\hat{\mathbf{C}} = \hat{\mathbf{C}}_{n\times n}$ which is defined by

$$\hat{\mathbf{C}}_{n\times n} = \frac{1}{J}\sum_{j=1}^{J}(\mathbf{y}_j - \bar{\mathbf{y}})(\mathbf{y}_j - \bar{\mathbf{y}})^T,$$

where $\bar{\mathbf{y}} = \frac{1}{J}\sum_{j=1}^{J}\mathbf{y}_j$. Note that from a population perspective, the covariance matrix $\hat{\mathbf{C}}_{n\times n}$ is defined by

$$\hat{\mathbf{C}}_{n\times n} = \frac{1}{J}\sum_{j=1}^{J}E[(\mathbf{y}_j - E[\mathbf{y}_j])(\mathbf{y}_j - E[\mathbf{y}_j])^T].$$

In Equation (3.4.2), the constraint guarantees that any information related to the k-th predictor passes through the filter while interferences from other predictors are reduced simultaneously by minimising the variance of projected data.

We implement the method of Lagrange multiplier to solve the optimisation problem in Equation (3.4.2) and obtain the optimal weight vector. If we denote the Lagrange multiplier by λ, then the Lagrangian function $\mathcal{L}$ is expressed as

$$\mathcal{L}(\mathbf{w}_k, \lambda) = \mathbf{w}_k^T\hat{\mathbf{C}}\mathbf{w}_k - \lambda(\mathbf{w}_k^T\mathbf{x}_k - 1).$$

We require to differentiate the Lagrangian $\mathcal{L}(\mathbf{w}_k, \lambda)$ with respect to $\mathbf{w}_k$. To attain the first and second order derivatives with respect to the vector $\mathbf{w}_k$, we invoke the following rules. For any matrix $\mathbf{A}$ and any vector $\mathbf{w}$, we have $\frac{\partial \mathbf{w}^T\mathbf{A}\mathbf{w}}{\partial \mathbf{w}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{w}$ and $\frac{\partial \mathbf{a}^T\mathbf{w}_k}{\partial \mathbf{w}_k} = \frac{\partial \mathbf{w}^T\mathbf{a}}{\partial \mathbf{w}} = \mathbf{a}$

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{w}_k, \lambda)}{\partial \mathbf{w}_k} &= \frac{\partial}{\partial \mathbf{w}_k}[\mathbf{w}_k^T\hat{\mathbf{C}}\mathbf{w}_k - \lambda(\mathbf{w}_k^T\mathbf{x}_k - 1)] \\ &= 2\hat{\mathbf{C}}\mathbf{w}_k - \lambda\mathbf{x}_k,\end{aligned}$$

setting the last equation above equal to zero gives us the following solution

$$\hat{\mathbf{w}}_k = \frac{\lambda}{2}\hat{\mathbf{C}}^{-1}\mathbf{x}_k, \tag{3.4.3}$$

substituting this $\hat{\mathbf{w}}$ into the constraint we get

$$\lambda = \frac{2}{\mathbf{x}_k^T\hat{\mathbf{C}}^{-1}\mathbf{x}_k},$$

finally substituting this λ in Equation (3.4.3) we get the optimal weight vector

$$\hat{\mathbf{w}}_k = \frac{\hat{\mathbf{C}}^{-1}\mathbf{x}_k}{\mathbf{x}_k^T\hat{\mathbf{C}}^{-1}\mathbf{x}_k}.$$

Note that the above optimal weight or direction vector depends on a estimator of the covariance matrix. In Section 3.4.3 we introduce different estimators of covariance matrix and in simulation studies, we demonstrate how these estimators affect the screening results.

Correspond to each predictor $\mathbf{x}_k$, the J-vector $\hat{\mathbf{w}}_k^T\mathbf{Y}$ obtains the amount of information that the k-th predictor carries about response variables. If we ignore the correlations between response variables, this is equivalent to regression coefficient estimates in linear regression modelling framework. In other words, if we replace $\hat{\mathbf{C}}$ by $\mathbf{I}_n$ then $\hat{\mathbf{w}}_k^T\mathbf{Y}$ is equivalent to the least square estimate $\hat{\mathbf{b}}_k$ expressed in Equation (3.3.3)

$$\hat{\mathbf{w}}_k^T\mathbf{Y} = \frac{\mathbf{x}_k^T\mathbf{Y}}{\mathbf{x}_k^T\mathbf{x}_k} \quad \cong \quad \hat{\mathbf{b}}_k = (\mathbf{x}_k^T\mathbf{x}_k)^{-1}\mathbf{x}_k^T\mathbf{Y}; \quad k = 1, \cdots, p,$$

Note that the minimisation of the variance is also inline with the minimisation of residual sum of squares in the setting of multivariate simple linear regression which is expressed in Equation (3.3.2).

Having found the direction that minimises the variation, we can now define the *Predictive Information Index* or *predictive power* for predictor $\mathbf{x}_k$ as

$$\hat{r}_k = \min_{\mathbf{w}_k^T \mathbf{x}_k = 1} \mathbf{w}_k^T \hat{\mathbf{C}} \mathbf{w}_k = \hat{\mathbf{w}}_k^T \hat{\mathbf{C}} \hat{\mathbf{w}}_k = \frac{1}{\mathbf{x}_k^T \hat{\mathbf{C}}^{-1} \mathbf{x}_k}, \quad k = 1, \cdots, p. \tag{3.4.4}$$

This index measures the amount of information that each predictor holds for response variable prediction. The higher this index, the more information the predictor carries about the response variable and therefore the more important the predictor.

Similarly, the predictive information index can be defined for a particular subset of predictors. This subset could be specified by some prior information or experts' knowledge. For example, in analysis of gene expressions, this subset could be a group of genes with the same pathways. Suppose $\mathbf{X}_{n \times p} = (\mathbf{x}_1 \cdots \mathbf{x}_p)$ denotes all predictors in the model and we are interested in finding the PII or predictive power for subset $\nu = \{k_1, \cdots, k_m\}$ of these predictors. Let $\mathbf{X}_\nu = (\mathbf{x}_{k_1}, \cdots, \mathbf{x}_{k_m})$ denote this subset in a matrix form. Then the joint predictive information index of this subset which is called *predictive information matrix*, can be found by solving the following optimisation problem

$$\min_{\mathbf{W}} \; (\mathbf{W}^T \hat{\mathbf{C}} \mathbf{W}), \quad s.t \quad \mathbf{W}^T \mathbf{X}_\nu = I_m, \tag{3.4.5}$$

where, $\mathbf{W}$ is an $n \times m$ direction matrix and the constraint matrix I_m is an $m \times m$ identity matrix. Note that $\mathbf{W}^T \mathbf{X}_\nu = I_m$ define m linear filters which null each other. Similar to what we did before, using Lagrange multiplier and differentiating with respect to the matrix $\mathbf{W}$, gives the optimised direction of $\hat{\mathbf{W}} = \hat{\mathbf{C}}^{-1} \mathbf{X}_\nu (\mathbf{X}_\nu^T \hat{\mathbf{C}}^{-1} \mathbf{X}_\nu)^{-1}$. Thus the joint predictive information matrix of subset $\mathbf{X}_\nu$ is defined by

$$\hat{r}_\nu = \left(\mathbf{X}_\nu^T \hat{\mathbf{C}}^{-1} \mathbf{X}_\nu\right)^{-1}. \tag{3.4.6}$$

In the next section we show how the introduced predictive information index can be standardised to define signal-to-noise ratio statistic.

3.4.2 Signal-to-noise ratio (SNR)

The predictive information index in equation (3.4.4) is often contaminated by the background noise or some unwanted interferences from undesired directions. In such cases, although the estimated predictive information index might be high, when compared to the noise level, it may not be considered as a high value anymore. Moreover, the background noise may be heterogeneous across the projected data. We address this issue by standardizing the PII for each predictor via dividing these values by the white noise. The obtained ratio is called signal-to-noise ratio. SNR is the ratio of the strength of a signal carrying information to that of unwanted interference (Sekihara and Nagarajan, 2008). In variable screening framework, SNR value can be regarded as a measure of how much useful information each predictor contains about the response variables. When observations on response variables are white noises, the predictive power of the kth predictor reduces to $\sigma^2 \hat{\mathbf{w}}_k^T \hat{\mathbf{w}}_k$. Accordingly, the SNR which is the ratio of the desired useful information to the level of background noise is defines as

$$\begin{aligned} \text{SNR}_k &= \frac{\hat{r}_k}{\sigma^2 \hat{\mathbf{w}}_k^T \hat{\mathbf{w}}_k} \\ &= \frac{1}{\sigma^2 \hat{\mathbf{w}}_k^T \hat{\mathbf{w}}_k (\mathbf{x}_k^T \hat{\mathbf{C}}^{-1} \mathbf{x}_k)} \end{aligned}$$

Since σ^2 is assumed to be the same for all response variables, we omit it from the calculations. If we substitute the optimised weight vector $\hat{\mathbf{w}}_k = \frac{\hat{\mathbf{C}}^{-1}\mathbf{x}_k}{\mathbf{x}_k^T \hat{\mathbf{C}}^{-1} \mathbf{x}_k}$ in the above equation, then the estimated SNR corresponding to each predictor

$\mathbf{x}_k$ is defined as

$$\mathrm{SNR}_k = \frac{\mathbf{x}_k^T \hat{\mathbf{C}}^{-1} \mathbf{x}_k}{\mathbf{x}_k^T \hat{\mathbf{C}}^{-2} \mathbf{x}_k} \quad k = 1, 2, \cdots, p. \tag{3.4.7}$$

Now, in order to find principal predictors, we calculate the SNR values corresponding to each predictor $\mathbf{x}_k; 1 \leq k \leq p$. Then these values are ranked and the highest ones are selected by thresholding. Predictors that the SNR indices of which are higher than the threshold value are classified as principal predictors. The higher the SNR index, the more important the predictor. If the covariance matrix $\hat{\mathbf{C}}$ is consistent with the covariance matrix of $\mathbf{y}_j$, then under certain conditions, screening by SNR can have the sure screening property such that for an appropriately chosen threshold all the true predictors can be detected with a probability one (Zhang and Oftadeh, 2016).

3.4.3 Shrinkage of covariance matrix

The PII and consequently the associated SNR value introduced above, are established upon an estimator of the response variable covariance matrix. The most well-known unbiased estimator of covariance matrix is the sample covariance matrix. We remind that the sample covariance matrix estimator utilised in SNR formulation is of the form

$$\hat{\mathbf{C}} = \frac{1}{J} \sum_{j=1}^{J} \mathbf{y}_j \mathbf{y}_j^T - \bar{\mathbf{y}} \bar{\mathbf{y}}^T = (\hat{c}_{ij}), \tag{3.4.8}$$

where $\bar{\mathbf{y}} = \sum_{j=1}^{J} \mathbf{y}_j / J = (\bar{y}_1, ..., \bar{y}_n)^T$ and $\hat{c}_{ij} = \sum_{t=1}^{J} (y_{it} - \bar{y}_i)(y_{jt} - \bar{y}_j)/J$. In spite of having some desirable properties, such as being maximum likelihood estimator and easy to compute, the sample covariance is known to perform poorly when the dimensionality is large. In many applied contexts, where the sample size is small relative to the number of variables, sample covariance is either singular or ill-conditioned. The reason is that sample eigenvalues are biased. The sample eigenvalues are positive real numbers therefore the smallest eigenvalues tend to zero while the largest tend to infinity. This causes the sample covariance matrix to become ill-conditioned or even singular. In particular, under high dimensions eigenvalues of sample covariance are known as poor estimates for the true ones (Daniels and Kass, 2001). One approach to address this problem is to shrink the eigenvalues. Shrinkage was first introduced by Stein (1955) and in the Stein-type shrinkage estimation, a convex combination of the sample covariance and a well-conditioned target matrix is used to estimate the covariance matrix, i.e.

$$\hat{\mathbf{C}}_{Stein} = (1-\lambda)\hat{\mathbf{C}} + \lambda T, \tag{3.4.9}$$

where $\lambda \in (0,1)$ is the shrinkage intensity and T is the target matrix. Through this shrinkage the eigenvalues of $\hat{\mathbf{C}}$ are shrunk to that of T which results in a positive definite, better conditioned and non-singular estimate for any dimensionality. The well-conditioned target matrix is chosen to be positive definite and representative of the true covariance matrix. The major drawback of shrinkage estimator is that the target and the intensity should be specified (Fisher and Sun, 2011). Ledoit and Wolf (2004) tackle this difficulty by introducing a well-conditioned covariance matrix estimator which is an optimal trade-off between the sample covariance matrix and the identity matrix. Optimality here means with respect to a quadratic loss function. Suppose Σ is the true covariance matrix. The goal is finding Σ^* which is a linear combination of the identity matrix and the sample covariance matrix such that the expected quadratic loss $E[\|\Sigma^* - \Sigma\|^2]$ is minimum. Accordingly, Ledoit and Wolf's shrinkage estimator is of the form

$$\hat{\mathbf{C}}_{opt} = \frac{b_n^2}{d_n^2}\mu_n \mathbf{I}_n + \frac{d_n^2 - b_n^2}{d_n^2}\hat{\mathbf{C}}, \tag{3.4.10}$$

where

$$\mu_n = \langle \hat{\mathbf{C}}, \mathbf{I}_n \rangle, \;\; d_n^2 = \langle \hat{\mathbf{C}} - \mu_n \mathbf{I}_n, \hat{\mathbf{C}} - \mu_n \mathbf{I}_n \rangle,$$
$$\bar{b}_n^2 = \frac{1}{J^2}\sum_{j=1}^{J} \langle \mathbf{Y}_j \mathbf{Y}_j^T - \hat{\mathbf{C}}, \mathbf{Y}_j \mathbf{Y}_j^T - \hat{\mathbf{C}} \rangle, \;\; b_n^2 = \min(\bar{b}_n^2, d_n^2)$$

and $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr } (\mathbf{A}\mathbf{B}^T)/n$ for any $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ and tr is the trace of a matrix. This estimator is consistent as the sample size and the dimension go to infinity together (Ledoit and Wolf, 2004). These estimators shrink the overdispersed sample covariance eigenvalues but they do not change the eigenvectors which are also inconsistent and they do not result in sparse estimators (Bickel and Levina, 2008). To resolve this drawback, Bickel and Levina (2008) threshold the sample covariance by thresholding the entries as follows

$$\hat{\mathbf{C}}_h = \hat{\mathbf{C}}(\tau_{nJ}) = (\hat{c}_{ij} I(|\hat{c}_{ij}| > h\tau_{nJ})),$$

where $I(\cdot)$ is the indicator and $\tau_{nJ} = \sqrt{\log(n)/J}$ with the tuning constant $h \geq 0$. Although under certain conditions the thresholded covariance matrix is consistent with the true covariance matrix (Zhang and Liu, 2015), it may still be degenerate when the dimension J is close to or smaller than the sample size n. In the work done by Zhang and Liu (2015), a thresholded estimator is used in calculating the beamformers. This thresholded estimator is defined as $\hat{\mathbf{C}}(\tau_{nJ}) = (\hat{c}_{ij} I(|\hat{c}_{ij}| > \tau_{nJ}))$ where τ_{nJ} is a varying constant in J and n. As pointed out above, this estimator may not be well-conditioned in high dimensions. Therefore following Bickel and Levina (2008) we first threshold the elements of the covariance matrix, then following Ledoit and Wolf (2004), we further shrink the thresholded covariance estimator to a diagonal matrix as follows:

$$\hat{\mathbf{C}}_{hs} = \frac{b_n^2}{d_n^2} \hat{\mu}_n I_n + \frac{d_n^2 - b_n^2}{d_n^2} \hat{\mathbf{C}}_h, \tag{3.4.11}$$

where

$$\begin{aligned}
\hat{\mu}_n &= < \hat{\mathbf{C}}_h, I_n >, \quad d_n^2 =< \hat{\mathbf{C}}_h - \hat{\mu}_n I_n, \hat{\mathbf{C}}_h - \hat{\mu}_n I_n >, \\
\bar{b}_n^2 &= \frac{1}{J^2} \sum_{k=1}^{J} \frac{1}{n} \left[\sum_{i=1}^{n} \sum_{j=1}^{n} (y_{ik} - \bar{y}_i)(y_{kj} - \bar{y}_j) - \hat{c}_{ij} \right]^2 I(|\hat{c}_{ij}| > h\tau_{nJ}), \\
b_n^2 &= \min\{\bar{b}_n^2, d_n^2\},
\end{aligned}$$

$\hat{\mathbf{c}}_{\cdot j}$ is the jth column of $\hat{\mathbf{C}}_h$ and $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr }(\mathbf{A}\mathbf{B}^T)/n$ for any $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ where tr (.) is the trace of a matrix. We also define the constant h by $h = \frac{c_0}{n} \sum_{i=1}^{n} c_{ii}$ where $c_0 \in \{0.01, 0.005, 0.001\}$. The estimator $\hat{\mathbf{C}}_{hs}$ not only shrinks the eigenvalues of the covariance matrix, but also sets small eigenvalues to zero which results in a sparse estimator. This sparsity is a result of thresholding the elements of the covariance matrix. The thresholded and shrunk $\hat{\mathbf{C}}_{hs}$ is the consistent estimator that we use in finding the SNR index.

Taking a different approach, we also shrink the covariance matrix by adding a $\lambda = \lambda_{k_0} \times 0.01$ to the sample covariance matrix

$$\hat{\mathbf{C}}_{eig1} = \hat{\mathbf{C}} + \lambda I_n \tag{3.4.12}$$

where λ_{k_0} satisfy the following inequality

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_{k_0}}{\lambda_1 + \lambda_2 + \cdots + \lambda_n} \geq 0.95,$$

and $\lambda_1, \lambda_2, \cdots, \lambda_n$ are eigenvalues of the sample covariance matrix. We also considered the following shrinkage estimator of covariance matrix.

$$\hat{\mathbf{C}}_{eig2} = \hat{\mathbf{C}} + (0.001 \times \lambda_{max}) I_n, \tag{3.4.13}$$

where λ_{max} denotes the maximum eigenvalue of the sample covariance matrix. These estimators are examined in simulation studies and through simulations we explore how they affect the screening accuracy.

3.5 Simulation studies

In this section, we conduct simulations to monitor the performance of two different screening tools proposed in this chapter: the Predictive Information Index (PII) and SNR. Our simulations serve three purposes:

(a) To investigate whether implementing different shrinkage approaches to shrink the covariance matrix influence the selection accuracy and if so which of the introduced shrinkage methods enhances the selection accuracy of both PII and SNR .

(b) To compare the screening performance of PII with SNR.

(c) To compare the screening accuracy of surpassing approach in part (b) with the likelihood-based marginal screening (LMS) discussed in Section 3.3.

3.5.1 Simulation setup

Scenario 1 (Highly correlated B): In each simulation, similar to the previous chapter, the covariate matrix $\mathbf{X}_{n\times p}$, was generated by sampling n number of p-vectors $\mathbf{x}_i, i = 1, \cdots, n$ from $N_p(\mathbf{0}, \mathbf{\Sigma}_{p\times p})$ where $\mathbf{\Sigma}$ is the variance-covariance matrix of gene expressions in our real data. Coefficient matrix $\mathbf{B}^h_{p\times J}$ with high correlations between its columns was generated by sampling p number of J-vectors from $N_J(\mathbf{0}, \Omega_h)$. The details about how we generated the covariance matrix Ω_h is explained in the previous chapter in Section 2.4.1. The error matrix, $\boldsymbol{\varepsilon}_{n\times J}$, was generated by sampling J number of n-vectors from a multivariate normal distribution $N(\mathbf{0}, \sigma^2\mathbf{I}_n)$, where $\sigma^2 = 0.1$. Finally, multivariate response variables were simulated according to the following multivariate regression model

$$\mathbf{Y}_{n\times J} = \mathbf{X}_{n\times p}\mathbf{B}_{p\times J} + \boldsymbol{\varepsilon}_{n\times J}. \tag{3.5.1}$$

Scenario 2 (Weakly correlated B): The data set $(\mathbf{Y}, \mathbf{X})$ for this setting was generated from the model (3.5.1) as described in setting 1 except the coefficient matrix. In this setting there is a weak correlation between columns of $\mathbf{B}$. Therefore the coefficient matrix $\mathbf{B}^l$ was generated form $N_J(\mathbf{0}, \Omega_l)$. Also the non-zero or active elements in the coefficient matrix were placed in columns with high correlations for scenario 1 and in columns with low correlations for scenario 2. For each simulation setup, we generated 50 datasets. In each dataset we set the number of active covariates $|T| = 10$. We applied each approach to each of 50 dataset, and obtained the sensitivity and specificity values that introduced in Section 2.3. To make a reasonable comparison, the specificity percentages were compared when the sensitivity values were fixed at the same level.

Suppose we want to evaluate the screening accuracy of SNR. For each scenario, we first calculated the SNR values corresponding to all covariates $\text{SNR}_i; i = 1, \cdots, p$. Then we thresholded these values at SNR values corresponding to each of active predictors as follows. We denote the SNR values corresponding to the 10 active covariates in an increasing order as $\text{SNR}_{(1)} \leq \text{SNR}_{(2)} \leq \text{SNR}_{(3)} \leq \cdots \leq \text{SNR}_{(10)}$. We thresholded the SNR indices $\text{SNR}_i; 1 \leq i \leq p$ at levels of $\text{SNR}_{(j)}; 1 \leq j \leq 10$ respectively. For instance, if we set the threshold level at $\text{SNR}_{(1)}$, we selected those predictors with SNR index not less than $\text{SNR}_{(1)}$. Note that $\text{SNR}_{(j)}$ values of active predictors were ordered increasingly. So by thresholding at the level of $\text{SNR}_{(1)}$, the selected subset of covariates contained all active covariates which gave a sensitivity value of 100%. Similarly, setting the threshold level at the largest value $\text{SNR}_{(10)}$ gave a sensitivity of 10%. This way, we obtained a set of 10 different sensitivity values of $10\%, 20\%, 30\%, \cdots, 100\%$. We then calculated the specificity values corresponding to each of these sensitivity values.

3.5.2 Results

(a) Comparing the PII and SNR with different covariance matrix estimators

In this section, we investigate how applying different covariance estimators influence the screening performance of the PII and SNR. To this aim, in calculating SNR and PII, each time we utilised one of the following covariance estimators: optimal estimator $\hat{\mathbf{C}}_{opt}$ introduced through Equation (3.4.10), the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ in Equation (3.4.11), $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$.

While calculating the thresholded and shrunk covariance estimator, $\hat{\mathbf{C}}_{hs}$, we considered three different values of tuning constant $h = 0.01, 0.005, 0.001$ denoted by hs1, hs2 and hs3 respectively. As pointed out earlier, we illustrate the comparisons between covariance shrinkage methods by comparing specificity percentages of predictive power and SNR values while we fix the sensitivity at levels $(100j/10)\%; 1 \leq j \leq 10$. The higher the specificity, the more accurate the screening. In other words, a high specificity percentage signifies that a high proportion of unimportant predictors have been detected and discarded correctly by the screening procedure. The following result show the specificity values when sensitivities are fixed.

Sensitivity-Specificity plots of Predictive Information Index (PII)

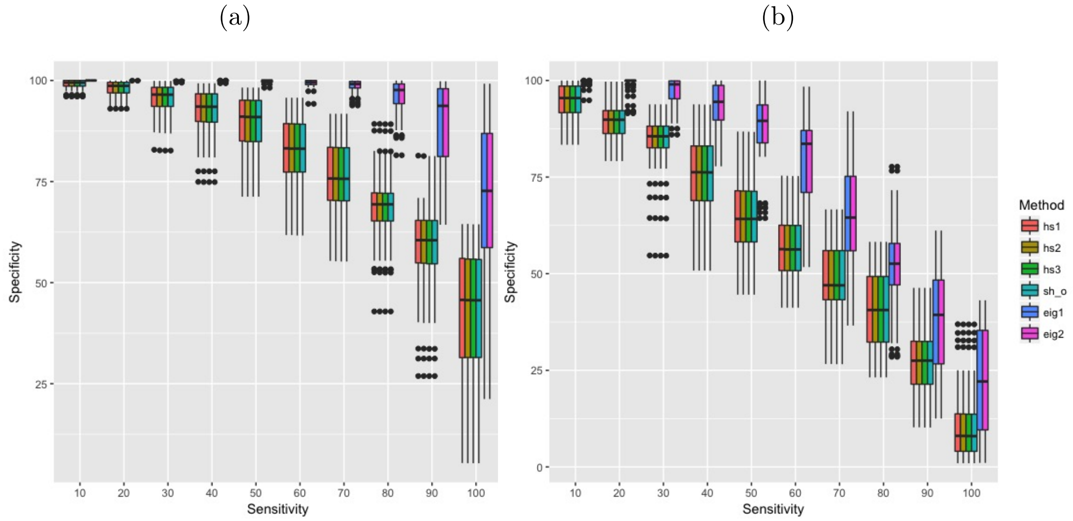
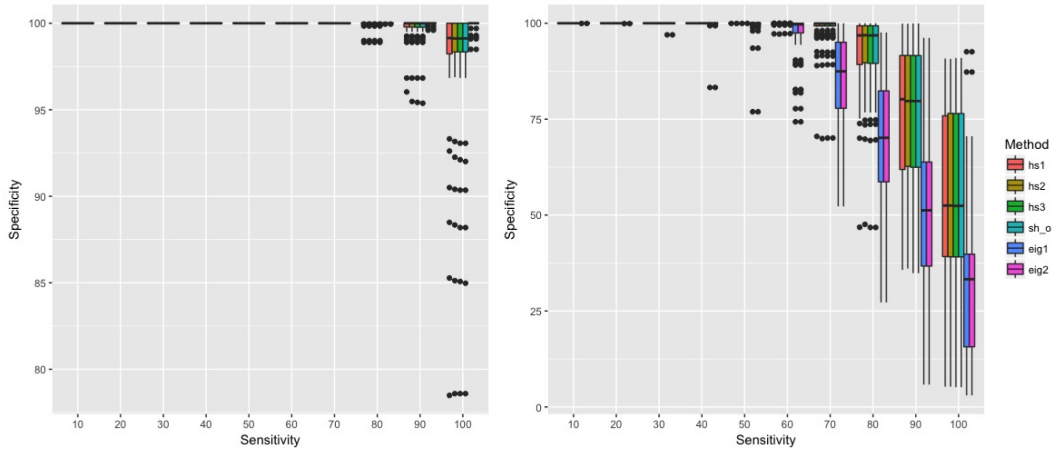


Figure 3.5.1: Comparing PII performance with different covariance matrix estimators. Results obtained from 50 simulations where $(p, n, J, |T|) = (2000, 88, 20, 10)$ for settings with (a) Highly correlated $\mathbf{B}$. (b) Weakly correlated $\mathbf{B}$. Here, sh$_o$ corresponds to PII built upon $\hat{\mathbf{C}}_{opt}$ estimator and hs1, hs2 and hs3 refers to PII built upon the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$ respectively. Also eig1 and eig2 correspond to the PII built upon $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$ estimators.

Simulation results in Figure 3.5.1 show that in both settings with highly correlated and weakly correlated $\mathbf{B}$, the PII has a higher specificity when it is built upon the estimators $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$. However, in setting 1 with highly correlated $\mathbf{B}$, Figure 3.5.1 (a), the specificity is significantly higher than the specificity obtained for setting with weakly correlated $\mathbf{B}$, Figure 3.5.1 (b). The reason that PII performs exactly the same for both $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$ estimators and same for other four estimators, is the particular simulation setting which is used here. Note that we cannot give a specific reason about why PII performs better with $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$ estimators and we just rely on numerical results. Justification of this phenomenon is a complicated task and beyond the scope of this thesis.

Sensitivity-Specificity plots of Signal-to-Noise Ratio (SNR)



(a)

(b)

Figure 3.5.2: Comparing SNR performance with different shrinkage methods applied on covariance matrix. Results obtained from 50 replicates where $(p, n, J, |T|) = (2000, 88, 20, 10)$ for settings with (a) Highly correlated $\mathbf{B}$. (b) Weakly correlated $\mathbf{B}$. Here, sh$_o$ corresponds to the SNR built upon $\hat{\mathbf{C}}_{opt}$ estimator and hs1, hs2 and hs3 refers to SNR built upon the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$ respectively. Also eig1 and eig2 correspond to the SNR built upon $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$ estimators.

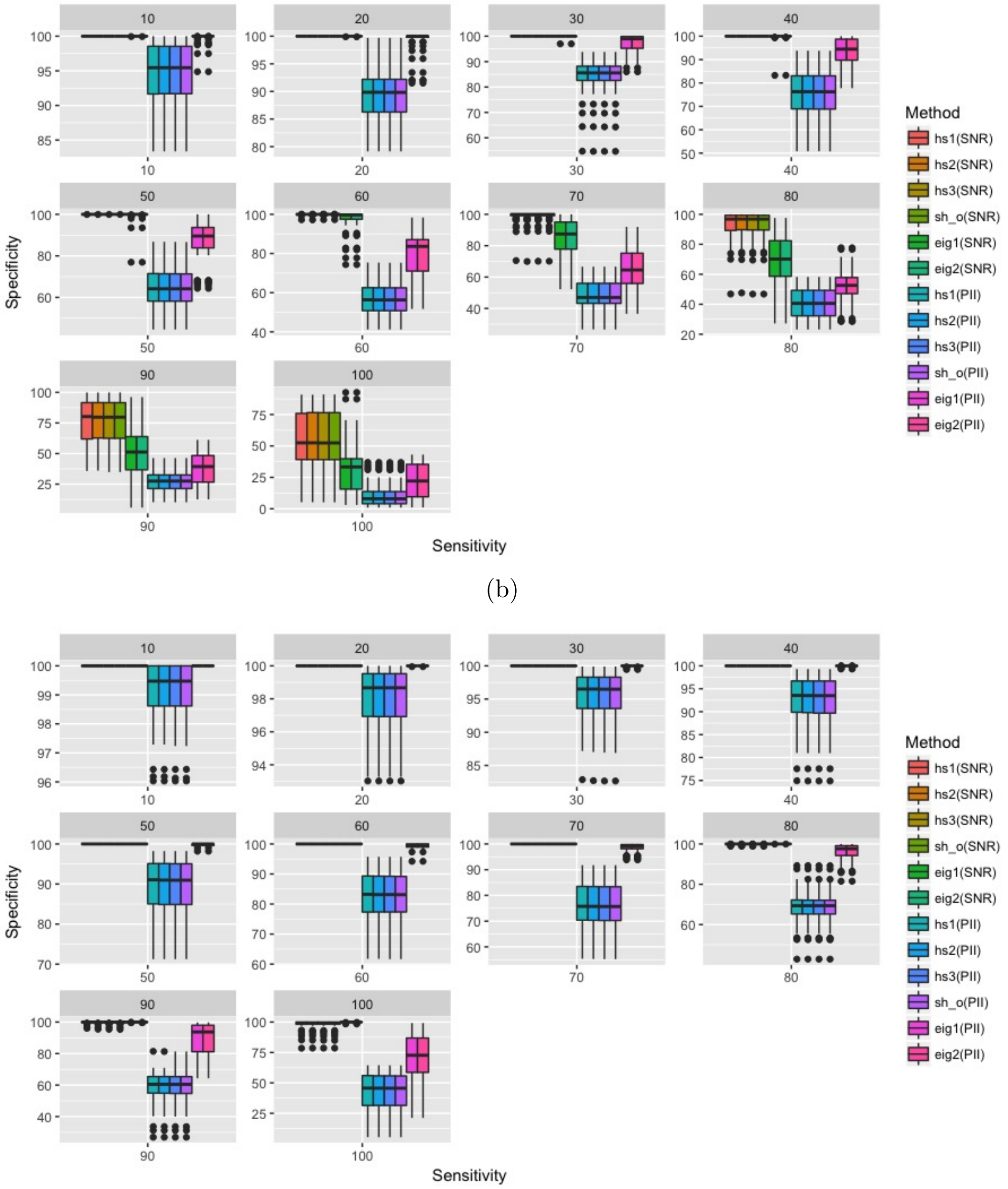
Simulation results in Figure 3.5.2 reveal that when columns of $\mathbf{B}$ are highly correlated there is no significant difference in SNR performance using different covariance estimators whereas, in the setting with weakly correlated $\mathbf{B}$ the influence of using different estimators is more noticeable. SNR screening based on the shrunk estimator $\hat{\mathbf{C}}_{opt}$ and also thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$, result in a more precise detection. These estimators shrink the overdispersed sample covariance eigenvalues more efficiently due to the particular shrinkage intensity placed on the target matrix which is discussed in Section 3.4.3. However, the screening accuracy does not change much by using $\hat{\mathbf{C}}_{hs}$ instead of $\hat{\mathbf{C}}_{opt}$. Note that the SNR value resulted from using $\hat{\mathbf{C}}_{hs}$ becomes h-dependent since we set the constant h to different values of $\{0.01, 0.001, 0.005\}$. Similar to part (a) the reason that SNR has the same performance based on both $\hat{\mathbf{C}}_{opt}$ and $\hat{\mathbf{C}}_{hs}$ for all values of h, is a result of our particular simulation setting. In settings that the noise part in data is more substantial, the difference becomes more noticeable (Zhang and Liu, 2015).

As it was mentioned earlier, our proposed method has explored the estimation of response covariance in each screening procedure. The results depicted above validate this declaration perfectly. Scenarios wherein there exist a high correlation in the coefficient matrix have much higher specificity level than the weakly correlated cases. The reason is that the simulated response inherits some high correlations from the coefficient matrix and this high correlation magnifies the screening accuracy of both PII and SNR. The reason is that these high correlations provide more information leading to a higher accuracy.

(b) Comparing PII with SNR performance

The optimum screening results for PII and SNR presented earlier are not based on the same covariance matrix estimator. In other words, PII performs better based on $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$ estimators, whereas SNR does not. This makes it difficult to make a faire comparison on the performance of these statistics. Therefore, to come to a final conclusion about which of these statistics gives more reliable screening results, we compared the specificity of each of these statistics based on all covariance estimators where we fixed the sensitivity at levels $(100j/10)\%; 1 \leq j \leq 10$. Results presented in Figure 3.5.3, uncover that in both scenarios with high and weak correlation structures in coefficient matrix, the SNR-based screening procedure outperforms the PII-based screening. Although the PII based on $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$ estimators performs well, it cannot gain the accuracy of the SNR all the time and specificity percentages justify the correctness of SNR screening in both scenarios with high and weak correlations. Accordingly, we opt for SNR as our screening statistic.

Box plots of Signal-to-Noise Ratio and Predictive Information Index


(a)

(b)

Figure 3.5.3: Comparing SNR with PII statistics based on different shrinkage methods applied on covariance matrix. Results obtained from 50 replicates where $(p, n, J, |T|) = (2000, 88, 20, 10)$ for settings with (a) weakly correlated $\mathbf{B}$ and (b) Highly correlated $\mathbf{B}$. Here, sh_o corresponds to the statistics built upon $\hat{\mathbf{C}}_{opt}$ estimator and hs1, hs2 and hs3 refers to the statistics built upon the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$ respectively. Also eig1 and eig2 correspond to the statistics built upon $\hat{\mathbf{C}}_{eig1}$ and $\hat{\mathbf{C}}_{eig2}$ estimators.

(c) Comparing SNR-based screening with likelihood-based marginal screening (LMS)

In this section, we compare the SNR-based screening based on $\hat{\mathbf{C}}_{hs}$ with the likelihood-based marginal screening procedure introduced in Section 3.3. The comparison was accomplished based on the corresponding sensitivity and specificity values for each method. We considered the multivariate regression model (3.5.1) with 10 nonzero or active coefficients. In order to carry out the LMS, we fitted a single multivariate regression to each covariate $\mathbf{x}_k; 1 \leq k \leq p$ and the multivariate response variable denoted in the matrix form as $\mathbf{Y}_{n \times J}$. We estimated the corresponding J-dimensional coefficient vector $\hat{\mathbf{b}}_k; 1 \leq k \leq p$ which is expressed in the Equation (3.3.3).

Having found the estimates, we calculated the squared Euclidean norm of estimated coefficient vectors $\|\hat{\mathbf{b}}_1\|_2^2, \cdots, \|\hat{\mathbf{b}}_p\|_2^2$. We took the same approach as previous Sections to threshold the norm values at different levels. We indicate the norm of 10 active predictors in an increasing order as $\|\hat{\mathbf{b}}_{(1)}\|_2^2 \leq \|\hat{\mathbf{b}}_{(2)}\|_2^2 \leq \cdots, \leq \|\hat{\mathbf{b}}_{(10)}\|_2^2$. We thresholded the values of $\|\hat{\mathbf{b}}_k\|; 1 \leq k \leq p$ at levels of $\|\hat{\mathbf{b}}_{(j)}\|_2^2; 1 \leq j \leq 10$ respectively. Therefore the reduced model corresponding to each level $1 \leq j \leq 10$ was obtained as

$$\boldsymbol{\mathcal{M}}_{(j)} = \{1 \leq k \leq p \ \text{ s.t } \ \|\hat{\mathbf{b}}_k\|_2^2 \geq \|\hat{\mathbf{b}}_{(j)}\|_2^2; 1 \leq j \leq 10\}, \qquad (3.5.2)$$

Since norm of active predictors are ordered increasingly, by thresholding at the level of $\|\hat{\mathbf{b}}_{(1)}\|_2^2$, the selected subset of covariates contained all active covariates which gave a sensitivity value of 100%. However, setting the threshold level at the largest value $\|\hat{\mathbf{b}}_{(10)}\|_2^2$ gave a sensitivity of 10%. This way, we obtained a set of 10 different sensitivity values of $10\%, 20\%, 30\%, \cdots, 100\%$. We then calculated the specificity values corresponding to each of these sensitivity values. We repeated this procedure for 50 simulations wherein data were simulated from a multivariate regression model according to the settings explained in Section 3.5.1. We then standardised the data according to what explained in Section 3.3 and set the number of active predictors to $|T| = (10, 100)$.

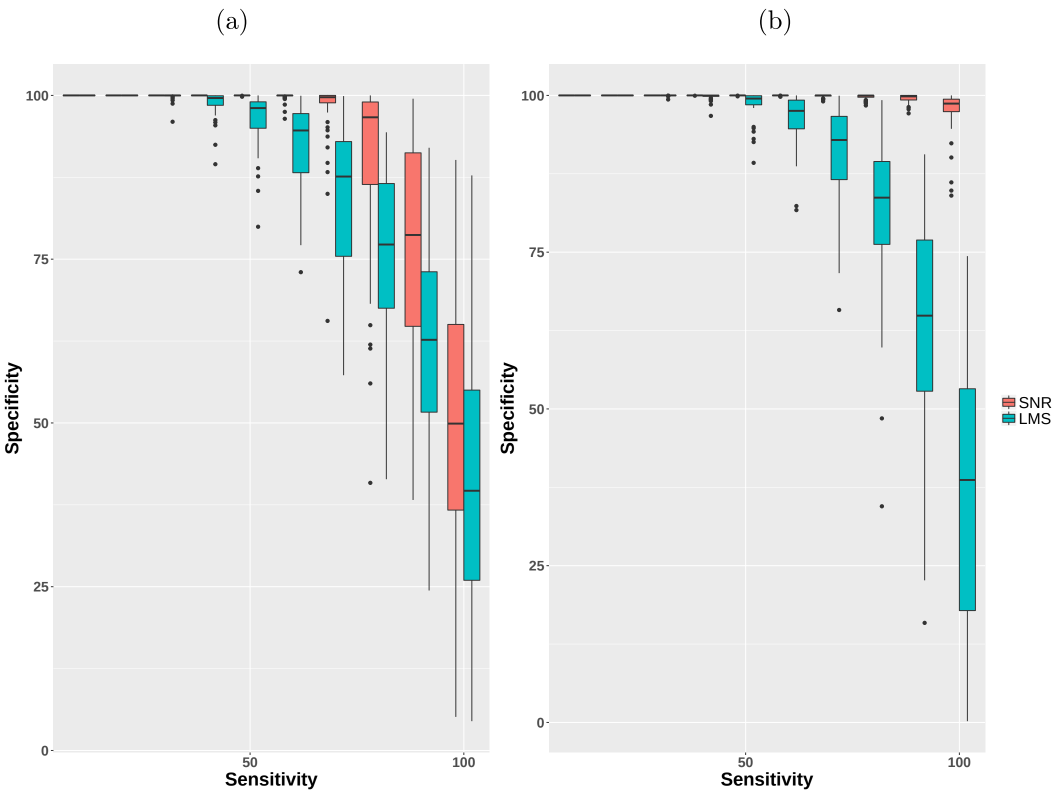


Figure 3.5.4: Box plots of specificity corresponding to SNR-based screening and likelihood-based marginal screening (LMS). Results obtained from 50 replicates where $(p, n, J, |T|) = (2000, 88, 20, 10)$ for settings with (a) Weakly correlated $\mathbf{B}$ (b) Highly correlated $\mathbf{B}$.

Results illustrated in both Figure 3.5.4 and Figure 3.5.5 are another evidence of SNR-based screening efficiency. These results also reflect the beneficial effect of employing the covariance matrix of the response variable in enhancing the screening accuracy. Existing high correlations in the coefficient matrix, and as a result in the response variable, does not improve the LMS performance since the correlation is not taken into account in this type of screening. However, this high correlation can substantially increase the SNR-based screening accuracy.

In the following results which are depicted in Figure 3.5.5, we can see that in the setting with a larger number of active variables $|T| = 100$, identifying active predictors becomes more difficult in both approaches. This is a result of a higher correlation structure caused by a larger number of active predictors. This effect is reflected in having lower specificity values for setting with 100 active coefficients. This phenomenon called mask effect is studied in more detail in Section 5.7.2.1

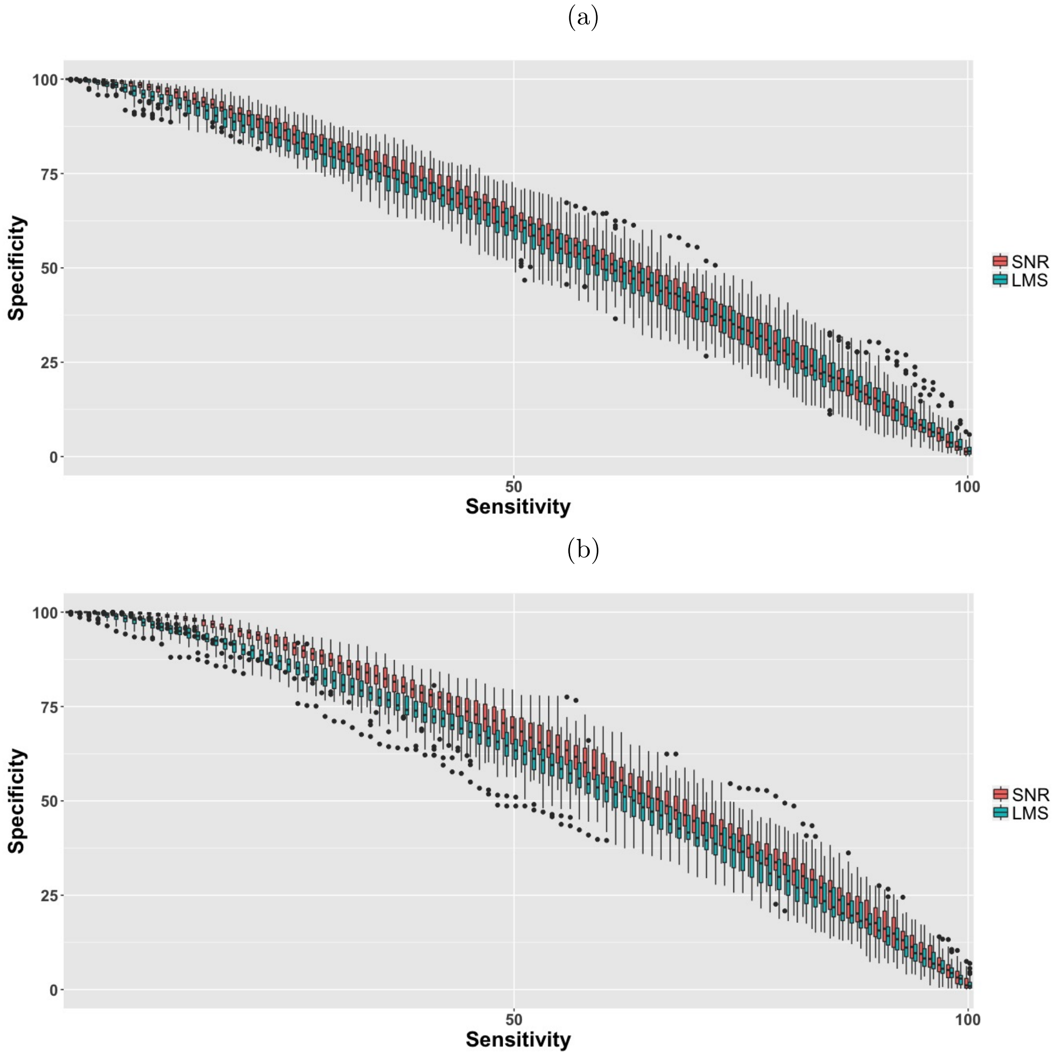


Figure 3.5.5: Box plots of specificity corresponding to SNR-based screening and likelihood-based marginal screening (LMS). Results obtained from 50 replicates where $(p, n, J, |T|) = (2000, 88, 20, 100)$ for settings with (a) Weakly correlated $\mathbf{B}$ (b) Highly correlated $\mathbf{B}$.

3.6 Real data application

All three introduced screening approaches were applied to real data where the predictor variables are of high dimension. The real data contain gene expression levels of 13321 genes and the (IC50) values of 131 drugs across 42 cell lines. All gene expression values and IC50 values were log-transformed. Let $\mathbf{X}_{42\times 13321}$

denote a design matrix, each column of which contain the expression levels of 42 cell lines and columns of $\mathbf{Y}_{42\times131}$ contain IC50 values of drugs across the same cell lines. Prior to any calculation data were standardised as explained in Section 3.3. We considered the multivariate multiple regression model (3.3.1) where $(p, J, n) = (13321, 131, 42)$ and we used the PII and the SNR statistics to screen the gene expressions. To this aim, we calculated the PII and SNR values corresponding to each gene expression $\mathbf{x}_k; k = 1, \cdots, 13321$. Then we sorted all these values in a decreasing order.

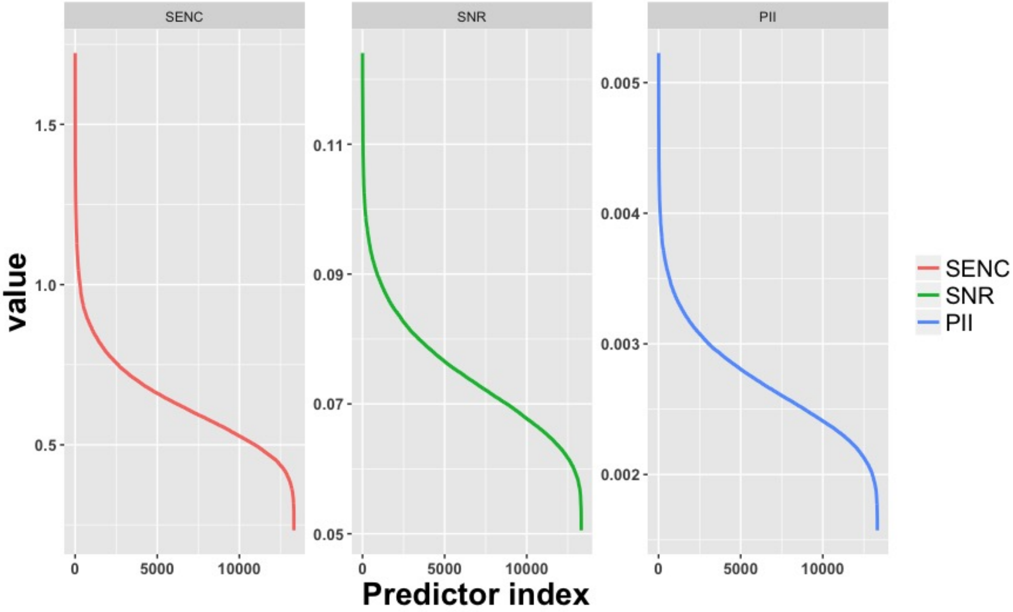
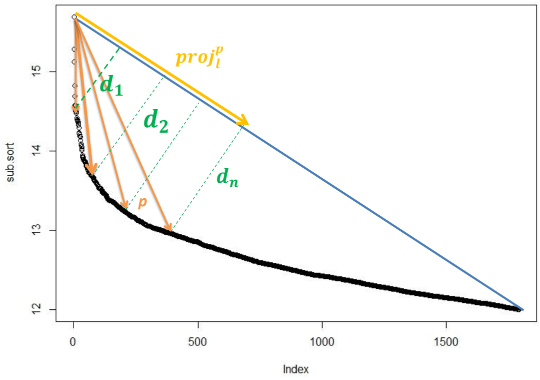


Figure 3.6.1: First and second from right: Predictive Information Index (PII), Signal-to-Noise Ratio (SNR) curves wherein PII snd SNR values obtained based on fitting a multivariate multiple regression to gene expressions and IC50 values. PII and SNR values obtained using shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.005$. First plot on the left shows the squared Euclidian norm of estimated coefficients (SENC) obtained from fitting multivariate single regression to each gene expression and IC50 values of all drugs. In all calculations the dimensions of the are $(p, n, J) = (13321, 42, 131)$.

We also performed likelihood-based marginal screening (LMS) by fitting a multivariate single regression to each predictor as it was explained in Section 3.3 and we found the corresponding estimates shown in Equation (3.3.3). Then we calculated the squared Euclidian norm of these estimates shown in Equation (3.3.5). The plots in Figure 3.6.1 show the SNR and PII values together with the squared Euclidian norm of estimated coefficients (SENC) for all gene expressions.

In order to select the most important predictors we thresholded the above values at the elbow point of each curve, i.e. the point at which the curvature is maximum. Elbow criterion is a graphical way to select the number of clusters in data. To find the elbow point we considered the vector between the highest and the lowest value on each curve then we found the distance between each point on the curve and this vector. The point on the curve with the largest distance from this vector was selected as elbow point. The following plot explains this on a subset of SNR values.

Figure 3.6.2: Finding elbow point on a curve



Suppose we have n sorted SNR values. If we denote the vector between the largest value and the i-th point on the curve by $\mathbf{p}_i$ (the dark orange vector), then the vector $\mathbf{d}_i = \mathbf{p}_i - \mathbf{proj}_l^{p_i}, i = 1, \cdots, n$ gives the distance vector corresponding to each point on the curve. We find the length of vectors $\mathbf{d}_i; i = 1, \cdots, n$ and the point on the curve with the maximum distance will be the desired elbow point.

Having obtained the PII, SNR and SENC values corresponding to all gene expressions, we then found the elbow point of each curve in Figure 3.6.1. We selected gene expressions that their PII, SNR and SENC values are larger than the elbow point. The following table shows the number of selected genes and the time taken for each screening procedure.

Method of screening	No.of selected genes	Time taken (s)
PII	1037	3.8
SNR	1316	3.6
LMS	1038	1.3

Table 1: The number of selected genes after applying different screening methods on real data containing $p = 13321$ expression levels of $n = 42$ cell lines together with IC50 values of $J = 131$ drugs across these cell lines. Time is recorded in seconds.

These methods have 215 genes in common which means that these 215 genes have been selected by all three methods after thresholding at elbow point. However, the number of common genes selected by SNR and PII method is 686.

Now we need to examine the screening validity which was performed by these methods on real data. Suppose we wish to evaluate the screening procedure conducted by PII. To this aim, we calculated the PII values corresponding to all predictors in real data. Then we sorted these values in a decreasing order and we chose a subset of predictors with the highest values of PII. For example, we chose predictors the PII values of which were among the first m highest values. Let these predictors form the design matrix $\mathbf{X}^*_{42\times m} = (\mathbf{x}^*_{(1)} \cdots \mathbf{x}^*_{(m)})$. We then fitted a multivariate multiple regression to $\mathbf{X}^*_{42\times m}$ and IC50 values, $\mathbf{Y}_{42\times 131}$. So we obtained the estimated coefficients $\mathbf{b}^*_{(1)}, \cdots, \mathbf{b}^*_{(m)}$ with the corresponding variance of $\sigma^2_{*(1)}, \cdots, \sigma^2_{*(m)}$. Then, we used these estimates to generate bootstrap samples. To simulate bootstrap samples we generated the bootstrap response variable $\mathbf{Y}^b_{42\times 131}$ from the following model

$$\mathbf{Y}^b_{42\times 131} = \mathbf{X}_{42\times 13321}\tilde{\mathbf{B}}_{13321\times 131} + \tilde{\mathbf{E}}_{42\times 131},$$

where $\tilde{\mathbf{B}}$ is a sparse matrix with all rows except $\mathbf{b}^*_{(1)}, \cdots, \mathbf{b}^*_{(m)}$ equal to zero. The error term $\tilde{\mathbf{E}}_{42\times 131}$ is simulated from the multivariate normal distribution $N(\mathbf{0}, \sigma^2\mathbf{I}_n)$ with $\sigma^2 = \frac{1}{m}\sum_{i=1}^m \sigma^2_{*(i)}$. This was followed by PII, SNR and likelihood-based marginal screening on these bootstrap data. We applied all three screening methods on 2000 sets of generated bootstrap data and calculated the corresponding sensitivity and specificity. The result of screening on bootstrap data is shown Figure 3.6.3. The first row shows the results when PII was applied to real data as we explained above. We repeated the above procedure for two other methods and generated bootstrap samples.

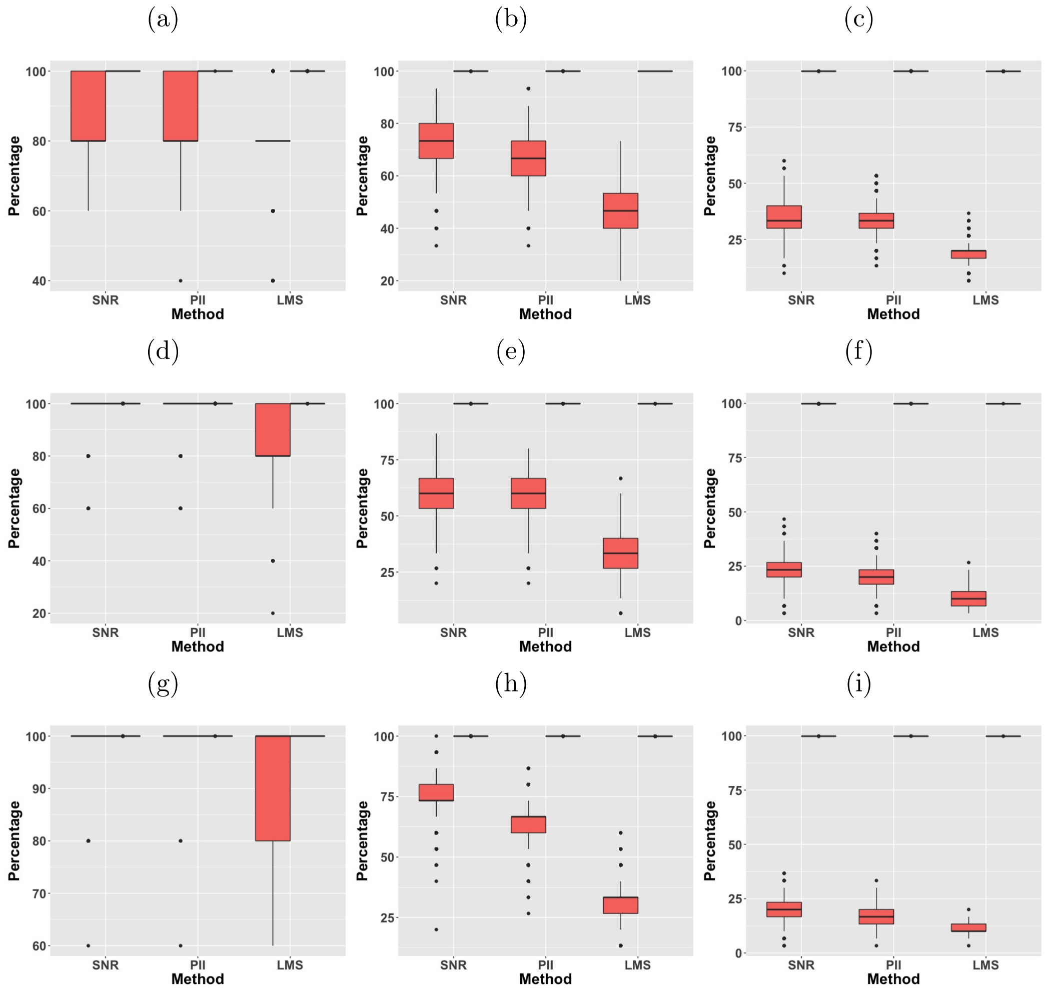

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 3.6.3: Box plots of sensitivity (red boxes) and fixed specificity values (straight dark lines) obtained from applying screening methods on 2000 bootstrap samples. The first row shows the results when PII was applied to real data then the first (a) $m = 5$, (b) $m = 15$, (c) $m = 30$ predictors were considered as active in bootstrap data. The second row shows the results when SNR was applied to real data then the first (d) $m = 5$, (e) $m = 15$, (f) $m = 30$ predictors were considered as active in bootstrap data. The third row shows the results when LMS was applied to real data then the first (g) $m = 5$, (h) $m = 15$, (i) $m = 30$ predictors were considered as active in bootstrap data.

The first row of Figure 3.6.3 shows the screening result when PII was first applied to the real data and the values sorted decreasingly and the first $m = 5, 15, 30$ predictors were selected to be used as active predictors in the bootstrap samples. The second and third row show the result when SNR and LMS first applied to the real data. As we expected from simulation studies reported in previous section, SNR-based screening surpasses both PII and LMS methods. Sensitivity percentages show the proportion of correctly identified active predictors and the specificity percentages quantify the proportion of correctly discarded non-active predictors. Since we set the number of active predictors to $m = 5, 15, 30$; for each m, the number of non-zero predictors and as a result the specificity levels are equal for all methods. However, the ability of recovering the true active predictors vary across the three methods leading to different values of sensitivity. Higher sensitivity levels corresponding to screening by SNR reflect that this method possesses a higher ability of detecting the true actives compare to other two methods.

The results obtained in simulation study and in bootstrapping from real data, approve that our final proposed screening method i.e. screening by SNR, is promising and this motivates us to further employ SNR in our variable selection procedure introduced in the next chapter.

Chapter 4

Principal Variable Analysis

4.1 Introduction

In the previous chapter, we have pointed out that some of the existing variable screening methods have not taken into account the correlation structure of multivariate response variables. This shortcoming may lead to a biased result in variable screening. To address this issue, we have proposed a selection procedure based on the response covariance matrix and showed that the proposed method improves the performance of variable screening.

Here, we further improve the above screening procedure, considering not only the correlation structure in the multivariate response, but also the high correlations between predictors. We reduce the effect of these correlations, by introducing a procedure called *principal variable analysis (PVA)*. In PVA we add more constraints to the optimisation procedure in order to suppress the interference with other predictors. This results in a more accurate selection.

4.2 Principal variable selection

The PVA contains the following steps. In the first step, we initialise the procedure by finding the maximum SNR value of predictors. In the second step, we iteratively run a forward nulling and selection until the stopping criterion in section 4.2.2 is met.

4.2.1 Forward nulling and selection

In the following, to facilitate the presentation, we first show the details of the first three iterations followed by a generalisation to any iteration.

Consider the dataset $(\mathbf{Y}, \mathbf{X})$ and the corresponding multivariate regression model (3.3.1) where $\mathbf{Y} = \mathbf{Y}_{n\times J} = (y_{ij}) = (\mathbf{y}_1\mathbf{y}_2\cdots\mathbf{y}_J)$, $\mathbf{X} = \mathbf{X}_{n\times p} = (x_{ik}) = (\mathbf{x}_1\cdots\mathbf{x}_p)$, and $\mathbf{y}_j; 1 \leq j \leq J$ and $\mathbf{x}_k; 1 \leq k \leq p$ are the vectors of n observations made on the response variables and the predictors. The process is designed as follows. In the first iteration, we optimise the following objective function

$$\min_{\mathbf{w}_k} \ (\mathbf{w}_k^T\hat{\mathbf{C}}\mathbf{w}_k), \quad s.t \quad \mathbf{w}_k^T\mathbf{x}_k \ = \ 1$$

and we find the optimal direction and obtain the predictive information index (PII) as it was calculated in the previous chapter. Then, we normalise the PII and obtain the SNR statistic. Therefore, in the first iteration the SNR values for each predictor $\mathbf{x}_k; k = 1\cdots p$ are calculated by

$$\text{SNR}_k^{(1)} = \frac{\mathbf{x}_k^T\hat{\mathbf{C}}^{-1}\mathbf{x}_k}{\mathbf{x}_k^T\hat{\mathbf{C}}^{-2}\mathbf{x}_k},$$

or equivalently, the SNR expression in the first iteration can be expressed in terms of the optimal weight vector as

$$\text{SNR}_k^{(1)} = \frac{\hat{\mathbf{w}}_k^{T(1)}\hat{\mathbf{C}}^{-1}\hat{\mathbf{w}}_k^{(1)}}{\hat{\mathbf{w}}_k^{T(1)}\hat{\mathbf{w}}_k^{(1)}}.$$

Let $\mathbf{x}_{k_1}$ be the predictor in which the SNR attains the maximum, in the sense that

$$\mathrm{SNR}_{k_1} = \max_{1\leq k\leq p} \frac{\hat{\mathbf{w}}_k^{T(1)}\hat{\mathbf{C}}^{-1}\hat{\mathbf{w}}_k^{(1)}}{\hat{\mathbf{w}}_k^{T(1)}\hat{\mathbf{w}}_k^{(1)}}.$$

Having found $\mathbf{x}_{k_1}$ as the predictor with the highest SNR, in the next (second) iteration, we null $\mathbf{x}_{k_1}$ and solve the following optimisation problem:

$$\begin{aligned} \min_{\mathbf{w}_k} \; (\mathbf{w}_k^T\hat{\mathbf{C}}\mathbf{w}_k), \quad s.t \quad \mathbf{w}_k^T\mathbf{x}_k &= 1 \\ \mathbf{w}_k^T\mathbf{x}_{k_1} &= 0. \end{aligned} \tag{4.2.1}$$

We utilise the method of Lagrange multiplier to solve the above optimisation problem. Let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ denote the Lagrange multiplier vector. Then the Lagrangian function $\mathcal{L}$ is of the form

$$\mathcal{L}(\mathbf{w}_k, \boldsymbol{\lambda}) \;=\; \mathbf{w}_k^T\hat{\mathbf{C}}\mathbf{w}_k - \boldsymbol{\lambda}\left[(\mathbf{x}_k, \mathbf{x}_{k_1})^T\mathbf{w}_k - (1,0)^T\right].$$

Differentiating the Lagrangian function $\mathcal{L}(\mathbf{w}_k, \lambda)$ with respect to $\mathbf{w}_k$ gives

$$\frac{\partial\mathcal{L}(\mathbf{w}_k, \boldsymbol{\lambda})}{\partial\mathbf{w}_k} \;=\; 2\hat{\mathbf{C}}\mathbf{w}_k - \boldsymbol{\lambda}(\mathbf{x}_k, \mathbf{x}_{k_1})^T.$$

Setting the above equation equal to zero yields the optimum direction vector

$$\hat{\mathbf{w}}_k = \frac{1}{2}(\lambda_1, \lambda_2)(\mathbf{x}_k, \mathbf{x}_{k_1})^T\hat{\mathbf{C}}^{-1}. \tag{4.2.2}$$

To obtain the Lagrange multiplier vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$, we substitute the optimal direction vector $\hat{\mathbf{w}}_k$ into the constraints, $(\mathbf{x}_k, \mathbf{x}_{k_1})^T\hat{\mathbf{w}}_k = (1,0)^T$. This results in

$$(\lambda_1, \lambda_2)^T = -2\left((\mathbf{x}_k, \mathbf{x}_{k_1})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k, \mathbf{x}_{k_1})\right)^{-1}(1,0)^T.$$

Finally, substituting this vector into (4.2.2), we have the optimal direction

$$\hat{\mathbf{w}}_k^{(2)} = \hat{\mathbf{C}}^{-1}(\mathbf{x}_k, \mathbf{x}_{k_1})\left((\mathbf{x}_k, \mathbf{x}_{k_1})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\ \mathbf{x}_{k_1})\right)^{-1}(1,0)^T.$$

Substituting $\hat{\mathbf{w}}_k^{(2)}$ in to the objective function $\mathbf{w}_k^T\hat{\mathbf{C}}\mathbf{w}_k$, we have

$$\text{SNR}_k^{(2)} = \frac{\hat{\mathbf{w}}_k^{T(2)}\hat{\mathbf{C}}^{-1}\hat{\mathbf{w}}_k^{(2)}}{\hat{\mathbf{w}}_k^{T(2)}\hat{\mathbf{w}}_k^{(2)}}, \tag{4.2.3}$$

where

$$\hat{\mathbf{w}}_k^{T(2)}\hat{\mathbf{C}}^{-1}\hat{\mathbf{w}}_k^{(2)} = (1,0)\left((\mathbf{x}_k,\mathbf{x}_{k_1})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\mathbf{x}_{k_1})\right)^{-1}(1,0)^T,$$

and

$$\begin{aligned}\hat{\mathbf{w}}_k^{T(2)}\hat{\mathbf{w}}_k^{(2)} &= (1,0)\left((\mathbf{x}_k,\mathbf{x}_{k_1})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\ \mathbf{x}_{k_1})\right)^{-1}(\mathbf{x}_k,\mathbf{x}_{k_1})^T\hat{\mathbf{C}}^{-2}(\mathbf{x}_k,\mathbf{x}_{k_1})\\ &\quad \left((\mathbf{x}_k,\mathbf{x}_{k_1})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\mathbf{x}_{k_1})\right)^{-1}(1,0)^T.\end{aligned}$$

Lastly, the second iteration is completed by finding k_2 such that

$$\text{SNR}_{k_2} = \max_{k\neq k_1}\text{SNR}_k^{(2)},$$

Now given that $\{\mathbf{x}_{k_1},\mathbf{x}_{k_2}\}$ are identified predictors in the previous steps, in the third iteration we solve

$$\begin{aligned}\min_{\mathbf{w}_k}\ (\mathbf{w}_k^T\hat{\mathbf{C}}\mathbf{w}_k),\quad s.t\quad \mathbf{w}_k^T\mathbf{x}_k &= 1\\ \mathbf{w}_k^T\mathbf{x}_{k_1} &= 0\\ \mathbf{w}_k^T\mathbf{x}_{k_2} &= 0.\end{aligned}$$

Similar to the second iteration, implementing the Lagrange multiplier method, the optimal direction in the third iteration is derived as

$$\hat{\mathbf{w}}_k^{(3)} = \hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})\left((\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})\right)^{-1}(1,0,0)^T,$$

accordingly, the SNR values in the third iteration are attained through the expression

$$\text{SNR}_k^{(3)} = \frac{\hat{\mathbf{w}}_k^{T(3)}\hat{\mathbf{C}}^{-1}\hat{\mathbf{w}}_k^{(3)}}{\hat{\mathbf{w}}_k^{T(3)}\hat{\mathbf{w}}_k^{(3)}}, \tag{4.2.4}$$

where

$$\hat{\mathbf{w}}_k^{T(3)}\hat{\mathbf{C}}^{-1}\hat{\mathbf{w}}_k^{(3)} = (1,0,0)\left((\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})\right)^{-1}(1,0,0)^T,$$

and

$$\begin{aligned}\hat{\mathbf{w}}_k^{T(3)}\hat{\mathbf{w}}_k^{(3)} &= (1,0,0)\left((\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\ \mathbf{x}_{k_1},\mathbf{x}_{k_2})\right)^{-1}\\ &\quad (\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})^T\hat{\mathbf{C}}^{-2}(\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})\\ &\quad \left((\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})^T\hat{\mathbf{C}}^{-1}(\mathbf{x}_k,\mathbf{x}_{k_1},\mathbf{x}_{k_2})\right)^{-1}(1,0,0)^T.\end{aligned}$$

This iteration is terminated by finding the predictor for which the following expression holds

$$\text{SNR}_{k_3} = \max_{k\notin\{k_1,k_2\}} \text{SNR}_k^{(2)}.$$

We now generalise the above process to the m-th iteration as follows. Suppose $\mathbf{X}_{n\times p} = (\mathbf{x}_1,\cdots,\mathbf{x}_p)$ represents all the predictors in the data and $\mathcal{S}_{m-1} = \{k_1,\cdots,k_{m-1}\}$ denotes the set of indices corresponding to predictors with maximum SNR values prior to the m-th iteration. Thus, the rest of predictors form the matrix $\check{\mathbf{X}} = \mathbf{X}_{n\times|\mathcal{S}_{m-1}^c|}$ where $\mathcal{S}_{m-1}^c = \{1 \le i \le p; i \notin \mathcal{S}_{m-1}\}$. Hereafter, we shall use $\mathbf{x}_{\{k\}\cup\mathcal{S}_{m-1}}$ to denote a vector, the first element of which is the predictor $\mathbf{x}_k$, $k \in \mathcal{S}_{m-1}^c$. The rest of the elements of $\mathbf{x}_{\{k\}\cup\mathcal{S}_{m-1}}$ are the identified predictors in the previous $m-1$ steps. More precisely, we have $\mathbf{x}_{\{k\}\cup\mathcal{S}_{m-1}} = (\mathbf{x}_k,\mathbf{x}_1,\cdots,\mathbf{x}_{m-1})$. In the the m-th iteration the optimization problem to be solved is of the form

$$\begin{aligned}\min_{\mathbf{w}_k}\ (\mathbf{w}_k^T\hat{\mathbf{C}}\mathbf{w}_k),\quad s.t\quad \mathbf{w}_k^T\mathbf{x}_k &= 1\\ \mathbf{w}_k^T\mathbf{x}_{k_1} &= 0\\ &\vdots\\ \mathbf{w}_k^T\mathbf{x}_{k_{m-1}} &= 0,\end{aligned} \tag{4.2.5}$$

where $k \in \mathcal{S}_{m-1}^c$. Hence, the Lagrangian function with the corresponding

Lagrange multiplier vector $\boldsymbol{\lambda}^{(m)} = (\lambda_1, \cdots, \lambda_m)$ takes the form

$$\mathcal{L}^{(m)}(\mathbf{w}_k, \boldsymbol{\lambda}^{(m)}) \quad = \quad \mathbf{w}_k^T \hat{\mathbf{C}} \mathbf{w}_k - \boldsymbol{\lambda}^{(m)} \left(\mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \mathbf{w}_k - \mathbf{e}_m^T \right),$$

where $\mathbf{e}_m = (1, 0, \cdots, 0)$ is an m-vector. The first element of $\mathbf{e}_m$, which is equal to one, guarantees that the information related to the k-th predictor under consideration passes through the filter. Zeros imply that no information from other predictors, which have already been detected in the previous steps, is included in the calculation. Differentiating the Lagrangian $\mathcal{L}^{(m)}(\mathbf{w}_k, \boldsymbol{\lambda}^{(m)})$ function with respect to $\mathbf{w}_k$ and setting the equation equal to zero, the optimal direction is obtained as

$$\hat{\mathbf{w}}_k = \frac{1}{2} \boldsymbol{\lambda}^{(m)} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \hat{\mathbf{C}}^{-1}. \tag{4.2.6}$$

To obtain the Lagrange multiplier vector $\boldsymbol{\lambda}^{(m)}$, we substitute the above optimal direction vector $\hat{\mathbf{w}}_k$ into the constraints, $\mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \hat{\mathbf{w}}_k = \mathbf{e}_m^T$ which yields

$$\boldsymbol{\lambda}^{(m)} = -2 \left(\mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \hat{\mathbf{C}}^{-1} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}} \right)^{-1} \mathbf{e}_m^T.$$

Substituting this vector into Equation (4.2.6), we obtain the optimal direction in the m-th iteration

$$\hat{\mathbf{w}}_k^{(m)} = \hat{\mathbf{C}}^{-1} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}} \left(\mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \hat{\mathbf{C}}^{-1} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}} \right)^{-1} \mathbf{e}_m^T.$$

Consequently, the *nulled predictive information index* in the m-th iteration, which is expressed as the variance of the projected data along $\hat{\mathbf{w}}$, is of the form

$$\hat{r}_{k|\mathcal{S}_{m-1}} = \hat{\mathbf{w}}_k^{T(m)} \hat{\mathbf{C}}^{-1} \hat{\mathbf{w}}_k^{(m)} = \mathbf{e}_m \left(\mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \hat{\mathbf{C}}^{-1} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}} \right)^{-1} \mathbf{e}_m^T.$$

and we have

$$\begin{aligned} \hat{\mathbf{w}}_k^{T(m)} \hat{\mathbf{w}}_k^{(m)} \quad &= \quad \mathbf{e}_m \left(\mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \hat{\mathbf{C}}^{-1} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \right)^{-1} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \hat{\mathbf{C}}^{-2} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}} \\ &\times \quad \left(\mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}}^T \hat{\mathbf{C}}^{-1} \mathbf{x}_{\{k\} \cup \mathcal{S}_{m-1}} \right)^{-1} \mathbf{e}_m^T. \end{aligned}$$

As a result, the SNR value in iteration m for each predictor $\mathbf{x}_k$, where $k \notin \mathcal{S}_{m-1}$, is calculated by

$$\mathrm{SNR}_k^{(m)} \propto \frac{\hat{\mathbf{w}}_k^{T(m)} \hat{\mathbf{C}}^{-1} \hat{\mathbf{w}}_k^{(m)}}{\hat{\mathbf{w}}_k^{T(m)} \hat{\mathbf{w}}_k^{(m)}}, \quad k \notin \mathcal{S}_{m-1} \tag{4.2.7}$$

Once we have found the above nulled $\mathrm{SNR}_k^{(m)}$ values, the algorithm proceeds by finding k_m such that

$$\mathrm{SNR}_{k_m} = \max_{k \notin \mathcal{S}_{m-1}} \mathrm{SNR}_k^{(m)}. \tag{4.2.8}$$

The predictor $\mathbf{x}_{k_m}$ is removed from $\check{\mathbf{X}}$. Accordingly, the index set $\mathcal{S}_{m-1} = \{k_1, \cdots, k_{m-1}\}$ is updated to the set $\mathcal{S}_m = \{k_1, \cdots, k_m\}$.

This is called the *forward nulling* phase since it is based on nulling the previously identified predictors by imposing multiple constraints on the minimum variance filter. Consequently, the minimum variance filters with such constraints are called *nulled-beamformers*. Indeed, the aim of the forward nulling and selection is to scan through the predictor space with a series of nulled-beamformers, each is tailored to a particular region in the space and resistant to interference effects originating from other regions. After a certain number of iterations, the SNR values start leveling off, such that the maximum SNR value does not differ substantially from the rest of the values anymore. This motivates us to define a stopping rule for the forward nulling.

4.2.2 Stopping criterion

Leveling off the SNR values gradually conveys that the remaining predictors are not outstanding any more and do not contain much information about the response variable. If no significant information is left in the remaining predictors we terminate the selection. To be concrete, in iteration m, after calculating $\mathrm{SNR}_k^{(m)}$ values for all predictors $\mathbf{x}_k; k \notin \mathcal{S}_{m-1}$, we sort these values decreasingly and identify the elbow point δ_m as discussed in Section 3.6.

Algorithm 4.1: Principal Variable Analysis (PVA)

```
1. Calculate $\textsc{snr}_k$ for all predictors $\mathbf{x}_k;\ k = 1, \cdots, p$.
2. In the first iteration, find $k_1$ such that $\textsc{snr}_{k_1} = \max_{1 \le k \le p} \textsc{snr}_k$ and define the nulled indices set $\mathcal{S}_1 = \{k_1\}$.
3. In iteration $m \ge 2$, calculate $\textsc{snr}_k^{(m)} | \mathcal{S}_{m-1} = \{\textsc{snr}_k^{(m)}; k \notin \mathcal{S}_{m-1}\}$ and find $k_m$ such that $\textsc{snr}_{k_m} = \max_{k \notin \mathcal{S}_{m-1}} \textsc{snr}_k^{(m)}$.
4. Order $\textsc{snr}_k^{(m)} | \mathcal{S}_{m-1}$ values and find the elbow point $\delta_m$ of this curve.
5. Identify the noise set $\mathcal{N}_{\delta_m} = \{\mathbf{x}_k \mid \underset{k \notin \mathcal{S}_{m-1}}{\text{SNR}_k^{(m)}} < \delta_m\}$.
6. Find $\mu_{\mathcal{N}_{\delta_m}}$ and $\sigma_{\mathcal{N}_{\delta_m}}$.
7. If $|\textsc{snr}_{k_m} - \mu_{\hat{I}^c}| \le a_0 \sigma_{\mathcal{N}_{\delta_m}}$ is true stop and set $\mathcal{S}_m = \mathcal{S}_{m-1} \cup \{k_m\}$; else return to step 3.
```

Predictors with SNR values higher than the the elbow point are considered as an approximate signal set, and the remainder is classified as noise. Hence, in iteration m, the approximate signal set $\mathcal{A}_{\delta_m}$ is defined as $\{\mathbf{x}_k \mid \underset{k \notin \mathcal{S}_{m-1}}{\text{SNR}_k^{(m)}} \ge \delta_m\}$ and the noise set $\mathcal{N}_{\delta_m}$ is defined as $\{\mathbf{x}_k \mid \underset{k \notin \mathcal{S}_{m-1}}{\text{SNR}_k^{(m)}} < \delta_m\}$. Now we check whether the maximum of SNR values in iteration m, SNR_{k_m}, satisfies the following condition

$$|\text{SNR}_{k_m} - \mu_{\mathcal{N}_{\delta_m}}| \le a_0 \sigma_{\mathcal{N}_{\delta_m}}, \tag{4.2.9}$$

where $\mu_{\mathcal{N}_{\delta_m}}$ is the mean and $\sigma_{\mathcal{N}_{\delta_m}}$ is the standard deviation of the noise set. In order to specify a_0 we considered three-sigma rule (Hazewinkel, 1993) and five-sigma rule (Collins, 2014) which is mostly used in practical experiments in physics (Acton, 2013). Accordingly, in PVA process we consider $a_0 \in \{5, 3\}$. If SNR_{k_m} falls into the above interval, iterations will stop. Otherwise k_m is merged with the set of nulled indices $\mathcal{S}_{m-1}$, i.e. $\mathcal{S}_m = \mathcal{S}_{m-1} \cup \{k_m\}$ and iterations will continue. Once the process stops, we obtain a list of highly ranked predictors called *principal predictors*. The steps of PVA are summarised in Algorithm 4.1.

4.3 Theoretical support

In this section we provide some theoretical support to show the properties of the predictive information index (PII) and the PVA. Some of the theories are beyond the scope of this thesis so the details are provided in the paper (Zhang and Oftadeh, 2016) which is based on this chapter and Chapter 3. Note that all theory in this part is based on PVA with an ideal setting where $\mathbf{C}$ is known. PVA with estimated $\mathbf{C}$ is discussed in the paper in details.

We remind that we consider a sample $(\mathbf{Y}, \mathbf{X})$ of size n on the response variables and predictors, where $\mathbf{Y} = \mathbf{Y}_{n\times J} = (y_{ij}) = (\mathbf{y}_1\mathbf{y}_2\cdots\mathbf{y}_J)$ and $\mathbf{X} = \mathbf{X}_{n\times p} = (x_{ik}) = (\mathbf{x}_1\cdots\mathbf{x}_p)$, and $\mathbf{y}_j; 1 \leq j \leq J$ and $\mathbf{x}_k; 1 \leq k \leq p$ are vectors of n observations made on the response variables and the predictors. Suppose the data follows the multivariate multiple regression model

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}, \tag{4.3.1}$$

where $\mathbf{B} = \mathbf{B}_{p\times J} = (b_{ij}) = (\mathbf{B}_1\mathbf{B}_2\cdots\mathbf{B}_J)$ and $\mathbf{E} = \mathbf{E}_{n\times J} = (\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_2\cdots\boldsymbol{\varepsilon}_J)$ with $\mathbf{b}_j$ and $\boldsymbol{\varepsilon}_j$ respectively denote the values of the random regression coefficients and the error terms related to the jth response variable. Assume that $\mathbf{B}$ and $\mathbf{E}$ are independent and that the covariance matrices of $\mathbf{y}_j$, $\mathbf{b}_j$ and $\boldsymbol{\varepsilon}_j$, denoted by $\mathbf{C} = (c_{ij})_{n\times n}$, $\Sigma = (\gamma_{ij})_{p\times p}$ and $\sigma^2 I_n$ respectively, where I_n is the $n \times n$ identity matrix. Also assume that these covariances are independent of index j. Therefore, for all $j = 1, \cdots, J$ we have $\mathbf{C} = \mathrm{cov}(\mathrm{y}_j)$, also form a population perspective we have $\mathrm{cov}(\mathbf{y}_j) = \mathbf{C} = E[(\mathbf{y}_j - E[\mathbf{y}_j])(\mathbf{y}_j - E[\mathbf{y}_j])^T]$. Since the regression coefficients are assumed to be random, the covariance structure of each response variable is determined by the covariance of random coefficients and the error term:

$$\mathbf{C} = \mathbf{X}\Sigma\mathbf{X}^T + \sigma^2 I_n. \tag{4.3.2}$$

Without loss of generality, assume that $\mathbf{x}_k^T\mathbf{x}_k = n, 1 \leq k \leq n$. Otherwise, this can be achieved by a standardisation procedure. Let γ_k^2 denote γ_{kk} in the covariance matrix of regression coefficients Σ, which corresponds to the regression coefficient at the predictor k. The variable selection aims to detect predictors that their regression coefficients have non-zero variances. Let set $\{1, 2, ..., p\}$ denote all the predictors in the data and ν_0 be the true predictor set. If $\nu = \{k_1, ..., k_{p_1}\}$ denotes any subset of predictors $\{1, 2, ..., p\}$, then the $(k_1, ..., k_{p_1})$th columns of $\mathbf{X}$ form the matrix $\mathbf{X}_\nu$. Let $\mathbf{e}_\nu$ be a $p \times p_1$ selection matrix in which for $1 \leq j \leq p_1$, its (k_j, j)-th entry takes value of 1 and the other entries take values of 0, then we can write $\mathbf{X}_\nu = \mathbf{X}\mathbf{e}_\nu$. Thus with this notation for the true predictor set we have

$$\mathbf{C} = \mathbf{X}_{\nu_0}\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0}\mathbf{X}_{\nu_0}^T + A_{\nu_0}, \tag{4.3.3}$$

where, A_{ν_0} denote the remainder of $\mathbf{C}$ after the term $\mathbf{X}_{\nu_0}\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0}\mathbf{X}_{\nu_0}^T$ is taken away. In the following proposition we show that the predictive information or the predictive power at ν_0, which was defined in (3.4.6), can be decomposed into the underlying predictive information matrix of the predictors in ν_0 plus the interferences from the predictors not in ν_0 and from the white noise.

Proposition 4.3.1. *If $\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0}$ and A_{ν_0} are invertible and $\mathbf{X}_{\nu_0}$ has the rank equal to the size of ν_0, then the predictive information matrix*

$$r_{\nu_0} = \mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0} + \left(\mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\right)^{-1}.$$

If $\gamma_k^2 = 0$, $k \notin \nu_0$ and as the sample size n is large enough, the minimum eigenvalue of $\mathbf{X}_{\nu_0}^T\mathbf{X}_{\nu_0}/n$ is bounded below from zero, then

$$r_{\nu_0} = \mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0} + O(1/n).$$

Proof: It follows from the definition that $\mathbf{C} \geq \sigma^2 I_n$ which implies C^{-1} has positive eigenvalues. Together with the fact that $\mathbf{X}_{\nu_0}$ has the rank equal to $|\nu_0|$, this shows that $\mathbf{X}_{\nu_0}^T\mathbf{C}^{-1}\mathbf{X}_{\nu_0}$ has positive eigenvalues. Similarly, we show that $\mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}$ is invertible. Now we invoke the Woodbury matrix identity. According to Hager (1989) Woodbury formula expresses the inverse of a matrix after a small rank perturbation in terms of the inverse of the original matrix. Using Woodbury formula for inverse of $\mathbf{C}$, we have

$$\mathbf{C}^{-1} = \mathbf{A}_{\nu_0}^{-1} - \mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\left((\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0})^{-1} + \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\right)^{-1}\mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}.$$

substituting the above expression for $\mathbf{C}^{-1}$ gives

$$\begin{aligned}
\mathbf{X}_{\nu_0}^T\mathbf{C}^{-1}\mathbf{X}_{\nu_0} &= \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0} - \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\left((\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0})^{-1} + \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\right)^{-1} \\
&\times \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0} \\
&= \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\left(\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0})^{-1} + \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\right)^{-1} \\
&\times \left((\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0})^{-1} + \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0} - \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\right) \\
&= \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\left(\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0})^{-1} + \mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0}\right)^{-1}(\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0})^{-1} \\
&= \left(\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0} + (\mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0})^{-1}\right)^{-1}.
\end{aligned}$$

By the definition, we have

$$r_{\nu_0} = \left(\mathbf{X}_{\nu_0}^T\mathbf{C}^{-1}\mathbf{X}_{\nu_0}\right)^{-1} = \mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0} + (\mathbf{X}_{\nu_0}^T\mathbf{A}_{\nu_0}^{-1}\mathbf{X}_{\nu_0})^{-1}.$$

When $\gamma_k^2 = 0$, $k \notin \nu_0$, we have $\mathbf{A} = \sigma^2 I_n$ and

$$\left(\mathbf{X}_{\nu_0}^T\mathbf{A}^{-1}\mathbf{X}_{\nu_0}\right)^{-1} = \frac{\sigma^2}{n}\left(\mathbf{X}_{\nu_0}^T\mathbf{X}_{\nu_0}/n\right)^{-1} = O\left(1/n\right).$$

The above proposition shows a local consistency of the predictive power with the underlying power $\mathbf{e}_{\nu_0}^T\Sigma\mathbf{e}_{\nu_0}$ at ν_0.

Most of the concepts and the related theory are not presented here as they are beyond the scope of this thesis. The following properties have been discussed and proved in the related paper (Zhang and Oftadeh, 2016).

(a) The global consistency property: The predictive power or predictive information index also has a global sparsistency property when the sample size tends to infinity. This property implies that for true active predictors that their regression coefficients have a positive variance, the predictive information index has a positive limit, whereas for non-active predictors the predictive information index tends to zero.

(b) Sure screening property: The screening procedure can have a sure screening property that for an appropriately chosen threshold, all predictors in ν_0 can be detected with a probability approaching to one. It can be shown that the forward nulling improves the accuracy of selection and the nulled predictive information index has higher values than non-nulled predictive information.

(c) Compared to the underlying predictive information matrix, $\mathbf{e}_{\nu_0}^T \Sigma \mathbf{e}_{\nu_0}$, the predictive information matrix r_{ν_0} may not be consistent if the correlation between the predictors do not converge to zero as n tends to infinity. It can be shown that under certain conditions, for any true predictor $k \in \nu_0$ the predictor does have positive predictive power although the power has deteriorated due to the interferences from other predictors. In the next section we introduce a concept which is used in the real data analysis.

4.4 Biological network

A biological network is a graphical representation through which several nodes are linked to each other. These nodes may be disease, protein, gene or other molecular characteristics. In the field of medicine, molecular networks play a crucial role in understanding human genetic disease by uncovering some hidden genomic associations. Human genetic disease are classified into three categories of monogenic, oligogenic and polygenic based on the number of genes that causes the disease. In monogenic disease mutation in single gene is essential and sufficient to cause the disease. Oligogenic diseases occur as a result of interaction between a few genes. Complex disease such as neurological and cancers which are multifactorial and involve many genes are called polygenic (Emmert-Streib, 2012). Because of the high influence that gene interactions may have in the disease progression, identifying disease causing genes and their associations can facilitate identifying the drug targets. Gene network is versatile tool for uncovering gene interactions. Through a detailed study of a gene network one can find potential drug targets for producing disease-specific treatments. Here, we focus on one type of network which is based on correlations between genes. We first give a mathematical definition of such a network and then we apply the method to the real data in the next section.

A *network* is a collection of inter-connected objects which is frequently presented as a number of points connected with a collection of lines. Networks conceptualise the interactions or relationships between a class of items. In mathematics, networks are often represented and referred to as *graphs*. A graph $G = (V, E)$ is a mathematical object composed by a set V of vertices or nodes which are connected by a set E of edges or links where elements of E are unordered pairs (u, v) of distinct vertices $u, v \in V$. The number of vertices is called the *order* and the number of edges is referred to as *size* of the graph. The connectivity of a graph is specified by the concept of *adjacency.* Two vertices $u, v \in V$ are said to be adjacent if they are connected by an edge in E. A vertex $v \in V$ is incident on an edge $e \in E$ if v is an endpoint of e therefore the degree of a vertex v is defined as the number of edges incident on v. A network is characterised by an adjacency matrix. Suppose graph $G = (V, E)$ corresponds to a network with N nodes. The adjacency matrix of graph G is an $N \times N$ symmetric matrix $A = (a_{ij})$ with entries

$$A_{ij} = \begin{cases} 1 & \text{if } i, j \in E \\ 0 & \text{otherwise,} \end{cases} \tag{4.4.1}$$

where an edge $e \in E$ is denoted by an unordered pair of vertices $i, j \in V$. $a_{ij} = 1$ if there is an edge between node i and j in graph G and a_{ij} is zero otherwise (Barabási, 2016).

4.5 Real data application

In the following sections we apply the PVA on the cancer data to find the principal genes. We then fit a multivariate regression to the selected set of genes and find the corresponding least square estimates. Then we identify a network of selected genes based on the correlation of their estimated coefficients.

4.5.1 Principal variable analysis on cancer data

As mentioned before, our real data contain log-expression levels of 13321 genes and the median inhibitory concentration (IC50) values of 131 drugs across 42 cell lines. Let $\mathbf{X}$ denote the log-expression levels and $\mathbf{Y}$ be the IC50 values. We consider the multivariate multiple regression model (4.3.1) for the data set $(\mathbf{Y}, \mathbf{X})$ where the sample size is $n = 42$ with $p = 13321$ predictors and $J = 131$ response variables. Evidently this is an ill-posed problem with high dimension predictors $p \gg n$ and $p \gg J$. We apply PVA to the data to identify the principal predictors. To this aim, we implement the PVA which is built upon the thresholded and shrunk covariance matrix $\hat{\mathbf{C}}_{hs}$ presented in Equation (3.4.11) with tuning constant $h = 0.001$. The result are not sensitive to the choice of c_0 for this particular data and we obtained the same set of selected predictors for different values of $h = 0.01, 0.005, 0, 001$. We also set the stopping rule to $a_0 = 5$, since choosing $a_0 = 3$ is computationally expensive. Simulation studies show that using three-sigma rather than five-sigma as stopping rule in PVA, does not improve the accuracy of selection substantially and just makes the process significantly longer. Therefore we set $a_0 = 5$ in application of PVA on real data which is more appropriate for such high dimensional data. The computational time for applying PVA on real data was 1.6 minutes on CPU with Intel Core i5-3470 processor and 8 GB RAM.

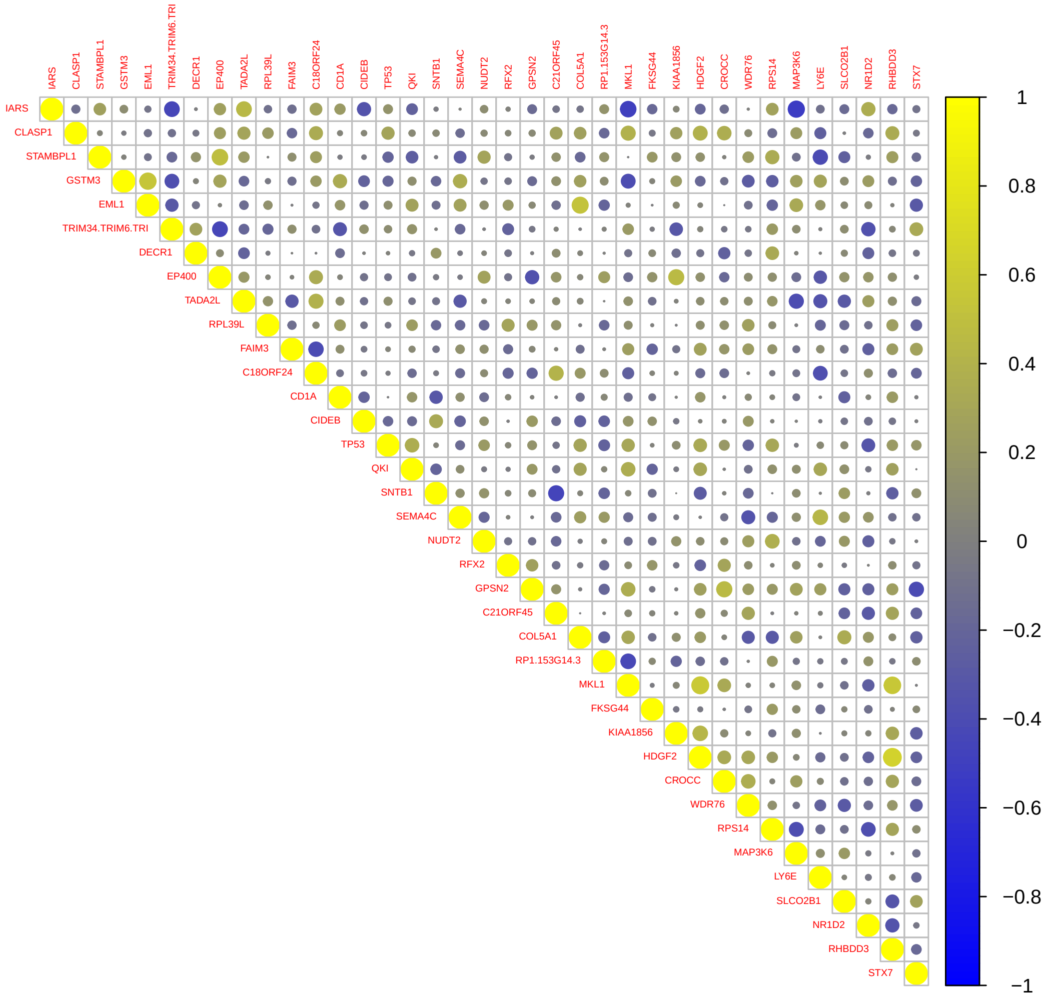


Figure 4.5.1: Correlations between 37 principal genes after performing variable selection by PVA. Circles show the magnitude of correlation between two variables, the darker the colour, the higher the correlation. Blue indicates negative correlation and yellow indicates negative correlation.

As a result of applying PVA on these data, 37 out of 13321 gene expressions were selected as principal predictors denoted by $\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_{37}$. Now we consider the regression model (4.3.1), but this time based on the data $(\tilde{\mathbf{X}}, \mathbf{Y})$ where the columns of the design matrix $\tilde{\mathbf{X}}_{42\times 37}$ contains the selected predictors by PVA and $\mathbf{Y}$ contains the IC50 values

$$\mathbf{Y}_{42\times 131} = \tilde{\mathbf{X}}_{42\times 37}\mathbf{B}_{37\times 131} + \mathbf{E}_{42\times 131}. \tag{4.5.1}$$

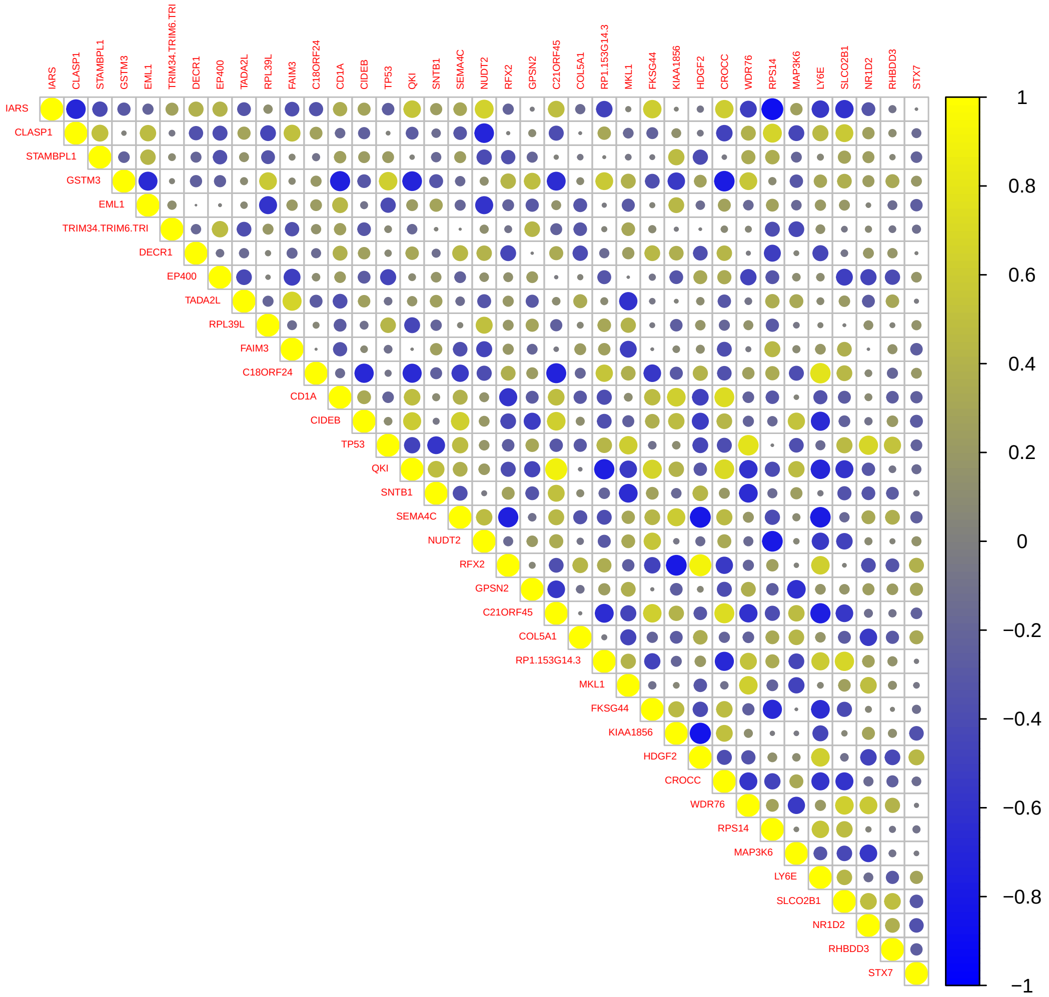


Figure 4.5.2: Correlations between least square coefficient estimates of principal genes. Circles show the magnitude of correlation between two predictors, the darker the colour, the higher the correlation. Blue indicates negative correlation and yellow indicates negative correlation.

The estimated least square coefficients form the matrix $\hat{\mathbf{B}}_{37\times 131}$. Figure 4.5.1 illustrates the correlation between 37 principal predictors $\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_{37}$. The correlation between least square estimates $\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_{37}$ is also shown in Figure 4.5.2. These correlation patterns uncover some appealing features about selected genes. These graphs demonstrate two entirely different correlation structure: a weak correlation between principal genes and a strong correlation structure between the estimated regression coefficients of these genes. The graphs show that although the principal genes are weakly correlated, once we take the IC50 values of drugs into account through the regression model (4.5.1), these selected genes become strongly correlated. In other words, the uncorrelated genes are highly correlated in their coefficients when they are linked to the IC50 values through the regression model. From biological point of view, this phenomena conveys valuable information about these genes. The fact that these genes are highly correlated based on their responses to cancer drugs, confirms that linking the gene expressions and IC50 values through a multivariate regression model is beneficial and can uncover some hidden information which cannot be recovered in the analysis of gene expressions. In the next section, we build a network of these genes for further investigation and extracting more information about these genes.

4.5.2 Predictive network of principal genes

In this section we intend to identify a network between the 37 principal genes selected by PVA. This network is based on the regression coefficients $\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_{37}$ of principal selected genes. We are interested in discovering more about the interactions between these genes, through establishing a network between their estimated coefficients. To this end, we consider the coefficient matrix $\hat{\mathbf{B}}_{37\times131}$, the rows of which $\hat{\mathbf{b}}_1, \cdots, \hat{\mathbf{b}}_{37}$ are 131-dimensional coefficient vectors correspond to the principal genes. We construct two different networks based on the coefficient matrix $\hat{\mathbf{B}}_{37\times131}$. One network is constructed with 37 nodes which are the rows of the coefficient matrix and another network is established based on the columns of this matrix with 131 nodes.

As explained in Section 4.4, a primary tool for constructing a network is an adjacency matrix. Therefore, we first define an adjacency matrix based on the correlations between the nodes. For the network with 37 nodes we find the pairwise Pearson correlation coefficient between the rows of regression coefficient matrix $\hat{\mathbf{B}}_{37\times131}$. Let $\mathbf{R}_{\tilde{p}\times\tilde{p}} = (r_{ij})$ denote the correlation matrix between $\tilde{p} = 37$ vectors. The idea is thresholding these correlations r_{ij} at some level of significance so that r_{ij}s with values higher than the threshold level are set to one in adjacency matrix and zero otherwise. So we need to perform a hypothesis test with $H_0 : r_{ij} = 0$ versus $H_1 : r_{ij} \neq 0$ to test the significance of r_{ij}s. To carry out this test we need to convert r_{ij}s to a normally distributed variable. Hence, we invoke Fisher's Z-transformation and calculate

$$z_{ij} = \frac{1}{2} \ln \left(\frac{1 - r_{ij}}{1 + r_{ij}} \right). \tag{4.5.2}$$

As a result, z_{ij}s are normally distributed and if they are independent and therefore uncorrelated, then z_{ij}s are i.i.d normally distributed variables $z_{ij} \approx N(0, 1/(J-3))$ where $J = 131$. Since $R_z = (z_{ij})$ is symmetric we just consider the upper diagonal elements of this matrix where $i < j$ and we test whether z_{ij} is significantly away from zero. If we consider the off-diagonal elements in $R_z = (z_{ij})$, there are $\tilde{p}(\tilde{p} - 1)/2$ tests to be carried out simultaneously.

It is known that in multiple testing where we perform a large number of hypothesis tests, it is very likely to have false discoveries just due to chance. In order to avoid making wrong decision multiple testing theory provides some approaches to control the error rates. According to McDonald (2009), multiple comparisons is an area of active research and there is no universally accepted approach for dealing with this issue.

Here, we invoke the classical yet widely used technique of Bonferroni correction which sets the significance cut-off at α/t to adjust the error rates, as pointed out in Norman and Streiner (2008). α is the desired significance level at which we want to test the set of hypotheses and t is the number of tests to be performed. Followed by this, in testing the hypotheses $H_0 : z_{ij} = 0$ versus $H_1 : z_{ij} \neq 0$ at $\alpha = 1\%$ significance level, by applying Bonferroni correction, this value is replaced with $\alpha^{'} = \alpha/t$; $t = \tilde{p}(\tilde{p} - 1)/2$. Since z_{ij}s are normally distributed, $z = \sqrt{J-3} z_{ij}\ N(0, 1)$. Therefore we claim that z_{ij} is significantly away from zero if $z > z_{\alpha'/2}$ and we can construct the adjacency matrix

$$\hat{A}_{ij} = \begin{cases} 1 & \text{if } |z| > z_{\alpha'/2} \\ 0 & \text{otherwise.} \end{cases} \tag{4.5.3}$$

The predictive network of principal genes with 37 nodes presented in Figure 4.5.3 is formed by assigning an edge between nodes i and j if $A_{ij} = 1$.

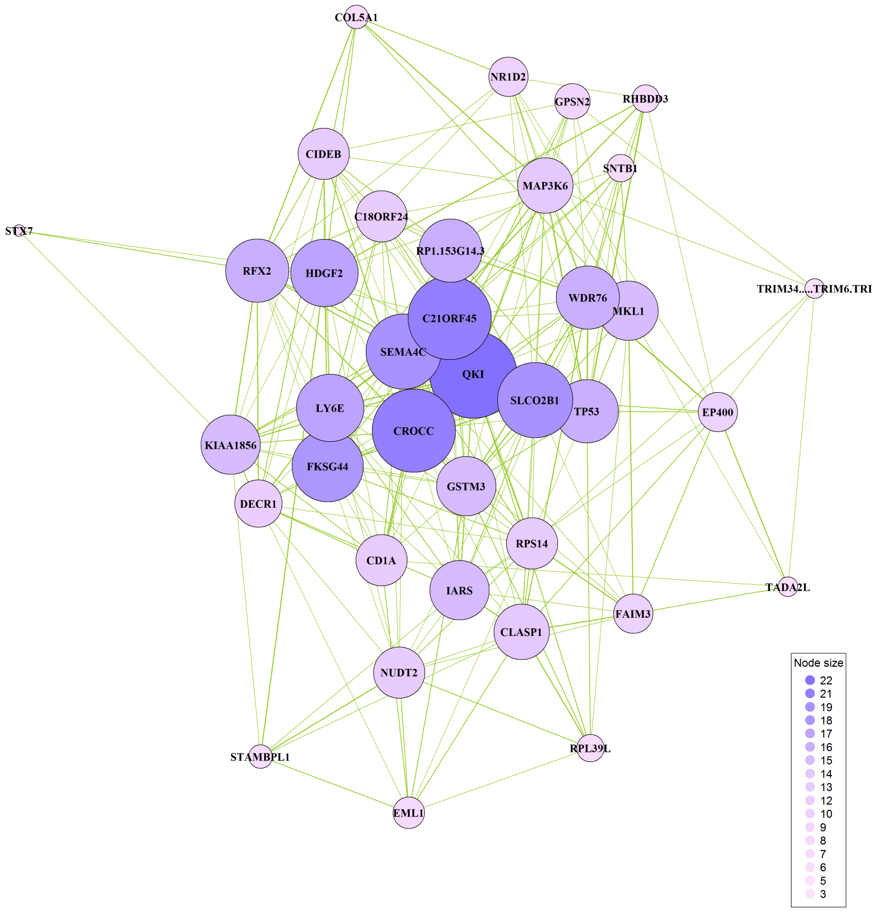


Figure 4.5.3: Network of estimated regression coefficients corresponding to 37 principal genes selected by PVA. These coefficient vectors are of dimension 131. Each node represent one of the selected genes. Size of each node depicts the degree of that node. Vertices with larger size are correlated with larger number of genes. The thickness of edges represent the magnitude of this correlations. The higher the correlation, the thicker the edge between two genes. The largest and smallest node size belongs to gene QKI with size 22 and gene STX7 with size 3, respectively.

This network is based on the regression coefficients of these genes so $A_{ij} = 1$ implies that these genes have a significant correlation based on their regression coefficients. The network is presented in Figure 4.5.3 is strongly connected which shows these genes are highly correlated based on their regression coefficients. The thickness of the lines show the magnitude of existing correlations between genes. Another network with 131 nodes is constructed based on the pairwise Pearson correlation coefficients between the columns of the coefficient matrix $\hat{\mathbf{B}}_{37\times131}$. Therefore for this case in the above procedure of finding adjacency matrix we have $J = 37$ and $\tilde{p} = 131$. The resulted network between 131 coefficient vectors is shown in Figure A.0.2 in Appendix A.

To reveal the roles played by these principal genes in different types of cancer, we investigated their protein staining in 20 common cancers as the protein products would dictate their functions (Stewart et al., 2017). The tables in Appendix A provide some information gathered from the Human Protein Atlas Portal at http://www.proteinatlas.org/cancer. In these tables, according to the information reported in the Portal, we classified the protein expression/staining levels into four categories: high, medium, low and not detected. We assigned the scores of $3, 2, 1$ and 0 to the four categories respectively. If a gene did not play a role in a cancer, it receives a score of zero as its protein staining at that cancer would be hardly detectable. We found that 34 of the selected genes had positive staining levels for at least one of these cancers. This implies that these genes might play certain functional roles in the growth of some of these cancers. In the Portal, there were no information available on the remaining 3 of the selected genes.

4.6 Simulation studies

In this section, we assess the ability of PVA procedure in identifying the informative and ruling out the uninformative predictors on simulated data. This assessment was carried out by calculating the sensitivity and specificity of the PVA which reflects the ability of correctly identifying the non-zero coefficients (sensitivity) and discarding the zero coefficients (specificity).

Similar to the previous chapter, we considered PVA based on four different covariance estimators introduced in Section 3.4.3. The Ledoit-Wolf's shrunk estimator $\hat{\mathbf{C}}_{opt}$ and the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$. Although results in Chapter 3 showed that SNR is not sensitive to these constants, here, we still use different values to explore how these constants affect the selection accuracy. Results are reported for two different values for the constant in the stopping criteria, $a_0 = 3, 5$.

Moreover, we compared the PVA performance with four well-known penalisation methods for multivariate regression by comparing the sensitivity values. These methods introduced in Section 2.2.2 are the multivariate group lasso (mgl), multivariate elastic net (menet). We used *glmnet* R-package (Friedman et al., 2010) to perform variable selection with these two methods. We also compared PVA with the multivariate lasso (ml) and multivariate sparse group lasso (msgl). Since the multivariate lasso in glmnet imposes a group lasso penalty and not a lasso penalty, we used the *lsgl* R-package (Vincent and Hansen, 2014) to apply the multivariate lasso and multivariate sparse group lasso penalties. The performances of these methods were examined by comparing the sensitivity values while specificity values were approximately the same. More details about how we fix this specificity is explained in Section 2.4. All simulations were programmed and conducted in R software.

4.6.1 Simulation setup

To investigate whether high correlations among response variables hinder or enhance the detection of true active predictors for the above variable selection methods, we designed two different settings.

Scenario 1 (Strongly correlated coefficient matrix): As it was explained in Section 2.4.1, the design matrix $\mathbf{X}_{n\times p}$ was generated by sampling n number of iid p-vectors $\mathbf{x}_i,\ i = 1, \cdots, n$ from a multivariate normal $N_p(\mathbf{0}, \mathbf{\Sigma}_{p\times p})$ where $\mathbf{\Sigma}$ is the covariance matrix of the gene expressions in our real data. The coefficient matrix $\mathbf{B}^h_{p\times J}$ was generated by sampling p number of J-vectors from $N_J(\mathbf{0}, \Omega_h)$ wherein the non-zero elements were placed in strongly correlated columns. The error matrix, $\mathbf{E}_{n\times J}$, was generated by sampling J times from a multivariate normal distribution $N_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$, where $\sigma^2 = 0.1$. Finally, the multivariate response variable was generated from multivariate regression model

$$\mathbf{Y}_{n\times J} = \mathbf{X}_{n\times p}\mathbf{B}_{p\times J} + \mathbf{E}_{n\times J}. \tag{4.6.1}$$

Scenario 2 (Weakly correlated coefficient matrix): For this setting, data were generated as explained in scenario 1 except that for this setting the coefficient matrix $\mathbf{B}^l_{p\times J}$ was generated by sampling p number of J-vectors from $N_J(\mathbf{0}, \Omega_l)$, wherein the non-zero elements are placed in weakly correlated columns.

In all simulations we generated 50 datasets of $(\mathbf{Y}_{n\times J}, \mathbf{X}_{n\times p})$ for each combination of $(n, p, J, |T|)$ where $n = 42, 88, 150$ is the sample size, $J = 20, 34, 131$ is the dimension of the response variable and $p = 2000$ is the dimension of the covariates. The non-zero elements or the active size was also set to $|T| = 10, 37, 70$. These simulations are based on stopping point $a_0 = 5$. We also conducted some simulations with stopping criterion $a_0 = 3$. Although the obtained results showed a negligible improvement in selection accuracy in terms of specificity, the computational cost was expensive and the selection process became significantly slow. Therefore, we chose $a_0 = 5$ in the following simulations and just report one setting in Figure 4.6.4 and Figure 4.6.5 where we set $a_0 = 3$.

4.6.2 Results

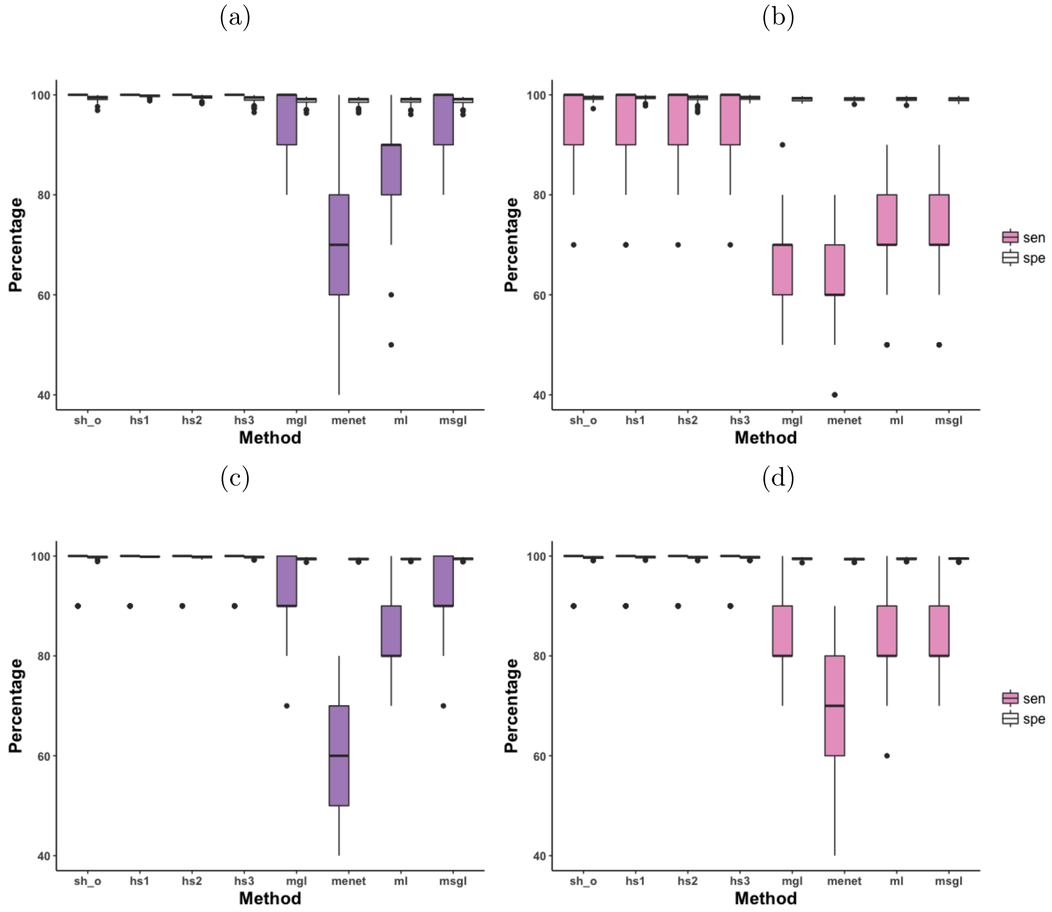


Figure 4.6.1: Box plots of the sensitivity percentages when specificity values are approximately the same. Here, sh$_o$ corresponds to PVA based on Ledoit-Wolf's shrunk estimator $\hat{\mathbf{C}}_{opt}$ and hs1, hs2 and hs3 refers to PVA based on the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$ respectively with the stopping criteria's constant $a_0 = 5$. Results obtained under (a) scenario 1 with strongly correlated coefficient matrix and (b) scenario 2 with weakly correlated coefficient matrix. Settings with $|T| = 10$, $n = 88$, $p = 2000$ with (a),(b) $J = 20$ and (c), (d) $J = 34$.

Results shown in Figure 4.6.1 illustrate that the sensitivity of PVA is much higher relative to other penalisation methods. This conveys that the accuracy of PVA in correctly detecting non-zero predictors is comparative. Similar to SNR-based screening, the selection accuracy is not affected by varying the tuning constant h. We also conducted more simulations with a larger number of active (non-zero) predictors to explore to what extent the selection process is influenced by the number of active predictors. Hence, we increased the number of active predictors to $|T| = 70$ and we report the result for different combinations of (n, J). The sensitivity and specificity percentages are depicted in Figure 4.6.2.

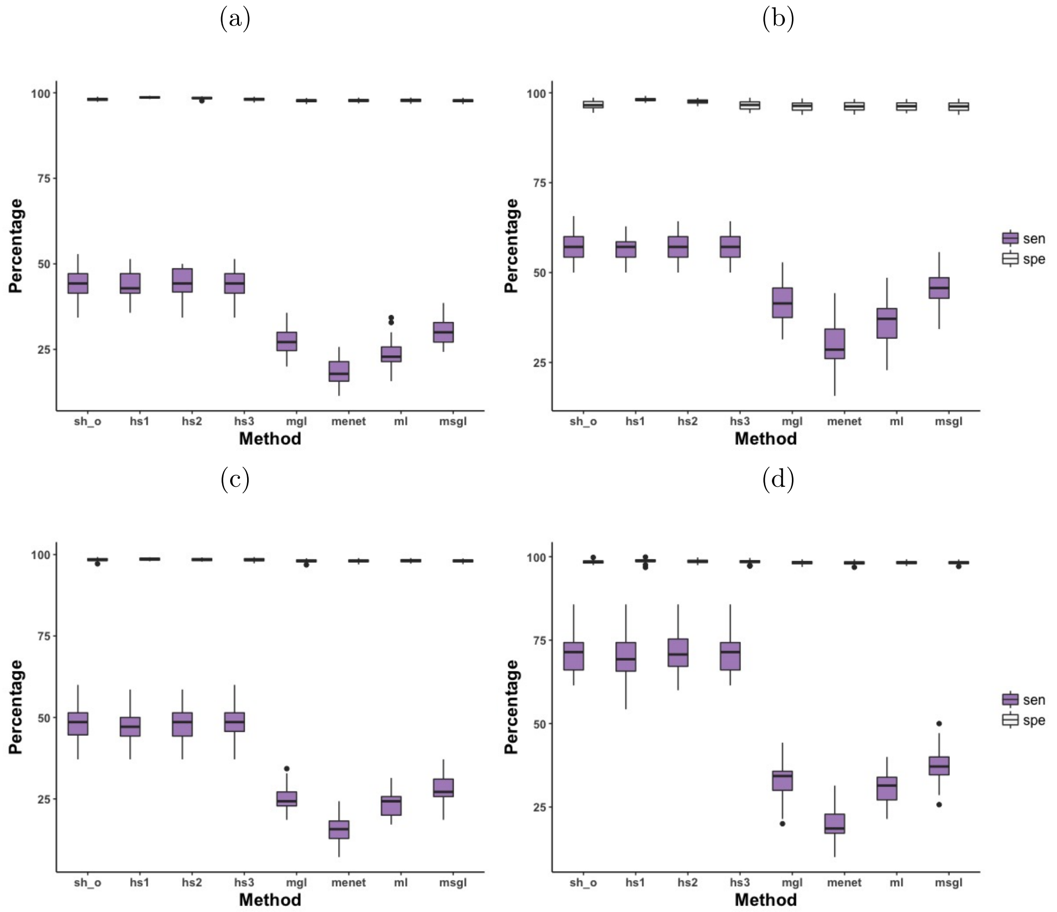
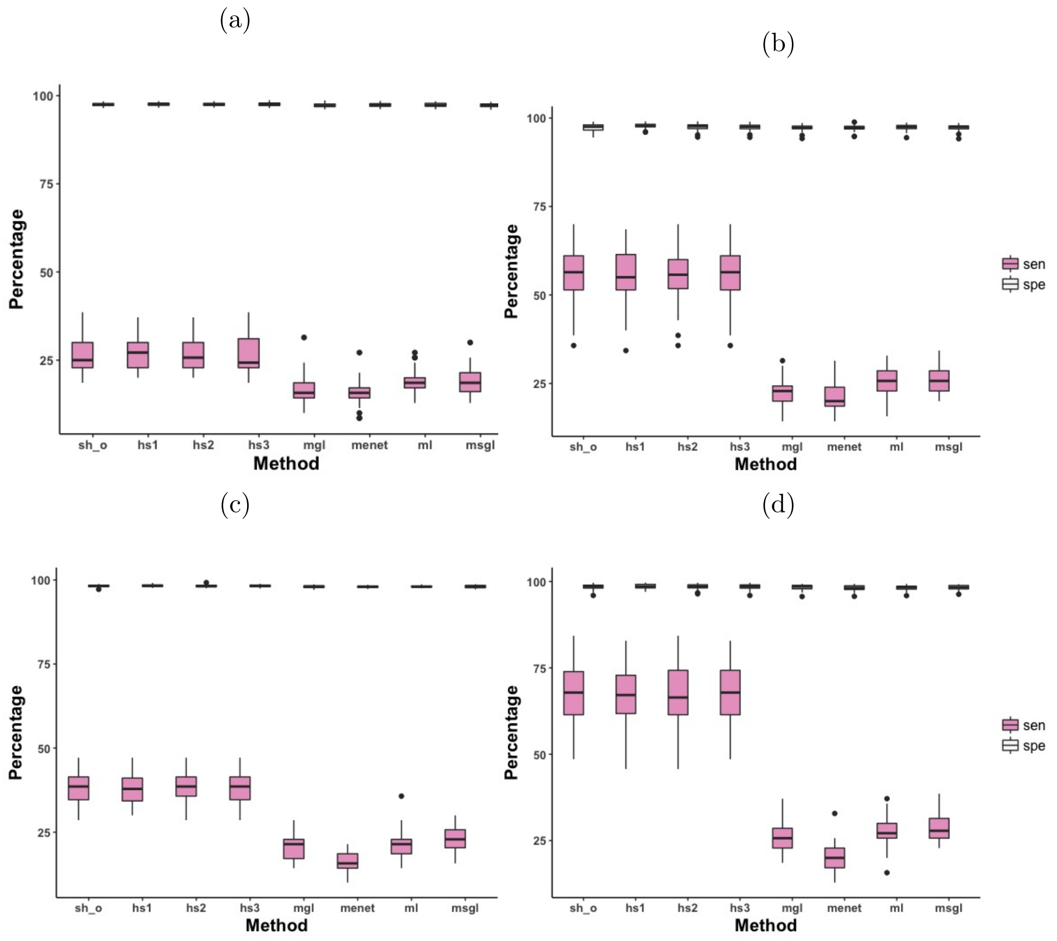


Figure 4.6.2: **Scenario 1 (Strongly correlated coefficient matrix):** Box plots of the sensitivity percentages when specificity values are approximately the same. Here, sh_o corresponds to PVA based on Ledoit-Wolf's shrunk estimator $\hat{\mathbf{C}}_{opt}$ and hs1, hs2 and hs3 refers to PVA based on the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$ respectively with the stopping criteria's constant $a_0 = 5$. In all settings $|T| = 70$ and $p = 2000$ where in (a) $J = 20$, $n = 88$; (b) $J = 20$, $n = 150$; (c) $J = 34$, $n = 88$; (d) $J = 34$, $n = 150$.

Comparing results presented above in Figure 4.6.2 (a), where the number of non-zero predictors is $|T| = 70$ with the same combinations of $(p, n, J) = (2000, 88, 20)$ in Figure 4.6.1 (a) where $|T| = 10$ verifies that increasing the number of active predictors reduces the accuracy of all methods significantly.

This is not surprising as this decline is the consequence of the mask effect caused by correlated predictors and as a result of these correlations, detecting the true non-zero predictors becomes more challenging. It can be seen that even when we increase the number of active predictors PVA still outperforms the competitors. Results obtained from the same combinations as Figure 4.6.2 but with weakly correlated coefficient matrix are presented in Figure 4.6.3.


(a)

(b)

(c)

(d)

Figure 4.6.3: **Scenario 2 (Weakly correlated coefficient matrix):** Box plots of the sensitivity percentages when specificity values are approximately the same. Here, sh$_o$ corresponds to PVA based on Ledoit-Wolf's shrunk estimator $\hat{\mathbf{C}}_{opt}$ and hs1, hs2 and hs3 refers to PVA based on the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$ respectively with the stopping criteria's constant $a_0 = 5$. In all settings $|T| = 70$ and $p = 2000$ where in (a) $J = 20$, $n = 88$; (b) $J = 20$, $n = 150$; (c) $J = 34$, $n = 88$; (d) $J = 34$, $n = 150$.

These results show much lower sensitivity level for PVA for all settings. This was expected since in PVA the correlation between response variables is taken into account. However, comparing the results in Figure 4.6.2 and Figure 4.6.3 reveals that other methods are not affected when there is a weaker correlation structure between response variables. This was also expected since these methods do not take the correlation between the response variables into account in the selection process. Moreover, comparing the results obtained in Figure 4.6.2 (a) with $n = 88$ and (b) with $n = 150$ reveals that increasing the sample size improves the performance of all methods. Also comparing the first and second row shows that increasing the number of columns J has a positive effect on the performance of these methods. PVA shows an outstanding accuracy compared to all other methods when the sample sizes J and n are large enough.

Since the number of selected predictors by PVA in real data is 37 and the sample size is 42 and we have 131 observations for the response variable, we conducted more simulations with the combinations of $(p, n, J, |T|) = (2000, 42, 131, 37)$.

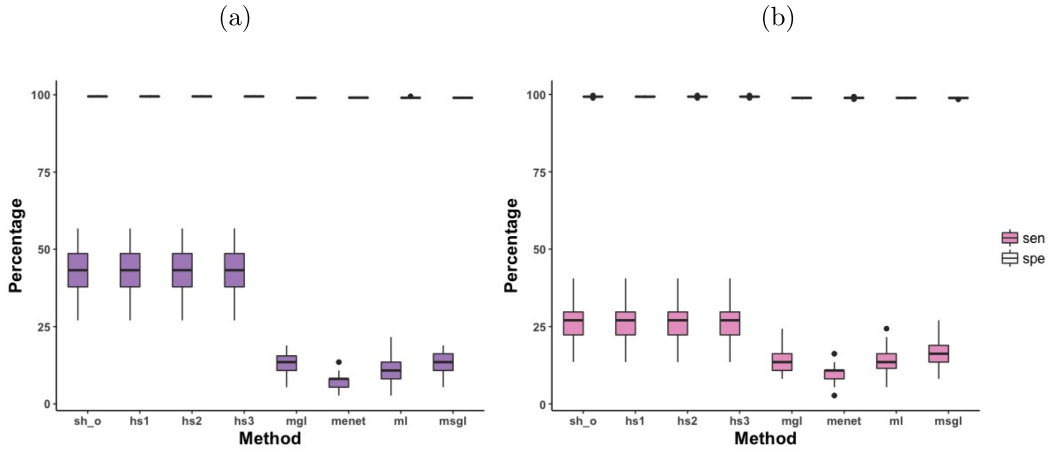


Figure 4.6.4: Box plots of the sensitivity percentages when specificity values are approximately the same. Here, sh_o corresponds to PVA based on Ledoit-Wolf's shrunk estimator $\hat{\mathbf{C}}_{opt}$ and hs1, hs2 and hs3 refers to PVA based on the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$ respectively and with the stopping criteria's constant $a_0 = 5$. Settings with $(p, n, J, |T|) = (2000, 42, 131, 37)$ (a) corresponds to scenario 1 with highly correlated $\mathbf{B}$ and (b) corresponds to scenario 2 with low correlations.

Here we also considered two different constants of $a_0 = 3, 5$ in the stopping criteria. Results presented in Figure 4.6.4 are obtained with $a_0 = 5$ and in Figure 4.6.5 with $a_0 = 3$.

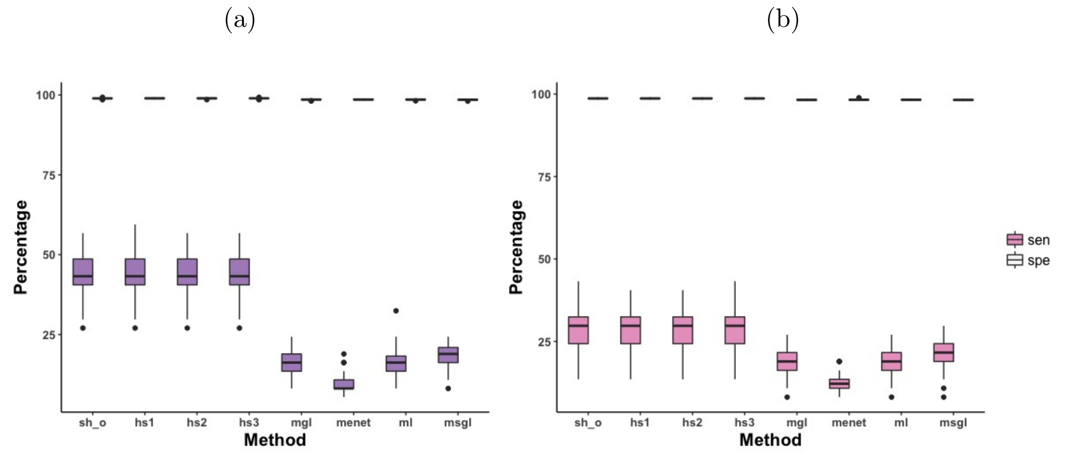


Figure 4.6.5: Box plots of the sensitivity percentages when specificity values are approximately the same. Here, sh_o corresponds to PVA based on Ledoit-Wolf's shrunk estimator $\hat{\mathbf{C}}_{opt}$ and hs1, hs2 and hs3 refers to PVA based on the thresholded and shrunk estimator $\hat{\mathbf{C}}_{hs}$ with tuning constants $h = 0.01, 0.005, 0.001$ respectively and with the stopping criteria's constant $a_0 = 3$. Settings with $(p, n, J, |T|) = (2000, 42, 131, 37)$ (a) corresponds to scenario 1 with highly correlated $\mathbf{B}$ and (b) corresponds to scenario 2 with low correlations.

To sum up, results presented in Figure 4.6.4 and Figure 4.6.5 together with previous previous outcomes also verifies that correlation structures in the coefficient matrix which impose the same correlation structure in the response variable can influence the selection accuracy of PVA. For example, in settings where the coefficient matrix is highly correlated PVA is more efficient and leads to a more accurate selection with higher sensitivity. The reason is that PVA takes the correlation structure of the response variable into account whereas other methods lack this interesting property. We can also see that in all scenarios with different simulation settings with $n > J$ and $n < J$, our proposed PVA surpasses all the penalisation methods.

Tables 2 and 3 show the mean and standard errors of all methods across 50 simulations. The improvement obtained by PVA against penalisation methods is also reported in these tables.

The upper part of these tables illustrate mean values of sensitivity percentages across 50 simulations together with standard errors resulted from dividing the standard deviation of sensitivity percentages across all simulations by 50. The lower part provides the improvement percentages by PVA obtained from the following ratio

$$\text{Improvement by PVA} = \frac{\text{mean}(\text{SEN}_{\text{PVA}}) - \text{mean}(\text{SEN}_{\text{PM}})}{\text{mean}(\text{SEN}_{\text{PM)}}},$$

where PM stands for the penalization method. In order to find standard errors we calculated the following ratio

$$d = \frac{\text{SEN}_{\text{PVA}} - \text{SEN}_{\text{PM}}}{\text{SEN}_{\text{PM}}},$$

then we calculated the standard error of the above ratio by calculating

$$\text{Standard error}_{\text{Improvement}} = \frac{\sigma_d}{\sqrt{50}}.$$

The improvement obtained by PVA in both scenarios with strong correlations and weak correlations against all penalisation methods under consideration is noticeable. For example, under scenario 1 with strongly correlated coefficient matrix, PVA improves the selection accuracy by 47%, 26%, 70%, 91% and 223% compared to the multivariate sparse group lasso (msgl) which is a well-known penalisation method. This improvement by PVA is also remarkable under scenario 2 with 41%, 116%, 66%, 135% and 62% increase against multivariate sparse group lasso method.

	$\lvert T\rvert = 70$				$\lvert T\rvert = 37$
	$J = 20$		$J = 34$		$J = 131$
	$n = 88$	$n = 150$	$n = 88$	$n = 150$	$n = 42$
Mean of sensitivity and standard error in %					
PVA	44.25	57.45	47.71	70.97	42.81
Standard error	(0.55)	(0.48)	(0.71)	(0.80)	(0.89)
mgl	27.40	41.77	25.31	33.20	12.81
Standard error	(0.58)	(0.68)	(0.52)	(0.70)	(0.44)
menet	18.2	29.54	15.51	19.48	7.62
Standard error	(0.46)	(0.84)	(0.50)	(0.66)	(0.35)
ml	23.37	36.31	23.51	30.45	11.72
Standard error	(0.55)	(0.79)	(0.54)	(0.61)	(0.53)
msgl	30.05	45.37	27.94	37.05	13.24
Standard error	(0.52)	(0.71)	(0.58)	(0.64)	(0.43)
Sensitivity improvement by PVA(hs3) and standard error in %					
Against mgl	61	37	88	113	234
Standard error	(4)	(2)	(5)	(6)	(20)
Against menet	143	94	207	264	461
Standard error	(7)	(6)	(14)	(18)	(51)
Against ml	89	58	102	133	264
Standard error	(4)	(4)	(5)	(6)	(35)
Against msgl	47	26	70	91	223
Standard error	(3)	(2)	(4)	(4)	(19)

Table 2: **Scenario 1 (Strongly correlated coefficient matrix):** Mean sensitivity and mean improvement obtained from 50 simulations when specificity is fixed approximately at the same level for all methods. PVA(hs3) is the PVA when the covariance matrix is estimated by the shrunk and thresholded estimator $\hat{\mathbf{C}}_{hs}$ with $h = 0.001$.

	$\lvert T\rvert = 70$				$\lvert T\rvert = 37$
	$J = 20$		$J = 34$		$J = 131$
	$n = 88$	$n = 150$	$n = 88$	$n = 150$	$n = 42$
Mean of sensitivity and standard error in %					
PVA	26.71	55.91	38.34	67.51	26.91
Standard error	(0.73)	(1.05)	(0.61)	(1.17)	(0.89)
mgl	16.80	22.65	20.51	25.82	14.74
Standard error	(0.51)	(0.56)	(0.47)	(0.59)	(0.56)
menet	15.41	20.97	16.08	20.17	9.83
Standard error	(0.44)	(0.54)	(0.40)	(0.58)	(0.43)
ml	18.65	25.42	21.31	27.45	14.70
Standard error	(0.46)	(0.52)	(0.58)	(0.60)	(0.59)
msgl	18.85	25.82	22.97	28.71	16.59
Standard error	(0.50)	(0.51)	(0.49)	(0.56)	(0.63)
Sensitivity improvement by PVA(hs3) and standard error in %					
Against mgl	59	146	86	161	82
Standard error	(6)	(8)	(5)	(8)	(11)
Against menet	73	166	138	234	173
Standard error	(8)	(9)	(8)	(15)	(20)
Against ml	43	119	79	145	83
Standard error	(5)	(6)	(6)	(9)	(13)
Against msgl	41	116	66	135	62
Standard error	(5)	(6)	(5)	(6)	(10)

Table 3: **Scenario 2 (Weakly correlated coefficient matrix):** Mean sensitivity and mean improvement obtained from 50 simulations when specificity is fixed approximately at the same level for all methods. PVA(hs3) is the PVA when the covariance matrix is estimated by the shrunk and thresholded estimator $\hat{\mathbf{C}}_{hs}$ with $h = 0.001$.

4.7 Discussion

Although according to the simulation studies, PVA possesses a higher selection accuracy compared to other methods, the number of variables selected by PVA is limited to the sample size. The selection process in PVA is based on forward nulling which is accomplished through null-beamformers. Setting specific constraints in these null-beamformers which assume that projections are orthogonal to each other imposes a limitation on PVA selection. Due to this orthogonality assumption, PVA selection becomes restricted to the sample size. Another factor that has an impact on the number of selected variables by PVA is the stopping rule which is used in Algorithm 4.1 in Section 4.2.2. For example, using 3-sigma rule as the stopping criterion is computationally more expensive than using 5-sigma rule but results in selecting a larger number of variables. PVA performance is also sensitive to the choice of tuning constant which is used in $\hat{\mathbf{C}}_{hs}$ covariance estimator yet this sensitivity is not substantial in our simulated data. PVA does not show any sensitivity to the choice of tuning parameter in real data application, and it selects the same set of predictors based on constants $h = \{0.01, 0.005, 0.001\}$.

Chapter 5

Likelihood Fusion for Multivariate Regression Models

5.1 Introduction

The study in this chapter is also motivated by the cancer drug data introduced earlier. We propose a two-stage mixture-based model and a procedure based on the proposed model to perform marginal variable screening and regression classification, simultaneously. The rationale behind the new proposal is as follows. Response variables in our real data are the IC50 values of different drugs. Naturally, it makes sense to assume that these drugs have a group structure. For example, these drugs can be classified into different groups based on the types of cancers which are treated by these drugs. Or they can be classified based on their effectiveness on different gene expressions.

The model that we propose resembles the mixture of regression models introduced in Chapter 2. We remind that the IC50 values form the matrix $\mathbf{Y}_{n\times J}$, which contain n observations on J response variables. We wish to cluster these response variables into groups. To this aim, we calculate the likelihood function for each response variable. Then we construct a mixture-based model wherein these likelihoods are regarded as density functions. Note that the idea is pulling the information from different columns, or response variables, together. This proposed mixture-based model is called *likelihood fusion*. This model is specified in detail in the next section.

5.2 Likelihood fusion models for multivariate regressions

In this section, we introduce likelihood fusion model and we implement this model in a two-stage screening and clustering procedure. This two-stage procedure was initially introduced by Zhang (2017) for screening and clustering of sparse regressions with finite non-Gaussian mixtures. The model we propose here is designed for multivariate regression models where we wish to regress several response variable against high dimensional predictors $p \gg n$. In the first stage of this procedure, we use the likelihood fusion to perform a marginal variable screening for multivariate regression models. In the second stage, the proposed model is fitted to the reduced predictors to classify regressions. Accordingly, through this procedure for high dimensional regressions, variable screening and classification is carried out simultaneously. We start by giving a definition of likelihood fusion model followed by model estimation. Then we explain the second stage of the procedure which is the classification stage.

Suppose we are interested in clustering J independent multivariate response variables $\mathbf{y}_j; j = 1, \cdots, J$ into K groups. For each of these response variables, n observations are recorded i.e. $\mathbf{y}_j = (y_{j1}, \cdots, y_{jn}), j = 1, \cdots, J$. Let matrix $\mathbf{X}_{n \times p}$ be a design matrix formed by n observations on p covariates. The dependence of $\mathbf{y}_j$ on $\mathbf{X}$ is expressed through the conditional distribution of $\mathbf{y}_j | \mathbf{X}$ which is modelled by the following mixture of regressions

$$f(\mathbf{y}_j|\mathbf{X}, \Phi) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{y}_j|\mathbf{X}\boldsymbol{\beta}_k, \sigma_k^2), \tag{5.2.1}$$

where $\Phi = (\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_K, \sigma_1, \cdots, \sigma_K, \pi_1, \cdots, \pi_K)$ and these parameters vary across the components. $f_k(\mathbf{y}_j|\mathbf{X}\boldsymbol{\beta}_k, \sigma_k^2)$ is the conditional density of $\mathbf{y}_j$ given

$(\mathbf{X}\boldsymbol{\beta}_k, \sigma_k^2)$ in the kth component. This implies that each $\mathbf{y}_j$ can come from any of these K components with probability of π_k. The mixture of regressions that is expressed through Equation (5.2.1) is quite different from the classical mixture of regressions model. Here, each density function $f_k(\mathbf{y}_j|\mathbf{X}\boldsymbol{\beta}_k, \sigma_k^2)$ is a likelihood or a joint distribution of n iid observations given by

$$\begin{aligned} f_k(\mathbf{y}_j|\mathbf{X}\boldsymbol{\beta}_k, \sigma_k^2) &= \prod_{i=1}^{n} f_k^i(y_{ij}|\mathbf{x}_i\boldsymbol{\beta}_k, \sigma_k^2) \\ &= \prod_{i=1}^{n} \frac{1}{(2\pi\sigma_k^2)^{1/2}} \exp\left\{-\frac{(y_{ij}-\mathbf{x}_i\boldsymbol{\beta}_k)^2}{2\sigma_k^2}\right\} \\ &= \frac{1}{(2\pi\sigma_k^2)^{n/2}} \exp\left\{-\frac{(\mathbf{y}_j-\mathbf{X}\boldsymbol{\beta}_k)^T(\mathbf{y}_j-\mathbf{X}\boldsymbol{\beta}_k)}{2\sigma_k^2}\right\} \end{aligned}$$

Accordingly, the above model is a mixture of multivariate normal density functions. In other words, this model is a weighted sum or a convex combination of K density functions. We refer to this method as *likelihood fusion.* In this model, regression coefficients and error terms are heterogeneous across the components.

Note that in the real data we have a small sample size. Due to the lack of information caused by the small sample size, we are not able to deal with correlations among response variables. Therefore, here we assume the ideal case that the response variables are independent and we construct the likelihood. Accordingly, what we calculate in the next section is actually a pseudo or artificial likelihood and can be regarded as an approximation of the dependent case.

5.3 Estimation of likelihood fusion models

In order to fit the model (5.2.1) we need to estimate all model parameters. We obtain these estimates by applying maximum likelihood method. The likelihood function corresponding to the model (5.2.1) and observations $\mathbf{y}_1, \cdots, \mathbf{y}_J$

is given by

$$\begin{aligned} L(\Phi) &= \prod_{j=1}^{J} f(\mathbf{y}_j|\mathbf{X},\Phi) \\ &= \prod_{j=1}^{J}\sum_{k=1}^{K}\pi_k f_k(\mathbf{y}_j|\mathbf{X}\boldsymbol{\beta}_k,\sigma_k^2). \end{aligned} \tag{5.3.1}$$

We estimate model parameters by maximizing the logarithm of this likelihood function

$$l(\Phi) \;=\; \sum_{j=1}^{J}\log\left[\sum_{k=1}^{K}\pi_k f_k(\mathbf{y}_j|\mathbf{X}\boldsymbol{\beta}_k,\sigma_k^2)\right]. \tag{5.3.2}$$

Due to the lack of information about group indices, the summation appeared in the log-likelihood function is not decomposable. In the next section, we invoke the Expectation-Maximization(EM) algorithm introduced in Chapter 2 to deal with this problem.

5.3.1 EM algorithm for computing the maximum fused-likelihood estimator

The likelihood function expressed in Equation (5.3.2) is called an incomplete-data log-likelihood. As it was explained in Section 2.5.1, to attain the maximum likelihood estimates, EM algorithm is an adequate tool to be utilised in the existence of latent variables in the data. To this aim, the incomplete-data $(\mathbf{y}_j,\mathbf{X})$ is augmented by defining a component label vector, $\mathbf{z}$. Having completed the missing part of the data with this component indicator variable, we can now construct a likelihood for complete data $(\mathbf{y}_j,\mathbf{X},\mathbf{z})$. The complete-data likelihood corresponding to the model (5.2.1) and complete-data $(\mathbf{y}_j,\mathbf{X},\mathbf{z})$ is given by

$$\begin{aligned} L(\Phi) &= \prod_{j=1}^{J}\prod_{k=1}^{K}\pi_k^{z_{jk}} f_k(\mathbf{y}_j|\mathbf{X}\boldsymbol{\beta}_k, \sigma_k^2)^{z_{jk}} \\ &= \prod_{j=1}^{J}\prod_{k=1}^{K}\pi_k^{z_{jk}}(2\pi\sigma_k^2)^{\frac{-nz_{jk}}{2}} \exp\left\{-\frac{z_{jk}(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k)^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k)}{2\sigma_k^2}\right\}, \end{aligned} \tag{5.3.3}$$

with the corresponding log-likelihood

$$l(\Phi) = \sum_{j=1}^{J}\sum_{k=1}^{K}\left[z_{jk}\log\pi_k - \frac{nz_{jk}}{2}\log(2\pi\sigma_k^2) - \frac{z_{jk}(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k)^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k)}{2\sigma_k^2}\right]. \tag{5.3.4}$$

Now we can invoke the EM algorithm on the complete-data log-likelihood in Equation (5.3.4).

E-Step

In the expectation phase of the algorithm we calculate the expectation of the complete-data log-likelihood, i.e. $E\log(L(\Phi))$. Note that there is still no information about the latent variables z_{jk}, but we can calculate the expectation of these variables instead. This is attainable through posterior probabilities of these variables, given the observed data and the model parameters. These variables are discrete so this probability can be found using the Bayes' theorem as follows. In the mth iteration of the EM algorithm we have

$$\begin{aligned} E(z_{jk}|\mathbf{y}_j, \Phi) &= w_{jk}^{(m)}(\mathbf{y}_j, \Phi^{(m)}) \\ &= \frac{\pi_k^{(m)} f_k(\mathbf{y}_j; \theta_k^{(m)})}{\sum_{t=1}^{K}\pi_t^{(m)} f_t(\mathbf{y}_j; \theta_t^{(m)})}, \end{aligned} \tag{5.3.5}$$

where $\Phi = (\boldsymbol{\pi}, \boldsymbol{\theta}) = (\pi_1, \cdots, \pi_K, \theta_1, \cdots, \theta_K)$ are model parameters. The vector $\boldsymbol{\pi}$ indicates mixing proportions and the parameters corresponding to K likelihood functions are denoted by $(\theta_1, \cdots, \theta_K) = (\boldsymbol{\beta}_1, \sigma_1)\cdots, (\boldsymbol{\beta}_K, \sigma_K)$. The above expectation defines the posterior probability of belonging for each observation j. This posterior yields the probability that observation j belongs to the group k. In order to avoid the overflow and underflow in numerical calculations of the EM algorithm, we re-arrange the Equation (5.3.5) as follows

$$
\begin{aligned}
w_{jk}^{(m)}(\mathbf{y}_j, \Phi^{(m)}) &= \frac{\frac{\pi_k^{(m)}}{(2\pi\sigma_k^{2(m)})^{n/2}} \exp\left\{-\frac{(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})}{2\sigma_k^{2(m)}}\right\}}{\sum_{s=1}^{K} \frac{\pi_s^{(m)}}{(2\pi\sigma_s^{2(m)})^{n/2}} \exp\left\{-\frac{(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_s^{(m)})^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_s^{(m)})}{2\sigma_s^{2(m)}}\right\}} \\
&= \frac{\pi_k^{(m)} \exp\left\{-\frac{(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})}{2\sigma_k^{2(m)}}\right\}}{\sum_{s=1}^{K} \pi_s^{(m)} \left(\frac{\sigma_k^{2(m)}}{\sigma_s^{2(m)}}\right)^{n/2} \exp\left\{-\frac{(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_s^{(m)})^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_s^{(m)})}{2\sigma_s^{2(m)}}\right\}}.
\end{aligned}
$$

Using the substitution $\left(\frac{\sigma_k^{2(m)}}{\sigma_s^{2(m)}}\right)^{n/2} = \exp\left(\frac{n}{2}\log(\frac{\sigma_k^{2(m)}}{\sigma_s^{2(m)}})\right)$ and re-arranging the latter expression we get

$$
\begin{aligned}
w_{jk}^{(m)}(\mathbf{y}_j, \Phi^{(m)}) &= \sum_{s=1}^{K} \frac{1}{\frac{\pi_s^{(m)}}{\pi_k^{(m)}} \exp\left\{\frac{n}{2}\log\left(\frac{\sigma_k^{2(m)}}{\sigma_s^{2(m)}}\right) - \frac{(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_s^{(m)})^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_s^{(m)})}{2\sigma_s^{2(m)}}\right\}} \\
&\times \frac{1}{\exp\left\{\frac{(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})}{2\sigma_k^{2(m)}}\right\}}.
\end{aligned}
$$

Now going back to the expectation step, in the complete log-likelihood (5.3.4), the indicators are replaced with their expectations obtained in (5.3.5). For this reason, this step is called Expectation step or referred to as E-step in the EM algorithm. The E-step for the complete data log-likelihood is obtained as follows. In the m-th iteration, the expectation of complete log-likelihood gives the following equation

$$\begin{aligned} Q(\Phi|\Phi^{(m)}) &= E\log(L(\Phi)) \\ &= \sum_{j=1}^{J}\sum_{k=1}^{K} E(z_{jk})\log\pi_k^{(m)} - \sum_{k=1}^{K}\sum_{j=1}^{J}\frac{n}{2}E(z_{jk})\log(2\pi\sigma_k^{2(m)}) \\ &- \sum_{k=1}^{K}\frac{\sum\limits_{j=1}^{J} E(z_{jk})(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})}{2\sigma_k^{2(m)}}. \end{aligned}$$

If we substitute the expression (5.3.5) in the above equation we obtain

$$\begin{aligned} Q(\Phi|\Phi^{(m)}) &= E\log(L(\Phi)) \qquad (5.3.6) \\ &= \sum_{j=1}^{J}\sum_{k=1}^{K} w_{jk}^{(m)}\log\pi_k^{(m)} - \sum_{k=1}^{K}\sum_{j=1}^{J}\frac{n}{2}w_{jk}^{(m)}\log(2\pi\sigma_k^{2(m)}) \\ &- \sum_{k=1}^{K}\frac{\sum\limits_{j=1}^{J} w_{jk}^{(m)}(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})}{2\sigma_k^{2(m)}}. \end{aligned}$$

The next step in EM algorithm is the Maximization step through which we maximize $Q(\Phi, \Phi^{(m)})$ with respect to all parameters in the model i.e. $\Phi = (\boldsymbol{\pi}, \boldsymbol{\theta})$.

M-Step

We start by finding the maximum likelihood estimate of π_k which can be obtained by solving the equation $\frac{\partial Q}{\partial \pi_k} = 0$ subject to $\sum\limits_{k=1}^{K}\pi_k = 1$ and $\pi_k \geq 0$. Since we have this condition, we use the Lagrange multiplier in order to solve this constrained optimisation problem. The corresponding Lagrangian function is of the form

$$\mathcal{L}(\pi_k, \lambda) = \sum_{j=1}^{J}\sum_{k=1}^{K} w_{jk}^{(m)}\log\pi_k - \lambda(\sum_{k=1}^{K}\pi_k - 1),$$

differentiating the Lagrangian function with respect to π_k and setting equal to zero gives

$$\begin{aligned}
\frac{\partial \mathcal{L}(\pi_k, \lambda)}{\partial \pi_k} &= \sum_{k=1}^{K} \sum_{j=1}^{J} \frac{w_{jk}^{(m)}}{\pi_k} - \lambda \\
&= \sum_{k=1}^{K} \sum_{j=1}^{J} w_{jk}^{(m)} - \lambda \pi_k \\
&= 0.
\end{aligned}$$

Solving the last equation leads to the optimum estimate of mixing probability

$$\hat{\pi}_k^{(m+1)} = \frac{\sum\limits_{j=1}^{J} w_{jk}^{(m)}}{\lambda}, \tag{5.3.7}$$

substituting the optimal weight (5.3.7) in constraint $\sum\limits_{k=1}^{K} \pi_k^{(m)} = 1$, the Lagrange multiplier is obtained as

$$\lambda = \sum_{j=1}^{J} \sum_{k=1}^{K} w_{jk}^{(m)},$$

and replacing the obtained λ in expression (5.3.7) gives

$$\hat{\pi}^{(m+1)} = \frac{\sum\limits_{j=1}^{J} w_{jk}^{(m)}}{\sum\limits_{j=1}^{J} \sum\limits_{k=1}^{K} w_{jk}^{(m)}},$$

finally, using the fact that w_{ik}s are probabilities, we have $\sum\limits_{k=1}^{K} w_{ik}^{(m)} = 1$, hence $\lambda = \sum\limits_{j=1}^{J} 1 = J$. Thus the maximum likelihood estimate of π_k in the m-th iteration is given by

$$\hat{\pi}_k^{(m+1)} = \frac{1}{J} \sum_{j=1}^{J} w_{jk}^{(m)}. \tag{5.3.8}$$

Now we are going to obtain the maximum likelihood estimates of parameters $\theta_k = (\boldsymbol{\beta}_k, \sigma_k^2)$ corresponding to the kth component which is derived by solving the equation $\frac{\partial Q}{\partial \theta_k} = 0$. To obtain the maximum likelihood estimate of regression coefficients $\boldsymbol{\beta}_k$ we differentiate the Equation (5.3.6) with respect to $\boldsymbol{\beta}_k^{(m)}$ as follows

$$\begin{aligned}\frac{\partial Q}{\partial \boldsymbol{\beta}_k^{(m)}} &= \frac{\partial}{\partial \boldsymbol{\beta}_k^{(m)}} \sum_{k=1}^{K}\sum_{j=1}^{J} \frac{w_{jk}^{(m)}(\mathbf{y}_j - X\boldsymbol{\beta}_k^{(m)})^T(\mathbf{y}_j - X\boldsymbol{\beta}_k^{(m)})}{2\sigma_k^{2(m)}} \\ &= \sum_{j=1}^{J} \frac{-2w_{jk}^{(m)}\mathbf{X}^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k^{(m)})}{2\sigma_k^{2(m)}}, \end{aligned} \tag{5.3.9}$$

setting the above equation equal to zero, in the m-th iteration, the optimal solution for regression coefficients $\hat{\boldsymbol{\beta}}_k^{(m+1)} = (\beta_{k1}, \cdots, \beta_{kp})$ corresponding to the kth component is obtained as

$$\hat{\boldsymbol{\beta}}_k^{(m+1)} = \frac{(\mathbf{X}^T\mathbf{X})^{-1} \sum\limits_{j=1}^{J} w_{jk}^{(m)}\mathbf{X}^T\mathbf{y}_j}{\sum\limits_{j=1}^{J} w_{jk}^{(m)}}. \tag{5.3.10}$$

Now to avoid the overflow and underflow issue in numerical applications, we re-arrange the obtained optimal estimate as follows

$$\hat{\boldsymbol{\beta}}_k^{(m+1)} = \frac{\frac{1}{J}(\mathbf{X}^T\mathbf{X})^{-1} \sum\limits_{j=1}^{J} w_{jk}^{(m)}\mathbf{X}^T\mathbf{y}_j}{\hat{\pi}_k^{(m+1)}}. \tag{5.3.11}$$

Lastly, differentiating the Equation (5.3.6) with respect to $\sigma_k^{2(m)}$ gives

$$\begin{aligned}\frac{\partial Q}{\partial \sigma_k^{2(m)}} &= -\frac{n}{2}\sum_{j=1}^{J}\frac{w_{jk}^{(m)}}{\sigma_k^{2(m)}} \\ &+ \frac{\sum\limits_{j=1}^{J} w_{jk}^{(m)}(\mathbf{y}_j - \mathbf{X}\hat{\boldsymbol{\beta}}_k^{(m+1)})^T(\mathbf{y}_j - \mathbf{X}\hat{\boldsymbol{\beta}}_k^{(m+1)})}{2\sigma_k^{4(m)}}.\end{aligned}$$

Setting the above equation equal to zero results in the maximum likelihood estimate of component errors in the m-th iteration

$$\hat{\sigma}_k^{2(m+1)} = \frac{\sum\limits_{j=1}^{J} w_{jk}^{(m)}(\mathbf{y}_j - \mathbf{X}\hat{\boldsymbol{\beta}}_k^{(m+1)})^T(\mathbf{y}_j - \mathbf{X}\hat{\boldsymbol{\beta}}_k^{(m+1)})}{n\sum\limits_{j=1}^{J} w_{jk}^{(m)}}. \tag{5.3.12}$$

In numerical applications we use the following expression for the estimated variance

$$\hat{\sigma}_k^{2(m+1)} = \frac{\frac{1}{nJ}\sum_{j=1}^{J} w_{jk}^{(m)}(\mathbf{y}_j - \mathbf{X}\hat{\boldsymbol{\beta}}_k^{(m+1)})^T(\mathbf{y}_j - \mathbf{X}\hat{\boldsymbol{\beta}}_k^{(m+1)})}{\hat{\pi}_k^{(m+1)}}.$$

To sum up, in the M-step of the EM algorithm, and in the iteration m with corresponding model parameters $\Phi^{(m)}$, the aim is finding a set of parameters $\Phi^{(m+1)}$ such that

$$\Phi^{(m+1)} = \underset{\Phi}{\operatorname{argmax}}\, Q(\Phi|\Phi^{(m)}).$$

The EM algorithm alternates between the above E-step and the M-step until convergence. The stopping rule that we impose to confirm the convergence is based on the relative change in the log-likelihood. We say the convergence is attained when the following inequality holds

$$|\frac{l(\Phi^{(m+1)}) - l(\Phi^{(m)})}{l(\Phi^{(m)})}| < \epsilon, \tag{5.3.13}$$

where ϵ is a reasonably small value such as 10^{-10} and $l(\Phi)$ is calculated by

$$l(\Phi) = \sum_{j=1}^{J} \log \sum_{k=1}^{K} \frac{\pi_k}{(2\pi\sigma_k^2)^{n/2}} \exp\left\{-\frac{(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k)^T(\mathbf{y}_j - \mathbf{X}\boldsymbol{\beta}_k)}{2\sigma_k^2}\right\}.$$

Once the algorithm has stopped, the optimal estimates of the model parameters and the optimal classification of observations $\mathbf{y}_j; j = 1, \cdots, J$ are obtained. We shall refer to the above procedure of estimating the likelihood fusion model as fitting the likelihood fusion model. In the next section we marginally fit the likelihood fusion model to perform variable screening.

5.4 Marginal variable screening by likelihood fusion

This section explains the first stage of our two-stage procedure which is variable screening for high dimensional multivariate regressions where $p \gg n$. We conduct variable screening by marginally fitting the likelihood fusion model to data as follows. Consider the pair of $(\mathbf{Y}_{n\times J}, \mathbf{X}_{n\times p})$ where each column $\mathbf{y}_j, j = 1, \cdots, J$ of $\mathbf{Y}_{n\times J}$, indicates J independent multivariate response variables for each of which n observations are recorded, therefore each column is a vector of the form $\mathbf{y}_j = (y_{j1}, \cdots, y_{jn}), j = 1, \cdots, J$. Let $\mathbf{X}_{n\times p}$ denotes the design matrix where each column $\mathbf{x}_t, t = 1, \cdots, p$ records n observations on covariate $\mathbf{x}_t$. Thus each column of $\mathbf{X}_{n\times p}$ is a vector of the form $\mathbf{x}_t = (x_{t1}, \cdots, \mathbf{x}_{tn})$. Also suppose that we deal with a high dimensional case where $p \gg n$. To screen out the unimportant covariates with weak influence on the response variables we marginally fit the likelihood fusion model to each covariate $\mathbf{x}_t, t = 1, \cdots, p$. Note that in the model (5.2.1) the conditional distribution of each response variable given the design matrix $\mathbf{X}$, i.e. $\mathbf{y}_j|\mathbf{X}$, is considered whereas in the marginal likelihood fusion, the marginal conditional distribution $\mathbf{y}_j|\mathbf{x}_t$ for $j = 1, \cdots, J$ is considered. This means for each $\mathbf{x}_t, t = 1 \cdots, p$ the conditional distribution of $\mathbf{y}_j|\mathbf{x}_t$ is expressed by a marginal likelihood fusion as

$$f_t(\mathbf{y}_j|\mathbf{x}_t, \Phi) = \sum_{k=1}^{K} \pi_k f_{kt}(\mathbf{y}_j|\mathbf{x}_t\beta_k, \sigma_k^2), \tag{5.4.1}$$

where $\Phi = (\beta_1, \cdots, \beta_K, \sigma_1, \cdots, \sigma_K, \pi_1, \cdots, \pi_K)$ is the vector of model parameters. Each function $f_{kt}(\mathbf{y}_j|\mathbf{x}_t\beta_k, \sigma_k^2)$ is a likelihood function formulated as

$$\begin{aligned} f_{kt}(\mathbf{y}_j|\mathbf{x}_t\beta_k, \sigma_k^2) &= \prod_{i=1}^{n} f_{kt}^i(y_{ij}|x_{ti}\beta_k, \sigma_k^2) \\ &= \prod_{i=1}^{n} \frac{1}{(2\pi\sigma_k^2)^{1/2}} \exp\left\{-\frac{(y_{ij} - x_{ti}\beta_k)^2}{2\sigma_k^2}\right\} \\ &= \frac{1}{(2\pi\sigma_k^2)^{n/2}} \exp\left\{-\frac{(\mathbf{y}_j - \mathbf{x}_t\beta_k)^T(\mathbf{y}_j - \mathbf{x}_t\beta_k)}{2\sigma_k^2}\right\}. \end{aligned}$$

Similar to what we did in Section 5.3, the incomplete-data is augmented by defining indicator variables $(z_{jk})_{j=1\cdots,J;k=1,\cdots,K}$ leading to a complete-data likelihood

$$l_t(\Phi) = \sum_{j=1}^{J}\sum_{k=1}^{K}\left[z_{jk}\log\pi_k - \frac{nz_{jk}}{2}\log(2\pi\sigma_k^2) - \frac{z_{jk}(\mathbf{y}_j - \mathbf{x}_t\beta_k)^T(\mathbf{y}_j - \mathbf{x}_t\beta_k)}{2\sigma_k^2}\right].$$

In the m-th iteration, the probability of belonging to component k for each observation j is phrased by

$$w_{jk}^{(m)}(\mathbf{y}_j, \Phi^{(m)}) = \frac{\frac{\pi_k^{(m)}}{(2\pi\sigma_k^{2(m)})^{n/2}}\exp\left\{-\frac{(\mathbf{y}_j - \mathbf{x}_t\beta_k^{(m)})^T(\mathbf{y}_j - \mathbf{x}_t\beta_k^{(m)})}{2\sigma_k^{2(m)}}\right\}}{\sum_{s=1}^{K}\frac{\pi_s^{(m)}}{(2\pi\sigma_s^{2(m)})^{n/2}}\exp\left\{-\frac{(\mathbf{y}_j - \mathbf{x}_t\beta_s^{(m)})^T(\mathbf{y}_j - \mathbf{x}_t\beta_s^{(m)})}{2\sigma_s^{2(m)}}\right\}},$$

therefore in the E-step of the EM algorithm for marginal model, the indicators are replaced with the above equation to form the function $Q(\Phi|\Phi^{(m)}) = E\log(L(\Phi))$ with parameters $\Phi = (\boldsymbol{\pi}, \boldsymbol{\theta}) = (\pi_1, \cdots, \pi_K, (\beta_1, \sigma_1)\cdots, (\beta_K, \sigma_K))$. This is followed by differentiating the Q-function with respect to the model parameters which yields the maximum likelihood estimates of the model parameters. In the m-th iteration the mixing proportion is given by

$$\hat{\pi}_k^{(m+1)} = \frac{1}{J}\sum_{j=1}^{J} w_{jk}^{(m)},$$

and the maximum likelihood estimates of the parameters corresponding to all components in the model $(\theta_1, \cdots, \theta_K) = (\beta_1, \sigma_1)\cdots, (\beta_K, \sigma_K)$ are defined by

$$\begin{aligned}\hat{\beta}_k^{(m+1)} &= \frac{(\mathbf{x}_t^T\mathbf{x}_t)^{-1}\sum_{j=1}^{J} w_{jk}^{(m)}\mathbf{x}_t^T\mathbf{y}_j}{\sum_{j=1}^{J} w_{jk}^{(m)}}\\ \hat{\sigma}^{2(m+1)} &= \frac{\sum_{j=1}^{J} w_{jk}^{(m)}(\mathbf{y}_j - \mathbf{x}_t\beta_k^{(m+1)})^T(\mathbf{y}_j - \mathbf{x}_t\beta_k^{(m+1)})}{n\sum_{j=1}^{J} w_{jk}^{(m)}}.\end{aligned}$$

The stopping criteria for EM algorithm in marginal model fitting is the same as Equation (5.3.13) where the likelihood is defined by

$$l_t(\Phi) = \sum_{j=1}^{J} \log \sum_{k=1}^{K} \frac{\pi_k}{(2\pi\sigma_k^2)^{n/2}} \exp\left\{ -\frac{(\mathbf{y}_j - \mathbf{x}_t\beta_k)^T(\mathbf{y}_j - \mathbf{x}_t\beta_k)}{2\sigma_k^2} \right\}.$$

To sum up, our proposed screening procedure through the model (5.4.1) starts with fitting the model marginally to each covariate $\mathbf{x}_t; t = 1, \cdots, p$ with different number of components, $1 \leq K \leq K^{'}$. Then we calculate BIC values $\text{BIC}_{t,1}, \cdots, \text{BIC}_{t,K'}$ corresponding to each covariate $\mathbf{x}_t$ and each K

$$\text{BIC}_{t,K} = \frac{1}{J}\left[-2l_t(\Phi) + (|\theta_k| \times K + K - 1)\log(J)\right], \qquad (5.4.2)$$

where $l_t(\Phi)$ is the maximum log-likelihood, $|\theta_k|$ denotes the number of component parameters and J is the sample size of the response variable that we are aiming to classify into groups. As we fit the model marginally to each of covariates, we have a simple regression corresponding to each component rather than a multiple regression. Therefore we need to estimate two parameters β_k, σ_k^2 corresponding to each component i.e $|\theta_k| = 2$ and $K - 1$ mixing proportions.

The screening process is continued by finding the minimum BIC value for each covariate t, i.e. $\hat{\text{BIC}}_{t,\hat{K}} = \text{BIC}_{t_{\min}} = \min_{1 \leq K \leq K'}(\text{BIC}_{t,K})$. Having found the optimal BIC value $\text{BIC}_{t_{\min}}$ for all covariates $t = 1, \cdots, p$, we then use the reciprocal of BIC values $\text{RBIC}_t = 1/\text{BIC}_{t_{\min}}$; $t = 1, \cdots, p$ as our criteria to select variables. To this aim, we classify the RBIC_t values into two groups using k-means classification.

The k-means classification starts by considering K randomly chosen observations as K initial cluster centres $\mu_1, \cdots, \mu_k$. Then at each iteration the distance between these means and each data point is calculated and the data point is assigned to the cluster with the smallest distance. Then the mean of the cluster to which the data point is mapped is updated. This process is repeated until there is no change in assignment. In this algorithm the aim is minimizing the sum of squared distances from each data point to its corresponding cluster centre (MacQueen et al., 1967).

In order to perform k-means classification we use the built-in function `kmeans` in R-software. We set the algorithm of the `kmeans` function to the default algorithm which is the algorithm of Hartigan and Wong (1979). We also set the number of clusters to two when we apply `kmeans` function. Therefore, we obtain two clusters, where each cluster has a different mean value. Let $\mathcal{G}_h$ denote the cluster with larger mean value and $\mathcal{G}_l$ denote the cluster with smaller cluster mean value. The screening process is completed by selecting the predictors which their corresponding RBIC belong to the cluster $\mathcal{G}_h$ with larger cluster mean. Therefore, the signal set $\mathcal{S}$ is defined as

$$\mathcal{S} = \{\mathbf{x}_t;\ 1 \leq t \leq p \ \mid \ \mathrm{RBIC}_t \in \mathcal{G}_h\} \tag{5.4.3}$$

This process is summarised through Algorithm 5.1.

Algorithm 5.1: Marginal variable screening by likelihood fusion

```
1. Repeat for $t = 1, \cdots, p$;
2. Repeat for $K = 1, \cdots, K'$;
3. Initialize $\beta_k, \sigma_k, \pi_k$ and calculate the initial log-likelihood.
4. **E step**. In the $m$-th iteration given the current parameters calculate posterior probabilities
   $w_{jk}^{(m)}(\mathbf{y}_j, \Phi^{(m)}) = \frac{\frac{\pi_k^{(m)}}{(2\pi\sigma_k^{2(m)})^{n/2}} \exp\left\{-\frac{(\mathbf{y}_j-\mathbf{x}_t\beta_k^{(m)})^T(\mathbf{y}_j-\mathbf{x}_t\beta_k^{(m)})}{2\sigma_k^{2(m)}}\right\}}{\sum\limits_{s=1}^{K} \frac{\pi_s^{(m)}}{(2\pi\sigma_s^{2(m)})^{n/2}} \exp\left\{-\frac{(\mathbf{y}_j-\mathbf{x}_t\beta_s^{(m)})^T(\mathbf{y}_j-\mathbf{x}_t\beta_s^{(m)})}{2\sigma_s^{2(m)}}\right\}},$
5. **M step**. Update parameters using posterior probabilities in 4.
   $\hat{\pi}_k^{(m+1)} = \frac{1}{J}\sum_{j=1}^{J} w_{jk}^{(m)}$
   $\hat{\beta}_k^{(m+1)} = \frac{(\mathbf{x}_t^T\mathbf{x}_t)^{-1}\sum_{j=1}^{J} w_{jk}^{(m)}\mathbf{x}_t^T\mathbf{y}_j}{\sum_{j=1}^{J} w_{jk}^{(m)}}$
   $\hat{\sigma}^{2(m+1)} = \frac{\sum_{j=1}^{J} w_{jk}^{(m)}(\mathbf{y}_j - \mathbf{x}_t\beta_k^{(m+1)})^T(\mathbf{y}_j - \mathbf{x}_t\beta_k^{(m+1)})}{n\sum_{j=1}^{J} w_{jk}^{(m)}}.$
6. Calculate new log-likelihood, if the convergence criterion is not satisfied return to step 4.
7. Calculate $\mathrm{BIC}_{t,K}$.
8. End repeat K.
9. Find $\min\limits_{1\le K\le K'}(\mathrm{BIC}_K)$
10. End repeat t.
```

5.5 Classification by likelihood fusion

We recall that the second stage of our proposed procedure is the classification stage. In this stage, the likelihood fusion model (5.2.1) is fitted to the selected predictors in the screening stage and the response variables. Let columns of matrix $\tilde{\mathbf{X}}$ contain the selected predictors $\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_{\tilde{p}}$ in the screening stage. Then, in the classification stage, the likelihood fusion model is fitted to data $(\mathbf{y}_j, \tilde{\mathbf{X}})$ as follows

$$f(\mathbf{y}_j|\tilde{\mathbf{X}}, \Phi) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{y}_j|\tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}_k, \sigma_k^2), \quad j = 1, \cdots, J, \tag{5.5.1}$$

where $\tilde{\boldsymbol{\beta}}_k$ is a $\tilde{p}$-dimensional coefficient vector. In classification stage we wish to find the optimal classification for the data. To this aim, we fit the model (5.3.1) with an EM algorithm (as it is explained in Section 5.3.1) to the data $(\mathbf{y}_j, \tilde{\mathbf{X}})$ with different number of components $1 \leq K \leq K'$ and calculate the corresponding BIC values $\text{BIC}_1, \cdots, \text{BIC}'_K$. These BIC values are calculated according to the following expression

$$\text{BIC}_K = \frac{1}{J}\left[-2l(\Phi) + (|\tilde{\theta}_k| \times K + K - 1)\log(J)\right], \tag{5.5.2}$$

Similar to the BIC calculated in screening stage, $l(\Phi)$ is the maximum log-likelihood, $|\tilde{\theta}_k|$ denotes the number of component parameters and J is the number of response variables that we wish to classify. Here, the regression coefficient vector is a $\tilde{p}$-dimensional vector $\tilde{\boldsymbol{\beta}} = (\tilde{\beta}_1, \cdots, \tilde{\beta}_{\tilde{p}})$. Therefore, the number of parameters corresponding to each component is $|\tilde{\theta}_k| = \tilde{p} + 1$. We also need to estimate $K - 1$ mixing proportions.

Having found these BIC values, the optimal number of components is the one with the minimum BIC value, i.e., $\hat{\text{BIC}}_{\hat{K}} = \min_{1 \leq K \leq K'} (\text{BIC}_K)$. Accordingly, the classification corresponding to $\hat{K}$ yields the optimal classification.

5.6 Initialisation of EM algorithm

Finding proper initial values to start the EM algorithm is challenging since the algorithm is sensitive to the choice of starting points. To get an idea we first carried out some random choices to initialise the algorithm and conducted a few simulations but the performance was very poor in both screening and clustering stage. Therefore, to choose more reasonable initial values, we fitted a simple linear regression to the dataset $(\mathbf{y}_j, \mathbf{x}_t)$ where $j = 1, \cdots, J$ and $t = 1, \cdots, p$ and both $\mathbf{y}_j$ and $\mathbf{x}_t$ are n-vectors. Corresponding to each single regression we recorded two estimated parameters $\hat{\beta}_{tj}$ and $\hat{\sigma}_{tj}$. This provided us with a set of size $|p \times J|$ contained regression coefficient estimates β_{tj} and a set of size $|p \times J|$ which contained σ^2_{tj} values. Then we applied the Mclust (Fraley et al., 2012) R-software to classify each set of parameters into different number of components, $K = 1, \cdots, K'$. This software fits a normal mixture model to classify the data. For each parameter and corresponding to each $1 \leq K \leq K'$, we extracted the K dimensional mean vector to be used as initial values. For example to fit a marginal likelihood fusion model with K components, the initial values for parameter β, were the elements of the mean vector $\boldsymbol{\beta}^K_{\text{mean}} = (\beta_1, \cdots, \beta_K)$. Likewise, the initial values for parameter σ^2, were the elements of the mean vector $\boldsymbol{\sigma}^{2K}_{\text{mean}} = (\sigma_1, \cdots, \sigma_K)$. Applying this technique slightly improved the accuracy of both screening and classification outcomes.

To enhance the performance of screening and classification, even more, we modified the above approach as following. We first explain how we did the initialisation for the screening stage. We first classified the response variables $\mathbf{y}_j; j = 1, \cdots, J$ into K number of components using Mclust (Fraley et al., 2012). Suppose that K components and their corresponding elements are denoted by

$$\begin{aligned} C_1 &= \{\mathbf{y}_{j1,i_1} \mid i_1 = 1, \cdots, n_1\} \\ C_2 &= \{\mathbf{y}_{j2,i_2} \mid i_2 = 1, \cdots, n_2\} \\ &\vdots \\ C_K &= \{\mathbf{y}_{jK,i_K} \mid i_K = 1, \cdots, n_K\}, \end{aligned} \tag{5.6.1}$$

where $\sum_{k=1}^{K} n_k = J$. Since in the screening stage we fit a marginal likelihood fusion model to each predictor $\mathbf{x}_t$, in the initialisation step, we also fitted a multivariate simple linear regression to the elements of each component and each $\mathbf{x}_t$. For example, consider the kth component C_k with n_k number of elements. Let these elements form the columns of matrix Y_{C_k}. The following multivariate simple regression was fitted to the predictor $\mathbf{x}_t$ and all response variables in the kth component Y_{C_k}

$$\mathbf{Y}_{C_k} = \mathbf{x}_t \boldsymbol{\beta}_{t,C_k} + \boldsymbol{\epsilon}_{C_k}, \tag{5.6.2}$$

therefore, for the kth component we got the set of estimated parameters $\{\hat{\boldsymbol{\beta}}_{t,C_k}, \hat{\boldsymbol{\sigma}}^2_{t,C_k}\}$ where $\hat{\boldsymbol{\beta}}_{t,C_k} = (\hat{\beta}_{t,1}, \cdots, \hat{\beta}_{t,n_k})$ and $\hat{\boldsymbol{\sigma}}^2_{t,C_k} = (\hat{\sigma}_{t,1}, \cdots, \hat{\sigma}_{t,n_k})$. Then the initial values for the parameters in the kth component, i.e. $(\beta_{t,k}, \sigma^2_{t,k})$ were set to $\beta_{t,k} = 1/n_k \sum_{i_1=1}^{n_k} \hat{\beta}_{t,i_k}$ and $\sigma_{t,k} = 1/n_k \sum_{i_1=1}^{n_k} \hat{\sigma}^2_{t,i_k}$, respectively. The parameters for the rest of components $(\beta_{t,2}, \sigma^2_{t,2}), \cdots, (\beta_{t,K}, \sigma^2_{t,K})$ were found similarly. These values were used as initial values to start the algorithm. With this approach the screening performance of our method was improved significantly. In the classification stage of our method we took the same path for initialisation but the fitted regression model was a multiple regression rather than a single regression.

Parameter initialisation for the classification stage also starts by classifying $\mathbf{y}_j; j = 1, \cdots, J$ into K number of components using Mclust. These components and the elements are shown in (5.6.1). Suppose that the set of selected covariates in the screening stage is denoted by $\tilde{X} = \{\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_{\tilde{p}}\}$. Also assume that the subset $\tilde{\mathbf{X}}_{10} = \{\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_{10}\}$ contains the first ten selected covariates with highest reciprocal BIC. Considering the kth component as an example, we fitted a multivariate multiple regression model to the dataset $(\mathbf{Y}_{C_k}, \tilde{\mathbf{X}}_{10})$

$$\mathbf{Y}_{C_k} = \tilde{\mathbf{X}}_{10}\mathbf{B}_{10,C_k} + \boldsymbol{\epsilon}_{C_k}, \tag{5.6.3}$$

where $\mathbf{Y}_{C_k}$ is a matrix, columns of which are the elements in the kth component i.e. $\{\mathbf{y}_{jk,i_k} \mid i_k = 1, \cdots, n_k\}$. Therefore we obtained the parameters corresponding to the kth component which were the coefficient estimates $\mathbf{B}_{10,C_k} = (\hat{\beta}_{ij})_{10\times n_k} = (\hat{\boldsymbol{\beta}}_1, \cdots, \hat{\boldsymbol{\beta}}_{10})^T$ where $\hat{\boldsymbol{\beta}}_i, i = 1, \cdots, 10$ is a n_k-dimensional vector. The estimated variance for the kth component was $\hat{\boldsymbol{\sigma}}^2_{C_k} = (\hat{\sigma}_1, \cdots, \hat{\sigma}_{n_k})$. Having found these estimates, we then set the initial values for the parameters of the kth component to $\boldsymbol{\beta}_{C_k} = (\beta_1, \cdots, \beta_{10})$ where $\beta_i = \frac{1}{n_k}\sum_{j=1}^{n_k}\hat{\beta}_{ij}, i = 1, \cdots, 10$ and $\sigma^2_{C_k} = \frac{1}{n_k}\sum_{i=1}^{n_k}\hat{\sigma}_i$. After finding component parameters for the rest of components $C_2, \cdots, C_K$, these parameters were used as initial values to fit the likelihood fusion model to the data $(\mathbf{y}_j, \tilde{\mathbf{X}}_{10}); j = 1, \cdots, J$ as follows

$$f(\mathbf{y}_j|\tilde{\mathbf{X}}_{10}, \Phi_{10}) = \sum_{k=1}^{K}\pi_k f_k(\mathbf{y}_j|\tilde{\mathbf{X}}_{10}\boldsymbol{\beta}_{10,k}, \sigma^2_k), \tag{5.6.4}$$

where $\boldsymbol{\beta}_{10,k}$ is a 10-dimensional coefficient vector. The optimal mixing proportion $\hat{\boldsymbol{\pi}} = (\pi_1, \cdots, \pi_K)$ obtained through fitting the above model to the first ten covariates was used to update the above classification, shown in (5.6.1), to $\tilde{C}_k, k = 1 \cdots, K$. Note that this classification is different from the one obtained by implementing Mclust. Consider the k-th component as an example, we fitted a multivariate regression model to the dataset $(\mathbf{Y}_{\tilde{C}_k}, \tilde{\mathbf{X}})$ as follows

$$\mathbf{Y}_{\tilde{C}_k} = \tilde{\mathbf{X}}\tilde{\mathbf{B}}_{\tilde{C}_k} + \boldsymbol{\epsilon}_{\tilde{C}_k}, \tag{5.6.5}$$

as pointed out before, $\mathbf{Y}_{\tilde{C}_k}$ is a matrix columns of which are the elements in the k-th component i.e. $\{\mathbf{y}_{jk,i_k} \mid i_k = 1, \cdots, n_k\}$. The columns of $\tilde{X}$ are the selected covariates in the screening stage. From fitting the above model the regression coefficient estimates $\tilde{\mathbf{B}}_{\tilde{p},\tilde{C}_k} = (\hat{\beta}_{ij})_{\tilde{p}\times n_k}$ and the estimated variances $\hat{\boldsymbol{\sigma}}^2_{\tilde{C}_k} = (\hat{\sigma}_1, \cdots, \hat{\sigma}_{n_k})$ were obtained. Similar to what we explained above we took the mean of regression coefficients across columns of $\tilde{\mathbf{B}}_{\tilde{p},\tilde{C}_k}$ as initial value of coefficient vector. Therefore, the initial value for the coefficient vector corresponding to the kth component was set to $\tilde{\boldsymbol{\beta}}_{\tilde{C}_k} = (\tilde{\beta}_1, \cdots, \tilde{\beta}_{\tilde{p}})$ where $\tilde{\beta}_i = \frac{1}{n_k}\sum_{j=1}^{n_k} \hat{\beta}_{ij}, i = 1, \cdots, \tilde{p}$. The initial value for the variance of the kth component was set to $\sigma^2_{C_k} = \frac{1}{n_k}\sum_{i=1}^{n_k} \hat{\sigma}_i$.

5.7 Simulation studies

Simulations conducted in this part serve the following purposes:

(a) To assess the accuracy of screening which is the first stage of our proposed two-stage procedure.

(b) To compare the marginal screening based on the likelihood fusion of regressions with SNR-based screening.

(c) To compare the classification performance of our method with that of normal mixture model introduced in Section 2.5.

5.7.1 Data generation

We conducted 30 simulations wherein 131 multivariate observations $\mathbf{y}_j = (y_{j1}, \cdots, y_{jn})^T$; $j = 1, \cdots, 131$ were generated from a data fusion mixture model with $K = 2$ and $K = 5$ components. Suppose that observation $\mathbf{y}_j$ is a member of component k, then the following regression model holds for any $\mathbf{y}_j$ which belongs to this component:

$$\mathbf{y}_j = \mathbf{X}\boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k. \tag{5.7.1}$$

In the above model, $\mathbf{X}$ is a $n \times p$ covariate matrix which is fixed across all components $k = 1, \cdots, K$. This matrix was generated by simulating n iid samples from $N_p(\mathbf{0}, \Sigma_{p\times p})$ where $\Sigma_{p\times p}$ is the variance-covariance matrix of a random subset of gene expressions in cancer data. The error term is an n-dimensional vector such that $\boldsymbol{\epsilon}_k \sim N_n(\mathbf{0}, \sigma_k^2\mathbf{I}_n)$ and $\boldsymbol{\beta}_k$ is a p-dimensional coefficient vector.

We designed two scenarios with different settings. Scenario 1 wherein component variances are not equal for all components in the model and scenario 2 with equal variances for all components. In each of these scenarios we also considered models with well separated and not well separated components. In scenarios where the components are well separated, there is no substantial differences between the classification performance of Mclust and likelihood fusion model (LFM). Therefore, we present those cases where LFM noticeably outperforms Mclust in classifications. For each of these scenarios various settings differing in sample size, the number of covariates and the number of true active covariates were considered.

Scenario 1 (Non-equal variances for components): In this scenario we considered various settings through the following combinations of (p, n, a) where $p = 500, 1000$ denotes the number of covariates, $n = 42, 84$ is the sample size and $a = 3, 8$ denotes the number of non-zero or active covariates in the model. The variances were set to $\sigma^2 = (1.1, 0.4, 0.9, 0.8, 0.2)$ and $\sigma^2 = (0.2, 0.7)$ for the model with five and two components respectively. Also mixing proportions were set to $\pi_k = 0.2$ for $k = 1, \cdots, 5$ and $\pi = (0.4, 0.6)$. The coefficient vector $\boldsymbol{\beta}_k$ is a sparse vector wherein non-zero elements were simulated from $N(\mu_k, 0.001)$. The position of these non-zero covariates were selected randomly. For $K = 2$, we set $\mu_1 = 0.8$ and $\mu_2 = 1$ and where we have $K = 5$ components we set $(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) = (0.4, 0.6, 0.8, 1, 1.2)$.

Scenario 2 (Equal variances for all components): In this scenario we considered the same various settings as scenario 1 but with $\sigma_k^2 = 0.7; k = 1, \cdots, 5$ for the model with five components and $\sigma^2 = (0.7, 0.7)$ for the model with two components.

5.7.2 Performance of likelihood fusion

As we mentioned earlier, we implement the likelihood fusion model to screen high dimensional predictors and then we apply the likelihood fusion to the reduced set of predictors and classify the response variables. In this section, in the first part, the marginal screening explained in Section 5.4 is applied to screen the predictors. To evaluate the performance of likelihood fusion, the screening accuracy is reported in terms of sensitivity and specificity percentages. Then in the second part, we perform the two-stage procedure. In the first stage, we implement the marginal screening to screen covariates then we fit the model (5.2.1) to the selected covariates and the response variables to classify the response variables in the data. To assess the performance of likelihood fusion classification, we compare the result with the classification performed by normal mixture model. This comparison is based on the adjusted Rand index (Hubert and Arabie, 1985) which was introduced in Section 2.5.2. In order to perform normal mixture modelling to classify the data, we implement Mclust package (Fraley et al., 2012) written in R-software. All the programming for this chapter is also written in R software.

5.7.2.1 Performance in screening

In this section, we intend to evaluate the screening performance of likelihood fusion model based on the accuracy of this approach in correctly detecting non-zero covariates, reported as sensitivity, and also in discarding zero covariates reported as specificity. We conducted 30 simulations for scenario1 (non-equal variances) and scenario 2 (equal variances). In each simulation we marginally fittd the likelihood fusion model to each covariate $\mathbf{x}_t; t = 1, \cdots p$ as it was explained in Section 5.4. After doing a few pilot studies we observed that the number of components alternated between $\hat{K} = 2$ and $\hat{K} = 3$. In other words, the lowest BIC for each predictor $\mathbf{x}_t$ was attained at $K = 2, 3$. Therefore to reduce the computational cost, we restricted the number of components to $1 \leq K \leq 4$ for the setting with $K = 2$. For the same reason in settings with

$K = 5$, we restricted the number of components to $3 \leq K \leq 7$. Through these simulations we also explore how the sample size, the dimension of covariates and the number of active covariates can influence the screening accuracy.

Effect of increasing active size on screening accuracy

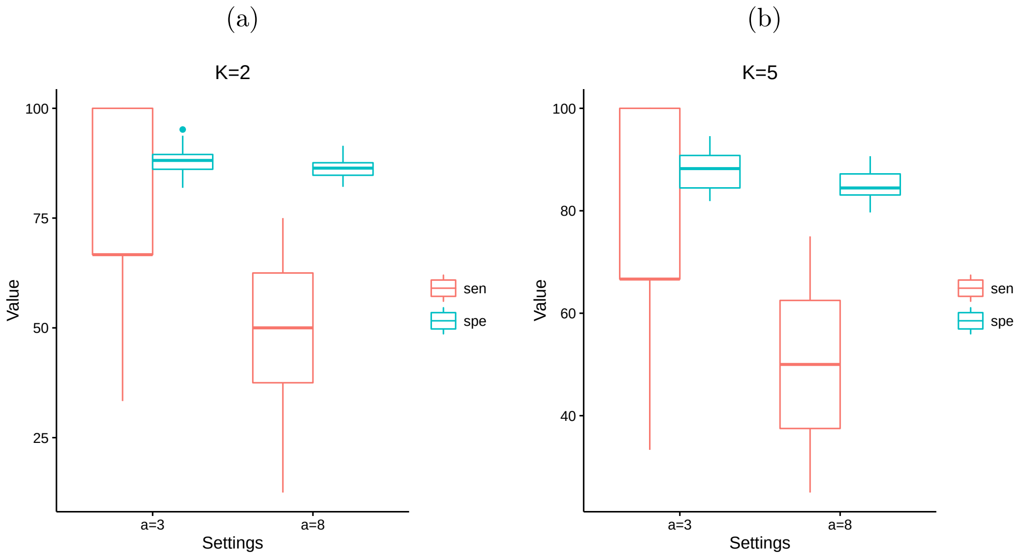


Figure 5.7.1: **Scenario 1 (Non-equal variances for components)**: Sensitivity and specificity percentages where all components in mixture model have different variances, with $p = 500, n = 42$ and $a = 3, 8$ active number of covariates where mixture model has (a) K=2 and (b) K=5 number of components.

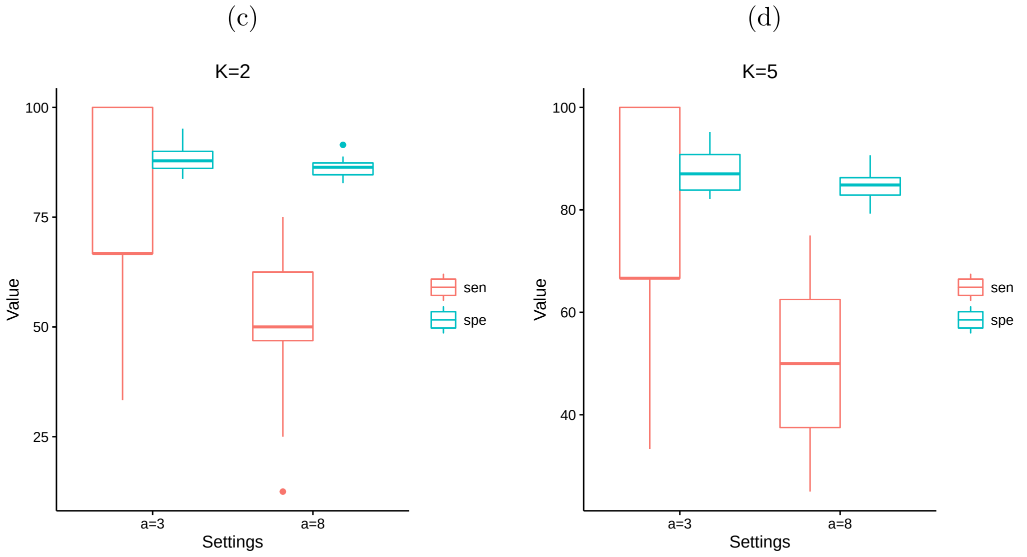


Figure 5.7.2: **Scenario 2 (Equal variances for all components)**: Sensitivity and specificity percentages where all components in mixture model have same variance, with $p = 500, n = 42$ and $a = 3, 8$ active number of covariates where mixture model has (a) K=2 and (b) K=5 number of components.

Results presented in Figure 5.7.1 and 5.7.2 uncover that increasing the number of active covariates leads to a noticeable decline in the sensitivity percentages. This means that the ability of the process in correctly capturing the active or non-zero covariates becomes poor when there are more active covariates in the model. The reason behind this reduction is the correlation between the active covariates, a phenomena called masking effect (Farrar and Glauber, 1967), (Wang, 1996). We study this in more details in the following.

Masking effect

In this part, we explore how the correlation between covariates can affect the accuracy of variable screening and as a result alter the sensitivity percentage. We show that when we increase the number of non-zero or active covariates in the model, the sensitivity decreases. Despite the fact that having more active covariates in the model should make it easier for the process to detect them, recognising the active covariates becomes more difficult. The following results illustrate a phenomena known as masking effect which is induced by correlations between covariates. These correlations act like a mask and cause some difficulties in identifying the important covariates in the variable screening procedure (Berry and Feldman, 1985). To monitor this effect more carefully we simulated data according to scenario 1 with the number of covariates $p = 500$ and the sample size $n = 42$ where the number of components is $K = 5$. We first set the number of active or non-zero covariates to $a = 20$ and after screening via marginal likelihood fusion model the obtained sensitivity was 30% with the corresponding specificity of 85.20%. We removed 3 of the covariates for which the correlation with at least one of the other covariates was higher than 0.5. Then we performed the marginal screening procedure again on this reduced set of covariates. As it was expected in the absence of these highly correlated covariates, the sensitivity percentage increased by 5% to the value of 35% with the corresponding specificity of 86.35%.

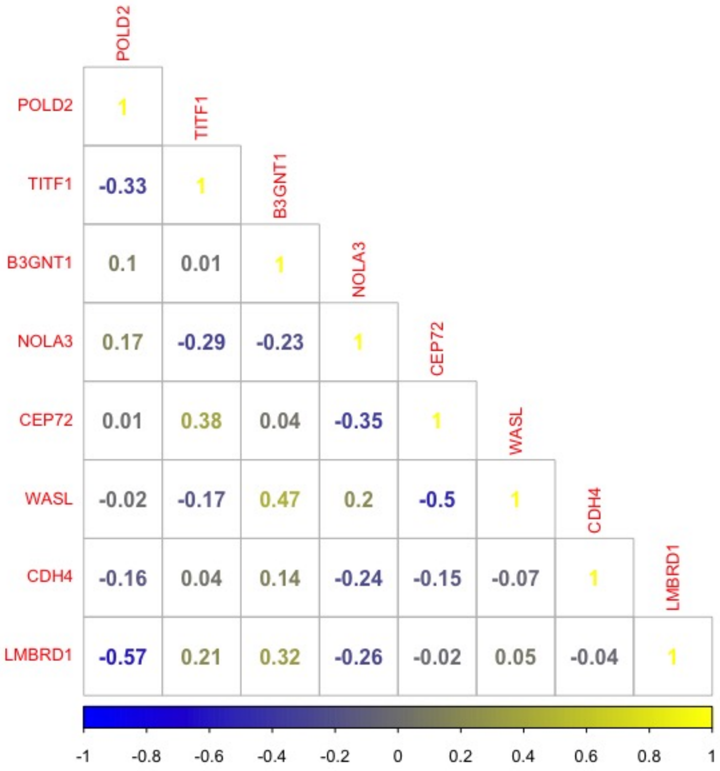


Figure 5.7.3: Correlation structures among active covariates with active size $a = 8$, $p = 500$ and $n = 42$ under scenario 1.

In another case, we set the number of non-zero covariates to $a = 8$ and we applied likelihood fusion to screen covariates. We obtained the sensitivity of 62.5% and the specificity of 83.9%. The correlation structure among these 8 active covariates is shown in Figure 5.7.3. We removed two of active covariates POLD2 and LMBRD1 with the correlation coefficient -0.57 and repeated the screening procedure on the model with 6 active covariates. Correlations among these 6 covariates is depicted in Figure 5.7.4. In the absence of these two correlated variables, the sensitivity improved to 83.3% with the corresponding specificity of 83.8%.

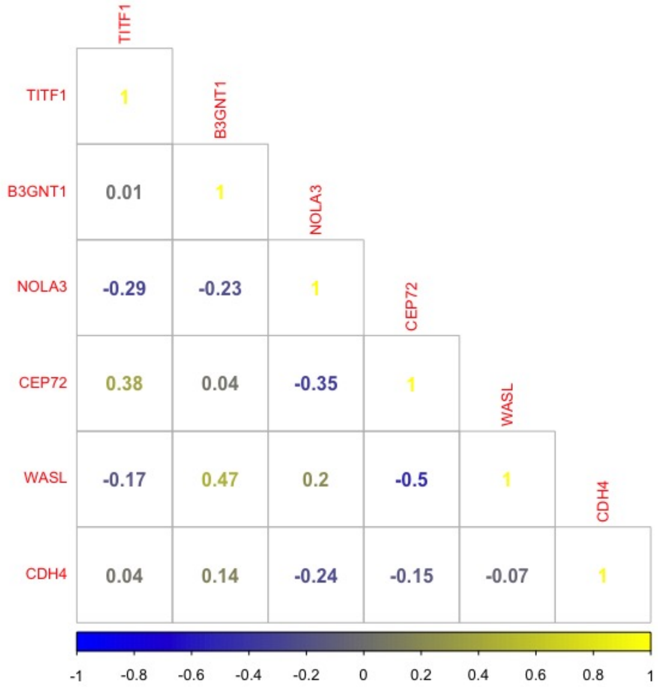


Figure 5.7.4: Correlation structures among active covariates with active size $a = 6$.

Now we investigate the mask effect in large scale simulations. We conducted 20 simulations with $p = 500$, $n = 42, a = 8$ and $K = 5$ under scenario 1. We performed the screening procedure when all 8 active covariates were included in the model and record the sensitivity values. In order to see the effect of correlations on the screening accuracy we removed the highly correlated predictors and recorded the sensitivity. To this aim, we identified the active predictors with a correlation higher than 0.5 with other active covariates. For each predictor $\mathbf{x}_t$, we recorded the number of highly correlated predictors with this predictor. Then we removed those predictors that were correlated with a larger number of predictors. Therefore, we obtained a reduced set of predictors with a weaker correlation structure. Then we performed screening on this reduced set of predictors and recorded the sensitivity values. The resulted sensitivity and specificity corresponding to the case with all 8 predictors in the model and corresponding to the reduced set with lower correlations is presented in Figure 5.7.5.

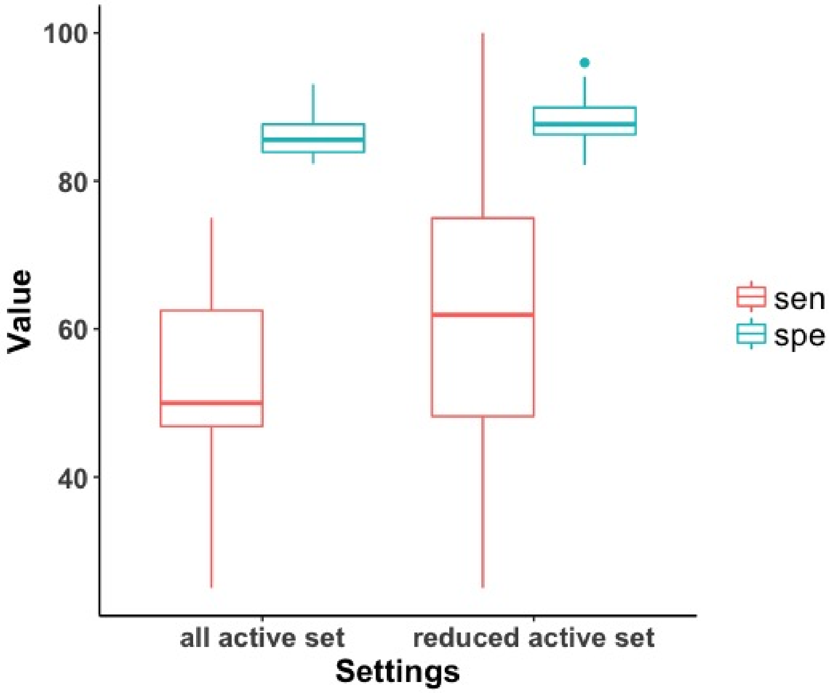
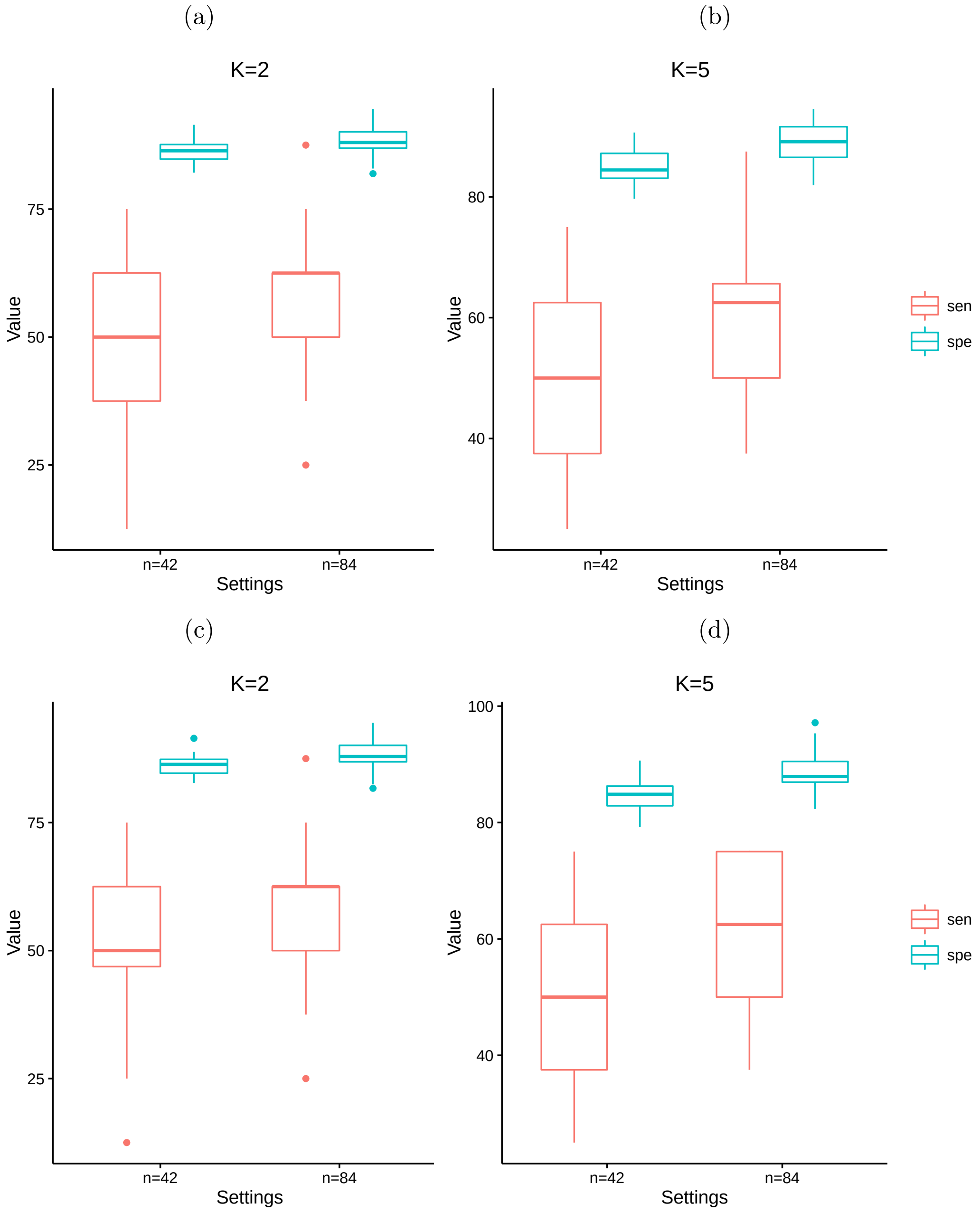


Figure 5.7.5: Box plots of sensitivity and specificity values resulted from screening when all active covariates are included in the model and when the highly correlated active covariates are removed from the model. Results obtained from 20 simulations $p = 500$, $n = 42, a = 8$ and $K = 5$ under scenario 1.

Another factor that can have an influence on screening accuracy is the sample size. In the following we wish to investigate how sample size affect the screening accuracy. To this aim, we conducted 20 simulations under scenario1 and scenario 2 with different sample sizes. Results presented in Figure 5.7.6 reveal that increasing the sample size can improve the selection accuracy under both scenarios with equal and non-equal component variances. Moreover, in settings with $n = 42$ even though the sample size is very small, the screening accuracy of our proposed method is larger than 50%. Through the following simulations we also show that when we have enough sample size, even when the number of predictors are as large as a thousand, the marginal screening performs well in identifying important predictors with the selection precision as high as 75%. These results are presented in Figure 5.7.7.

Effect of increasing sample size on sensitivity and specificity


(a)

(b)

(c)

(d)

Figure 5.7.6: Sensitivity and specificity percentages for settings with $p = 500, n = \{42, 84\}, a = 8, K = \{2, 5\}$ under (a), (b) scenario 1 where components in the mixture model have different variances. Results obtained in (c) and (d) are under scenario 2 where all components have the same variance.

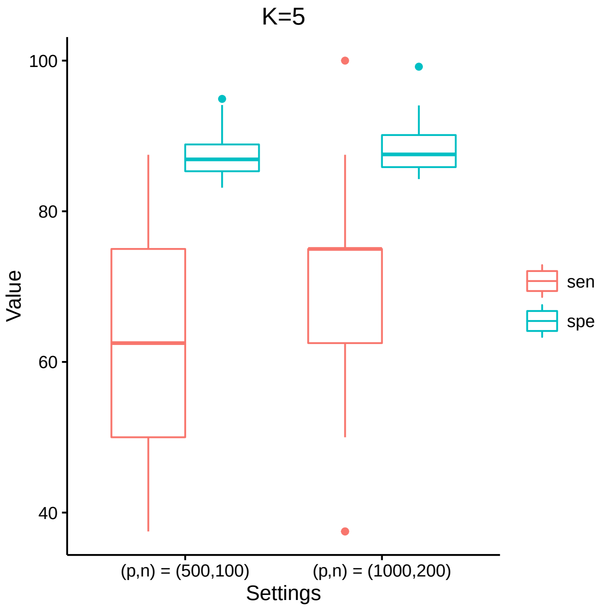


Figure 5.7.7: Sensitivity and specificity percentages under scenario 1 with non-equal component variances for setting with $a = 8, K = 5$ where we have $(p, n) = (500, 100)$ and $(p, n) = (1000, 200)$.

In the following we compare the performance of our two proposed screening methods. Screening through the likelihood fusion model (LFM) and screening via SNR. To this aim, we generated 30 datasets as explained in Section 5.7.1 under both scenario 1 and scenario 2 with $p = 500, n = 42$ and $a = 8$ and applied the aforementioned screening methods on these data. We compared the screening performance by comparing specificity values for both methods while sensitivities were fixed. We explain how we fixed sensitivities in screening by marginal likelihood fusion. The same procedure was applied to fix the sensitivities in SNR-based screening. We remind that the marginal screening explained in Section 5.4 is completed by finding the optimal $\text{BIC}_{t_{\min}}$ for each covariate $\mathbf{x}_t; t = 1, \cdots, p$ and then thresholding the reciprocal of BIC values $\text{RBIC}_t = 1/\text{BIC}_{t_{\min}}$; $t = 1, \cdots, p$. We thresholded the RBIC_t values at levels of $\text{RBIC}_{(j)}; j = 1, \cdots, 8$ corresponding to the 8 active covariates. We ordered the $\text{RBIC}_{(j)}$ values increasingly thus by thresholding RBIC_t values at the level of $\text{RBIC}_{(1)}$, the selected subset of covariates by the screening process contained all active covariates which gave a sensitivity value of 100%. Similarly, setting the threshold level at the largest value $\text{RBIC}_{(10)}$ gave a sensitivity of 10%. This way, we obtained a set of sensitivity values as $12.5\%, 25\%, 37\%, 50\%, 62\%, 75\%, 37\%, 87.5\%, 100\%$.

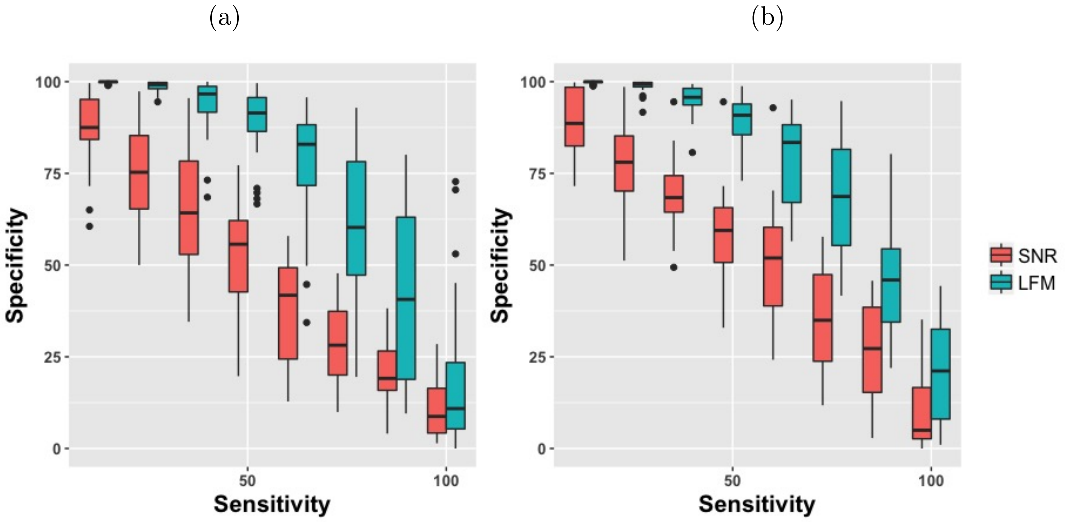


Figure 5.7.8: **Scenario 1 (Non-equal variances for components)**: Sensitivity and specificity percentages resulted from two screening procedures: marginal screening through the likelihood fusion model (LFM) and screening by SNR. Simulations were conducted where $p = 500, n = 42$ and $a = 8$ for the model with (a) K=2 and (b) K=5 components.

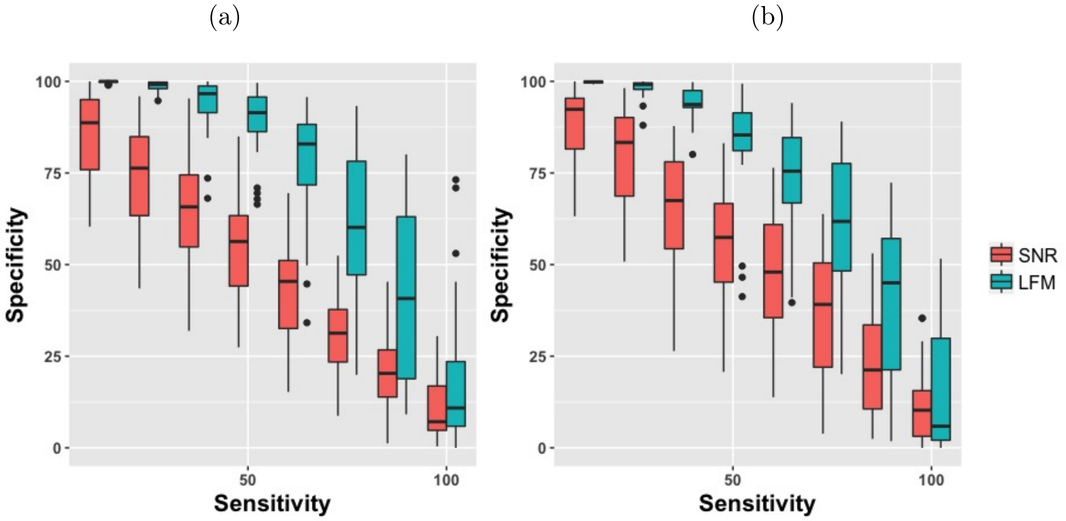


Figure 5.7.9: **Scenario 2 (Non-equal variances for all components)**: Sensitivity and specificity percentages resulted from two screening procedures: marginal screening through the likelihood fusion model (LFM) and screening by SNR. Simulations were conducted where $p = 500, n = 42$ and $a = 8$ for the model with (a) K=2 and (b) K=5 components.

The obtained results in Figure 5.7.8 and Figure 5.7.9 show that under both scenarios with equal and non-equal component variances, when the data have a hidden group structure, screening by likelihood fusion leads to a more accurate selection compare to the SNR-based screening.

5.7.2.2 Performance in clustering

In this part we examine the classification performance of the likelihood fusion model. Suppose we are interested in classifying multivariate response variables $\mathbf{y}_1, \cdots, \mathbf{y}_J$ into groups. To this aim, we implement the likelihood fusion model to classify $\mathbf{y}_1, \cdots, \mathbf{y}_J$. Then we compare the classification obtained using likelihood fusion with that of the finite mixture model. The advantage of applying our method is that we can take the covariates into account in the process of classification and this leads to a more reliable classification.To classify multivariate response variables by fitting finite mixture model we use the Mclust software. We assess the precision of these two classifications by calculating the adjusted Rand index of clustering for each approach.

Consider the data set $(\mathbf{Y}_{n\times J}, \mathbf{X}_{n\times p})$ where $p \gg n$ and let $\mathbf{y}_1, \cdots, \mathbf{y}_J$ denote the columns of matrix $\mathbf{Y}$. We intend to cluster $\mathbf{y}_1, \cdots, \mathbf{y}_J$ through our proposed two-phase procedure. We generated 20 datasets under both scenario 1 and scenario 2 according to what was explained in the Section 5.7.1 and applied our two-phase procedure to the simulated data. In the first stage, we fitted the marginal likelihood fusion model (5.4.1) to perform a variable screening and reduce the dimension of covariates. In the second stage we fitted the full model (5.2.1) to the dataset $(\mathbf{Y}_{n\times J}, \tilde{\mathbf{X}}_{n\times\tilde{p}})$, where $\tilde{\mathbf{X}}_{n\times\tilde{p}}$ is a matrix columns of which are the selected predictors by screening in the first stage. We imitated our real data in some of simulation settings and set the sample size $n = 42$. In such cases, even after screening, the number of selected covariates $\tilde{p}$ was still larger than n and the problem was ill-posed. Therefore we did a forward selection to further reduce the number of selected predictors prior to moving to the classification stage.

Forward selection

Suppose the set of selected covariates in the screening stage is denoted by $\{\tilde{\mathbf{x}}_l, l = 1, \cdots, \tilde{p}\}$ and their corresponding optimal BIC values $\text{RBIC}_l; l = 1, \cdots, \tilde{p}$ is ordered decreasingly. Forward selection starts by considering a model which includes the predictor $\mathbf{x}_1$ with the largest optimal BIC value. To select the second predictor to be included in the model, for each $\tilde{\mathbf{x}}_i$, $i = 2, \cdots, \tilde{p}$, we fitted the model (5.4.1) with different number of components $1 \leq K \leq K'$ to the dataset $(\mathbf{y}_j, \tilde{\mathbf{x}}_i)$; $j = 1, \cdots, J$. Then we calculated the BIC values $\text{BIC}_{i,K}, i = 2, \cdots, \tilde{p}$ and we found the optimal BIC value corresponding to each predictor such that $\hat{\text{BIC}}_i = \min_{1 \leq K \leq K'} (\text{BIC}_{i,K})$. Therefore we obtained a set of optimal BIC values $\hat{\text{BIC}}_i, i = 2, \cdots, \tilde{p}$. Then the predictor which had the smallest $\hat{\text{BIC}}$ was selected to be added to the model. This process was repeated until the number of the predictors in the model reaches 40. Since the sample size is 42, we fixed the number of predictors to 40 because otherwise $\mathbf{X}^T\mathbf{X}$ was not invertible for some of simulation settings. Then we moved on to the second stage, which was the classification stage, and fitted the full model (5.2.1) to $\mathbf{y}_1, \cdots, \mathbf{y}_J$ and the predictors selected in the forward selection stage. Once the likelihood converged as explained in the very end of Section 5.3.1 then the EM algorithm stopped iterating and the optimal classification was obtained. Then we calculated the adjusted Rand index for the optimal classification. We also applied the mixture model to classify the response variable and calculate the corresponding adjusted Rand index. The result of these classifications are presented in the following graphs.

Graphs in Figure 5.7.10 and 5.7.11, show the resulted adjusted Rand index corresponding to each classification method. The results were obtained under settings with $p = 500$, $a = 3, 8$ and $n = 42, 84$. In settings with small sample size where $n = 42$, forward selection was performed after screening stage. For example, in results shown in Figure 5.7.10, since the number of selected predictors after screening was larger than 42, forward selection was applied to reduce the dimension of predictors to 40. Then in the classification stage, the LFM was fitted to these 40 predictors selected with forward selection. In settings with sample size $n = 84$, since the number of selected predictors in the screening stage do not exceed the sample size, no forward selection was applied. In this case after screening, in the classification stage, the LFM model was fitted to the reduced set of predictors. In results shown in Figure 5.7.11, in settings with $n = 42$, forward selection was applied after screening stage. Then LFM was fitted to the 40 predictors that were selected by forward selection.

According to the adjusted Rand index shown in Figure 5.7.10 and 5.7.11, LFM outperforms Mclust in all scenarios. This means that the classification obtained by applying likelihood fusion model results in a higher accuracy than that of normal mixture model. The reason is that through likelihood fusion we take the covariates into account in the classification process and this substantially improves the classification accuracy. Unlike what we observed in screening procedure, increasing the number of active covariates results in a more accurate classification. This is because by having more non-zero covariates, more information is contributed to the classification process which leads to a more accurate classification.

Effect of increasing active size on clustering performance

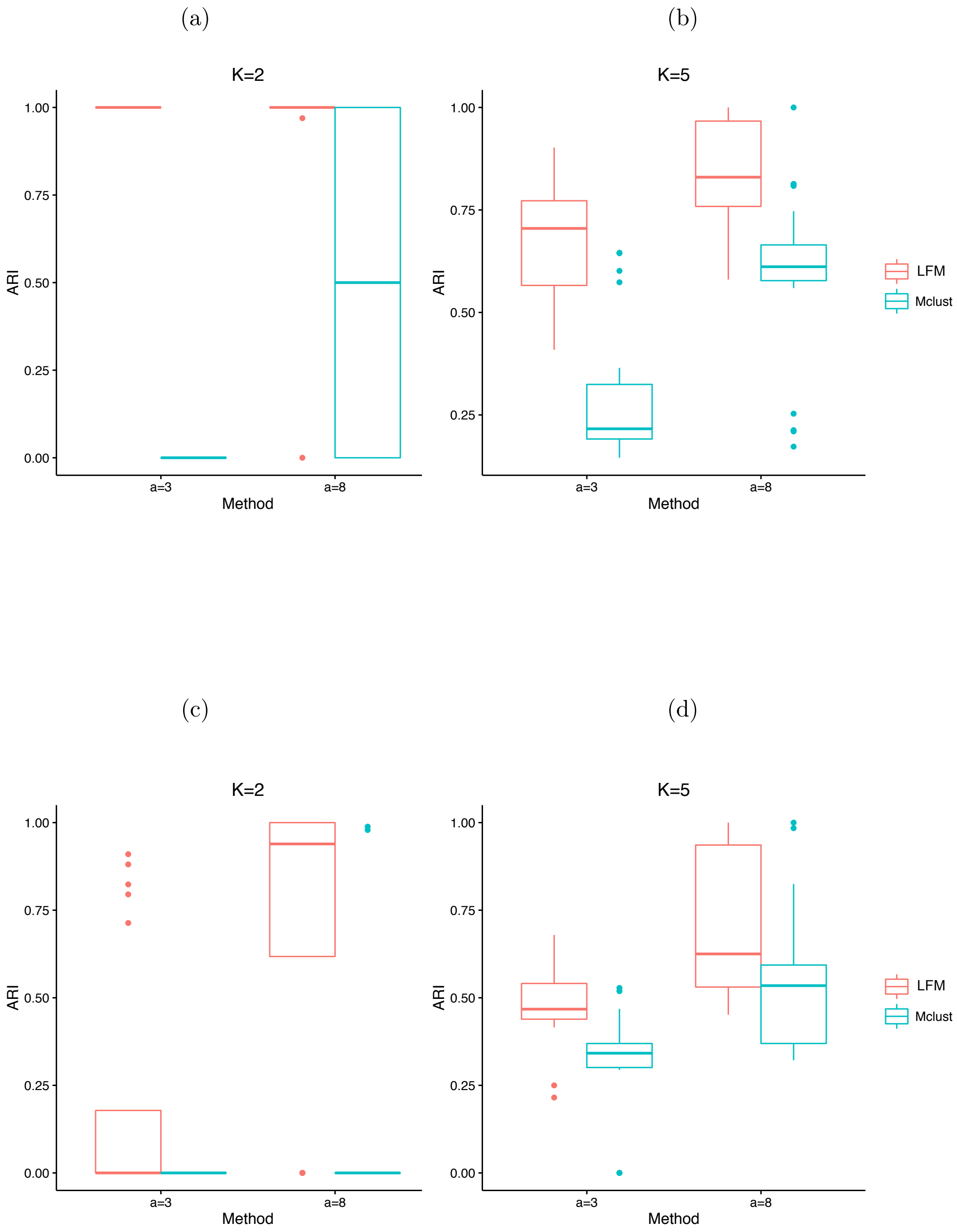


Figure 5.7.10: Box plots of the Adjusted Rand index obtained from 30 simulations. (a) and (b) under scenario1 with non-equal component variances. (c) and (d) under scenario 2 with equal variances for all components. The graphs compare the Adjusted Rand index of clustering via likelihood fusion model (LFM) and clustering via mixture model (Mclust) where $p = 500, a = 8, 3$ with sample of size $n = 42$. Here, forward selection was applied after screening stage then LFM was fitted to the set of selected predictors by forward selection.

Effect of increasing the sample size on clustering performance

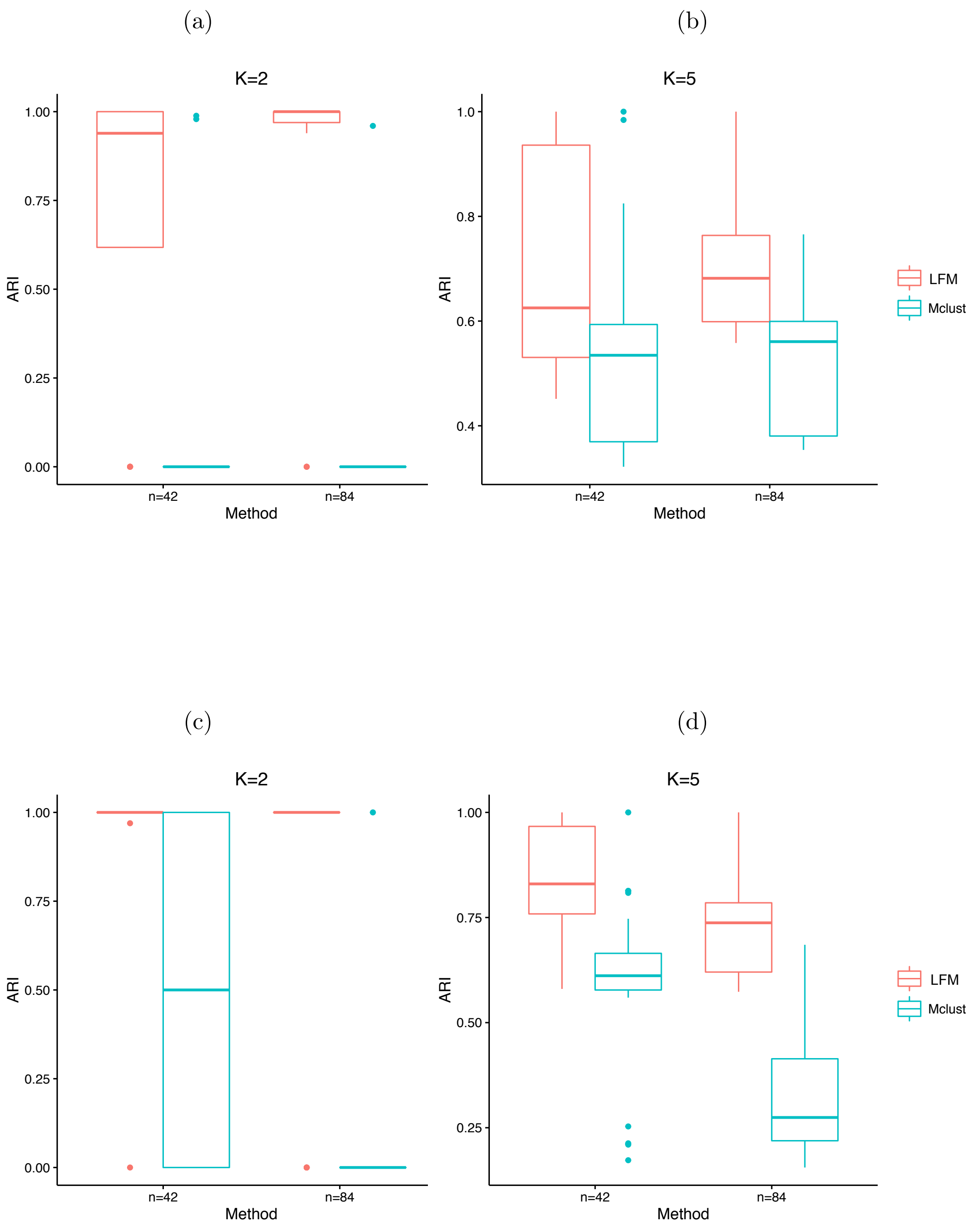


Figure 5.7.11: Box plots of the Adjusted Rand index obtained from 30 simulations. (a) and (b) under scenario1 with non-equal component variances. (c) and (d) under scenario 2 with equal variances for all components. The graphs compare the Adjusted Rand index of clustering via likelihood fusion model (LFM) and clustering via mixture model (Mclust) where $p = 500, a = 8$ with sample of size $n = 42, 84$. In settings with $n = 42$ forward selection was applied after screening stage then LFM was fitted to the set of selected predictors by forward selection.

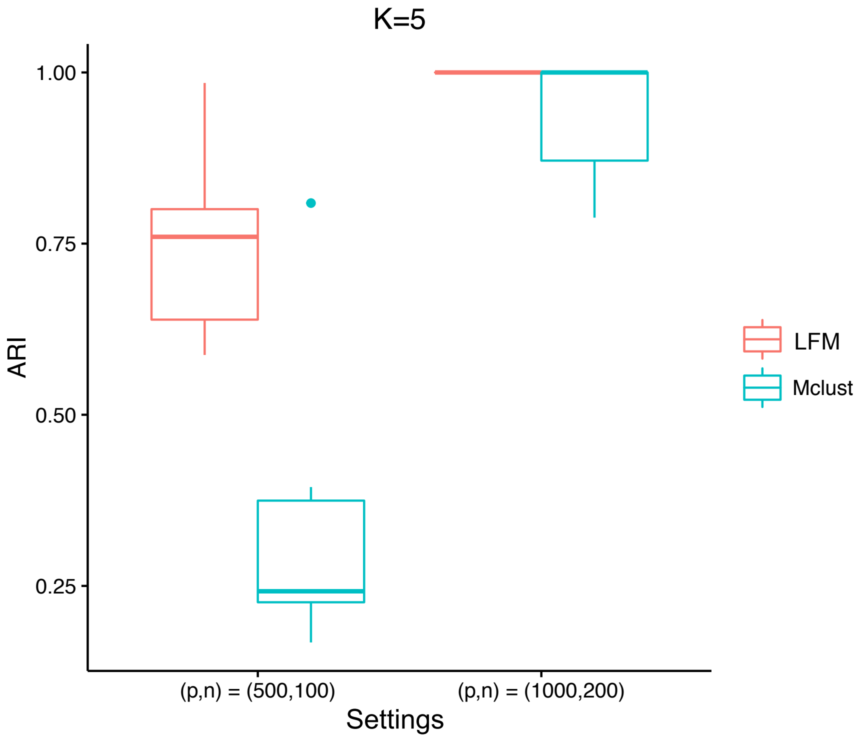


Figure 5.7.12: Adjusted Rand index for classifications obtained through applying likelihood fusion model (LFM) and Mclust for 20 simulations. There are $a = 8$ active covariates in the model. Simulations were run under scenario 1 and without forward selection.

The above results show that having enough sample size can improve the performance significantly even when the number of predictors is high. In these cases that the sample size is enough there is no need to do forward selection and the classification stage can be performed after screening.

5.8 Real data application

In this section, our two-phase proposed approach introduced in the previous sections was applied to the real data. As mentioned before the cancer drug data contain a high dimensional design matrix $\mathbf{X}_{42\times 13321}$ formed by gene expression levels across 42 cell lines. These data also include the IC50 values of 131 drugs across the same cell lines which form 42 observations on 131 response variables denoted by $\mathbf{Y}_{42\times131}$. Let vectors $\mathbf{y}_j, j = 1, \cdots, 131$ indicate the columns of matrix $\mathbf{Y}$ and vectors $\mathbf{x}_t, t = 1, \cdots, 13321$ denote the columns of $\mathbf{X}$. Prior to applying our method on real data, gene expressions were log-transformed and normalised and also log-transformed IC50 values were centralised by column mean. Since $42 \ll 13321$ we face the issue of high dimensionality in the analysis of these data. To resolve this issue we implemented the marginal likelihood fusion model to screen the predictors (which are gene expression levels) and reduced the dimension. To this aim, for each $\mathbf{x}_t$ we fitted the marginal likelihood fusion model (5.4.1) to the data $(\mathbf{y}_j, \mathbf{x}_t), j = 1, \cdots, 131$ with different number of components $1 \le K \le 10$. Then we found the minimum BIC value for each covariate t, i.e. $\hat{\text{BIC}}_{t,\hat{K}} = \text{BIC}_{t_{\min}} = \min_{1\le K\le 10}(\text{BIC}_{t,K})$.

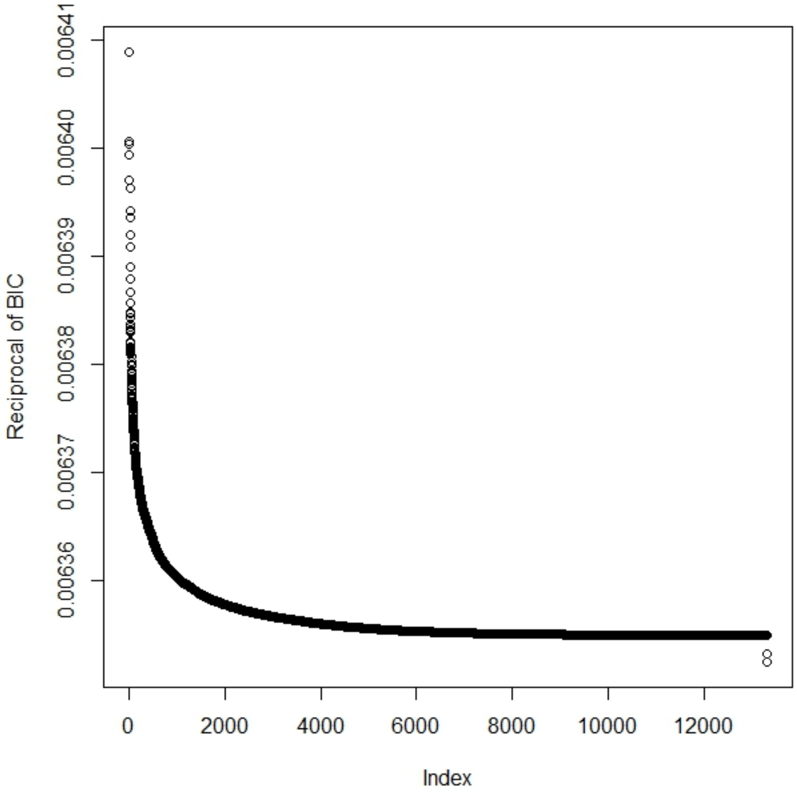
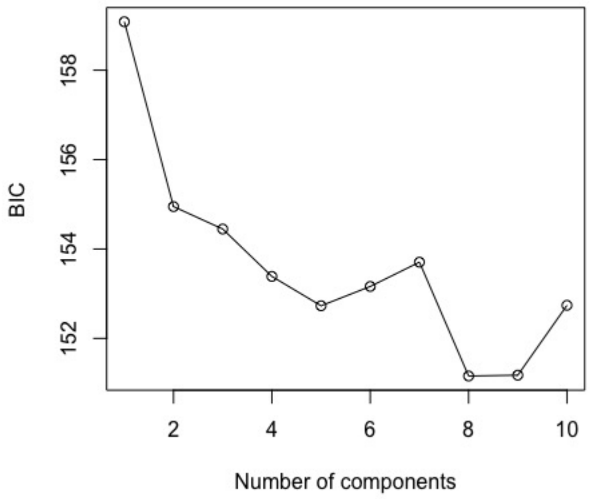


Figure 5.8.1: Reciprocal of optimal BIC values corresponding to all genes in real data. The result obtained after applying the marginal likelihood fusion to the real data.

After finding the optimal BIC values $\text{BIC}_{t_{\min}}$ corresponding to all covariates $t = 1, \cdots, 13321$, we then calculated the reciprocal of BIC values $\text{RBIC}_t = 1/\hat{\text{BIC}}_{t,\hat{K}}$; $t = 1, \cdots, 13321$ these values are plotted in Figure 5.8.1. Then we clustered these values into two groups using k-means classification. The predictors corresponding to the group with higher mean were regarded as important predictors. As a result of k-means classification, a set of size 2179 was selected as signal set. This screening was performed on CPU with Intel Core i5-3470 processor and 8 GB RAM. Time taken this screening to run on real data was 13.65 hours.

Since the sample size is very small we had to reduce these selected covariates before moving to the classification stage. Thus we selected 40 out of the 2179 selected covariates by forward selection as explained in Section 5.7.2.2. Then in the classification stage we fitted the model (5.2.1) to the data using 40 selected covariates after forward selection. We varied the number of components and fitted the likelihood fusion model for $1 \leq K \leq 10$. The following plot shows BIC values corresponding to different number of components.

Figure 5.8.2: BIC values for different components



According to BIC values and comparing $\sigma_1, \cdots, \sigma_K$ it is reasonable to classify the drugs $\mathbf{y}_1, \cdots, \mathbf{y}_{131}$ into 5 groups. The mixing proportions and the variances of components corresponding to this classification are $\boldsymbol{\pi} = (0.3, 0.16, 0.30, 0.17, 0.06)$ and $\boldsymbol{\sigma^2} = (0.96, 1.54, 2.84, 1.49, 4.31)$.

In comparison to the screening performed by SNR where 1316 genes were selected, there are 230 genes in common out of 2179 genes that have been selected by likelihood fusion. Computational time for screening performed by SNR is 3.6 seconds which is much faster than screening by likelihood fusion which takes 13.65 hours. According to the simulation study in the previous section, the screening accuracy of likelihood fusion is much higher than SNR-based screening. There are two aspects to be considered in deciding on what approach we should take in the analysis of such data . From biological point of view, drugs often have a group structure according to the type of disease that they can cure. On the other hand, mixture models are known as powerful models to capture the heterogeneity in data and discover the hidden group structure. Therefore, it seems more reasonable to conclude that for these particular data, applying the likelihood fusion model to screen the gene expressions and to classify the drugs is an appropriate choice.

Chapter 6

Bayesian Inference of Finite Mixture Models

6.1 Introduction

As pointed out before, one of the prominent applications of finite mixture models is in cluster analysis where the aim is exploring groups in the data (Everitt et al., 2011). This application of finite mixture models has attracted much attention in both frequentists and Bayesian paradigm. The main advantage of the Bayesian over the frequentist approach is the possibility of putting priors on the component parameters. Frühwirth-Schnatter (2006) states some of the advantages that the Bayesian approach for clustering possesses. The parameter estimation in fitting mixture model under frequentist framework is carried out via EM algorithm which might lead to degenerate solutions. This is unlikely in the Bayesian estimation since some prior information is set on the variance of components. Moreover, there exist a more principled way of posterior classification of the objects into clusters under the Bayesian paradigm. Also cluster analysis under Bayesian framework could be performed through finding the marginal posterior distribution of the hidden allocation vector without estimating the component parameters. In Bayesian clustering without parameter estimation, an observation is allocated to the component with the highest posterior probability. This inference can be performed under two assumptions: when the number of components is known and when the number of components is unknown. Here, we focus on the former assumption and perform the analysis when the number of components is assumed to be known. We aim to apply Bayesian inference to classify observations through mixture models. Hence, we utilize Bayesian finite mixture models with normal components to obtain the posterior distribution of the allocation vector. We calculate this posterior by applying two different type of priors on component parameters.

6.2 Bayesian mixture models

Suppose that we have N observations $y_1, \cdots, y_n$, on a univariate random variable Y which comes from a population with K groups. Let the vector $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K, \boldsymbol{\eta})$ denote parameters of the mixture model then the mixture density function is given by

$$p(y_i|\boldsymbol{\vartheta}) = \sum_{k=1}^{K} \eta_k p(y_i|\boldsymbol{\theta}_k),$$

where $p(y_i|\boldsymbol{\theta}_k)$ is the component density with parameters $\boldsymbol{\theta}_k$. Fitting the above mixture model is equivalent to estimating the parameters of the model. Under a Bayesian framework, all the information about these unknown parameters contained in the data $\mathbf{y} = (y_1, \cdots, y_N)$ is extracted through finding the posterior density $p(\boldsymbol{\vartheta}|\mathbf{y})$. Suppose the prior $p(\boldsymbol{\vartheta})$ is known. According to the Bayes' theorem the posterior density is defined as

$$p(\boldsymbol{\vartheta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\vartheta})p(\boldsymbol{\vartheta}).$$

We discussed in section 2.6 that the allocation vector or group indicator vector $\mathbf{S}$ is the missing part of the data. Therefore, after completing data, the complete-data likelihood is combined with a prior density $p(\boldsymbol{\vartheta})$ to obtain the mixture posterior density $p(\boldsymbol{\vartheta}|\mathbf{y})$. So the first step to be taken in Bayesian framework for analysing mixture model is putting prior on the unknown parameters of the model, i.e. $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K, \boldsymbol{\eta})$. Choosing priors is a very important task in Bayesian inference of mixture models.

6.2.1 Prior specification for model parameters

The priors for parameters can be chosen in two different manners. If the prior is chosen such that it brings some background information into the problem, then the prior is called subjective or informative. On the other hand, if the prior is vague or flat and have almost no impact on the posterior distribution, it is called noninformative or objective (Press, 2009). A prior is called improper if it is not integrable over the parameter space. Usually improper priors are used where noninformative priors are desired and they lead to proper posteriors. However, in a mixture context objective or noninformative priors are not suitable choices for mixture models as they lead to improper posteriors (Roeder and Wasserman, 1997);(Richardson and Green, 1997). Besides bringing additional information to the analysis, subjective priors are proper and generally well behaved analytically (Press, 2009). Subjective priors for mixture modes are often obtained by choosing conjugate priors to the complete data likelihood. These subjective priors are highly dependent on the parameters of the priors or hyperparameters. Therefore, to control the influence of these hyperparameters on the analysis it is usual to set priors on these hyperparameters. In section 2.6.1 we introduced the hierarchical priors for parameters of component densities. We now introduce the priors for the allocation vector and the mixing proportions.

The allocation vector $\mathbf{S}$ is multinomially distributed with probability $\boldsymbol{\eta}$. Therefore the proper prior for the mixing proportions is the Dirichlet distribution which is defined by

$$\text{Dir}(x_1, \cdots, x_K; \alpha_1, \cdots, \alpha_K) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_{i-1}},$$

where $x_i \in (0,1)$ and $\sum_{i=1}^{K} x_i = 1$. The Dirichlet distribution is the conjugate prior for multinomial data (Richardson and Green, 1997). A Dirichlet prior for the weights with the concentration parameter e_0, is of the form

$$\begin{aligned} p(\boldsymbol{\eta}|e_0) &= \frac{\Gamma(\sum_{k=1}^{K} e_0)}{\prod_{k=1}^{K}\Gamma(e_0)} \prod_{k=1}^{K} \eta_k^{e_0-1} \\ &= \frac{\Gamma(Ke_0)}{\Gamma(e_0)^K} \prod_{k=1}^{K} \eta_k^{e_0-1}. \end{aligned} \tag{6.2.1}$$

Now by integrating out the mixing proportions, $\boldsymbol{\eta}$, the prior on allocation variables is given by

$$p(\mathbf{S}) = \int p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\eta}|e_0)d\boldsymbol{\eta},$$

where the distribution of allocations is multinomial and is given by

$$p(\mathbf{S}|\boldsymbol{\eta}) = \prod_{k=1}^{K} \eta_k^{N_k(S)},$$

where $N_k(S)$ is the number of elements in component k. Hence, the prior for allocations is obtained as follows

$$\begin{aligned} p(\mathbf{S}) &= \int \frac{\Gamma(\sum_{k=1}^{K} e_0)}{\prod_{k=1}^{K}\Gamma(e_0)} \prod_{k=1}^{K} \eta_k^{e_0-1} \prod_{k=1}^{K} \eta_k^{N_k(S)} d\eta_k \\ &= \frac{\Gamma(Ke_0)}{\Gamma(e_0)^K} \int \prod_{k=1}^{K} \eta_k^{N_k(S)+e_0-1} d\eta_k, \end{aligned}$$

which according to definition of Dirichlet distribution can be rewritten as

$$\begin{aligned} p(\mathbf{S}) &= \frac{\Gamma(Ke_0)}{\Gamma(e_0)^K} \frac{\prod_{k=1}^{K}\Gamma(N_k(S)+e_0)}{\Gamma(\sum\limits_{k=1}^{K} N_k(S)+e_0)} \\ &= \frac{\Gamma(Ke_0)\prod_{k=1}^{K}\Gamma(N_k(S)+e_0)}{\Gamma(Ke_0+N)\Gamma^K(e_0)}. \end{aligned}$$

6.2.2 Bayesian clustering with mixture of Normals

Similar to any clustering technique, the main purpose in Bayesian clustering is finding an optimal classification of data or the optimal allocation vector $\mathbf{S}$. One of the pioneering works in the field of Bayesian clustering is the work done by Binder (1978). In this work the Bayesian clustering is formulated under a bayesian decision theoretic framework and is based on a loss function $R(\mathbf{S}, \hat{\mathbf{S}})$. This function measures the amount of loss which is caused by choosing the allocation vector $\hat{\mathbf{S}}$ over the true classification $\mathbf{S}$. Binder (1978) believes that finding the best allocation vector $\hat{\mathbf{S}}$ by maximizing the posterior does not take into account how different this allocation is from the true one.

Our interest in this part is to take an alternative way and implement mixture of normals to classify observations by finding the optimal allocation vector $\hat{\mathbf{S}}$ with the highest posterior. This clustering is preformed without estimating the component parameters and is just based on the marginal posterior of the allocation vector $p(\mathbf{S}|\mathbf{y})$. Therefore, the aim is finding the optimal allocation vector $\hat{\mathbf{S}}$ for data by maximising this posterior distribution. In order to evaluate how different $\hat{\mathbf{S}}$ is from the true allocation vector we measure the adjusted Rand index Rand (1971) corresponding to $\hat{\mathbf{S}}$. We consider two different set of hierarchical priors and calculate this posterior. In inference with mixture of normals it is common to choose priors for the component parameters which are independent of the mixing proportions $\boldsymbol{\theta}_k$. The posterior corresponding to such priors is calculated in Section 6.2.2.1. Alternatively, in Section 6.2.2.2, we propose a different prior for the mean of each component μ_k which is not independent of the weight η_k.

6.2.2.1 Mixing-proportion independent priors for components

We consider the mixture model (6.2.1) where $p(y_i|\boldsymbol{\theta}_k)$ is a normal density with the corresponding parameters $\boldsymbol{\theta}_k = (\mu_k, \sigma_k^2)$. Diebolt and Robert (1994) set the following hierarchical prior distributions on parameters.

$$\begin{aligned} \mu_k|\sigma_k^2 &\sim N(\mu_{k0}, \sigma_k^2/N_0), \\ \sigma_k^2 &\sim IG(a_0, b_0). \end{aligned}$$

We also remind that the allocation vector corresponding to observations $\mathbf{y} = (y_1, \cdots, y_N)$ is denoted by $\mathbf{S} = (S_1, \cdots, S_N)$. Our aim is finding the optimal allocation that gives the best classification. The optimal allocation vector is obtained by finding the vector with the highest posterior distribution.

Let $(\mathbf{y}, \mathbf{S})$ denote the complete data and $\boldsymbol{\theta} = (\mu_1, \cdots, \mu_k, \sigma_1^2, \cdots, \sigma_k^2)$. Then, the joint distribution of data and parameters is expressed by the following factorization:

$$p(\mathbf{y}, \mathbf{S}, \boldsymbol{\theta}, \boldsymbol{\eta}) = p(\mathbf{y}|\mathbf{S}, \boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta}) \tag{6.2.2}$$

As pointed out earlier, the Bayesian clustering is based on the posterior distribution $p(\mathbf{S}|\mathbf{y})$ of the allocation vector. In order to find the marginal distribution of allocations, i.e. $p(\mathbf{S}|\mathbf{y})$ from expression (6.2.2), we integrate out all other parameters in the model denoted by vector $\boldsymbol{\vartheta}$. Therefore, we need to calculate the following integration:

$$\begin{aligned} p(\mathbf{S}|\mathbf{y}) &= \iint p(\mathbf{y}|\boldsymbol{\eta}, \mathbf{S}, \boldsymbol{\theta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\eta})p(\boldsymbol{\theta})d\boldsymbol{\theta}d\boldsymbol{\eta} \\ &= \int \left(\int p(\mathbf{y}|\boldsymbol{\eta}, \mathbf{S}, \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} \right) p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\eta})d\boldsymbol{\eta} \\ &= \int \left(\prod_{k=1}^{K} \iint \prod_{i:S_i=k} (p(y_i|\mu_k, \sigma_k)p(\mu_k, \sigma_k)d\mu_k d\sigma_k \right) p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\eta})d\boldsymbol{\eta} \end{aligned}$$

Since $p(y_i|S_i = k, \boldsymbol{\vartheta}) = p(y_i|, \boldsymbol{\theta}_k)$ and $Pr(S_i = k|\boldsymbol{\vartheta}) = \eta_k$ the complete-data likelihood function is given by

$$p(\mathbf{y}|\mathbf{S}, \boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\eta}) = \prod_{i=1}^{N} \prod_{k=1}^{K} \left(p(y_i|\mu_k, \sigma_k^2)\eta_k \right)^{I_{S_i=k}}$$

Therefore

$$p(\mathbf{y}|\mathbf{S},\boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})$$

$$= \prod_{i=1}^{N}\prod_{k=1}^{K}\left(p(\mathbf{y}_i|\mu_k,\sigma_k^2)\eta_k\right)^{I_{S_i=k}}\prod_{k=1}^{K}p(\mu_k|\sigma_k^2)p(\sigma_k^2)p(\eta_k)$$

$$= \prod_{k=1}^{K}\left(\prod_{i:S_i=k}\frac{1}{\sqrt{2\pi\sigma_k^2}}\exp\left\{-\frac{\sum\limits_{i:S_i=k}(y_i-\mu_k)^2}{2\sigma_k^2}\right\}\right)\left(\prod_{k=1}^{K}\eta_k^{\sum\limits_{i=1}^{N}I_{S_i=k}}\right)$$

$$\times\prod_{k=1}^{K}\left(\frac{N_0}{2\pi\sigma_k^2}\right)^{1/2}\exp\left\{-\frac{N_0}{2\sigma_k^2}(\mu_k-\mu_0)^2\right\}$$

$$\times\prod_{k=1}^{K}\frac{b_0^{a_0}}{\Gamma(a_0)}(\sigma_k^2)^{-a_0-1}\exp\left\{-b_0/\sigma_k^2\right\}$$

$$\times\frac{\Gamma(\sum_{k=1}^{K}e_0)}{\prod_{k=1}^{K}\Gamma(e_0)}\prod_{k=1}^{K}\eta_k^{e_0-1}$$

$$= \left(\frac{1}{2\pi}\right)^{\frac{\sum\limits_{k=1}^{K}N_k(S)}{2}}\left(\frac{N_0}{2\pi}\right)^{K/2}\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^{K}\frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}$$

$$\times\prod_{k=1}^{K}\exp\left\{-\frac{\sum\limits_{i:S_i=k}(y_i-\mu_k)^2+N_0(\mu_k-\mu_0)^2+2b_0}{2\sigma_k^2}\right\}$$

$$\times\prod_{k=1}^{K}\eta_k^{e_0+N_k(S)-1}\prod_{k=1}^{K}\frac{1}{\sigma_k^{2(a_0+1)+N_k(S)+1}}$$

The expression $\sum\limits_{i:S_i=k}(y_i-\mu_k)^2$ in the above equations can be rearranged as follows

$$\begin{aligned}\sum_{i:S_i=k}(y_i-\mu_k)^2 &= \sum_{i:S_i=k}[(y_i-\bar{y})^2+(\bar{y}-\mu_k)^2]\\ &= \sum_{i:S_i=k}(y_i-\bar{y})^2+2(\bar{y}-\mu_k)\sum_{i:S_i=k}y_i-\bar{y}+\sum_{i:S_i=k}(\bar{y}-\mu_k)^2,\end{aligned}$$

substituting the variance $S^2_{y,k} = \frac{1}{N_k(\mathbf{S})}\sum_{i:S_i=k}(y_i - \bar{y}_k(\mathbf{S}))^2$ in the above expression we get

$$\sum_{i:S_i=k}(y_i - \mu_k)^2 = N_k(S)S^2_{y,k} + N_k(S)(\bar{y} - \mu_k)^2,$$

hence

$$p(\mathbf{y}|\mathbf{S},\boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})$$

$$= \left(\frac{1}{2\pi}\right)^{\frac{\sum\limits_{k=1}^{K} N_k(S)}{2}} \left(\frac{N_0}{2\pi}\right)^{K/2} \left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^K \frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}$$

$$\times \prod_{k=1}^{K} \exp\left\{-\frac{N_k(\mathbf{S})S^2_{y,k}(\mathbf{S}) + N_k(\mathbf{S})(\bar{y}_k(\mathbf{S}) - \mu_k)^2 + N_0(\mu_k - \mu_{k0})^2 + 2b_0}{2\sigma_k^2}\right\}$$

$$\times \prod_{k=1}^{K} \eta_k^{e_0+N_k(S)-1} \prod_{k=1}^{K} \frac{1}{\sigma_k^{2(a_0+1)+N_k(S)+1}}$$

$$= \left(\frac{1}{2\pi}\right)^{N/2} \left(\frac{N_0}{2\pi}\right)^{K/2} \left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^K \frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}$$

$$\times \prod_{k=1}^{K} \exp\left\{-\frac{(N_k(\mathbf{S}) + N_0)\left[\mu_k - \frac{N_k(\mathbf{S})\bar{y}_k(\mathbf{S}) + N_0\mu_{k0}}{N_k(\mathbf{S}) + N_0}\right]^2}{2\sigma_k^2}\right\}$$

$$\times \prod_{k=1}^{K} \exp\left\{-\frac{N_k(\mathbf{S})S^2_{y,k}(\mathbf{S}) + 2b_0 + \frac{N_k(\mathbf{S})N_0}{N_k(\mathbf{S})+N_0}(\bar{y}(\mathbf{S}) - \mu_{k0})^2}{2\sigma_k^2}\right\}$$

$$\times \prod_{k=1}^{K} \eta_k^{e_0+N_k(S)-1} \prod_{k=1}^{K} \frac{1}{\sigma_k^{2(a_0+1)+N_k(S)+1}}$$

Now we have to integrate out all the parameters in the model to obtain $p(\mathbf{S}|\mathbf{y})$. We start from integrating out the parameter μ_k by calculating the integral

$$\prod_{k=1}^{K} \int \exp\left\{-\frac{(N_k(\mathbf{S}) + N_0)\left[\mu_k - \frac{N_k(\mathbf{S})\bar{y}_k(\mathbf{S}) + N_0\mu_{k0}}{N_k(\mathbf{S}) + N_0}\right]^2}{2\sigma_k^2}\right\} d\mu_k,$$

which is the kernel of a normal density. Hence

$$\int p(\mathbf{y}, \mathbf{S}|\boldsymbol{\vartheta}) p(\mathbf{S}|\boldsymbol{\eta}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) d\boldsymbol{\mu}$$

$$= \left(\frac{1}{2\pi}\right)^{N/2} \left(\frac{N_0}{2\pi}\right)^{K/2} \left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^K \frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}$$

$$\times \prod_{k=1}^{K} \exp\left\{-\frac{N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}) + 2b_0 + \frac{N_k(\mathbf{S})N_0}{N_k(\mathbf{S})+N_0}(\bar{y}(\mathbf{S}) - \mu_{k0})^2}{2\sigma_k^2}\right\}$$

$$\times \prod_{k=1}^{K} \eta_k^{e_0+N_k(S)-1} \prod_{k=1}^{K} \frac{1}{\sigma_k^{2(a_0+1)+N_k(S)+1}} \prod_{k=1}^{K} \sqrt{\frac{2\pi\sigma_k^2}{N_k(\mathbf{S}) + N_0}}$$

Integrating out σ_k^2 and η_k reads

$$\begin{aligned} p(\mathbf{S}|\mathbf{y}) &= \left(\frac{1}{2\pi}\right)^{N/2} N_0^{K/2} \left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^K \frac{\Gamma(Ke_0)}{\Gamma(e_0)^K} \prod_{k=1}^{K} \sqrt{\frac{1}{N_k(\mathbf{S}) + N_0}} \\ &\times \int_0^1 \prod_{k=1}^{K} \eta_k^{N_k(\mathbf{S})+e_0-1} d\eta \\ &\times \int \prod_{k=1}^{K} \exp\left\{-\frac{N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}) + 2b_0 + \frac{N_k(\mathbf{S})N_0}{N_k(\mathbf{S})+N_0}(\bar{y}(\mathbf{S}) - \mu_{k0})^2}{2\sigma_k^2}\right\} \\ &\times \prod_{k=1}^{K} \sigma_k^2 (\frac{1}{\sigma_k^2})^{\frac{N_k(\mathbf{S})+3}{2}+a_0} d\sigma_k^2 \end{aligned}$$

In the above equation, the first integral is the kernel of the Dirishlet distribution expressed in Equation (6.2.1). Therefore,

$$\int_0^1 \prod_{k=1}^{K} \eta_k^{N_k(\mathbf{S})+e_0-1} d\eta = \left(\frac{\Gamma(\sum_{k=1}^{K} N_k(S) + e_0)}{\prod_{k=1}^{K} \Gamma(N_k(S) + e_0)}\right)^{-1}. \qquad (6.2.3)$$

Now if we use the following notation

$$\mathcal{B} = N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}) + 2b_0 + \frac{N_k(\mathbf{S})N_0}{N_k(\mathbf{S}) + N_0}(\bar{y}(\mathbf{S}) - \mu_{k0})^2,$$

then the second integral in Equation (6.2.3) can be reformed as

$$\prod_{k=1}^{K}\int \exp\left\{-\frac{\mathcal{B}}{\sigma_k^2}\right\}(\sigma_k^2)^{-(a_0+\frac{N_k(\mathbf{S})}{2})-1}d\sigma_k^2,$$

the above integral is the kernel of Inverse Gamma density with parameters $(a_0+\frac{N_k(\mathbf{S})}{2}, \frac{1}{2}\mathcal{B})$ therefore,

$$\begin{aligned} p(\mathbf{S}|\mathbf{y}) &= \int\left(\prod_{k=1}^{K}\iint\prod_{i:S_i=k}(p(y_i|\mu_k,\sigma_k)p(\mu_k,\sigma_k)d\mu_k d\sigma_k\right)p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\eta})d\boldsymbol{\eta} \\ &= 2^{Ka_0}N_0^{K/2}\left(\frac{1}{\pi}\right)^{N/2}\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^{K}\frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}\prod_{k=1}^{K}\sqrt{\frac{1}{N_k(\mathbf{S})+N_0}} \\ &\times \frac{\prod_{k=1}^{K}\Gamma(N_k(S)+e_0)}{\Gamma(Ke_0+N)}\prod_{k=1}^{K}\mathcal{B}^{-(a_0+\frac{N_k(\mathbf{S})}{2})}\prod_{k=1}^{K}\Gamma(a_0+\frac{N_k(\mathbf{S})}{2}) \end{aligned} \tag{6.2.4}$$

The logarithm of the allocation posterior obtained in (6.2.4) is of the form

$$\begin{aligned} \log(p(\mathbf{S}|\mathbf{y})) &= K/2\log(N_0) - N/2\log(\pi) + Ka_0\log(b_0) - K\log\Gamma(a_0) \\ &+ \log\Gamma(Ke_0) - K\log\Gamma(e_0) + Ka_0\log(2) - \Gamma(N+Ke_0) \\ &+ \sum_{k=1}^{K}\log\Gamma(N_k(\mathbf{S})+e_0)\sum_{k=1}^{K}\log\Gamma(a_0+\frac{N_k(\mathbf{S})}{2}) \\ &- \sum_{k=1}^{K}\frac{1}{2}\log(N_k(\mathbf{S})+N_0) - \sum_{k=1}^{K}(a_0+N_k(\mathbf{S})/2)\log\mathcal{B} \end{aligned} \tag{6.2.5}$$

6.2.2.2 Mixing-proportion dependent priors for components

We consider the mixture model where $p(y_i|\boldsymbol{\theta}_k)$ is a normal density with the corresponding parameters $\boldsymbol{\theta}_k = (\mu_k, \sigma_k^2)$. Here, we consider impose some dependency between the parameters. Therefore we set a prior for the mean of each component μ_k which depends on the weight η_k. To this aim, we propose the following prior distributions on parameters.

$$\begin{aligned}
\mu_k|\sigma_k^2, \eta_k &\sim N(\mu_{k0}, \frac{\sigma_k^2}{N_0\eta_k}), \\
\sigma_k^2 &\sim IG(a_0, b_0). \\
\eta_k &\sim D(e_0, \cdots, e_0)
\end{aligned}$$

According to the above hierarchical model, the joint distribution of the data and the model parameters is formulated as

$$\begin{aligned}
& p(\mathbf{y}|\mathbf{S}, \boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta}) \\
& = \prod_{i=1}^{N}\prod_{k=1}^{K}\left(p(\mathbf{y}_i|\mu_k, \sigma_k^2)\eta_k\right)^{I_{S_i=k}}\prod_{k=1}^{K}p(\mu_k|\sigma_k^2, \eta_k)p(\sigma_k^2)p(\eta_k) \\
& = \prod_{k=1}^{K}\left(\prod_{i:S_i=k}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{\sum\limits_{i:S_i=k}(y_i-\mu_k)^2}{2\sigma_k^2}\right\}\right)\left(\prod_{k=1}^{K}\eta_k^{\sum\limits_{i=1}^{N}I_{S_i=k}}\right) \\
& \times\prod_{k=1}^{K}\left(\frac{N_0\eta_k}{2\pi\sigma_k^2}\right)^{1/2}\exp\left\{-\frac{N_0\eta_k}{2\sigma_k^2}(\mu_k-\mu_0)^2\right\} \\
& \times\prod_{k=1}^{K}\frac{b_0^{a_0}}{\Gamma(a_0)}(\sigma_k^2)^{-a_0-1}\exp\left\{-b_0/\sigma_k^2\right\} \\
& \times\frac{\Gamma(\sum_{k=1}^{K}e_0)}{\prod_{k=1}^{K}\Gamma(e_0)}\prod_{k=1}^{K}\eta_k^{e_0-1}.
\end{aligned}$$

Therefore,

$$
\begin{aligned}
&p(\mathbf{y}|\mathbf{S},\boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})\\
&=\left(\frac{1}{2\pi}\right)^{\frac{\sum\limits_{k=1}^{K} N_k(S)}{2}}\left(\frac{N_0}{2\pi}\right)^{K/2}\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^{K}\frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}\\
&\times\prod_{k=1}^{K}\exp\left\{-\frac{\sum\limits_{i:S_i=k}(y_i-\mu_k)^2+N_0\eta_k(\mu_k-\mu_0)^2+2b_0}{2\sigma_k^2}\right\}\\
&\times\prod_{k=1}^{K}\eta_k^{e_0+N_k(S)-1/2}\prod_{k=1}^{K}\frac{1}{\sigma_k^{2(a_0+1)+N_k(S)+1}}
\end{aligned}
$$

After doing some simple algebra we get

$$
\begin{aligned}
&p(\mathbf{y}|\mathbf{S},\boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})\\
&=\left(\frac{1}{2\pi}\right)^{N/2}\left(\frac{N_0}{2\pi}\right)^{K/2}\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^{K}\frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}\\
&\times\prod_{k=1}^{K}\exp\left\{-\frac{(N_k(\mathbf{S})+N_0\eta_k)\left[\mu_k-\frac{N_k(\mathbf{S})\bar{y}_k(\mathbf{S})+N_0\eta_k\mu_{k0}}{N_k(\mathbf{S})+N_0\eta_k}\right]^2}{2\sigma_k^2}\right\}\\
&\times\prod_{k=1}^{K}\exp\left\{-\frac{N_k(\mathbf{S})S_{y,k}^2(\mathbf{S})+2b_0+\frac{N_k(\mathbf{S})N_0\eta_k}{N_k(\mathbf{S})+N_0\eta_k}(\bar{y}_k(\mathbf{S})-\mu_{k0})^2}{2\sigma_k^2}\right\}\\
&\times\prod_{k=1}^{K}\eta_k^{e_0+N_k(S)-1/2}\prod_{k=1}^{K}\frac{1}{\sigma_k^{2(a_0+1)+N_k(S)+1}}
\end{aligned}
$$

Now we are going to find the marginal posterior distribution of the allocation vector $p(\mathbf{S}|\mathbf{y})$ by integrating out all parameters. We first integrate out μ_k from the above expression and we get

$$
\begin{aligned}
&\prod_{k=1}^{K}\int p(\mathbf{y},\mathbf{S}|\boldsymbol{\vartheta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})d\mu_k \\
&= N_0^{K/2}\left(\frac{1}{2\pi}\right)^{N/2}\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^{K}\frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}\prod_{k=1}^{K}\sqrt{\frac{1}{N_k(\mathbf{S})+N_0\eta_k}} \\
&\times\prod_{k=1}^{K}\exp\left\{-\frac{N_k(\mathbf{S})S_{y,k}^2(\mathbf{S})+2b_0+\frac{N_k(\mathbf{S})N_0\eta_k}{N_k(\mathbf{S})+N_0\eta_k}(\bar{y}_k(\mathbf{S})-\mu_{k0})^2}{2\sigma_k^2}\right\} \\
&\times\prod_{k=1}^{K}\eta_k^{e_0+N_k(S)-1/2}\prod_{k=1}^{K}\frac{1}{\sigma_k^{2(a_0+1)+N_k(S)+2}}
\end{aligned}
$$

Finally integrating out σ_k and η_k, the posterior $p(\mathbf{S}|\mathbf{y})$ is obtained as

$$
\begin{aligned}
p(\mathbf{S}|\mathbf{y}) &= \int_0^1\iint p(\mathbf{y}|\boldsymbol{\eta},\mathbf{S},\boldsymbol{\theta})p(\mathbf{S}|\boldsymbol{\eta})p(\boldsymbol{\eta})p(\boldsymbol{\theta})d\boldsymbol{\theta}d\boldsymbol{\eta} \\
&= N_0^{K/2}\left(\frac{1}{\pi}\right)^{N/2}\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^{K}\frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}2^{Ka_0} \\
&\times \frac{\prod_{k=1}^{K}\Gamma(N_k(\mathbf{S})+e_0+1/2)}{\Gamma(N+Ke_0+K/2)}\prod_{k=1}^{K}\Gamma(N_k(\mathbf{S})/2+a_0) \\
&\times \prod_{k=1}^{K}\int_0^1\frac{\mathcal{B}(\eta_k)^{-a_0-\frac{N_k(\mathbf{S})}{2}}}{(N_k(\mathbf{S})+N_0\eta_k)^{1/2}}d\eta_k,
\end{aligned}
\tag{6.2.6}
$$

where

$$
\mathcal{B}(\eta_k) = N_k(\mathbf{S})S_{y,k}^2(\mathbf{S})+2b_0+\frac{N_k(\mathbf{S})N_0\eta_k}{N_k(\mathbf{S})+N_0\eta_k}(\bar{y}_k(\mathbf{S})-\mu_{k0})^2.
$$

As we can see for the case with dependent hierarchical priors there is no explicit form for the posterior $p(\mathbf{S}|\mathbf{y})$. Due to this formulation, we faced some challenges in calculating the integration in the expression (6.2.6). Calculating this integration is not always possible in a usual way as a result of overflow or underflow, depending on simulation settings. To address this issue we calculate this definite integral by calculating Reimann sums over a partition of $[0, 1]$.

Suppose P partitions $[a,b]$ into equal length subintervals $[x_{i-1}, x_i]; i = 1, \cdots, n$. For a continuous and integrable function f on interval $[a,b]$ the definite integral of f from a to b can be computed by

$$\int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i^*)\Delta x_i$$

for any $x_i^* \in [x_{i-1}, x_i]$ with $\Delta x_i = \frac{b-a}{n}$ and $x_i^* = a + i\Delta x_i$; $i = 1, \cdots, n$ (McGregor et al., 2010). Now we aim to calculate the following integration using the above Reimann sums. We have

$$\int_0^1 f(\eta_k)d\eta_k =$$
$$\int_0^1 \frac{(N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}) + 2b_0 + \frac{N_k(\mathbf{S})N_0\eta_k}{N_k(\mathbf{S})+N_0\eta_k}(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2)^{-a_0-\frac{N_k(\mathbf{S})}{2}}}{(N_k(\mathbf{S}) + N_0\eta_k)^{1/2}},$$

we rearrange the above integrand expression as follows

$$f(\eta_k) = \frac{(N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}))^{-a_0-\frac{N_k(\mathbf{S})}{2}} D(\eta_k)^{-a_0-\frac{N_k(\mathbf{S})}{2}}}{N_k(\mathbf{S})^{1/2}\left(1 + \frac{N_0\eta_k}{N_k(\mathbf{S})}\right)^{1/2}}, \tag{6.2.7}$$

where

$$D(\eta_k) = 1 + \frac{1}{N_k(\mathbf{S})S_{y,k}^2(\mathbf{S})}\left(2b_0 + \frac{N_k(\mathbf{S})N_0\eta_k}{N_k(\mathbf{S}) + N_0\eta_k}(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2\right) \tag{6.2.8}$$

We consider partition P of $[0,1]$ into subintervals $[x_0, x_1], [x_1, x_2], \cdots, [x_{n-1}, x_n]$. Hence, $\Delta x_i = 1/n$ and $x_i^* = i\Delta x_i$. This leads to

$$\int_0^1 f(\eta_k)d\eta_k = \sum_{i=1}^{n} \frac{1}{n} f_k(x_i^*).$$

Even using the above approximation did not completely solve the problem of overflow and underflow and we still got some infinity values in numerical calculations. To tackle this problem we divide all summands by the largest element which is $f_k(x_n^*) = f_k(1)$. Therefore we calculate

$$\int_0^1 f(\eta_k) d\eta_k = \frac{f_k(x_n^*)}{n} \sum_{i=1}^{n} \frac{f_k(x_i^*)}{f_k(x_n^*)}, \tag{6.2.9}$$

where according to the Equation (6.2.7) we have

$$\frac{f_k(x_i^*)}{f_k(x_n^*)} = \frac{(1+\frac{N_0}{N_k(\mathbf{S})})^{1/2}}{(1+\frac{N_0 x_i^*}{N_k(\mathbf{S})})^{1/2}(\frac{D_k(x_i^*)}{D_k(1)})^{a_0+\frac{N_k(\mathbf{S})}{2}}}.$$

Now according to the expression (6.2.8) we have

$$\begin{aligned}\frac{D_k(x_i^*)}{D_k(1)} &= \frac{N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}) + 2b_0 + \frac{N_k(\mathbf{S})N_0 x_i^*}{N_k(\mathbf{S})+N_0 x_i^*}(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2}{N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}) + 2b_0 + \frac{N_k(\mathbf{S})N_0}{N_k(\mathbf{S})+N_0}(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2} \\ &= \frac{S_{y,k}^2(\mathbf{S}) + \frac{2b_0}{N_k(\mathbf{S})} + \frac{N_0 x_i^*}{N_k(\mathbf{S})+N_0 x_i^*}(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2}{S_{y,k}^2(\mathbf{S}) + \frac{2b_0}{N_k(\mathbf{S})} + \frac{N_0}{N_k(\mathbf{S})+N_0}(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2}\end{aligned}$$

In order to use the latter expression in computational programming and avoid any possible underflow issue, we further rearrange the latter expression to get

$$\begin{aligned}\frac{D_k(x_i^*)}{D_k(1)} &= 1 - \frac{(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2 \left(\frac{N_0}{N_k(\mathbf{S})+N_0} - \frac{N_0 x_i^*}{N_k(\mathbf{S})+N_0 x_i^*}\right)}{S_{y,k}^2(\mathbf{S}) + \frac{2b_0}{N_k(\mathbf{S})} + \frac{N_0}{N_k(\mathbf{S})+N_0}(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2} \\ &= 1 - \frac{\frac{N_0}{N_k(\mathbf{S})+N_0}\left(1 - \frac{(N_k(\mathbf{S})+N_0)x_i^*}{N_k(\mathbf{S})+N_0 x_i^*}\right)(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2}{S_{y,k}^2(\mathbf{S}) + \frac{2b_0}{N_k(\mathbf{S})} + \frac{N_0}{N_k(\mathbf{S})+N_0}(\bar{y}_k(\mathbf{S}) - \mu_{k0})^2}.\end{aligned} \tag{6.2.10}$$

Now the integration in the Equation (6.2.11) is obtained by calculating the following summation

$$\begin{aligned}\frac{f_k(x_n^*)}{n}\sum_{i=1}^{n}\frac{f_k(x_i^*)}{f_k(x_n^*)} &= \frac{1}{n}\frac{(N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}))^{-a_0-\frac{N_k(\mathbf{S})}{2}} D(1)^{-a_0-\frac{N_k(\mathbf{S})}{2}}}{N_k(\mathbf{S})^{1/2}} \\ &\times \sum_{i=1}^{n} \frac{1}{(1+\frac{N_0 x_i^*}{N_k(\mathbf{S})})^{1/2}(\frac{D_k(x_i^*)}{D_k(1)})^{a_0+\frac{N_k(\mathbf{S})}{2}}}\end{aligned} \tag{6.2.11}$$

Substituting the above expression in the allocation posterior results in

$$
\begin{aligned}
p(\mathbf{S}|\mathbf{y}) &= N_0^{K/2}\left(\frac{1}{\pi}\right)^{N/2}\left(\frac{b_0^{a_0}}{\Gamma(a_0)}\right)^{K}\frac{\Gamma(Ke_0)}{\Gamma(e_0)^K}2^{Ka_0} \\
&\times \frac{\prod_{k=1}^{K}\Gamma(N_k(\mathbf{S})+e_0+1/2)}{\Gamma(N+Ke_0+K/2)}\prod_{k=1}^{K}\Gamma(N_k(\mathbf{S})/2+a_0) \\
&\times \frac{1}{n}\frac{(N_k(\mathbf{S})S_{y,k}^2(\mathbf{S}))^{-a_0-\frac{N_k(\mathbf{S})}{2}}D(1)^{-a_0-\frac{N_k(\mathbf{S})}{2}}}{N_k(\mathbf{S})^{1/2}} \\
&\times \sum_{i=1}^{n}\frac{1}{(1+\frac{N_0x_i^*}{N_k(\mathbf{S})})^{1/2}(\frac{D_k(x_i^*)}{D_k(1)})^{a_0+\frac{N_k(\mathbf{S})}{2}}}
\end{aligned}
\tag{6.2.12}
$$

Taking the log of this posterior, we get

$$
\begin{aligned}
\log(p(\mathbf{S}|\mathbf{y})) &= K/2\log(N_0) - N/2\log(\pi) + Ka_0\log(b_0) - K\log\Gamma(a_0) \\
&+ \log\Gamma(Ke_0) - K\log\Gamma(e_0) + Ka_0\log(2) - \Gamma(N+Ke_0+K/2) \\
&+ \sum_{k=1}^{K}\log\Gamma(N_k(\mathbf{S})+e_0+1/2) + \sum_{k=1}^{K}\log\Gamma(a_0+\frac{N_k(\mathbf{S})}{2}) \\
&- \sum_{k=1}^{K}(a_0+N_k(\mathbf{S})/2)\left[\log(N_k(\mathbf{S}))+\log(S_{y,k}^2)+\log(D(1))\right] \\
&- \sum_{k=1}^{K}\log(n) + \sum_{k=1}^{K}\log\sum_{i=1}^{n}\frac{1}{(1+\frac{N_0x_i^*}{N_k(\mathbf{S})})^{1/2}(\frac{D_k(x_i^*)}{D_k(1)})^{a_0+\frac{N_k(\mathbf{S})}{2}}}
\end{aligned}
\tag{6.2.13}
$$

6.3 Simulation studies

In this section, we conduct simulations to compare the classification accuracy of Bayesian normal mixture model with that of normal mixture models. We implement the Bayesian normal mixture model based on both independent priors and dependent priors. In order to compare the classification performance of Bayesian mixture model with the frequentist model we use the Mclust software which was also used in previous chapter. This software performs classification by applying normal mixture models. The optimal classification is a result of applying EM algorithm for finding the maximum likelihood estimates of the model parameters.

6.3.1 Data generation

We generated data from a normal mixture model with three components. We used the same setting as used in one of the examples in Frühwirth-Schnatter (2006) to generate the data. The weights or mixing proportions denoted by vector $\boldsymbol{\eta} = (\eta_1, \eta_2, \eta_3)$ are set to $\boldsymbol{\eta} = (0.3, 0.2, 0.5)$. Also, each component is a normal mixture model with mean μ_k and variance σ_k^2. These component parameters were set to $\boldsymbol{\mu} = (-3, 0, 2)$ and $\boldsymbol{\sigma^2} = (1, 0.5, 0.8)$.

6.3.2 Results

We remind that our interest in this analysis is exploring the classification performance of Bayesian mixture models. We utilised the Bayesian mixture model under the following hierarchical priors where the component mean depends on the weight or mixing proportion corresponding to that component

$$\begin{aligned} \mu_k|\sigma_k^2, \eta_k &\sim N(\mu_{k0}, \frac{\sigma_k^2}{N_0\eta_k}), \\ \sigma_k^2 &\sim IG(a_0, b_0). \\ \eta_k &\sim D(e_0, \cdots, e_0), \end{aligned}$$

which results in the allocation posterior in Equation (6.2.12). We also implemented the Bayesian mixture model with hierarchical priors where the mean of each component was independent of the weight or mixing proportion as following

$$\begin{aligned}
\mu_k|\sigma_k^2 &\sim N(\mu_{k0}, \frac{\sigma_k^2}{N_0}), \\
\sigma_k^2 &\sim IG(a_0, b_0), \\
\eta_k &\sim D(e_0, \cdots, e_0),
\end{aligned}$$

The allocation posterior can be regarded as a function of hyperparameters $N_0, a_0, b_0, e_0, \mu_{k0}$. We followed Richardson and Green (1997) in order to choose some of these hyperparameters. Richardson and Green (1997) set μ_{k0} to the median of the data. The hyperparameters are chosen as $a_0 = 2$ and $e_0 = 1$ and for the parameter b_0 they consider the prior $b_0 \sim G(0.2, 10/R^2)$ where R^2 is the length of the interval of the variation of the data. In order to choose N_0 we followed Raftery et al. (1996) and set $N_0 = 2.6/(y_{max} - y_{min})$.

We found the optimal classification by maximising the logarithm of allocation posterior. To this aim, we applied an iterative algorithm through which we updated the allocation vector in each iteration and we calculated the posterior corresponding to the updated allocation vector. The algorithm started with an initial allocation vector $\mathbf{S}^{(0)} = \mathbf{S}_{current}$. We chose this initial vector by implementing Mclust to classify the data. This classification was obtained by fitting a normal mixture model to the data. Let $\mathbf{S}^{(0)} = (S_1, \cdots, S_N)$ denote the allocation vector of observations y_i; $i = 1, \cdots, N$ obtained by Mclust. In the first iteration we updated the allocation S_1 corresponding to the first observation y_1 while allocation index of the rest of observations were fixed. To update the allocation we generated a random number from the uniform distribution $\mathcal{U}[0, 1]$. If $U < p_1$, the observation y_1 was assigned to the first component, if $U < p_1 + p_2$, the observation was assigned to the second component, otherwise y_1 was assigned to the third component, where $p_1 = p_2 = p_3 = 1/3$. This resulted in an updated allocation vector $\mathbf{S}^{(1)} = \mathbf{S}_{updated}$. As a result of this update, the number of elements in each component changed. If $S_1^{new} = S_1 = k$, then no moving occurred whereas, if the observation moved to another component, say l, then the number of observations in each component were updated as

$$N_k(S_1^{new}, \mathbf{S}_{-1}) = N_k(\mathbf{S}) - 1, \quad N_l(S_1^{new}, \mathbf{S}_{-1}) = N_l(\mathbf{S}) + 1,$$

correspondingly, the mean $\bar{y}_k(\mathbf{S})$ and the variance $S_{y,k}(\mathbf{S})$ of each component were updated. Then the log-posterior $p((S_1^{new}, \mathbf{S}_{-1})|\mathbf{y})$ corresponding to the updated allocation vector was calculated according to the expression (6.2.13). The updated allocation for the first observation was accepted if the updated posterior was greater than the current posterior, i.e. $p((S_1^{new}, \mathbf{S}_{-1})|\mathbf{y}) > p(\mathbf{S}^{(0)}|\mathbf{y})$. If the new allocation was accepted, then this updated allocation was used as the current allocation in the next iteration $\mathbf{S}_{current} = \mathbf{S}_{updated}$ and the observation was moved to the component l. Otherwise, the observation was kept in the current component k and the algorithm moved to the next observation y_2. These steps were repeated until all observations $i = 1 \cdots, N$ were updated and also the posterior converges to the maximum possible value. Then the allocation vector corresponding to this optimal posterior was recorded and compared with that of Mclust classification by finding the adjusted Rand index (Rand, 1971) which was introduced in the introduction chapter.

We simulated 300 datasets from a mixture of normals with three components as explained earlier and we applied the above algorithm to find the optimal classification for these data. While calculating posteriors in the above process, we considered both dependent case (6.2.13) and independent case (6.2.5) posteriors.

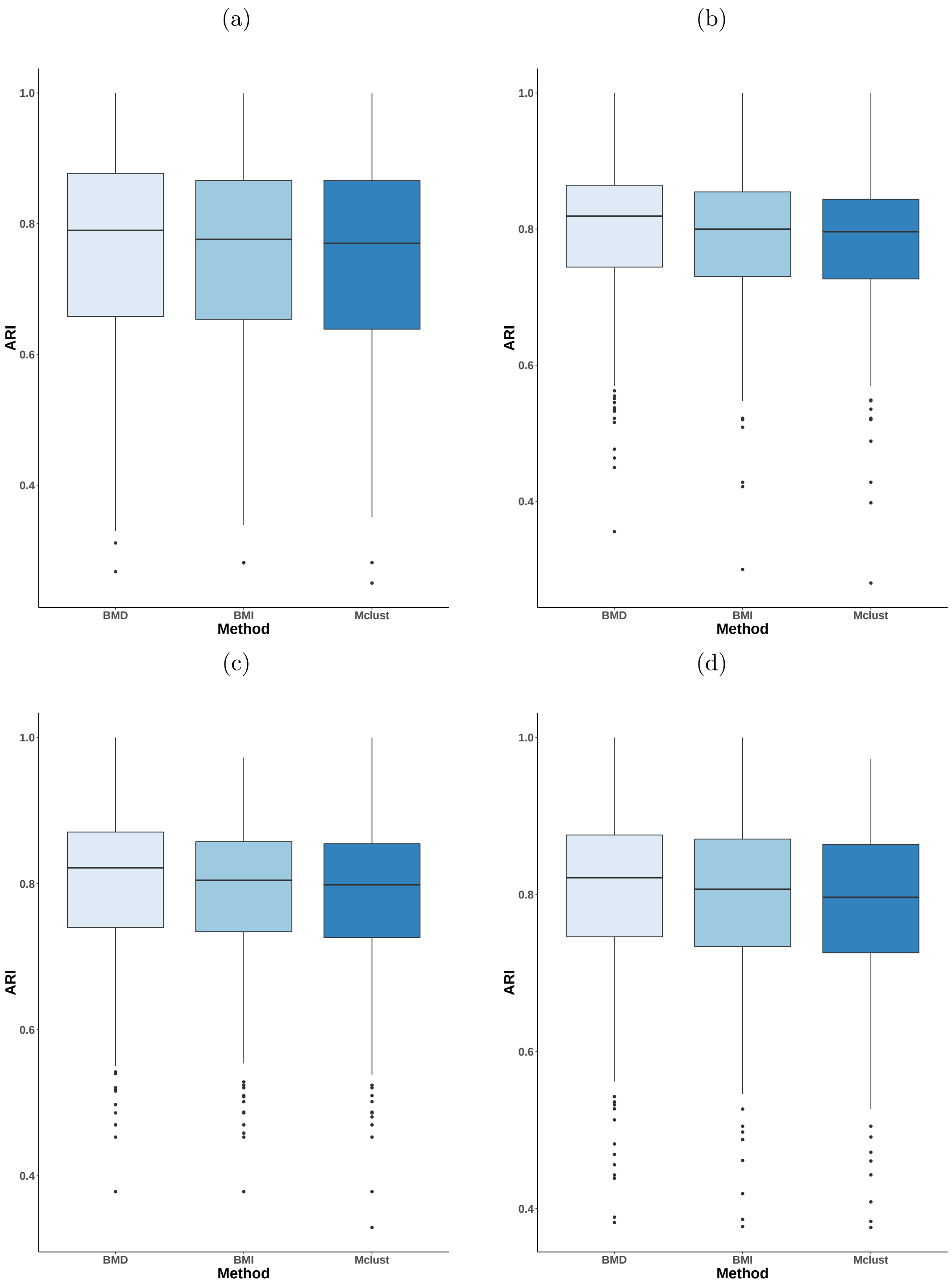


Figure 6.3.1: Box plots of adjusted Rand index values corresponding to the classifications performed by applying Bayesian mixture model with dependent priors (BMD), Bayesian mixture model with independent priors (BMI) and the non-Bayesian mixture of normals (Mclust) where $N_0 = 2.6/(y_{max} - y_{min})$ and $b_0 \sim G(0.2, 10/R^2)$ where R^2 is the length of the interval of the variation of the data. Other hyperparameters and sample size are chosen as (a) $N = 50$, $a_0 = 2$, $e_0 = 1$, (b) $N = 100$, $a_0 = 2$, $e_0 = 1$, (c) $N = 100, a_0 = 5$, $e_0 = 1$, (d) $N = 100, a_0 = 5$, $e_0 = 2$

Results presented in Figure 6.3.1 show that the Bayesian classification results in a more accurate classification particularly when the priors are dependent. Due to the imposed dependency on priors, more information is brought to the inference which leads to a more accurate classification. These results show that the idea of setting dependent prior is promising and could be a potential area for more investigation.

Note that the search strategy applied here to find the optimal allocation vector is a naive way. The more principled way to find the optimal allocation vector, particularly when the sample size is large, is applying powerful methods of MCMC. Unfortunately due to the time limit this part could not be completed during this course and is remained for the future work.

Chapter 7

Conclusion and Future Work

In analysing cancer drug data, in order to follow the recent "multiple genes, multiple drugs" paradigm, we have suggested to use a multivariate multiple linear regression which can accommodate all the response variables simultaneously. In this regression model there are a relatively large number of predictors compared to the sample size. To address this high dimensionality issue, in Chapter 3 we have proposed the SNR-based screening method to reduce the number of predictors in the above multivariate regression model. In this method, the SNR values of predictors are calculated. Then these values are ranked and thresholded. The predictors with higher SNR values are selected. Through simulation studies we showed that this approach outperformed the Sure Independence Screening (SIS) method of Fan and Lv (2008). SNR-based screening results in a higher accuracy by having higher sensitivity values. Since in the SNR-based screening the correlation structure in the response variables is taken into account, the ability of SNR-based screening is higher than SIS in identifying the informative predictors. SNR-based screening also has the sure screening property.

Based on the SNR statistic, in Chapter 4 we have developed a novel variable selection method, called principal variable analysis (PVA)(Zhang and Oftadeh, 2016). Since high dimensional predictors are often highly correlated, we have introduced a forward-nulling procedure to reduce the interferences from other predictors, while we calculate the SNR value for one predictor. By PVA, we try to find a small number of principal variables to explain the maximum amount of variation in the data. We have compared the PVA performance with some of the existing methods through simulation studies. These methods are multivariate lasso, multivariate elastic net, multivariate group-lasso and multivariate sparse group-lasso. The result have shown that PVA has a much higher selection accuracy compare to these methods.

The only limitation of PVA is that the number of iterations in the selection algorithm is restricted to the sample size n. We have also assumed that the error terms are iid distributed with mean zero and equal variances. However, this assumption may be restrictive in some applications. Generalising this algorithm to the one that is not restricted to the sample size is the next step for this research.

In Chapter 5 we have introduced a mixture-based model which is called likelihood fusion. Then we have used the likelihood fusion model in a two-stage procedure wherein the screening of predictors and the classification of the response variables have been preformed simultaneously. In this inference the aim is to classify the response variable in a dataset which contains several response variables with high dimensional predictors. Through the two-stage procedure we have first screened the predictors in some way and then classified the response variables using the selected predictors. Since the number of selected predictors may exceed the sample size n, to reduce the size of predictors even further, we have applied a forward selection method after performing screening. Then the reduced selected predictors have been used to classify the response variables. Although forward selection could solve the issue and reduce the size of selected predictors, it is not an optimal method. The reason is that forward selection is computationally expensive. The forward selection stage could be improved by penalising the likelihood fusion model. This could be a potential research area for further investigation.

We have also compared the SNR-based screening with the screening performed through likelihood fusion model. The results have shown that the mixture-based screening outperforms the SNR screening when the data have a group structure. Mixture-based screening has resulted in a higher accuracy by taking into account the information from each group of response variables.

In the last chapter we have studied a Bayesian inference clustering by applying the mixture of normal distributions. The aim is to classify the data without estimating the component parameters. Therefore in this inference we wish to find the allocation vector of the data which yields the optimal classification. To find this allocation vector we have maximised the posterior of the allocation vector. To this aim, we have considered two different set of hierarchical priors and calculated the posterior of the allocation vector. In inference with mixture of normals it is common to choose priors for the component parameters which are independent of the mixing proportions $\boldsymbol{\theta}_k$. Alternatively, we have proposed a different prior for the mean of each component μ_k which is not independent of the weight η_k. Simulation results have shown that the classification obtained by the mixing proportion dependant prior is more accurate than the commonly used prior which is not dependant on the mixing proportion. The search strategy applied in Chapter 6 to find the optimal allocation vector is a naive way. The more principled way to find the optimal allocation vector, particularly when the sample size is large, is applying powerful methods of MCMC. Unfortunately due to the time limit this part could not be completed during this course and is remained for the future work.

Appendix A

Protein staining levels of the selected genes in 20 common cancers

In this appendix we present some information about the genes which were selected by applying PVA to cancer drug data. To reveal the roles played by these selected genes in cancer, we investigated their protein staining in 20 common cancers. The Tables 4 ∼8 presented here provide some information gathered from the Human Protein Atlas Portal http://www.proteinforlas.org/cancer. In these tables, as in the Portal, we classified the protein expression/staining levels into four categories: high, medium, low and not detected. We assigned the scores of $3, 2, 1$ and 0 to the four categories respectively. If a gene did not play a role in a cancer, it would receive a score of zero as its protein staining for that cancer would be hardly detectable. We found that 34 of the selected genes had positive staining levels on at least one of these cancers. This implies that these genes might play certain functional roles in the growth of some of these cancers. In the Portal, there were no information available on the remaining 3 of the selected genes.

Table 4: Mean scores of protein staining for cancers with standard errors in brackets. For each cancer type, the highest mean score across all genes is highlighted. Genes that have the highest mean score for at least one type of cancer are shown in colour.

	Cancer type							
Gene	Breast		Carcinoid		Cervical		Colorectal	
IARS	0.9	(0.15)	0.25	(0.21)	0.33	(0.24)	1.36	(0.26)
CLASP1	0.9	(0.15)	1.75	(0.21)	0.3	(0.2)	1.12	(0.32)
STAMBPL1	2	(0.00)	1.25	(0.41)	1.6	(0.25)	2.25	(0.12)
GSTM3	0.75	(0.29)	0	(0.00)	0	(0.00)	0	(0.00)
EML1	0	(0.00)	0	(0.00)	0.09	(0.08)	0	(0.00)
TRIM6-TRIM34	NA	NA	NA	NA	NA	NA	NA	NA
DECR1	2	(0.11)	1.25	(0.25)	0.83	(0.23)	2.16	(0.16)
EP400	1.27	(0.22)	1.25	(0.41)	1.41	(0.22)	1.27	(0.31)
TADA2L	NA	NA	NA	NA	NA	NA	NA	NA
RPL39L	0.16	(0.16)	0	(0.00)	0.08	(0.08)	0	(0.00)
FAIM3	2.08	(0.08)	1.75	(0.54)	1.81	(0.22)	2.45	(0.15)
C18ORF24 (SKA1)	2.25	(0.12)	1.5	(0.55)	1.8	(0.12)	2.09	(0.20)
CD1A	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
CIDEB	1.2	(0.23)	1.25	(0.21)	0.75	(0.17)	1	(0.47)
TP53	0.41	(0.27)	0	(0.00)	1	(0.31)	1.75	(0.33)
QKI	0	(0.00)	0	(0.00)	0.25	(0.12)	0.25	(0.17)
SNTB1	0.45	(0.15)	0	(0.00)	0.33	(0.13)	0.25	(0.12)
SEMA4C	0.75	(0.20)	0	(0.00)	0.58	(0.24)	1.36	(0.19)
NUDT2	1.81	(0.12)	1.5	(0.25)	1.5	(0.22)	1.6	(0.19)
RFX2	0	(0.00)	0	(0.00)	0	(0.00)	0.08	(0.08)
GPSN2(TECR)	0.1	(0.10)	0	(0.00)	0.58	(0.24)	0.33	(0.18)
C21ORF45 (MIS18A)	1.45	(0.20)	1.25	(0.21)	0.5	(0.18)	1.7	(0.24)
COL5A1	0.44	(0.16)	0	(0.00)	1.33	(0.27)	0.6	(0.15)
RP1.153G14.3 (ZNF391)	2	(0.00)	2	(0.00)	1.25	(0.20)	2	(0.00)
MKL1	1.08	(0.22)	0.25	(0.21)	0.91	(0.22)	0.91	(0.25)
FKSG44	2.16	(0.11)	2	(0.35)	2	(0.00)	2.91	(0.09)
KIAA1856	2.63	(0.15)	2	(0.00)	2	(0.12)	2.4	(0.15)
HDGF2	NA	NA	NA	NA	NA	NA	NA	NA
CROCC	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
WDR76	0.3	(0.14)	0	(0.00)	0	(0.00)	0.54	(0.23)
RPS14	2.16	(0.16)	2	(0.00)	2.08	(0.08)	2.63	(0.15)
MAP3K6	1.25	(0.23)	1.5	(0.43)	1.83	(0.11)	1.41	(0.22)
LY6E	1.11	(0.36)	1.75	(0.21)	0.58	(0.18)	0.81	(0.28)
SLCO2B1	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
NR1D2	1.45	(0.20)	0.5	(0.25)	1.75	(0.12)	2	(0.00)
RHBDD3	1.58	(0.25)	0.5	(0.25)	0.5	(0.29)	0.81	(0.25)
STX7	0.16	(0.16)	0.66	(0.27)	0.27	(0.18)	0.5	(0.15)

Table 5: Mean scores of protein staining for cancers with standard errors in brackets. For each cancer type, the highest mean score across all genes is highlighted. Genes that have the highest mean score for at least one type of cancer are shown in colour.

	Cancer type							
Gene	Endometrial		Glioma		Hand and neck		Liver	
IARS	0.58	(0.27)	0.08	(0.08)	0.3	(0.28)	1.58	(0.27)
CLASP1	0.9	(0.27)	0.33	(0.18)	0.25	(0.21)	1.4	(0.20)
STAMBPL1	1.72	(0.14)	1.5	(0.22)	2	(0.00)	0.83	(0.25)
GSTM3	0	(0.00)	0	(0.00)	0	(0.00)	0.08	(0.08)
EML1	0	(0.00)	0.09	(0.08)	0	(0.00)	0	(0)
TRIM6-TRIM34	NA	NA	NA	NA	NA	NA	NA	NA
DECR1	2.16	(0.16)	2	(0.16)	1.75	(0.21)	2.83	(0.11)
EP400	1.18	(0.25)	1.41	(0.30)	1.5	(0.25)	1	(0.26)
TADA2L	NA	NA	NA	NA	NA	NA	NA	NA
RPL39L	0	(0.00)	0.4	(0.15)	0	(0.00)	0.83	(0.23)
FAIM3	1.91	(0.25)	1.08	(0.25)	1.75	(0.21)	1.36	(0.32)
C18ORF24 (SKA1)	1.41	(0.22)	2	(0.12)	1.66	(0.28)	1.5	(0.25)
CD1A	0	(0.00)	0	(0.00)	0	(0.00)	0.16	(0.16)
CIDEB	1.3	(0.13)	0.91	(0.25)	1.5	(0.25)	1.25	(0.12)
TP53	0.25	(0.23)	1.08	(0.38)	1.25	(0.64)	0.3	(0.14)
QKI	0.08	(0.08)	2.83	(0.11)	0.75	(0.41)	0	(0.00)
SNTB1	0.08	(0.08)	0.25	(0.17)	0	(0.00)	0.25	(0.12)
SEMA4C	0.7	(0.14)	0.25	(0.12)	0.5	(0.25)	0.91	(0.22)
NUDT2	1.36	(0.26)	1.90	(0.16)	2	(0.00)	1.63	(0.23)
RFX2	0	(0.00)	0.3	(0.20)	0	(0.00)	0.08	(0.08)
GPSN2(TECR)	0.36	(0.14)	0	(0.00)	0	(0.00)	0.58	(0.18)
C21ORF45 (MIS18A)	1.16	(0.16)	0.18	(0.11)	1	(0.00)	1.75	(0.17)
COL5A1	0.54	(0.19)	0.09	(0.08)	2.3	(0.35)	0.16	(0.16)
RP1.153G14.3 (ZNF391)	1.83	(0.11)	1.09	(0.20)	1.25	(0.41)	2	(0.00)
MKL1	0.41	(0.18)	0.83	(0.28)	1.33	(0.72)	0.75	(0.20)
FKSG44	1.90	(0.20)	1.83	(0.11)	2.25	(0.21)	1.37	(0.39)
KIAA1856	1.33	(0.29)	1.91	(0.09)	2.5	(0.25)	1.4	(0.32)
HDGF2	NA	NA	NA	NA	NA	NA	NA	NA
CROCC	0.27	(0.18)	0	(0.00)	0	(0.00)	0.09	(0.08)
WDR76	0	(0.00)	0.25	(0.23)	0.3	(0.28)	0	(0.00)
RPS14	2.25	(0.12)	2.18	(0.17)	2.33	(0.28)	1.81	(0.12)
MAP3K6	1	(0.26)	1.6	(0.28)	1.75	(0.21)	1.41	(0.18)
LY6E	0.54	(0.23)	1	(0.22)	0.75	(0.41)	0	(0.00)
SLCO2B1	0	(0.00)	0	(0.00)	0	(0.00)	0.27	(0.13)
NR1D2	1.16	(0.26)	0.91	(0.25)	1.75	(0.21)	1.2	(0.23)
RHBDD3	1.5	(0.3)	0.36	(0.23)	0.75	(0.21)	1.08	(0.32)
STX7	0.1	(0.09)	0.00	(0.00)	(0.25)	(0.21)	0.25	(0.12)

Table 6: Mean scores of protein staining for cancers with standard errors in brackets. For each cancer type, the highest mean score across all genes is highlighted. Genes that have the highest mean score for at least one type of cancer are shown in colour.

	Cancer type							
Gene	Lung		Lymphoma		Melanoma		Ovarian	
IARS	0.2	(0.17)	0.16	(0.10)	0.1	(0.10)	0.75	(0.26)
CLASP1	1.33	(0.21)	1.5	(0.22)	1.5	(0.22)	0.58	(0.21)
STAMBPL1	1.4	(0.20)	0.9	(0.26)	2.58	(0.14)	2.5	(0.14)
GSTM3	0.09	(0.08)	0	(0.00)	0	(0.00)	0.18	(0.17)
EML1	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
TRIM6-TRIM34	NA	NA	NA	NA	NA	NA	NA	NA
DECR1	1	(0.33)	1.41	(0.18)	1.8	(0.13)	1.81	(0.12)
EP400	1.2	(0.23)	1.18	(0.30)	0.9	(0.30)	1.41	(0.18)
TADA2L	NA	NA	NA	NA	NA	NA	NA	NA
RPL39L	0.08	(0.08)	0	(0.00)	0	(0.00)	0.08	(0.08)
FAIM3	1.33	(0.24)	1.5	(0.20)	1	(0.20)	1.2	(0.23)
C18ORF24 (SKA1)	1.16	(0.23)	0.66	(0.21)	1.9	(0.19)	1.41	(0.18)
CD1A	0	(0.00)	0	(0.00)	0.08	(0.08)	0	(0.00)
CIDEB	0.91	(0.18)	0.41	(0.14)	1.63	(1.19)	1.2	(0.15)
TP53	1.08	(0.32)	0.25	(0.12)	0.90	(0.24)	1.5	(0.43)
QKI	0	(0.00)	0.08	(0.08)	0.91	(0.22)	0.27	(0.18)
SNTB1	0.25	(0.12)	0	(0.00)	0	(0.00)	0.16	(0.16)
SEMA4C	0.41	(0.22)	1	(0.25)	0.41	(0.41)	0.75	(0.23)
NUDT2	1	(0.31)	1.66	(0.21)	1.36	(0.19)	1.41	(0.14)
RFX2	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
GPSN2(TECR)	0.4	(0.25)	0	(0.00)	0.3	(0.20)	0.45	(0.15)
C21ORF45 (MIS18A)	0.63	(0.28)	0.66	(0.18)	1.66	(0.24)	0.83	(0.19)
COL5A1	0.18	(0.11)	0	(0.00)	0.54	(0.26)	0.7	(0.20)
RP1.153G14.3 (ZNF391)	1.54	(0.20)	0.33	(0.13)	1.63	(0.25)	1.91	(0.09)
MKL1	0.08	(0.08)	1.27	(0.29)	0.16	(0.10)	0.27	(0.13)
FKSG44	2	(0.20)	0.6	(0.15)	1.7	(0.22)	1.91	(0.27)
KIAA1856	0.8	(0.36)	0.83	(0.19)	1.83	(0.20)	1.41	(0.22)
HDGF2	NA	NA	NA	NA	NA	NA	NA	NA
CROCC	0	(0.00)	0	(0.00)	0.08	(0.08)	0	(0.00)
WDR76	0	(0.00)	0.09	(0.08)	0.25	(0.17)	0	(0.00)
RPS14	2.1	(0.20)	2.36	(0.19)	2.54	(0.15)	2.41	(0.15)
MAP3K6	1.44	(0.27)	0.16	(0.10)	0.66	(0.24)	1	(0.26)
LY6E	0.33	(0.21)	0	(0.00)	0.5	(0.22)	0.91	(0.25)
SLCO2B1	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
NR1D2	1.66	(0.14)	0.66	(0.21)	1.54	(0.23)	1.16	(0.20)
RHBDD3	1.33	(0.29)	0	(0.00)	0.9	(0.32)	1.2	(0.30)
STX7	0.2	(0.12)	2.09	(0.27)	2.45	(0.30)	0.5	(0.25)

Table 7: Mean scores of protein staining for cancers with standard errors in brackets. For each cancer type, the highest mean score across all genes is highlighted. Genes that have the highest mean score for at least one type of cancer are shown in colour.

	Cancer type							
Gene	Pancreforic		Prostfore		Renal		Skin	
IARS	0.16	(0.10)	0.33	(0.22)	0	(0.00)	0.08	(0.08)
CLASP1	1.11	(0.29)	1.9	(0.24)	1.83	(0.11)	0.3	(0.14)
STAMBPL1	2.1	(0.30)	1.25	(0.26)	0.16	(0.10)	1.5	(0.30)
GSTM3	0	(0.00)	0	(0.00)	0	(0.00)	0.08	(0.08)
EML1	0	(0.00)	0	(0.00)	0.36	(0.23)	0	(0.00)
TRIM6-TRIM34	NA	NA	NA	NA	NA	NA	NA	NA
DECR1	1.3	(0.40)	2	(0.12)	1.58	(0.25)	0.09	(0.08)
EP400	1	(0.22)	1.33	(0.35)	1	(0.33)	1.22	(0.21)
TADA2L	NA	NA	NA	NA	NA	NA	NA	NA
RPL39L	0	(0.00)	0.08	(0.08)	0	(0.00)	0	(0.00)
FAIM3	1.8	(0.14)	0.8	(0.23)	1.90	(0.10)	0.33	(0.18)
C18ORF24 (SKA1)	1.41	(0.22)	2	(0.11)	1.41	(0.25)	1.54	(0.23)
CD1A	1.08	(0.08)	0	(0.00)	0.54	(0.19)	0	(0.00)
CIDEB	1.25	(0.17)	0.25	(0.12)	0.54	(0.19)	1.12	(0.12)
TP53	2	(0.31)	0	(0.00)	0.09	(0.08)	0.90	(0.27)
QKI	0	(0.00)	0	(0.00)	0	(0.00)	0.5	(0.21)
SNTB1	0.18	(0.11)	0.72	(0.13)	0	(0.00)	0	(0.00)
SEMA4C	0.41	(0.14)	0.33	(0.18)	0.08	(0.08)	0.41	(0.14)
NUDT2	1.90	(0.10)	1.91	(0.09)	1.18	(0.25)	1.2	(0.18)
RFX2	0.2	(0.12)	0	(0.00)	0	(0.00)	0	(0.00)
GPSN2(TECR)	0.11	(0.10)	0.54	(0.29)	0	(0.00)	0.5	(0.16)
C21ORF45 (MIS18A)	1.1	(0.17)	1.66	(0.16)	0.5	(0.18)	0.9	(0.22)
COL5A1	0	(0.00)	0.22	(0.13)	0.2	(0.12)	1	(0.28)
RP1.153G14.3 (ZNF391)	1.90	(0.10)	2.16	(0.11)	1.16	(0.16)	0.5	(0.14)
MKL1	0.41	(0.18)	0.5	(0.22)	0	(0.00)	0.2	(0.18)
FKSG44	1.5	(0.22)	1.72	(0.23)	0.25	(0.12)	1.33	(0.18)
KIAA1856	2.4	(0.27)	1.7	(0.27)	1.36	(0.19)	2	(0.16)
HDGF2	NA	NA	NA	NA	NA	NA	NA	NA
CROCC	0.1	(0.10)	0	(0.00)	0	(0.00)	0	(0.00)
WDR76	0.09	(0.08)	0	(0.00)	0	(0.00)	0.09	(0.08)
RPS14	1.90	(0.20)	1.5	(0.17)	1.36	(0.19)	2.6	(0.15)
MAP3K6	1.41	(0.22)	1.33	(0.21)	1.66	(0.21)	0.58	(0.24)
LY6E	1.2	(0.27)	0.16	(0.16)	0.08	(0.08)	0.58	(0.21)
SLCO2B1	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
NR1D2	1.4	(0.25)	1.41	(0.18)	0.75	(0.17)	1.25	(0.20)
RHBDD3	1.5	(0.32)	1.71	(0.26)	0.08	(0.08)	0.27	(0.13)
STX7	0.09	(0.08)	0	(0.00)	0.3	0.16	0.45	(0.19)

Table 8: Mean scores of protein staining for cancers with standard errors in brackets. For each cancer type, the highest mean score across all genes is highlighted. Genes that have the highest mean score for at least one type of cancer are shown in colour.

	Cancer type							
Gene	Stomach		Testis		Thyroid		Urothelial	
IARS	0.1	(0.09)	0.5	(0.21)	1	(0.00)	0.54	(0.29)
CLASP1	1.27	(0.22)	1.16	(0.23)	1.5	(0.55)	1.41	(0.22)
STAMBPL1	2	(0.11)	1.45	(0.20)	1.75	(0.59)	2.09	(0.15)
GSTM3	0	(0.00)	0	(0.00)	0	(0.00)	0.5	(0.32)
EML1	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
TRIM6-TRIM34	NA	NA	NA	NA	NA	NA	NA	NA
DECR1	1.9	(0.17)	1.27	(0.29)	2.75	(0.21)	2	(0.31)
EP400	1.54	(0.30)	1	(0.16)	1.25	(0.21)	0.90	(0.27)
TADA2L	NA	NA	NA	NA	NA	NA	NA	NA
RPL39L	0	(0.00)	0	(0.00)	0.25	(0.21)	0	(0.00)
FAIM3	1.18	(0.11)	0.91	(0.25)	2.25	(0.21)	1.66	(0.21)
C18ORF24 (SKA1)	1.58	(0.25)	1.5	(0.21)	1.66	(0.28)	1.18	(0.31)
CD1A	1.45	(0.23)	0.83	(0.10)	0	(0.00)	1.09	(0.15)
CIDEB	1.41	(0.14)	1.33	(0.18)	1.5	(0.75)	1.08	(0.22)
TP53	1	(0.49)	1.16	(0.23)	0	(0.00)	1.81	(0.33)
QKI	0.1	(0.09)	0.2	(0.14)	0	(0.00)	0.16	0.16
SNTB1	0	(0.00)	0	(0.00)	0.5	(0.25)	0.08	(0.08)
SEMA4C	0.18	(0.11)	0.5	(0.22)	0.75	(0.41)	0.75	(0.20)
NUDT2	1.58	(0.14)	1.83	(0.11)	1	(0.35)	1.83	(0.11)
RFX2	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
GPSN2(TECR)	0	(0.00)	0.09	(0.008)	0.3	(0.28)	0.18	(0.11)
C21ORF45 (MIS18A)	0.54	(0.32)	1.6	(0.32)	2	(0.35)	1	(0.23)
COL5A1	0.27	(0.13)	0.4	(0.17)	0.25	(0.21)	0.91	(0.27)
RP1.153G14.3 (ZNF391)	1.90	(0.10)	1.63	(0.15)	1	(0.00)	1.9	(0.09)
MKL1	0.09	(0.08)	1	(0.22)	0	(0.00))	1	(0.15)
FKSG44	1.66	(0.40)	0.4	(0.17)	1	(0.00)	1.33	(0.21)
KIAA1856	1.6	(0.25)	2.5	(0.25)	2.5	(0.25)	2.09	(0.20)
HDGF2	NA	NA	NA	NA	NA	NA	NA	NA
CROCC	0.1	0.10	0	(0.00)	0	(0.00)	0	(0.00)
WDR76	0	(0.00)	0.7	(0.28)	0.25	(0.21)	0	(0.00)
RPS14	2.09	(0.15)	1.88	(0.12)	2.6	(0.43)	1.72	(0.14)
MAP3K6	1.58	(0.18)	1.33	(0.24)	2	(0.00)	2	(0.16)
LY6E	0.72	(0.31)	0.36	(0.14)	0	(0.00)	0.58	(0.18)
SLCO2B1	0	(0.00)	0	(0.00)	0	(0.00)	0	(0.00)
NR1D2	1.72	(0.14)	2	(0.00)	1.75	(0.21)	1.9	(0.26)
RHBDD3	0.6	(0.28)	0.54	(0.26)	0.75	(0.41)	0.63	(0.23)
STX7	0.5	0.27	0.3	0.20	0.75	(0.41)	0.45	(0.15)

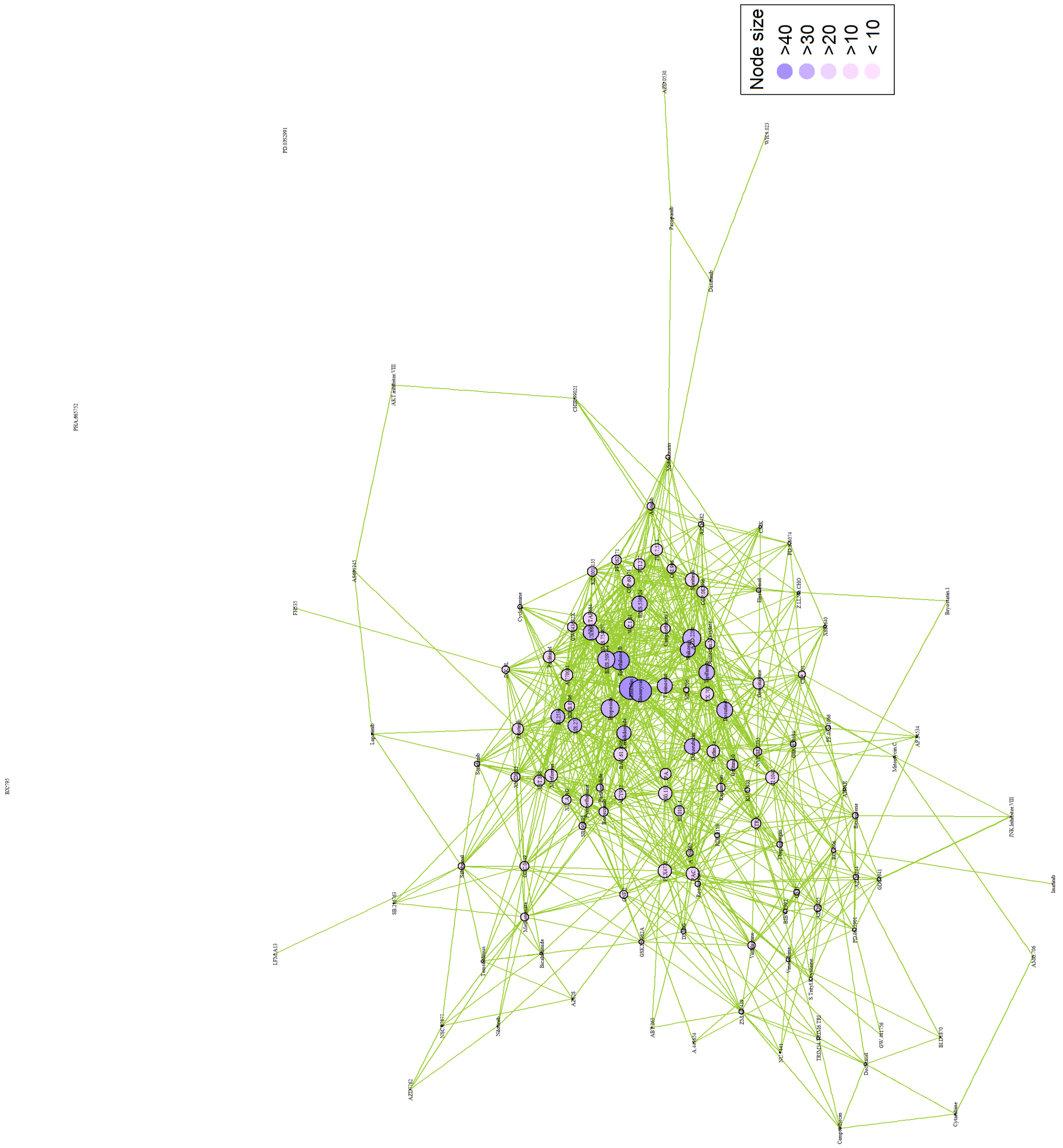


Figure A.0.2: Network between columns of estimated coefficient matrix. Each vertex corresponds to one drug and is a vector of dimension 37. Size of each node represents the degree of each node. This is a strongly connected network which shows that drugs are highly correlated.

Bibliography

Abraham, B. and Merola, G. (2005), 'Dimensionality reduction approach to multivariate prediction', *Computational statistics & data analysis* **48**(1), 5–16.

Acton, Q. (2013), *Issues in Nuclear, High Energy, Plasma, Particle, and Condensed Matter Physics: 2013 Edition*, ScholarlyEditions.
URL: *https://books.google.co.uk/books?id=CoRFM_ Ou9kEC*

Al-Subaihi, A. A. et al. (2002), 'Variable selection in multivariable regression using sas/iml', *Journal of Statistical Software* **7**(i12).

Almeida, C. A. and Barry, S. A. (2011), *Cancer: basic science and clinical aspects*, John Wiley & Sons.

Asche, C. V. (2015), *Applying Comparative Effectiveness Data to Medical Decision Making: A Practical Guide*, Springer.

Barabási, A.-L. (2016), *Network science*, Cambridge university press.

Bayes, M. and Price, M. (1763), 'An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs', *Philosophical Transactions (1683-1775)* pp. 370–418.

Bebek, G. (2012), 'Identifying gene interaction networks', *Statistical Human Genetics: Methods and Protocols* pp. 483–494.

Berry, W. D. and Feldman, S. (1985), *Multiple regression in practice*, number 50, Sage.

Bickel, P. J. and Levina, E. (2008), 'Covariance regularization by thresholding', *The Annals of Statistics* pp. 2577–2604.

Binder, D. A. (1978), 'Bayesian cluster analysis', *Biometrika* **65**(1), 31–38.

Breiman, L. (1995), 'Better subset regression using the nonnegative garrote', *Technometrics* **37**(4), 373–384.

Breiman, L. and Friedman, J. H. (1997), 'Predicting multivariate responses in multiple linear regression', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **59**(1), 3–54.

Bühlmann, P. and Van De Geer, S. (2011), *Statistics for high-dimensional data: methods, theory and applications*, Springer Science & Business Media.

Carter, B. Z., Mak, P. Y., Mu, H., Zhou, H., Mak, D. H., Schober, W., Leverson, J. D., Zhang, B., Bhatia, R., Huang, X. et al. (2016), 'Combined targeting of bcl-2 and bcr-abl tyrosine kinase eradicates chronic myeloid leukemia stem cells', *Science translational medicine* **8**(355), 355ra117–355ra117.

Collins, H. (2014), *Gravity's Ghost and Big Dog: Scientific discovery and social analysis in the twenty-first century*, University of Chicago Press.

Cook, R. D. (2007), 'Fisher lecture: Dimension reduction in regression', *Statistical Science* pp. 1–26.

Cook, R. D. and Forzani, L. (2009), 'Likelihood-based sufficient dimension reduction', *Journal of the American Statistical Association* **104**(485), 197–208.

Daniels, M. J. and Kass, R. E. (2001), 'Shrinkage estimators for covariance matrices', *Biometrics* **57**(4), 1173–1184.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the royal statistical society. Series B (methodological)* pp. 1–38.

Dey, D. K. and Rao, C. R. (2005), *Bayesian thinking, modeling and computation*, Vol. 25, Gulf Professional Publishing.

Dhaenens, C. and Jourdan, L. (2016), *Metaheuristics for big data*, John Wiley & Sons.

Diebolt, J. and Robert, C. P. (1994), 'Estimation of finite mixture distributions through bayesian sampling', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 363–375.

Emmert-Streib, F. (2012), *Statistical Diagnostics for Cancer: Analyzing high-dimensional data*, John Wiley & Sons.

Everitt, B. S., Landau, S., Leese, M. and Stahl, D. (2011), 'Cluster analysis: Wiley series in probability and statistics'.

Fan, J. and Lv, J. (2008), 'Sure independence screening for ultrahigh dimensional feature space', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(5), 849–911.

Fan, J., Samworth, R. and Wu, Y. (2009), 'Ultrahigh dimensional feature selection: beyond the linear model', *Journal of Machine Learning Research* **10**(Sep), 2013–2038.

Fan, K. and Hoffman, A. J. (1955), 'Some metric inequalities in the space of matrices', *Proceedings of the American Mathematical Society* **6**(1), 111–116.

Faria, S. and Soromenho, G. (2010), 'Fitting mixtures of linear regressions', *Journal of Statistical Computation and Simulation* **80**(2), 201–225.

Farrar, D. E. and Glauber, R. R. (1967), 'Multicollinearity in regression analysis: the problem revisited', *The Review of Economic and Statistics* pp. 92–107.

Fisher, T. J. and Sun, X. (2011), 'Improved stein-type shrinkage estimators for the high-dimensional multivariate normal covariance matrix', *Computational Statistics & Data Analysis* **55**(5), 1909–1918.

Fraley, C., Raftery, A. E., Murphy, T. B. and Scrucca, L. (2012), 'mclust version 4 for r: Normal mixture modeling for model-based clustering, classification, and density estimation'.

Friedman, J. H. and Tukey, J. W. (1974), 'A projection pursuit algorithm for exploratory data analysis', *IEEE Transactions on computers* **100**(9), 881–890.

Friedman, J., Hastie, T., Tibshirani, R., Hastie, M. T. and Matrix, D. (2010), 'Package ?glmnet?', *Journal of Statistical Software* **33**, 1.

Frühwirth-Schnatter, S. (2006), *Finite Mixture and Markov Switching Models: Modeling and Applications to Random Processes*, Springer.

Garnett, M. J., Edelman, E. J., Heidorn, S. J., Greenman, C. D., Dastur, A., Lau, K. W., Greninger, P., Thompson, I. R., Luo, X., Soares, J. et al. (2012), 'Systematic identification of genomic markers of drug sensitivity in cancer cells', *Nature* **483**(7391), 570–575.

Hager, W. W. (1989), 'Updating the inverse of a matrix', *SIAM review* **31**(2), 221–239.

Hall, P. and Miller, H. (2009), 'Using generalized correlation to effect variable selection in very high dimensional problems', *Journal of Computational and Graphical Statistics* **18**(3), 533–550.

Hartigan, J. A. and Wong, M. A. (1979), 'Algorithm as 136: A k-means clustering algorithm', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1), 100–108.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J. and Tibshirani, R. (2009), *The elements of statistical learning*, Vol. 2, Springer.

Hazewinkel, M. (1993), 'Encyclopaedia of mathematics: Volume 9 stochastic approximation e zygmund class of functions'.

Hoerl, A. E. and Kennard, R. W. (1970), 'Ridge regression: Biased estimation for nonorthogonal problems', *Technometrics* **12**(1), 55–67.

Hubert, L. and Arabie, P. (1985), ‘Comparing partitions’, *Journal of classification* **2**(1), 193–218.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013), *An introduction to statistical learning*, Vol. 112, Springer.

Jolliffe, I. T. (1986), Principal component analysis and factor analysis, *in* ‘Principal component analysis’, Springer, pp. 115–128.

Kidd, B. A., Readhead, B. P., Eden, C., Parekh, S. and Dudley, J. T. (2015), ‘Integrative network modeling approaches to personalized cancer medicine’, *Personalized medicine* **12**(3), 245–257.

Koch, I. (2013), *Analysis of Multivariate and High-Dimensional Data*, Vol. 32, Cambridge University Press.

Ledoit, O. and Wolf, M. (2004), ‘A well-conditioned estimator for large-dimensional covariance matrices’, *Journal of multivariate analysis* **88**(2), 365–411.

Lee, M.-L. T. (2007), *Analysis of microarray gene expression data*, Springer Science & Business Media.

Li, X. and Xu, R. (2008), *High-dimensional data analysis in cancer research*, Springer Science & Business Media.

Liu, J., Zhong, W. and Li, R. (2015), ‘A selective overview of feature screening for ultrahigh-dimensional data’, *Science China Mathematics* **58**(10), 1–22.

MacQueen, J. et al. (1967), Some methods for classification and analysis of multivariate observations, *in* ‘Proceedings of the fifth Berkeley symposium on mathematical statistics and probability’, Vol. 1, Oakland, CA, USA., pp. 281–297.

McDonald, J. H. (2009), *Handbook of biological statistics*, Vol. 2.

McGregor, C., Nimmo, J. and Stothers, W. (2010), *Fundamentals of university mathematics*, Elsevier.

McLachlan, G. J. and Basford, K. E. (1988), *Mixture models: Inference and applications to clustering*, Vol. 84, Marcel Dekker.

McLachlan, G. and Krishnan, T. (2007), *The EM algorithm and extensions*, Vol. 382, John Wiley & Sons.

McLachlan, G. and Peel, D. (2004), *Finite mixture models*, John Wiley & Sons.

Miller, A. J. (1984), 'Selection of subsets of regression variables', *Journal of the Royal Statistical Society. Series A (General)* pp. 389–425.

Moosa, J. M., Kaykobad, M., Rahman, M. S. and Shakur, R. (2016), 'Gene selection for cancer classification with the help of bees', *BMC medical genomics* **9**(2), 47.

Norman, G. R. and Streiner, D. L. (2008), *Biostatistics: the bare essentials*, PMPH-USA.

Peng, J., Zhu, J., Bergamaschi, A., Han, W., Noh, D.-Y., Pollack, J. R. and Wang, P. (2010), 'Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer', *The annals of applied statistics* **4**(1), 53.

Press, S. J. (2009), *Subjective and objective Bayesian statistics: principles, models, and applications*, Vol. 590, John Wiley & Sons.

Quandt, R. E. (1975), 'Estimating mixtures of normal distributions and switching regressions', *parameters* **2**, 2.

Raftery, A. E. et al. (1996), 'Hypothesis testing and model selection via posterior simulation', *Markov chain Monte Carlo in practice* pp. 163–188.

Rand, W. M. (1971), 'Objective criteria for the evaluation of clustering methods', *Journal of the American Statistical association* **66**(336), 846–850.

Rao, C. R. and Rao, M. B. (1998), *Matrix algebra and its applications to statistics and econometrics*, World Scientific.

Richardson, S. and Green, P. J. (1997), 'On bayesian analysis of mixtures with an unknown number of components (with discussion)', *Journal of the Royal Statistical Society: series B (statistical methodology)* **59**(4), 731–792.

Roeder, K. and Wasserman, L. (1997), 'Practical bayesian density estimation using mixtures of normals', *Journal of the American Statistical Association* **92**(439), 894–902.

Schwab, M. (2008), *Encyclopedia of cancer*, Springer Science & Business Media.

Sekihara, K. and Nagarajan, S. S. (2008), *Adaptive spatial filters for electromagnetic brain imaging*, Springer Science & Business Media.

Simon, N., Friedman, J. and Hastie, T. (2013), 'A blockwise descent algorithm for group-penalized multiresponse and multinomial regression', *arXiv preprint arXiv:1311.6529* .

Stewart, B., Wild, C. P. et al. (2017), 'World cancer report 2014', *Health* .

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 267–288.

Van Veen, B. D., Van Drongelen, W., Yuchtman, M. and Suzuki, A. (1997), 'Localization of brain electrical activity via linearly constrained minimum variance spatial filtering', *IEEE Transactions on biomedical engineering* **44**(9), 867–880.

Vincent, M. and Hansen, N. R. (2014), 'Sparse group lasso and high dimensional multinomial classification', *Computational Statistics & Data Analysis* **71**, 771–786.

Vogel, H. G., Maas, J. and Gebauer, A. (2010), *Drug discovery and evaluation: methods in clinical pharmacology*, Springer Science & Business Media.

Wang, G. C. (1996), 'How to handle multicollinearity in regression modeling', *The Journal of Business Forecasting* **15**(1), 23.

Wang, L., Wang, Y., Hu, Q. and Li, S. (2014), 'Systematic analysis of new drug indications by drug-gene-disease coherent subnetworks', *CPT: pharmacometrics & systems pharmacology* **3**(11), 1–9.

Wang, X., Leng, C. and Dunson, D. B. (2015), On the consistency theory of high dimensional variable screening, *in* 'Advances in Neural Information Processing Systems', pp. 2431–2439.

Xu, Q., Wagstaff, K. L. et al. (2005), Active constrained clustering by examining spectral eigenvectors, *in* 'International Conference on Discovery Science', Springer, pp. 294–307.

Yang, W., Soares, J., Greninger, P., Edelman, E. J., Lightfoot, H., Forbes, S., Bindal, N., Beare, D., Smith, J. A., Thompson, I. R. et al. (2013), 'Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells', *Nucleic acids research* **41**(D1), D955–D961.

Yuan, M., Ekici, A., Lu, Z. and Monteiro, R. (2007), 'Dimension reduction and coefficient estimation in multivariate linear regression', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**(3), 329–346.

Yuan, M. and Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 49–67.

Zhang, J. (2017), 'Screening and clustering of sparse regressions with finite non-gaussian mixtures', *Biometrics* **73**(2), 540–550.

Zhang, J. and Liu, C. (2015), 'On linearly constrained minimum variance beamforming', *Journal of Machine Learning Research* **1**, 1–35.

Zhang, J. and Oftadeh, E. (2016), 'Principal variable analysis: Multivariate variable selection through use of null-beamforming', *Kent Academic Repository* .

Zhong, W. and Zhu, L. (2015), ‘An iterative approach to distance correlation-based sure independence screening’, *Journal of Statistical Computation and Simulation* **85**(11), 2331–2345.

Zhu, L.-P., Li, L., Li, R. and Zhu, L.-X. (2011), ‘Model-free feature screening for ultrahigh-dimensional data’, *Journal of the American Statistical Association* **106**(496), 1464–1475.

Zou, H. and Hastie, T. (2005), ‘Regularization and variable selection via the elastic net’, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(2), 301–320.