Partial Least Squares Regression for High Dimensional and Correlated Data

Mohammed Abdullah A Alshahrani

Department of Statistics

University of Leeds

A thesis submitted for the degree of

Doctor of Philosophy

May 2019

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

This thesis is dedicated to my parents Mr. Abdullah Alshahrani, Mrs. Jamlaa Alshahrani, my wife Mrs. Jamilah, my lovely son Abdullah and all my family members.

Acknowledgements

First of all, appreciation is due to Allah whose guidance and powers have granted me with the health, knowledge and patience to finish this work.

I would like to express my deepest gratitude and appreciation to my supervisor Dr Arief Gusnanto, for his great supervision, continuous support, helpful conversations, advice throughout the research of my PhD, sharing knowledge and expertise. In fact, words cannot express my heartfelt gratitude, appreciation and thanks for all the support, guidance and time he had provided during my stay in Leeds. My gratitude extends to Professor Charles Taylor, for his valuable advice and suggestions during my whole PhD journey. This thesis would not be done without their help and effort.

I would like to express my deepest gratitude and special thanks to my parents, for their continuous love, tolerance, support, praying and the sacrifices they made for me in order to receive my higher education. I am very much thankful to my dear wife, Jamilah Alshahrani and my son Abdullah, for their continuous love, patience, understanding, praying and continuing support that always kept me going. Also, I express my thanks to my brothers and sisters for their assistance and valuable prayers.

Finally, I would like to give my special thanks to all staff in the School of Mathematics for their help, cooperation and constant encouragement during my studies. My deepest appreciation to Mrs. Margaret Jones for her patience dealing with my many enquiries. I would like to thanks Prince Sattam bin Abdulaziz University and for the financial support for my PhD.

Abstract

This thesis focuses on the investigation of partial least squares (PLS) methodology to deal with high-dimensional correlated data. Current developments in technology have enabled experiments to produce data that are characterised by, first, the number of variables that far exceeds the number of observations and, second, variables that are substantially correlated between them. These types of data are common to be found in, first, chemometrics where absorbance levels of chemical samples are recorded across hundreds of wavelengths in a calibration of near-infrared (NIR) spectrometer. Second, they are also common to be found in genomics where copy number alterations (CNA) are recorded across thousands of genomic regions from cancer patients. PLS is a well-known method to employ in the analysis of high-dimensional data as a regression method in chemometric data or as a classification method in genomic data. It deals with those characteristics of the data by constructing latent variables, called components, to represent the original variables. However, there are some challenges in the application of PLS for such analysis and, in this research, there are several areas of investigation that we have performed to deal with them. The first one is that there are three main PLS algorithms with potentially different interpretation of relevant quantities. We deal with this problem by consolidating those three algorithms and identify the case in which those three algorithms would give the same estimates. The second one is the unusual negative shrinkage factors (or "filter factors") that PLS experiences in the model fitting. One of the main reasons PLS can deal with high-dimensional data is that the estimates experience a shrinkage. Unlike ridge regression or principal component regression that experience shrinkage factors between zero and one, PLS can experience shrinkage factors more than one or even negative (hence, more appropriate to be called “filter factors” than “shrinkage factors”). To our knowledge, there has been no previous meaningful investigation on the negative filter factors (NFF) in PLS. In this research we present a novel result whereby we identify the condition for NFF to happen and investigate characteristics of the data that are associated with NFF to get an insight. Lastly, the main challenge of the application of PLS is in the interpretation of weights associated with the predictors. With hundreds and thousands of predictors, each and every predictor variable has non-zero weight. However, we expect that only some predictor variables are contributing to the association with the outcome variable. We therefore resort to the sparse estimation of predictor weights where some weights are zero estimated and the other weights are non-zero. A (standard) lasso estimation has a weakness in dealing with correlated variables as it picks up one variable within a correlation “block” without knowing the reason. A novel approach is needed to take into account the dependencies between predictor variables in estimating the weights. We propose a new method where a new penalty function is introduced in the likelihood function associated with the estimation of weights. The penalty function is a combination of a lasso penalty that imposes sparsity and a penalty based on Cauchy distribution with a smoother matrix to take into account dependencies between genomic regions. The results show that the estimates of the weights are sparse: many weights are zero estimated, and those non-zero estimates are grouped and exhibit smoothness within them. The interpretation on genomic regions becomes easy and identification of important regions for each component can be done simultaneously with prediction in a single modelling framework. We investigate the relation between PLS and graphical modelling using the information in the weights to construct the graph with unsuccessful results.

Abbreviations

MLR	Multiple Linear Regression
SVD	Singular Value Decomposition
OLS	Ordinary Least Squares
RR	Ridge Regression
PCA	principal Component Analysis
PCR	principal Component Regression
PLS	Partial Least Squares
m	A specific component
M	Total of the number of components
q	A column vector
w	A column vector
NIPALS	Nonlinear Iterative Partial Least Squares
PLS1	Univariate (one response) PLS regression
PLS2	Multivariate (many responses) PLS regression
LDA	Linear Discriminant Analysis
QDA	Quadratic Discriminant Analysis
SVM	Support Vector Machines
RMSE	Root Mean Squared Error
RMSEP	Root Mean Squared Error Prediction
RMSEP-LOO	Root Mean Squared Error Prediction
NFF	Negative Filter Factors
SPLS-L1	Sparse Partial Least Squares regression with L_1 penalty
HL	Hierarchical Likelihood
SPLS-HL	Sparse PLS using the second NIPALS algorithm with HL penalty
SPLS2	Sparse PLS using the second NIPALS algorithm with HL penalty
SSPLS	Sparse Smoothed PLS using the first NIPALS algorithm
SSPLS2	Sparse Smoothed PLS using the second NIPALS algorithm
MSECV	Mean Square Error of cross-validation
MERCV	Misclassification Error Rate of cross-validation
MSPE	Mean Squared Prediction Error
RMSPE	Square Root of Mean Squared Prediction Error
RMSPE-LOO	RMSPE with Leave-One-Out cross-validation

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

A common statistical problem is to find a relationship between a set of predictors and a dependent variable which is called regression. When the number of predictors is highly exceeding the number of samples, the classical regression solution cannot be achieved. High-dimensional data can be found in many application areas such as chemometrics, bioinformatics, brain mapping or social science. Genomic data in bioinformatics and spectroscopic data in chemometrics are high-dimensional data because the number of genes (p) in the genomic data exceeds the number of observations (samples) (n), and the number of wavelengths (p) is larger than the number of objects (n) in spectroscopic data. One more feature of these data is that the correlation between the covariates is very large. For examples, copy number alterations (CNA) data has a very large number of covariates (genomic regions), and multi-component spectroscopic (NIR) data has many covariates (wavelengths). These data are high in dimensional such that $p \gg n$ and the covariates are highly-correlated. In this case, classical statistical methods cannot handle these data.

Some methods, which are discussed in Section 1.2.1, called variable selection where these methods aim to select only a subset of the predictors to be included in the regression model. Variable selection has two main approaches which are discrete or shrinkage. Variable selection by shrinkage tries to set some of the coefficients in the model to zero by penalising their magnitude (e.g. Tibshirani (1996)). On the other hand, discrete feature selection involves relevant variables from a set of variables that have an association with response variable. One more group of approaches, which are discussed in Section 1.2.2, called feature extraction or dimension reduction. This approach tries to reduce the high-dimensional (p) genes or wavelengths space to a lower-dimensional space with a dimension called components (m). This m is chosen usually by cross-validation (Geladi & Kowalski, 1986). There are two common methods of this approach which are unsupervised method called principal component regression (PCR) (e.g, Massy (1965)) and the supervised method named partial least squares (PLS) regression developed by (Wold (1966)).

Ordinary least squares (OLS) is an attempt to maximise the correlation between the predictor matrix, X, and the response matrix, Y, while PCR tries to maximise the variances between the predictors in X without taking into account the response matrix in Y. However, PLS regression is a generalisation of OLS and PCR because it tries to maximise the covariance between the predictors, X, and responses, Y, and it can also analyse data with high-collinearity, and numerous X-predictors (Frank & Friedman, 1993). PLS is a multivariate method that can reduce the dimensionality by projecting the original data matrix (X) which is high in dimension to a lower dimension by taking information from both the X and Y matrices. Also, PLS method eliminates the collinearity between the predictors using the lower dimension which has orthogonal components. With this type of data sets and because our interest is in prediction and the interpretation of the coefficients in the model, PLS regression is the optimal method. In spite of that, PLS becomes more reliable for removing the collinearity which OLS method cannot do that. In addition, PLS regression is preferred than PCR when prediction is important (Höskuldsson, 1988).

The original work developing the PLS regression method was done by Wold (1966) in the field of econometrics. PLS method has been used for analyses of high-dimensional data in many research fields including chemometrics and bioinformatics (see Höskuldsson (1988), Worsley (1997) and Hulland (1999)). Moreover, the PLS method has been found to be a useful dimension reduction technique in chemometrics by the groups of (Wold, 1975) and (Wold *et al.*, 1983). Also, in the gene expression data, Nguyen & Rocke (2002) showed how PLS regression is a powerful dimension reduction method for these genomic data. They then used logistic regression or discrimination (LD), linear discriminant analysis (LDA), and quadratic discriminant analysis (QDA) for classifying human tumor samples. PLS is a useful method for prediction especially when the number of predictors in X are huge and the sample size is very small (Höskuldsson, 1988).

With ordinary PLS the number of coefficients is still over thousands in genomic data as CNA data which makes the interpretation of them hard. A sparse solution is needed to select the variables that are more relevant to the response variable. Moreover, NIR and CNA data sets have blocks of correlated covariates which need a method to tackle this problem.

This chapter is organised as follows. In Section 1.2 we review some methods that have been used and developed with high-dimensional and highly-correlated data with normal or binary outcomes. Data sets that are used in this thesis are described in Section 1.3 with a brief discussion about a biological background. In Section 1.4, we provide the motivation and contributions for this thesis. Finally, the structure of the thesis is given in Section 1.5.

1.2 Literature review

There are two main approaches to deal with high-dimensional and highly-correlated data. The first approach is based on variable selection as described in Section 1.2.1, and the second one is based on feature extraction described in Section 1.2.2.

1.2.1 Methods based on variable selection

To deal with high-dimensional data in terms of covariates block, choosing a subset of the variables is one way. It is very important to select the features that are expected to be predictive and significant for the response variable in the model. Some common approaches for feature selection involve discrete feature selection and shrinkage.

Univariate selection is a basic approach for variable selection (feature selection). In this approach, the variables that have a high association with the response variable are ranked based on a score test. There are several tests that have been used in the literature such as t statistic (Hedenfalk *et al.* (2001)) or Wilcoxon's rank sum statistic ((Dettling & Bühlmann, 2003)). Then, we include the top variables in the model depending on their p-values. Using the Bonferroni correction, we can adjust the p-values. But a certain error rate can be achieved such as false discovery rate or family-wise error rate (see (Benjamini & Hochberg, 1995)).

To improve the univariate model described above, we can take into account the correlation between genes by including the genes sequentially in a multivariate model. This type of method is called a forward stepwise selection. forward stepwise selection method starts by having the null model, then adding the genes that have largest score test value. Finally, we continue until we include the m covariates that are highly associated with response variable. Huang *et al.* (2005) compared five statistical methods which are PLS, penalised partial least squares, Lasso, nearest shrunken centroids and random forest using a binary response data for classification problem. Using two real data sets, Huang *et al.* (2005) found that all proposed methods perform similarly.

Discrete variable selection methods may not well the joint effects of multiple covariates which may result in low accuracy of prediction. To overcome such difficulties, another approach of the variable selection is regularisation procedures which can be called shrinkage methods. These methods aim to maximise the penalised log-likelihood with a penalty for a mixed model. Several penalties have proposed in the literature with gene expression data (high-dimensional data) for a classification purpose such as an L_2 penalty leading to ridge regression see (Ghosh, 2003).

With L_1 penalty which yields to Lasso solution in order to regularise log likelihood, a sparse solution can be achieved (e.g. Tibshirani (1996); Huang *et al.* (2005); Kalina (2014)). The problem with the Lasso is that with highly-correlated data, the Lasso will tend to identify one of the features that are associated with the response variable. This can be a desirable problem for interpretation and loosing some of important information from a set of correlated variables. Zou & Hastie (2005) proposed the EN to the linear regression in order to improve the ability of the Lasso to be able to identify sets of correlated genes associated with the response variable. Zou & Zhang (2009) proposed an adaptive EN method with an application on a high-dimensional data. Algamal & Lee (2015) proposed the adjusted adaptive EN for gene selection in high-dimensional data for classifying cancer data. The major drawback of the variable selection methods is the lack of stability (see (Breiman *et al.*, 1996)).

1.2.2 Methods based on feature extraction

Feature extraction is an alternative way which works as searching for combinations of variables, without excluding any of the variables. These methods project the original space p (whole data) to a lower space of dimension m where $m < p$. The most common feature extraction methods are PCR e.g, (Massy, 1965) and PLS e.g, (Lee *et al.*, 2011) and (Fort & Lambert-Lacroix, 2005).

PCR is an unsupervised method where its goal is to find orthogonal linear combinations of the original variables which have high variance. PCR has been used by Chiaromonte & Martinelli (2002) with microarray data for binary classification of leukemia into acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Since it is an unsupervised method, it may be inappropriate for predicting the outcome either a binary or continuous response variable. That is, it might be possible to have the top PCs capture the variability between the variables but they are not associated with response variable.

If prediction is desired, an alternative supervised method is PLS. It has been previously used for linear regression by Wold *et al.* (1993) whereas for classification by many researches such as Nguyen & Rocke (2002), Pérez-Enciso & Tenenhaus (2003), Boulesteix (2004) and Fort & Lambert-Lacroix (2005). For classification, several applications have been explored; for example Nguyen & Rocke (2002) proposed a novel procedure for classifying (predicting) human tumor samples using microarray gene expression data. This procedure involves dimension reduction using PLS and classification using Logistic Discrimination (LD) and Quadratic Discriminant Analysis (QDA). They suggested PLS rather than PCR for prediction purposes. Boulesteix (2004) applied a classical boosting algorithm (AdaBoost) in the framework of PLS dimension reduction.

The latent variables can be extracted using one of several algorithms that can be used for PLS regression such as the kernel algorithm proposed by Höskuldsson (1988), the SIMPLS algorithm De Jong (1995), and the NIPALS algorithm. The algorithm given here is one of the most complete and elegant ones if prediction is important and numerically stable. Also, the NIPALS usually converges in the case of non-convergence where there may be two or more of the eigenvalues are very close to each other (Geladi & Kowalski, 1986).

PLS1 is denoted for PLS with univariate response variable (y), and PLS2 denotes PLS with multivariate response variables (Y). Garthwaite (1994) provided an interpretation of PLS1 and PLS2 of forming prediction equations and it can be better than other methods in prediction.

There are two NIPALS algorithms based on two different views of how the latent variable (X-weights), denoted as (w), is calculated as described in Chapter 2, Sections 2.4.3 and 2.4.4. For PLS2, there are three different versions of the NIPALS algorithms that differ in the normalisation of the latent variables as described in Chapter 2, Section 2.5.

Lingjaerde & Christophersen (2000) and Frank & Friedman (1993) investigated the connection of the shrinkage factors between three shrinkage methods which are ridge regression (RR), PCR and PLS regression for the case $n > p$. The properties of the shrinkage factors of the PLS estimator have been studied in the literature e.g. Lingjaerde & Christophersen (2000), Butler & Denham (2000) and Rosipal & Krämer (2006). The shrinkage factors of the PLS estimator oscillating around 1 while for PCR and RR they are between 0 and 1.

Chun & Keleş (2010) reformulated the sparse PLS (SPLS) criterion by generalising the regression formulation of SPCA using the elastic net penalty of (Zou *et al.*, 2006). In Lee *et al.* (2011), sparse PLS (SPLS) methods with two penalties are proposed. The first SPLS using the L_1 penalty (SPLS-L1), with the same version of the NIPALS algorithm and different penalty hierarchical likelihood (HL), they proposed (SPLS-HL) method. Moreover, based on a different NIPALS algorithm with HL penalty, they proposed second method of SPLS based on the second NIPALS algorithm. Lee *et al.* (2011) argued that SPLS proposed by Chun & Keleş (2010) is a two stage procedure, like the preliminary-test for the estimation of the direction vector. Lee *et al.* (2011) compared their three proposed methods to the SPLS by Chun & Keleş (2010) and the standard PLS, they found that SPLS-HL method outperform than other competitive methods.

SPLS has been widely used for problems with high-dimensional data in bioinformatics field recently using the L_1 penalty such as (Sutton *et al.*, 2018) and (Ajana *et al.*, 2019). Moreover, Colombani *et al.* (2012) compared between the standard PLS and SPLS in genomic selection in French dairy cattle. Using the adaptive Lasso proposed by Zou (2006), Durif *et al.* (2017) proposed a method called adaptive sparse PLS.

However, Lee *et al.* (2011) did not consider the correlation between neighbouring genes or variables. Since the data sets, we are using, are high in dimension and highly-correlated, PLS can deal with high-dimensional data and for variable selection, L_1 penalty is one of the best methods. For data sets like CNA which have blocks of correlated covariates, SPLS methods identify randomly one covariate out of the block. L_1 penalty tends to pick only one variable out of the variables that are associated with the outcome. This is a well known problem of SPLS methods with Lasso solution. Huang *et al.* (2009) considered the smoothed logistic regression model for classification directly. They assumed that the regression coefficients are correlated random effects that follow a mixture of two distributions. They assume one of the mixture distributions is Cauchy to deal with the jumps in the underlying pattern. Gusnanto & Pawitan (2015) employed Cauchy distribution as a random effect model to obtain a sparse solution.

1.3 Data sets

We have been using two data sets, and they are high in dimension and highly-correlated. First, the corn spectroscopic data known as near infrared (NIR) data when the response is on a continuous scale with a univariate normal response and multivariate normal responses. Second, the tumour subtypes of lung cancer data when the response is binary. These are described in detail more with a section about a biological background below.

1.3.1 Corn spectroscopic data (NIR Spectra)

This data is available online on eigenvector.com (Blackburn, 2005). It consists of 80 samples of corn measured on 3 different NIR spectrometers. The wavelength range is 1100-2498 nm at 2 nm intervals (700 channels). Using three near infrared (NIR) spectrometers called “m5”, “mp5”, and “mp6” to measure these predictors or variables (wavelengths) represented by the columns in an X matrix, we obtain correspondingly three predictor matrices called “m5spec”, “mp5spec”, and “mp6spec”, respectively (Fu *et al.*, 2011). The predictors are generally strongly correlated with each other. For example, 93.4% of the variables have correlation coefficients more than 0.92, and 49.4% of the variables have correlation coefficients more than 0.99 for ”m5spec” data set (Fu *et al.*, 2011). The moisture, oil, protein and starch values are the response variables

(Y) for each of the samples is also included. A number of NBS glass standards were also measured on instrument mp5. The data was originally taken at Cargill. Thus, the X matrix has dimension (80×700) and the response matrix is (80×4). This data has been previously used and analysed in Li *et al.* (2009) and Fu *et al.* (2011) with PLS regression method. For sparse PLS this data are used by Lee *et al.* (2011) with two methods of SPLS. In our analysis in this thesis we use “mp5spec” data with the first column in the Y matrix. Thus, we will be dealing with a univariate response variable which is the moisture.

1.3.2 Genomic CNA data

The genome is made up of deoxyribonucleic acid (DNA) which carries the genetic information in all cellular form of life (see (Alberts, 2008)). The DNA in the nucleus is split up into a set of different chromosomes (see (Alberts, 2008)). There are 24 chromosomes in the human genome. There are two copies of DNA in each chromosome normally and each of these copies has two strands of DNA sequences. The base pairs are located in these two strands where A is paired with T, and C with G ((Alberts, 2008)). For more details, the reader is referred to (Shendure & Ji, 2008) and (Alberts, 2008).

DNA sequence technology has been widely used for high-dimensional data collection. Parallel DNA sequencing platforms recently have become available in order to reduce the cost of DNA sequencing. Next Generation Sequencing (NGS) is an incredible platform that can accelerate biological and biomedical research potentially, by utilising different technologies such as SOLiD and 454 sequencing (see (Alberts, 2008)). In order to build a genomic library, the NGS process starts by chopping the DNA into short fragments. After that, the human reference genome is used to map and sequence these fragments which called reads. Finally, these reads are counted per window across the genome. The result, which is a quantitative data set, called read count. Estimating the optimal number of windows can be done using the *NGSoptwin* package proposed by Gusnanto *et al.* (2014).

Seventy-six lung cancer patients had surgery at the Department of Thoracic Surgery at Leeds Teaching Hospitals in Leeds(UK). These patients comprise two groups which are squamous carcinoma for (38 patients), and (38 patients) for adeno carcinoma

(ADC). The clinical characteristics for the patients such as age and gender as covariates can be found in Gusnanto *et al.* (2015) with other information, but they are not used in the analysis of this data in this thesis. We only focus on the CNA genomic regions excluding the clinical characteristics for the patients.

Gusnanto *et al.* (2011) performed a normalisation step using their *CNAnorm* package to obtain the CNA estimates. Using two segmentation methods to estimate the CNA yield two different forms of the CNA estimates. The first estimate called smoothed estimate where CNA is estimated as smooth segmented lines obtained (Huang *et al.* (2007)). The second estimate of CNA is DNACopy where CNA is estimated as circular binary segmented lines (CBS) algorithm (Olshen *et al.* (2004)). We have more than 20000 windows that cover the whole genome (Gusnanto *et al.*, 2015). In the analysis of this thesis, we exclude the sex chromosomes. We also remove the genomic windows that have more than two missing values, and for those windows with less than two missing values, we replace the missing values by the average of that window (column) for all patients. Therefore, we have CNA estimates for 17694 genomic windows from the patients which can be summarised in a matrix denoted by X of size 76 by 17694. The binary tumour histological subtype is the response variable and denoted by y. This response variable is a binary vector where the first 38 take zero values for those patients who have squamous carcinoma and one values for adeno carcinoma (i.e. squamous carcinoma = 1, ADC = 0).

In this thesis we consider the smooth segmented CNA data for the analysis in Chapters 3 and 5. We use both data which are smoothed segmented CNA estimates and DNACopy CNA data in the analysis presented in Chapter 2.

1.4 Motivation and contribution

Predicting the outcome or classifying the subtype of a cancer have been studied widely in the past 10 years. Several methods have been applied to high-dimensional and highly-correlated data sets ($p \gg n$) as indicated in Sections 1.2.1 and 1.2.2, but they have some weaknesses. Although the variable selection method is easy to implement, it is hard to interpret the results, and it does not take advantage of the lower dimensional structure in the data which can result in a worse prediction performance than using PLS Lee *et al.* (2011). On the other hand, PLS regression is a powerful tool that reduces the dimension of the predictors block. PLS use an iterative algorithm called nonlinear iterative partial least square (NIPALS) to calculate the latent variables. There is a misunderstanding between three different versions of NIPALS algorithm.

The shrinkage of the estimator of three common shrinkage methods such as RR, PCR and PLS used for low-dimensional data in the literature by Frank & Friedman (1993), Lingjaerde & Christophersen (2000) towards the OLS solution. When the data is high-dimensional, shrinkage does not exist and so shrinkage factors are replaced by filter factors. However, the filter factors of the PLS estimator may have strange behaviour in that they can take negative values which are referred to as negative filter factors (NFF). Although NFF in the PLS estimator was mentioned by e.g Lingjaerde & Christophersen (2000) and Butler & Denham (2000), it has not been investigated in detail nor the conditions in which it may occur. The structure of the data and the variance-covariance matrix of the data has a relationship with NFF. It is important to find this connection.

Moreover, NIR data has high-correlation between variables and CNA data has dependencies between neighbouring genomic windows. If the feature selection or derived variable methods had been used on these data sets, the dependencies would have been ignored as described for CNA data in (Huang *et al.*, 2009). Therefore, these methods are unsuitable for NIR and CNA data since they ignore the correlation between variables or windows. Furthermore, feature extraction method does not include variable selection which will give a poor prediction if a large number of the irrelevant variables are included in the model.

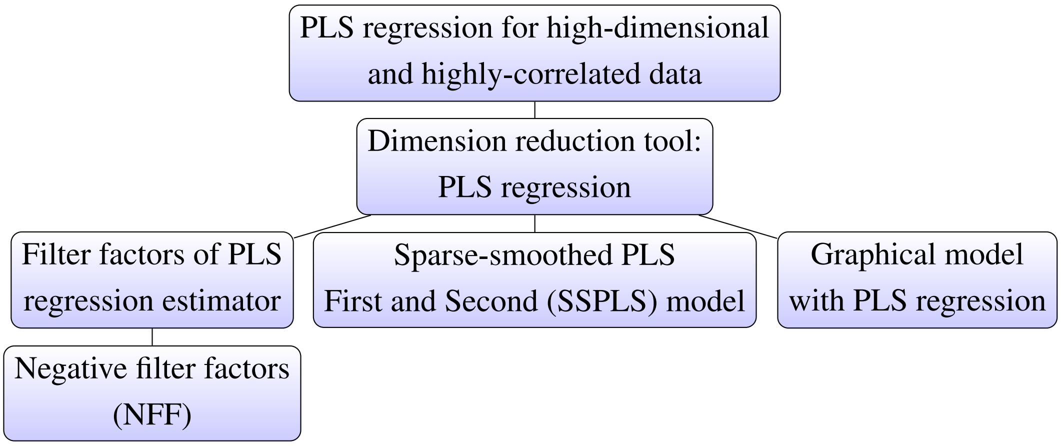
PLS method does not automatically perform variable selection for the relative variables with the outcome because PLS constructs latent variables that are linear combinations of the original covariates. Thus, the performance is expected to be reduced if a large number of covariates are not in fact related to the outcome Lee *et al.* (2011). Sparse PLS (SPLS) methods have been of interest with high-dimensional data in genomics data and spectroscopic data (see Lê Cao *et al.* (2008) Chun & Keleş (2010), Chung & Keles (2010), Fu *et al.* (2011), Lee *et al.* (2011), Lee *et al.* (2013), Sutton *et al.* (2018), Ajana *et al.* (2019)). Since we are dealing with CNA and NIR data where these data has blocks of covariates and applying SPLS cannot handle this type of data because in each block L_1 penalty tends to pick randomly one covariate of the block that is associated with outcome. This problem needs a method that tackle the dependencies between genomic regions or covariates within each block of covariates.

1.4.1 Contribution

Our contribution in this thesis can be summarised in some points as follows. In Chapter 2, we have proved that three different versions of NIPALS algorithm give the same estimator of PLS regression with discussion about "pls2-nipals" function in chemometrics package in R. In Chapter 3, we have investigated the filter factors of three common methods which are RR, PCR and PLS regression methods in the high-dimensional data case and showed that they have a common formula. In Chapter 4, we have investigated and proposed some new conditions on the negative filter factors (NFF) occurrence followed by showing the NFF using two real data sets with normal and binary response variables. We also have investigated some of the cases where NFF occurs with simulation for different structure of the covariance matrix of X. In Chapter 5, we have proposed the sparse-smoothed PLS with first NIPALS algorithm using the penalised likelihood approach. Specifically, we assumed the direction vectors (w) to follow a mixture of two distributions: Cauchy for second-differences of w (to achieve smoothness), and Laplace (to gain sparseness). Also, we have proposed another sparse-smoothed PLS based on the second NIPALS algorithm with the penalised likelihood model. We assumed the same penalty in the sparse-smoothed PLS using the first NIPALS algorithm but the change is only in the conditional likelihood part. Furthermore, the optimal tuning parameters is chosen based on two alternative ways for both methods of SSPLS. We have a local model which is based on w, and a global model based on the estimators of PLS ($\hat{\beta}$). Moreover, we applied these two proposed methods on two different data sets with two different response category (real-valued response as in NIR data) and binary response (CNA data) for classification. Further, we generalised the gradient algorithm that Goeman (2010) proposed by generalising his idea which follows the gradient of the likelihood from a given starting value which uses the full gradient at each step. Finally, in Chapter 6, we investigated the connection between graphical modelling and PLS in order to get some insights of the graph by interpreting the direction vectors w in each component.

1.5 Structure of the thesis

This thesis involves three main aspects of using PLS regression method based on the characteristic of the data which are high in dimension and highly-correlated. The first part comprises Chapters 3 and 4which emphasise on the filter factors of PLS and the negative filter factors (NFF). The second part, to deal with dependences and to provide a sparse solution, we provide sparse-smoothed PLS model as presented in 5. Finally, the PLS is combined with the graphical modelling to interpret each component and how the information in w would correspond to that of the graphical model as described in Chapter 6. A flow chart of this thesis is presented below.


PLS regression for high-dimensional and highly-correlated data
Dimension reduction tool: PLS regression
Filter factors of PLS regression estimator
Sparse-smoothed PLS First and Second (SSPLS) model
Graphical model with PLS regression
Negative filter factors (NFF)

In Chapter 2 we review PLS regression method and PCR for univariate and multivariate response variable(s). We also showed how PLS model is built. We have investigated three different versions of the most common algorithm that has been used by many researchers which is NIPALS algorithm with normalising the loadings of (X and y) or not. We provide a proof of the equivalence of their PLS estimators when the inner regression α is included in the estimator. Moreover, we applied the standard PLS method to two real data sets one with a real-valued response using NIR spectra data, and with a binary response using CNA data for classification. We discussed the variance of the PLS estimator and how it is not linear since it depends on the response variable y, so a numerical approach is used.

In Chapter 4 we have modified the results in Frank & Friedman (1993) and Lingjaerde & Christophersen (2000) for comparing the most common alternative methods to the OLS solution in the literature which are RR, PCR and PLS, for high-dimensional data case. We illustrated the behaviour of the filter factors of the three methods using both data sets. We discussed in detail more the behaviour of the filter factors of the PLS estimator.

In Chapter 3 we considered the behaviour of the filter factors of PLS estimator. Specifically, one of the strange behaviours of the filter factors that some cases negative values occur and that is discussed deeply in Chapter 4. We started by providing an example when the NFF occur from a simulated data set. We proposed conditions for the occurrence of NFF in each component based on the combination of the eigenvalues of two matrices (X^TX and $W_m^TX^TXW_m$), where the direction vectors are the columns in W_m matrix with m components. Furthermore, we presented some reasons of having NFF by simulating some different settings of the eigenvalues of X^TX. Moreover, we have used a small example to investigate NFF based on different structures of variance-covariance matrix of X.

In Chapter 5 we propose a model that deals with high-dimensional and highly-correlated data. We used the first version of NIPALS algorithm by assuming the direction vector (w) is a random effect. We first assume that w is a random effect follows a Cauchy distribution because the data is highly-correlated and it is difficult to interpret the significant variables that are associated with the outcome without smoothness. Then, we added the variable selection to the model by assuming (w) is a random effect model follows a mixture of two distributions (Cauchy to gain smoothness) and (Laplace to gain sparseness). We proposed another SSPLS model based on the second NIPALS algorithm which can be faster than the first one for high-dimensional data (Lee *et al.*, 2011). The first SSPLS method looks alike sparse principal component analysis SPCA since w is trying to find the maximum eigenvalue of $Z_m^TZ_m$, where $Z = Y^TX$ (Lee *et al.*, 2011). The second SSPLS method resembles sparse canonical covariance analysis since w is trying to find the maximum singular value of Z_m (Lee *et al.*, 2011).

Moreover, we optimised the tuning parameters in both methods by two approaches. The first approach is based on the local model where each component is similar to be treated separately which means that each component has the optimal tuning parameter that may not be the same for the following component. The second approach is based on the global model which uses the estimate of the coefficients by PLS estimator (β_{pls}).

In this Chapter, we present a full gradient ascent algorithm for maximising the penalised likelihood by generalising the idea of Goeman (2010). Furthermore, we applied the both models for SSPLS with both scenarios of choosing the optimal tuning parameters on the simulated data, NIR data (where the response is on a continuous scale), and CNA data (where the response is binary).

Furthermore, we compared our proposed methods first and second sparse-smoothed PLS (SSPLS) with local and global models with the first and second sparse PLS with HL penalty presented in Lee *et al.* (2011). We followed the simulation setting of Lee *et al.* (2011).

In Chapter 6 we tried to find one way for interpreting the direction vectors w_m for each component. We used the idea of conditional independence between the predictors and the outcome (response variable). We combined graphical modelling with PLS method in order to interpret w_m in each component by simulating some graphical models from the inverse covariance matrix. We applied PLS method using NIPALS algorithm (first version) on a data that is simulated from the inverse covariance matrix.

Finally, Chapter 7 gives the overall conclusion of the contents of this thesis with future work for some improvements.

Chapter 2

Overview on Partial Least Squares Regression

2.1 Overview

Partial least squares (PLS) is a multivariate method that can reduce the dimensionality by constructing latent variables where each latent variable is a linear combination of the original data matrix (X). Also, PLS method deals with the collinearity between the predictors using the latent variables which are orthogonal.

The genomic data in bioinformatics and the spectroscopic data in chemometrics are high-dimensional data because the number of genes, (p), in the genomic data exceeds the number of observations (samples), (n), and the number of wavelengths, (p), is larger than the number of objects (n) in spectroscopic data. In this situation (i.e. when $n \ll p$), dimension reduction is needed to reduce the high-dimensional (p) genes or wavelengths space to a lower-dimensional space with dimension m. This m is chosen usually by cross-validation (Geladi & Kowalski, 1986). Recently, numerous applications of classification using PLS methods for gene expression data have been done as in Nguyen & Rocke (2002), Boulesteix (2004) and Fort & Lambert-Lacroix (2005).

Recall, PLS1 is denoted for PLS with univariate response variable (y), and PLS2 denotes PLS with multivariate response variables (Y). There are two NIPALS algorithms based on two different views of how the latent variable (X-weights), denoted as (w), is calculated as described in Sections 2.4.3 and 2.4.4. For PLS2, there are three different versions of the NIPALS algorithms that differ in the normalisation of the latent variables as described in Section 2.5. The equivalence between the estimators of β_{pls2} using these three versions of NIPALS algorithms for PLS2 have not been investigated in the literature.

Traditional statistical methodology for prediction such as multiple linear regression (MLR) does not work when there are more variables than samples. For instance, in genomic data the number of genes is much larger than the number of samples. Thus, methods that are able to reduce the dimensionality of the data are necessary. Principal component regression (PCR) and PLS are the most popular methods in this sense, but they are different in the definition of reducing the dimensionality. PCR tries to reduce the dimension in the X block without taking any information from the Y block. However, PLS regression tries to take the information from both blocks.

Although PLS was not designed for classification problems, it has been used by many researchers. PLS can be used for classification as Barker & Rayens (2003) showed the connection between PLS and linear discriminant analysis (LDA). The theoretical connection between the PLS and traditional classification methods such as LDA is described in Section 2.9. The reader is referred for more details to (Barker & Rayens, 2003).

This chapter is organised as follows. Section 2.2 is a brief description of multiple linear regression (MLR) when the response/s is/are univariate and multivariate. Section 2.3 is a brief description of the principal component analysis (PCA) and principal component regression (PCR). Section 2.4 is a brief description of PLS regression for PLS1 and PLS2. Section 2.5 is about three different versions algorithms of the NIPALS algorithms with some investigation and a theoretical proof of their equivalence in terms of regression parameters, $\hat{\beta}_{\text{pls2}}$. Section 2.6 is about the relationship between (the eigenvalue and eigenvector structure) and PLS regression latent variables. Section 2.7 provides some results based on two real data sets with the criteria to choose the optimal number of components (m) to be included in the regression model. In Section 2.8, there is a discussion about how the estimated variance of the estimated regression parameters, $\hat{\beta}_{\text{pls1}}$ can be calculated numerically but not theoretically. Finally, Section 2.9 provides the connection between PLS and linear discriminant analysis (LDA).

2.2 Multiple linear regression (MLR)

From Equation (2.1), one can see that there are three cases need to be discussed. First case, if the number of predictors (p) is larger than the number of observations (n) there is an infinite number of solutions of β by applying some methods to select variables or to reduce the dimension. In the second case, the number of observations (n) equals to the number of predictors (p) there is one unique solution for β by providing that X has full rank. Third case, the number of predictors (p) is smaller than the number of observations, n, one can get a solution for β using "least-squares method". Although the number of predictors, p, is smaller than the number of observations, n, MLR has a problem in the inverse of X^TX is unstable because of collinearity between predictors (Geladi & Kowalski, 1986). For data that has continuous response variable(s) and continuous predictors, the first instance to be considered is linear regression and this can be done using MLR regression model.

2.2.1 MLR with univariate response

MLR attempts to model the relationship between two or more predictors and a response variable by fitting a linear equation to observed data. The model of MLR in the univariate regression (univariate response) is

$$y = X\beta_{\text{ols}} + \epsilon, \tag{2.1}$$

where y is an $(n \times 1)$ vector of response variable, X is an $(n \times p)$ data matrix, β_{ols} is a $(p \times 1)$ vector of parameters, ϵ is an $(n \times 1)$ vector of errors, and randomly distributed with mean and σ^2.

One can get solution for β by minimising the sums of squared errors, ϵ, using the most popular method ”least-squares method” (Draper & Smith, 1981) (Gunst & Mason, 1980) (Mardia *et al.*, 1979). The least-squares solution when $(n > p)$ is

$$\hat{\beta}_{\text{ols}} = (X^TX)^{-1}X^Ty. \tag{2.2}$$

Stone & Brooks (1990) showed how the regression procedure as two steps. A subspace of the original space is defined, then applying the regression on the subspace with the condition that the regression parameter lies in the subspace. The subspace in ordinary least squares (OLS) is defined by the single unit vector that maximises the sample correlation squared between linear combination of the predictor variables ($w^T X$), and the response (y) (Frank & Friedman, 1993) and (Stone & Brooks, 1990).

$$w_{\text{ols}} = \underset{w^T w=1}{\operatorname{argmax}} \quad \operatorname{corr}^2(y, w^T X), \tag{2.3}$$

where y is an $(n \times 1)$ vector of the response variable, X is an $(n \times p)$ data matrix, and w is a $(p \times 1)$ vector that spans the prescribed subspace.

When the number of predictor variables, p, is more than the number of observations, n, in the data matrix, X, the inverse of $X^T X$ does not exist. That is because of collinearity, zero determinant, or singularity are all the reason of non-invertible of the $X^T X$. So, we should use multivariate methods to solve this problem Geladi & Kowalski (1986).

2.2.2 MLR with multivariate response

MLR with more than one response variable or multivariate regression is the case when we have more than one response variable. This can be seen as follows:

$$Y = X\beta_{\text{ols}} + \varepsilon, \tag{2.4}$$

where Y is an $(n \times q)$ matrix of response variables, X is an $(n \times p)$ data matrix, β_{ols} is a $(p \times q)$ matrix of parameters, ε is an $(n \times q)$ matrix of errors, and randomly normally distributed with mean 0 and S_{qq}.

The least-squares solution for this case when $(n > p)$ is

$$\hat{\beta}_{\text{ols}} = (X^T X)^{-1} X^T Y, \tag{2.5}$$

2.3 Principal Component Regression (PCR)

Technically in principal component analysis (PCA), X is decomposed using its singular value decomposition as

$$X = UDV^T,$$

with:

$$U^T U = V^T V = I,$$

where U is the left singular vectors and V is the right singular vectors, and D is being a diagonal matrix with the singular values as diagonal elements. The singular vectors are ordered according to their corresponding singular values which correspond to the square root of the variance of X explained by each singular vector. The columns of U are then used to predict Y using standard regression because the orthogonality of the singular vectors eliminates the multicollinearity problem (Abdi, 2010).

The PCA model is a method that writes the data matrix X as the product matrix of a scores matrix T and a loadings matrix P as in Equation (2.6). The components are calculated using an iteration called NIPALS algorithm Geladi & Kowalski (1986) and Risvik (2007).

$$X = TP^T, \tag{2.6}$$

where X is an $(n \times p)$ data matrix, T is an $(n \times m)$ scores matrix, P^T is an $(m \times p)$ transpose of the loadings matrix, and m is the number of components that should be used. If $m = rank(X)$, then there is no error. Otherwise there is an error in the reduced dimension model. m is equal the minimum of the rank of X, $m \leq \min(n-1, p)$.

In Equation (2.6), T, the successive scores are orthogonal, and P, the successive loadings are orthonormal. These components can be extracted using an iteration algorithm. Here we are going to use the NIPALS algorithm (see appendix A for the algorithm) (Wold *et al.*, 2001).

The principal components here are trying to maximise the variances in X, without taking into account the response variable. The component m can be written as follows, where $m = 1, 2, \ldots, M$, and $M \leq \min(n-1, p)$.

$$\delta_m = t_m p_m^T \tag{2.7}$$

The principal component regression (PCR) uses the X matrix (predictors) without taking into account the (univariate) response variable (y). The results from PCA, which are the scores T and the loadings P, can be used to explain the principal component transformation of the data matrix X. This is a representation of X as its scores matrix T with a lower dimension, $m < p$ as shown in Equation (2.8).

$$y = TP^T\beta_{\text{pcr}} + \epsilon, \tag{2.8}$$

where ϵ is the error term with mean 0 and covariance $\Sigma = \sigma^2 I_n$ and I_n is the identity matrix, and the loadings p_m are chosen to maximise the variance in X as

$$p_{\text{pcr}_m} = \underset{p_m^T p_m = 1}{\text{argmax}} \quad \text{var}(p_m^T X), \qquad p_m \perp p_h, m > h \tag{2.9}$$

Since P are orthogonal, the transformation of Equation (2.6) is Geladi & Kowalski (1986)

$$T = XP (= TP^T P = TI_n),$$

where P is a matrix of X-loadings that maximise the variance between the predictors, and $p_{m_{\text{pcr}}}$ is the X-loading for one component, subject to $||p_m^T p_m|| = 1, \forall m$, and $< p_m, p_h >= 0$ for $m > h$.

So now the univariate regression formula, Equation (2.8) can be written as

$$y = XP\beta_{\text{pcr}} + \epsilon. \tag{2.10}$$

The least-squares solution for this case univariate regression (univariate response) is

$$\hat{\beta}_{\text{pcr}} = (T^T T)^{-1} T^T y. \tag{2.11}$$

The same procedures will be used for multivariate case with multivariate responses. The multivariate regression formula, Equation (2.10) can be written in terms of the new few components, X-scores, T as

$$Y = T\beta + \varepsilon. \tag{2.12}$$

where ε is the error term with mean 0 and covariance Σ_{nn}. The least-squares solution for this case multivariate regression (multivariate responses) is

$$\hat{\beta}_{\text{pcr}} = (T^T T)^{-1} T^T Y. \tag{2.13}$$

The variables of X are replaced by new ones that are orthogonal and also span the space of X. The inversion of $T^T T$ should not be a problem anymore because of the successive scores are orthogonal. To avoid the collinearity problems from influencing the solution, score vectors corresponding to small eigenvalues should not be used in the new model in Equation (2.10). PCR handles the collinearity problem and produce an invertible matrix in the estimation of β_{pcr} (Geladi & Kowalski, 1986).

Although PCR solves the collinearity problem and produce an invertible matrix in the estimation of the regression parameters, the problem of choosing an optimum subset of predictors remains. A possible strategy is to keep only a few of the first components. But these components are chosen to explain X rather than Y, and so, nothing guarantees that the principal components, which explain X, are relevant for Y (Abdi, 2010).

2.4 Partial least squares regression (PLS)

Partial least squares regression (PLS) was constructed to solve the multicollinearity problem in the regression model. The original work in the PLS method was introduced by Wold (1966) in the field of econometrics. Then, the use of PLS method has been found to be a useful dimension reduction technique in the chemometrics field by Wold *et al.* (1983) with a new term for PLS which is "projection to latent structures" (Abdi, 2010). PLS method is used to find a linear relation between predictors (X) and the responses (Y). PLS model is trying to find the maximum covariance between the predictors and the responses (Stone & Brooks, 1990). Thus, PLS is preferred to PCR if prediction is the goal of the analysis (Chin & Frye, 2003). By looking at the second step of PLS algorithm, we can see that PLS is equivalent to the conjugate gradient algorithm of forming an inverse of X^TX (Wold *et al.*, 1984).

PLS is mostly used in chemometrics and related fields. Recently, it has been used in bioinformatics and biology. PLS1 is used when the response variable is univariate for PLS and PLS2 is used when the response variables are multivariate (Garthwaite, 1994).

PLS model introduces X-weights to get orthogonal X-scores, T as can be seen in Equation (2.14).

$$X = TW^T, \tag{2.14}$$

where X is an $(n \times p)$ data matrix, T is an $(n \times m)$ scores matrix, W^T is an $(m \times p)$ transpose of the weights matrix, and m is the number of components that should be used. If all components are used $m = rank(X)$, then there is nothing left over. Otherwise there is an error in the reduced dimension model, where m is equal the minimum of the rank of X ($m \leq \min(n-1, p)$).

2.4.1 PLS with univariate response (PLS1)

PLS1 regression is the PLS regression when the response is univariate. PLS1 can be done by using an iterative algorithm called NIPALS. This algorithm is used to calculate the scores, loadings, and weights for X and y, and the parameters. Then, the parameters are used to calculate the predicted values in the future. The PLS regression model can be written as follows:

$$y = T\beta_{\text{pls1}} + \epsilon, \tag{2.15}$$

where ϵ is the error term with mean 0 and covariance $\Sigma = \sigma^2 I_n$ and I_n is the identity matrix, and the direction vectors w_m are chosen to maximise the covariance between X and y as

$$w_{\text{pls}_m} = \underset{w_m^T w_m = 1}{\text{argmax}} \quad \text{corr}^2(y, w_m^T X)\text{var}(w_m^T X), \qquad p_m \perp p_h, m > h. \tag{2.16}$$

Since the X-weights, W, are orthogonal, the transformation of Equation (2.14) is shown in Equation (2.17), where W is a matrix of X-weights that maximise the covariance between the response and the X-score, t_m, $\text{cov}(y, w_m^T X)$, and the direction vectors w_m is the X-weight for one component for $m = 1, \ldots, M$, subject to $||w_m^T w_m|| = 1, \forall m$, and $< w_m, w_h >= 0$ for $m > h$.

2.4.2 PLS1 regression model building

The variables in X are replaced by new ones that have better properties (orthogonality), and also span the space of X. The PLS model is built on properties of the NIPALS algorithm.

The general form for any multiple linear model with a univariate response is in Equation (2.2). The linear PLS model finds a few “new” variables, which estimate the latent variables on their rotations. These new variables are called X-scores and denoted by (t_m). The X-scores are predictors of y and can model X as in Equation (2.6). They are estimated as linear combinations of the original variables in X with the weights (w_m) where m is the number of components that should be used in the model, $(M \leq \min(n-1, p))$. Equation (2.17) shows the estimation of the scores matrix (T) which can represent the original data matrix X as follows Andersson (2009).

$$T = XW, \tag{2.17}$$

where T is an $(n \times m)$ scores matrix of X, X is an $(n \times p)$ data matrix, W is an $(m \times p)$ matrix of X-weights, and m is the number of components that should be used in the model, $(m = \min(n-1, p))$.

The X-scores (T) are good predictors of y, i.e.:

$$y = Tq^T + \epsilon, \tag{2.18}$$

where y is an $(n \times 1)$ a response variable, T is an $(n \times m)$ scores matrix of X, q^T is an $(m \times 1)$ transpose of the vector of the loadings of y where $(m < p)$ is the number of components that should be used in the model, and ϵ is the error term. Since each t_m is a linear combination of the X, and the scores, T, are uncorrelated, the residuals are uncorrelated. Thus, the regression parameters can be estimated using ordinary least squares (OLS) Helland (1988) and Wold *et al.* (1984).

By combining Equation (2.17), and Equation (2.18), it can be seen that

$$y = XWq^T + \epsilon. \tag{2.19}$$

So, the parameters of the above model is β_{pls1} which can be written as:

$$\beta_{\text{pls1}} = Wq^T. \tag{2.20}$$

The residual vector, ϵ, in Equations (2.1), (2.15), (2.18), and Equation (2.19) are the same, even though the models are written in different ways. PLS1 is a least squares method for minimising $\epsilon^T\epsilon$, given the characteristic structure of the vectors in W. We can solve Equation (2.18) for q, as solving Equation (2.15) for β. The solution for q in Equation (2.18) is Andersson (2009).

$$q^T = (T^T T)^{-1} T^T y. \tag{2.21}$$

By combining Equation (2.17), Equation (2.20), and Equation (2.21), one can write parameters vector in terms of the X-weights, W Helland (2001). Thus, we have

$$\beta_{\text{pls1}} = W(W^T X^T X W)^{-1} W^T X^T y. \tag{2.22}$$

To calculate the regression parameters, we can use the outputs from the NIPALS algorithm see Section 2.4.3. Using Equation (2.22) with the output matrices from the NIPALS algorithm, with their orthogonality properties and the relations between them, and Because $P = X^T T$, and $T = XW$, we can substitute $W^T X^T X$ by P^T, and $q^T = W^T X^T y$. Therefore, the regression parameters vector in Equation (2.22) can be written in a totally different-looking but equivalent as Helland (2001).

$$\hat{\beta}_{\text{pls1}} = W(P^T W)^{-1} q^T \tag{2.23}$$

Given that $Z = Y^T X$ for multivariate responses, the optimisation problem Equation (2.16) can be viewed in two different ways Lee *et al.* (2011). The first view which is similar to PCA where we try to find the eigenvector w that corresponding to the maximum eigenvalue of $Z_m^T Z_m$. Whereas the second view is similar to CCA where we try to find a right singular vector w corresponding to the maximum singular value of Z_m. Therefore, we have two different versions of NIPALS algorithm for PLS regression because of these two views. However, both algorithms give the same results for w when ordinary PLS regression is applied. Both versions of NIPALS algorithm calculate the parameters from the output of the iteration as in Equation (2.23) after calculating the W matrix in both NIPALS algorithms Lee *et al.* (2011). Although the response is univariate, we still have two different NIPALS algorithms based on the above two views.

2.4.3 First NIPALS algorithm for PLS1

This version of NIPALS algorithm is suitable for a univariate response as given in Lee *et al.* (2011).
X is an $(n \times p)$ data matrix, y is an $(n \times 1)$ vector of response variable, X and y are mean centred and scaled.
Initialisation: Set $X_1 = X$, $z = y^T X$, t_1 is the first column of X, and m=1.

Algorithm 1 The first NIPALS algorithm for PLS1

```
$w_m = z_m^T y_m^T t_m / (t_m^T y_m y_m^T t_m)$.
$w_m = w_m / \sqrt{w_m^T w_m}$        (normalisation)
$t_m = X_m w_m$
$X$-loadings: $p_m = X_m^T t / (t_m^T t_m)$
$y$-loadings: $q_m = y_m^T t_m / (t_m^T t_m)$
$X$-scores: $t_m = X_m w_m$
$X$-weights: $w_m = X_m y_m$
Update $X_{m+1} = X_m - (t_m p_m^T)$
Update $y_{m+1} = y_m - (t_m q_m^T)$
Update $z_{m+1} = y_m^T X_m$
```

The regression parameter: $\hat{\beta}_{\text{pls1}} = WM(P_M^T W_M)^{-1} q_M$, where q_M is a column vector.

After first component is calculated, X and y have to be replaced by their residuals.

2.4.4 Second NIPALS algorithm for PLS1

NIPALS algorithm is suitable for a univariate response as given in Wold *et al.* (1983). Also, the X-weight vectors, w_m form an orthonormal set, and the X-score vectors, t_m are orthogonal to each other, where m is the number of components that should be used in the model, $m = 1, 2, \ldots, M$, and $M \leq \min(n-1, p)$.

Initialisation: Set $X_1 = X$, and m=1.

Algorithm 2 The Second NIPALS algorithm for PLS1

```
$w_m = X_m^T y_m/(y_m^T y_m)$
$w_m = w_m/\sqrt{w_m^T w_m}$        (normalisation)
$t_m = X_m w_m$
$X$-loadings: $p_m = X_m^T t/(t_m^T t_m)$
$y$-loadings: $q_m = y_m^T t/(t_m^T t_m)$
$X$-scores: $t_m = X_m w$
$X$-weights: $w = X_m y_m$
Update $X_{m+1} = X_m - (t_m p_m^T)$
Update $y_{m+1} = y_m - (t_m q_m^T)$
```

The regression parameter: $\hat{\beta}_{\text{pls1}} = W_M(P_M^T W_M)^{-1} q_M$, where q_M is a column vector.

After first component is calculated, X and y have to be replaced by their residuals. Note that, in the first version w is the eigenvector corresponding to the largest eigenvalue of $Z^T Z$ while in the second version of NIPALS algorithm, the w is the right singular vector corresponding to the largest singular value of Z. That means the first version uses the eigenvalue-vector solution while the second version uses the singular value decomposition. However, both give the same solution for w hence, the same estimator of $\hat{\beta}$.

2.4.5 PLS with multivariate responses (PLS2)

PLS2 regression is a generalisation of MLR and PLS1. PLS2 can handle multivariate responses or univariate response unlike PLS1 which can handle only univariate response. The NIPALS algorithm is used in PLS2 as well. In PLS2, we will have Y-scores, U. There are three versions of the NIPALS algorithm of PLS2 case, and they are discussed in details below in Section 2.5 with a theoretical proof of their equivalent in the estimated regression parameters, $\hat{\beta}_{\text{pls2}}$.

2.4.6 PLS2 regression model building

The variables in X are replaced by new ones that have better properties (orthogonality), and also span the space of X. The PLS model is built on properties of the NIPALS algorithm. The general form for any linear model with multivariate responses, MLR is in Equation (2.4).

PLS decomposes the X and Y matrices into the form Geladi & Kowalski (1986).

$$\begin{aligned} X &= TP^T + F \\ Y &= UQ^T + \varepsilon. \end{aligned} \tag{2.24}$$

The $(n \times p)$ matrix X is the data matrix, and the $(n \times q)$ matrix Y is the responses matrix, and m is the number of components that should be used in the model, $(M \leq \min(n-1, p))$. T and U are the $(n \times m)$ scores of X and Y respectively. P is an $(m \times p)$ loadings matrix of X, and Q is an $(m \times q)$ loadings matrix of Y. ε is an $(n \times q)$ matrix of the X residuals, and F is an $(n \times p)$ matrix of the Y residuals.

The main interest is to describe Y as well as is possible to make ε as small as possible and at the same time, get a useful relation between X and Y. The inner relation can be made by regressing the Y block score, u, on the X block score, t, for every component. The simplest model for this relation is a linear one as in Equation (2.25):

$$u = \alpha t, \tag{2.25}$$

where α will be the inner relation, which is a regression of u on t with no intercept. However, this model is not the best possible because the components are calculated separately, so they have a weak relation to each other. It would be good to give them information about each other by swapping the scores of each other Geladi & Kowalski (1986).

$$T = XW^*A, \tag{2.26}$$

where A is a diagonal matrix of the inner relation (α) with dimension $(m \times m)$. As a result of that, using Equation (2.26), the mixed relation can be given in Equation (2.27):

$$Y = XW^*AQ^T + \varepsilon = Y = X\beta_{\text{pls2}} + \varepsilon. \tag{2.27}$$

Using the output matrices from the NIPALS algorithm for PLS2 in Appendix A, W^* is computed as

$$W^* = W(P^TW)^{-1}.$$

Therefore, the estimated regression parameters matrix can be written as

$$\hat{\beta}_{\text{pls2}} = W(P^TW)^{-1}AQ^T, \tag{2.28}$$

where $(M \leq \min(n-1, p))$ is the number of components that should be used in the model. Note that A is the identity matrix in the first version of NIPALS algorithm for PLS2 because of the X-loadings, Q is not normalised. However, A in the second and third versions of NIPALS algorithms is not the identity matrix. Thus, in the first version some researchers would write the parameters estimation solution as $\beta_{\text{pls2}} = W(P^TW)^{-1}Q^T$.

The need for the inner regression comes from the normalisation of q_m to $||q_m|| = 1, \forall m$ as used in the second algorithm Höskuldsson (1988) and third algorithm Geladi & Kowalski (1986). On the other hand, we believe that Sjöström *et al.* (1983) made an error not using the inner regression in the prediction step. These authors make still another choice of normalisation Manne (1987). In short, either normalisation for the Y-loadings (Q) or not, we should use the inner regression at the prediction stage. Any version of the NIPALS algorithm in Appendix A calculates the parameters from the output of the iteration as in Equation (2.28).

2.5 Investigating all three algorithms of PLS2

We have investigated three versions of the NIPALS algorithm for PLS2. The first algorithm does not normalise the loadings for X and Y as given in Wold *et al.* (1984), so the inner regression between the X-score vector, t_m, and the Y-score vector, u_m for each component equals one. The second algorithm normalises only Y-loadings, Q as given in Höskuldsson (1988), (i.e. they are scaled to have unit length, $||q|| = 1$), so the inner regression will not be one as in the first algorithm (simple version of NIPALS algorithm). The third algorithm normalises X-loadings and Y-loadings as given in Geladi & Kowalski (1986). We found that all three algorithms estimate the regression parameters in the same way after including the inner regression, α.

2.5.1 Theoretical proof of equivalence of the parameter estimations

In the first algorithm X-loadings, P and Y-loadings, Q are not normalised as given in Wold *et al.* (1984). In the second algorithm X-loadings, P is normalised, but Y-loadings, Q is not as given in Höskuldsson (1988). In the third algorithm X-loadings,

P and Y-loadings, Q are normalised as given in Geladi & Kowalski (1986). We first show that $\beta^{(1)}_{\text{pls2}} = \beta^{(2)}_{\text{pls2}}$

$$\beta^{(1)}_{\text{pls2}} = W^{(1)}(P^{(1)^T}W^{(1)})^{-1}A^{(1)}Q^{(1)^T},$$

and

$$\beta^{(2)}_{\text{pls2}} = W^{(2)}(P^{(2)^T}W^{(2)})^{-1}A^{(2)}Q^{(2)^T}.$$

From the NIPALS algorithms in appendix A, we can see that $W^{(1)} = W^{(2)}$, which means that the W matrix in the first and the second algorithms are the same. Also, $P^{(1)} = P^{(2)}$, which means that the P matrix in the first and the second algorithms are the same. However, $A^{(1)} \neq A^{(2)}$, which means that the A matrix in the first and the second algorithms are not the same. And, $Q^{(1)} \neq Q^{(2)}$, which means that the Q matrix in the first and the second algorithms are not the same. Thus, we need to show that Equation (2.29) is true for one component, then we can prove that $\beta^{(1)}_{\text{pls2}} = \beta^{(2)}_{\text{pls2}}$. (see the proof in appendix B).

$$\alpha^{(1)}q^{(1)^T} = \alpha^{(2)}q^{(2)^T} \tag{2.29}$$

Nevertheless, $\beta^{(1)}_{\text{pls2}} = \beta^{(2)}_{\text{pls2}}$ since $\alpha^{(1)}q^{T^{(1)}} = \alpha^{(2)}q^{T^{(2)}}$ as shown in Equation (2.29) for one component and for m components. We now show that $\beta^{(2)}_{\text{pls2}} = \beta^{(3)}_{\text{pls2}}$

$$\beta^{(2)}_{\text{pls2}} = W^{(2)}(P^{(2)^T}W^{(2)})^{-1}A^{(2)}Q^{(2)^T},$$

and

$$\beta^{(3)}_{\text{pls2}} = W^{(3)}(P^{(3)^T}W^{(3)})^{-1}A^{(3)}Q^{(3)^T}.$$

From the NIPALS algorithms in appendix A, we can see that $W^{(2)} \neq W^{(3)}$, which means that the W matrix in the second and the third algorithms are not the same. Also, $P^{(2)} \neq P^{(3)}$, which means that the P matrix in the second and the third algorithms are not the same. However, $A^{(2)} \neq A^{(3)}$, which means that the A matrix in the second and the third algorithms are not the same. And, $Q^{(2)} = Q^{(3)}$, which means that the Q matrix in the second and the third algorithm are the same. Thus, if we can show that $w^{(2)}(p^{(2)^T}w^{(2)})^{-1}\alpha^{(2)} = w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)}$ for one component, we can prove that $\beta^{(2)}_{\text{pls2}} = \beta^{(3)}_{\text{pls2}}$. (see the proof in appendix B).

To show that $w^{(2)}(p^{(2)^T}w^{(2)})^{-1}\alpha^{(2)} = w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)}$, we need to write $w^{(3)}$ in terms of $w^{(2)}$, and the same for $p^{(3)}$ and $\alpha^{(3)}$ for one component.
After some substitution and cancelation, we have:

$$w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)} = w^{(2)}\sqrt{(X^Tu^{(2)})^T(X^Tu^{(2)})}\sqrt{p^{(2)^T}p^{(2)}} \\ \left(\frac{p^{(2)^T}}{\sqrt{p^{(2)^T}p^{(2)}}}w^{(2)}\sqrt{(X^Tu^{(2)})^T(X^Tu^{(2)})}\sqrt{p^{(2)^T}p^{(2)}}\right)^{-1}\frac{\alpha^{(2)}}{\sqrt{p^{(2)^T}p^{(2)}}}. \tag{2.30}$$

Some terms will be canceled out. Therefore, we will have that:

$$w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)} = w^{(2)}(p^{(2)^T}w^{(2)})^{-1}\alpha^{(2)}. \tag{2.31}$$

Nonetheless, $\beta^{(2)}_{\text{pls2}} = \beta^{(3)}_{\text{pls2}}$ since $w^{(2)}(p^{(2)^T}w^{(2)})^{-1}\alpha^{(2)} = w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)}$ as shown in Equation (2.31) for one component and for m components. Since $\beta^{(1)}_{\text{pls2}} = \beta^{(2)}_{\text{pls2}}$ and $\beta^{(2)}_{\text{pls2}} = \beta^{(3)}_{\text{pls2}}$, then $\beta^{(1)}_{\text{pls2}} = \beta^{(3)}_{\text{pls2}}$. Therefore, β_{pls2} for all three NIPALS algorithms for multivariate responses are the same even if they are different in terms of normalisations or not.

2.5.2 Discussion about (pls2-nipals) function in Chemometrics Package in R

There is a function "pls2-nipals" in chemometrics package in R. This function is similar to the second version of NIPALS algorithm for PLS2 6, which normalises the Y-loadings (Q). This function is different to the second version of NIPALS algorithm for PLS2 6 in two ways. First, this function uses one more term which is called the Y-weights (C). Second, this function calculates the Y-loadings after convergence using Q. In contrast, the second method uses only one term instead of two terms, namely the Y-loadings (Q). This Y-loadings, Q is similar and equal to the Y-weights (C), in the "pls2-nipals" function. The Y-loadings (Q), in the "pls2-nipals" function is not used to calculate the regression parameters, $\hat{\beta}$. By looking at the inner regression in the second version of NIPALS algorithm for PLS2 6 and in the "pls2-nipals" function, we found that they are the same. However, the way that "pls2-nipals" calculates $\hat{\beta}$ does not take into account the inner regression. That could be right if the inner regression is equal to one, which is true in the first version of the NIPALS algorithm for PLS2 5 without normalising the Y-loadings (Q) or Y-weights (C), as written in "pls2-nipals" function. In short, it would be great when the Y-loadings (Q) or Y-weights (C) are normalised to consider and take into account the inner regression between the Y-scores (U) and the X-scores (T) in the calculation of the parameters estimation.

2.6 NIPALS and Eigenvalues/Eigenvectors Equations

2.6.1 NIPALS and Eigenvalues/Eigenvectors Equations in PLS

Regarding the vectors as determined only up to length allows the NIPALS loop to be replaced by an Eigen problem. The relation between vector q at step m, (q_m) with that at step $m-1$, (q_{m-1}) can be written as (see Appendix A for the NIPALS algorithm steps). We consider here the first version of the NIPALS algorithm of PLS2 Höskuldsson (1988):

$$\begin{aligned}
q_m &= Y^T t_m / t_m^T t_m \quad \text{from (step 5)}, \\
&= Y^T X w_{m-1} / [(t_m^T t_m)(w_{m-1}^T w_{m-1})] \quad \text{from (step 4)}, \\
&= Y^T X X^T u_{m-1} / [(t_m^T t_m)(w_{m-1}^T w_{m-1})(u_{m-1}^T u_{m-1})] \quad \text{from (step 2)}, \\
&= Y^T X X^T Y q_{m-1} / [(t_m^T t_m)(w_{m-1}^T w_{m-1})(u_{m-1}^T u_{m-1})(q_{m-1}^T q_{m-1})] \quad \text{from (step 7)}.
\end{aligned}$$

After convergence, we write:

$$Y^T X X^T Y q_1 = \lambda_q q_1. \tag{2.32}$$

This is the eigenvalue-eigenvector equation as used in the classical calculation. Where, λ_q is the largest eigenvalue of $Y^T X X^T Y$, and q_1 is the eigenvector corresponding to the largest eigenvalue, λ_q.

Similarly, we can do the same way to get the rest vectors, which are u_1, w_1, t_1 as:

$$Y Y^T X X^T u_1 = \lambda_u u_1,$$

$$X^T Y Y^T X w_1 = \lambda_w w_1,$$

$$X X^T Y Y^T t_1 = \lambda_t t_1,$$

where λ_u is the largest eigenvalue of YY^TXX^T, and u_1 is the eigenvector corresponding to the largest eigenvalue, λ_u. Also, λ_w is the largest eigenvalue of X^TYY^TX, and w_1 is the eigenvector corresponding to the largest eigenvalue, λ_w. Again, λ_t is the largest eigenvalue of XX^TYY^T, and t_1 is the eigenvector corresponding to the largest eigenvalue, λ_t. Hence, the latent components can be derived from the eigenvalue-eigenvector equation (Wold *et al.*, 2001).

2.7 Results: real data sets

Since the structure of the data sets described earlier in Section 1.3 are high in dimension and highly-correlated, we have applied the PLS method for analysing these data sets. We applied the ordinary PLS1 using the second NIPALS algorithm 2. For PLS2, we used all three versions of the NIPALS algorithm for PLS2 (5, 6 and 7 presented in appendix A) when we have multivariate responses as in spectra NIR data. For DNA-Copy CNA and smooth CNA data sets, we have applied the ordinary PLS1 where the response is univariate using the second NIPALS algorithm 2. For the NIR data, the predicted values is done by splitting the data randomly into two groups; a training set (40 samples) and a validation set (40 samples). Using the training set with 8 components in the second NIPALS algorithm then using the estimation of $\hat{\beta}$ in Figure 2.3 with the validation set to get the predicted values. For CNA data, the predicted class is done after splitting the data randomly in two groups: training data (38 samples) and validation data (38 samples). Applying the PLS1 on the training set to get the estimation of $\hat{\beta}$ in Figures 2.7 and 2.8 with validation set of the predictors to have the predicted values. Then, we classify the new samples based on their values if it is greater than zero squamous type, otherwise, class ADC.

2.7.1 Number of components selection

If the relation between X and Y is a linear model, the number of components that should be used in the PLS regression model is equal to the dimension of the model. All components should not be used even though it is possible to calculate as many PLS components as the rank of X matrix. This is because of the collinearity that components with small eigenvalues might bring collinearity to the regression model.

Thus, it is suggested that components with small eigenvalues should not be used to avoid the collinearity Geladi & Kowalski (1986). The number of components ($M \leq \min(n-1, p)$).

Nevertheless, there are several methods to decide when to stop choosing the components for the model. One possible criterion is cross-validation to decide the number of components that needed in the model. Stone & Brooks (1990) and Wold *et al.* (1984) showed examples of deciding the number of components to be included in the regression model. One is called the mean squared prediction error (MSPE) when the responses are normal which can be calculated as follows:

$$\text{MSPE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \frac{1}{n}\sum_{i=1}^{n} e_i^2. \tag{2.33}$$

We have used the MSPE with five-fold cross-validation to choose the optimal number of components as

$$\text{MSPECV}(m) = \frac{1}{n}\sum_{f=1}^{5} ||y_{[s]} - X_{[s]}\hat{\beta}_m^{-s}||^2. \tag{2.34}$$

Where $\hat{\beta}_m^{-s}$ is the coefficient estimates using the m number of components from the s-th training sets. The optimal m is chosen that correspond to the minimum value of the MSPECV.

On the other hand, when the response is binary, one can calculate the misclassification error rate using five-fold cross-validation (MERCV). The classification is done by classifying the sample to class Squamous carcinoma if the predicted value is greater than zero and to adeno carcinoma if the predicted value is less than zero. We use zero value as a classifier value because the both X and y are centred.

Figure 2.1 shows the MSPECV using five-fold cross-validation for NIR data with a univariate real-valued response (moisture).

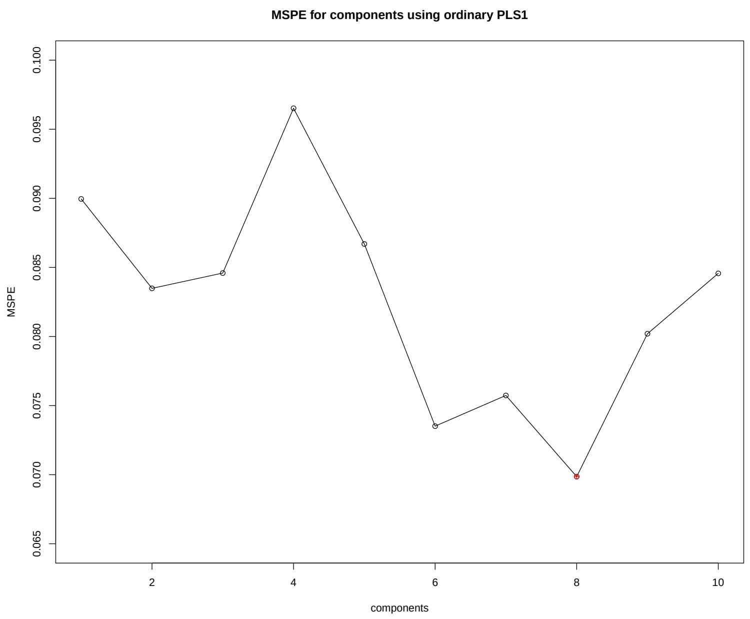
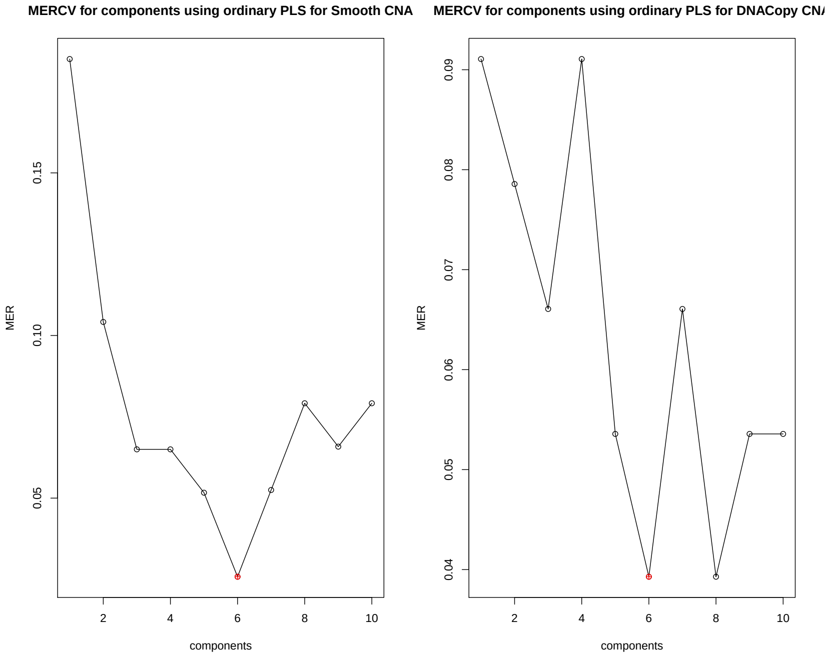


Figure 2.1: MSPECV with five-fold cross-validation for PLS1 using NIR data with a univariate normal response, red point is the optimal number of components (m_{opt} =8).

It can be seen from Figure 2.1 that the optimal number of components is 8 components for this data with only one response variable. This is based on five-fold cross-validation with MSPECV as in Equation (2.34).

Figure 2.2 shows the misclassification error using five-fold cross-validation for PLS1 using lung cancer for DNACopy CNA and smooth CNA data with a binary response.

Figure 2.2: MERCV for PLS1 using lung cancer. Left panel using DNACopy CNA and right panel using smooth CNA. The optimal number of components is coloured by red and for both data is 6 components (m_{opt}=6).



From Figure 2.2, we can see that using DNACopy CNA data (left panel) the optimal number of component using MERCV with five-fold is 6 components. Also, using the smooth CNA data (right panel) the optimal number with the same measurement is 6 components.

2.7.2 NIR data

Figure 2.3 shows the estimated parameters, $\hat{\beta}_{\text{pls1}}$, for PLS1 using normal response.

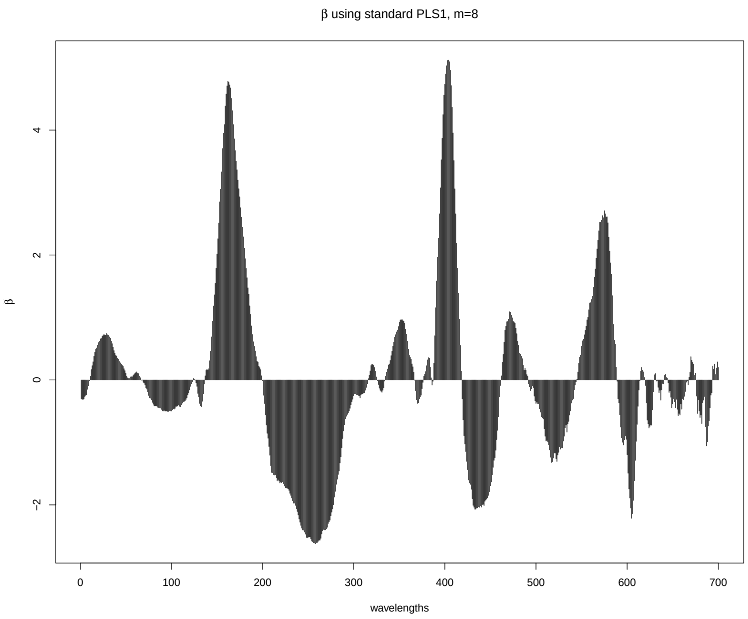


Figure 2.3: $\hat{\beta}_{\text{pls1}}$ from PLS1 using NIR data with optimal number of components $m_{opt} = 8$.

It can be seen from Figure 2.3 that the estimation of the coefficients which is based on the optimal number of components in the model. Positive values of the estimation indicate that wavelengths are positively affecting the response variable (moisture). In contrast, negative values indicate the wavelengths have a negative impact on moisture.

Figure 2.4 shows the estimate of w_1 and w_2, for PLS1 using normal response.

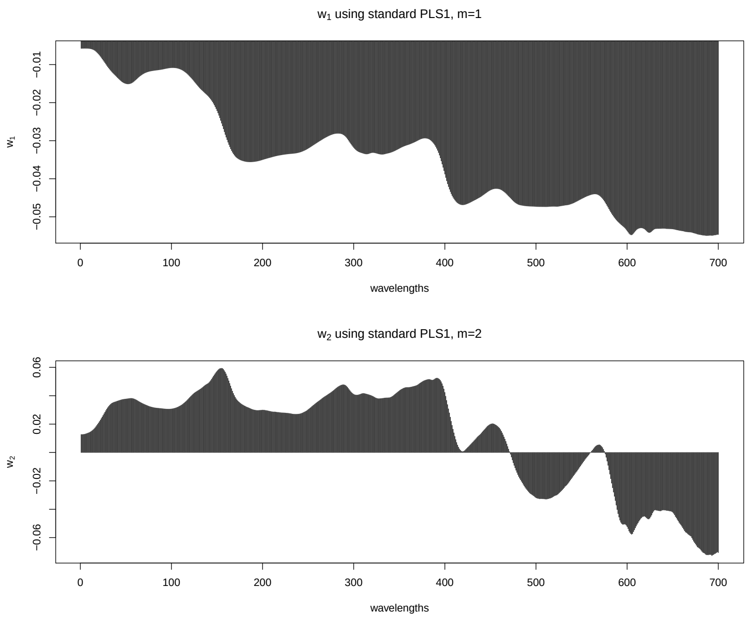


Figure 2.4: w_1 (top panel) and w_2 (bottom panel) from PLS1 using NIR data where the response is on a continuous scale.

It can be seen from top panel in Figure 2.4 the first component of w_1. The estimation of w_1 indicate those wavelengths that are highly associated to the moisture more than the others. The bottom panel of Figure 2.4 illustrates the estimation of w_2. w_1 and w_2 with others w_m are used to calculate the estimation of β_{pls}.

Figure 2.5 shows the predicted values for PLS1 using normal response.

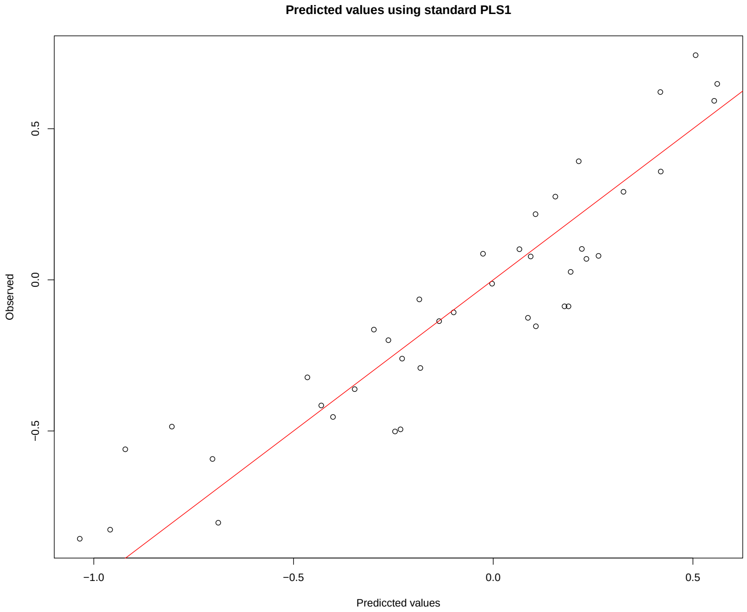


Figure 2.5: Predicted values for PLS1 for NIR data with real-valued response, the red line is the fitted line.

It can be seen from Figure 2.5 the predicted values versus the validation values in the validation set. It can be seen that the predicted values are close to the fitted line.

Figure 2.6 shows the predicted values for all three algorithms of PLS2 for NIR data with real-valued responses.

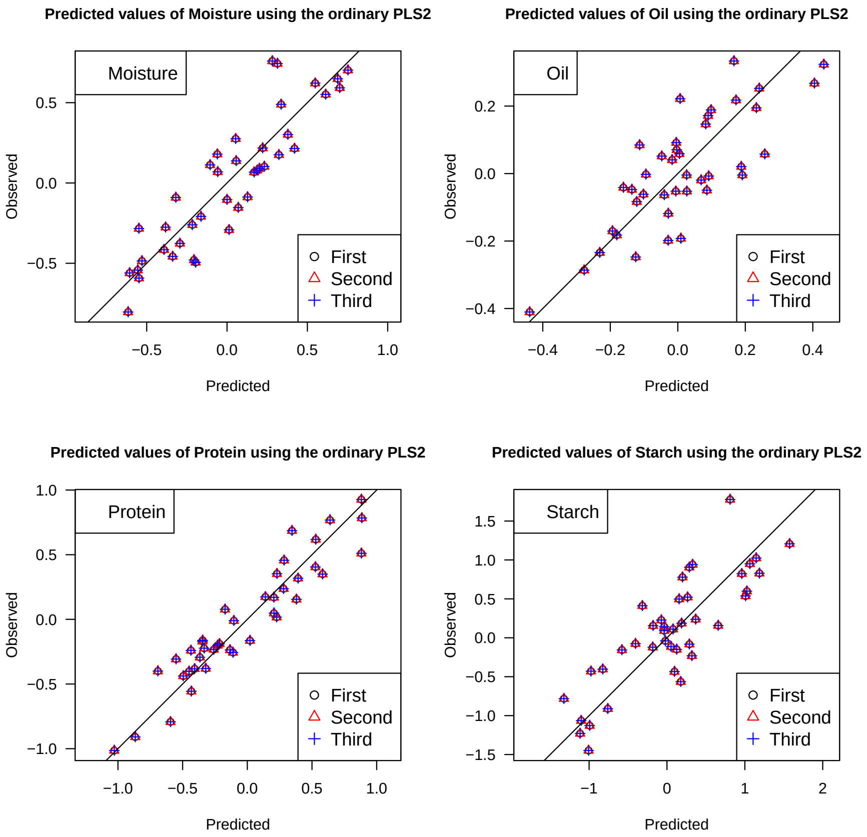


Figure 2.6: Predicted values for all three algorithms of PLS2 for NIR data with real-valued responses. Predicted values of Moisture (top left panel), Oil (top right panel), Protein (bottom left panel), and Starch (bottom right panel).

It can be seen from Figure 2.6 using three versions of the NIPALS algorithm for PLS2 that their predicted values are the same. This is because the inner regression is taken into account when calculating the estimation of PLS ($\hat{\beta}_{\text{pls}}$). The optimal number pf components is 15 components using PLS2. Looking at the predicted values in Figure 2.6, we can confirm that their estimations of $\hat{\beta}$ are the same.

2.7.3 CNA data

Figure 2.7 shows the estimated parameters ($\hat{\beta}_{\text{pls1}}$) using PLS1 for lung cancer DNA-Copy CNA data.

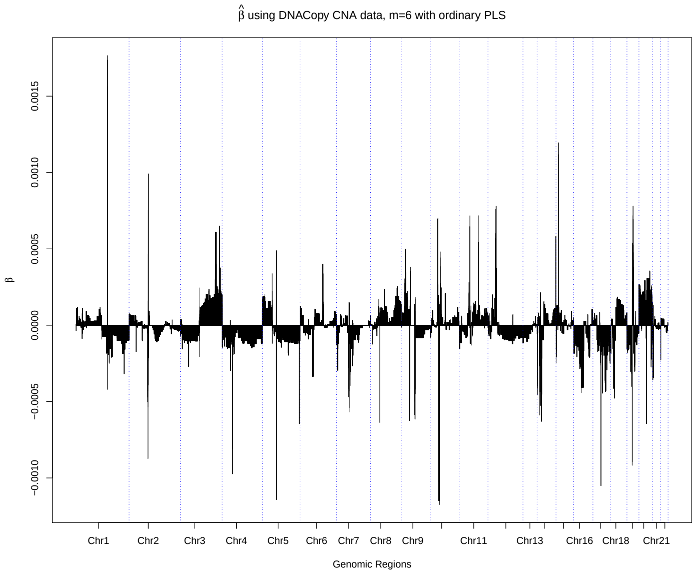


Figure 2.7: $\hat{\beta}_{\text{pls1}}$ for DNACopy CNA data where the response variable is binary using PLS1 with 6 components (m_{opt}=6). The sex chromosomes are excluded in the analysis.

It can be seen from Figure 2.7 that the positive values of the genomic windows indicate that these genomic contribute more to the squamous carcinoma class. On the other hand, negative values of the estimation of $\hat{\beta}$ contribute more to ADC class.

Figure 2.8 shows the estimated parameters ($\hat{\beta}_{\text{pls}}$) using PLS1 for lung cancer smooth CNA data.

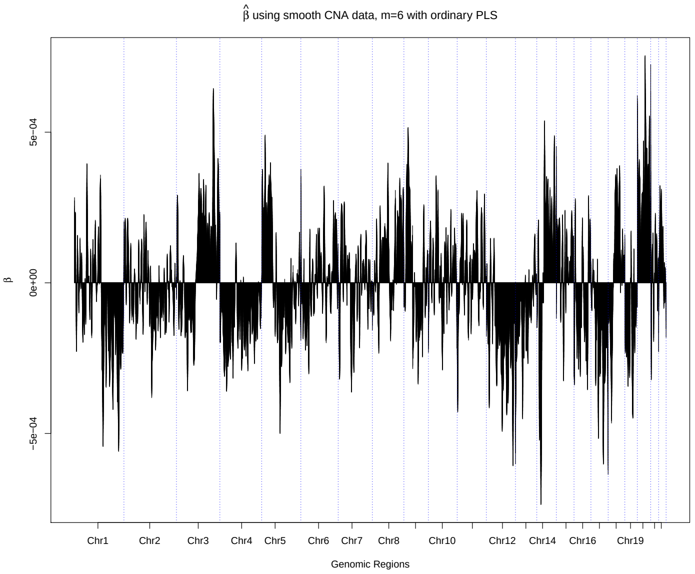


Figure 2.8: $\hat{\beta}_{\text{pls1}}$ for PLS1 for lung cancer smoothing data with binary response. The optimal number of components is 6 components (m_{opt}=6). The sex chromosomes are excluded in the analysis.

We can see from Figure 2.8 that the estimation of $\hat{\beta}$ using PLS1 where the response is binary. Since X and y are centred, the thresholding for the classification is zero. Positive values of the estimation of $\hat{\beta}$ indicate that these genomic regions contribute more to the squamous carcinoma. Negative values of the estimation of $\hat{\beta}$ indicate that the negative genomic regions contribute more to ADC class.

Figure 2.9 shows the w_1 and w_2 using PLS1 for lung cancer DNACopy CNA data where the response is binary.

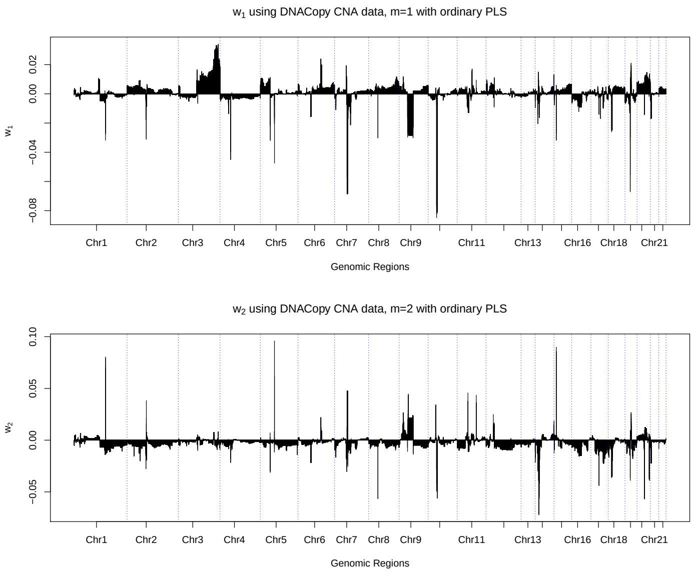


Figure 2.9: w_1 (top panel) and w_2 (bottom panel) for PLS1 for lung cancer DNACopy CNA data with a binary response. The sex chromosomes are excluded in the analysis.

It can be seen from the top panel of Figure 2.9, the first component of the X-weights (w_1). We can see that chromosome 3, 7, 10 and 19 have some genomic regions with large copy number in absolute value indicating these chromosomes may have an association with the cancer type. In the bottom panel of Figure 2.9, it can be seen that in chromosomes 1, 5, 7, 13 and 14 with large number of copy in absolute value.

Figure 2.10 shows the w_1 and w_2 for PLS1 for lung cancer smooth CNA data where the response variable is binary.

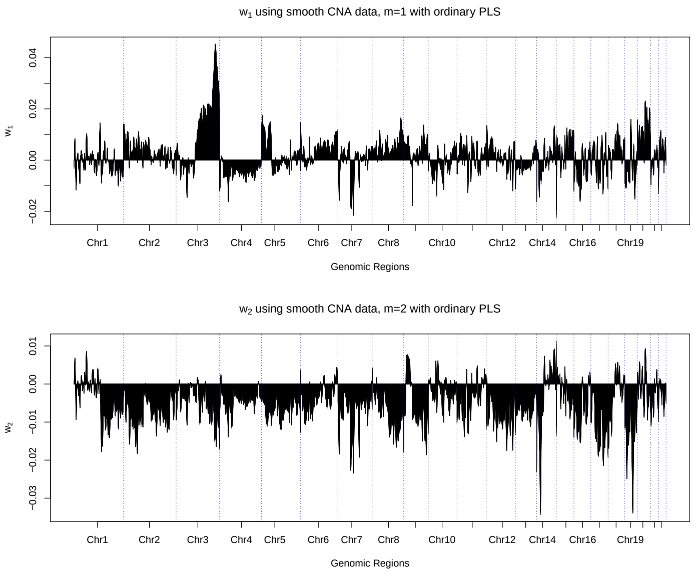


Figure 2.10: w_1 (top panel) and w_2 (bottom panel) for PLS1 for Lung cancer smooth CNA data with binary response. The sex chromosomes are excluded in the analysis.

It can be seen that from the top panel of Figure 2.10 using w_1 the genomic regions in chromosomes 3 and 7 has a large value in absolute value which indicates that these may be associated with the cancer type. In the bottom panel of Figure 2.10, we can see that in w_2 the genomic regions with large value in absolute values as in chromosomes 14 and 19 for example.

Figure 2.11 shows the predicted values class using PLS1 for lung cancer DNACopy CNA data (left panel) and smooth CNA data (right panel) with a binary response. The predicted class are plotted based on the optimal number of components which is 6 components.

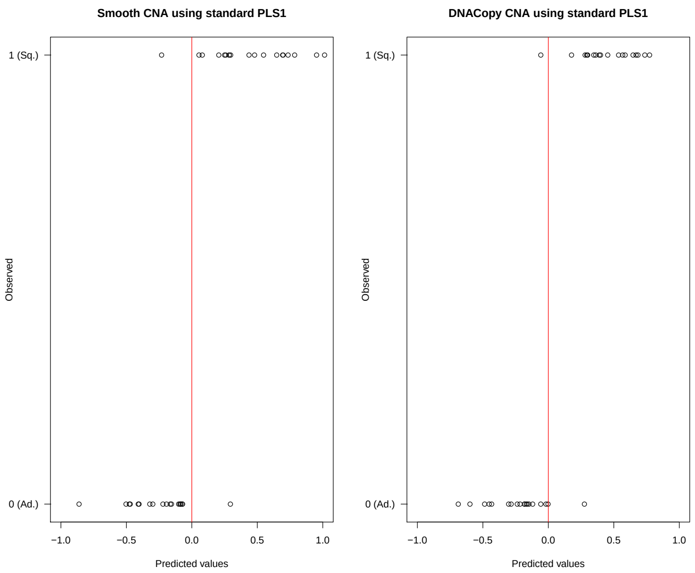


Figure 2.11: Predicted values (class) using PLS1 for lung cancer DNACopy CNA data (left panel) and smooth CNA data (right panel) with a with binary response. (m_{opt} =6).

It can be seen from the left panel of Figure 2.11 the predicted class using the DNACopy CNA data with only two misclassified samples. Using the smooth CNA data (right panel) of Figure 2.11 , we can see that only 3 misclassified samples. The predicted values (class) versus the validation samples in the validation set are plotted.

2.8 The estimated variance of the PLS1 estimator

It is difficult if it is not impossible to calculate the variance of $\hat{\beta}_{\text{pls1}}$ theoretically because the distribution of the estimated parameters of PLS1 is unknown. Also, all the factors that are extracted from NIPALS algorithm such as w_m, p_m, and t_m are unknown distribution. Therefore, previous researchers have calculated it numerically using computationally intensive procedures like bootstrapping to get the confidence intervals for predictions. In some cases, interval estimates are not calculated, only point estimates are calculated. Another way, Phatak *et al.* (1993) introduced an approach based on the linearisation of the PLS1 estimator to allow them to construct approximate confidence intervals for predictions from PLS1.

The estimator of the univariate PLS ($\hat{\beta}_{\text{pls1}}$) is a biased estimator, which is not similar to the estimator in the OLS because the estimator of PLS1 is a non-linear function of the response variable, y. Thus, it is difficult or impossible to derive the exact distribution of the estimator. Approximating the distribution of the estimator is a useful way to construct the confidence intervals instead of only the point estimates Denham (1997). We have used the numerical approach bootstrapping since the theoretical derivation and linear approximation is beyond our scope in this research. Interested readers can refer to Denham (1997), Phatak *et al.* (1993) and Phatak & de Hoog (2002) for more details in the approximated distribution of the estimated parameters of PLS1.

2.9 The connection between PLS regression and LDA

PLS was not originally constructed for classification problems. However, recent researches have used PLS for classification problems. Specifically, in genomic data there have been numerous applications of classification using PLS methods for gene expression data:

- Discriminating human heart failure etiology using gene expression profiles Huang *et al.* (2005).
- Linear regression and two-class classification with gene expression data Huang & Pan (2003).
- Classification of acute leukemia subtypes based on gene expression data Cho *et al.* (2002).
- Prediction of outcome with microarray data: a partial least squares discriminant analysis (PLS-DA) approach Pérez-Enciso & Tenenhaus (2003).
- Molecular classification of cancer: class discovery and class prediction by gene expression monitoring Golub *et al.* (1999).
- Classification of human ovarian tumors using multivariate data analysis of polypeptide expression patterns Alaiya *et al.* (2000).

- Potentialities of multivariate approaches in genome-based cancer research: identification of candidate genes for new diagnostics by PLS discriminant analysis Musumarra *et al.* (2004).

- Evaluating methods for classifying expression data Man *et al.* (2004).

- Tumor classification by partial least squares using microarray gene expression data Nguyen & Rocke (2002).

- Classification using partial least squares with penalised logistic regression Fort & Lambert-Lacroix (2005).

Nguyen & Rocke (2002) introduced an approach that needs two steps to use PLS in classification. To use this approach one needs to choose the number of components to be used in the model in the first step, and choose the classification method for the second step. The steps are: First, using PLS as a dimension reduction method. Secondly, the PLS components or latent structures are used as predictors in a classical discrimination method such as logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA). Boulesteix (2004) used only LDA for the further studied using the same approach that Nguyen & Rocke (2002) used with a comparison to other dimension reduction. PLS-LDA was the best classification method among all the other classification methods with nearest centroids approach by Tibshirani *et al.* (2002) and the support vector machines (SVM) for all eight studied cancer data sets.

In LDA, Fisher was interested in the following optimisation problem as in Equation (2.35):

$$\underset{w^T w=1}{\operatorname{argmax}} \left\{ \frac{(w^T H w)}{w^T E w} \right\}, \tag{2.35}$$

where H denotes the among-groups sums-of-squares and cross-products matrix and E the pooled within-groups sums-of-squares and cross-products matrix.

PLS components are defined as follows, with the constraint that the components are orthogonal in the X block Barker & Rayens (2003).

It is well known that the CCA directions are the Fisher LDA directions when CCA is performed on the data matrix X and the coded matrix representing the response variable y as below Barker & Rayens (2003).

In the binary case where we have two groups or classes, we can write the response variable (Y) in two ways as in Barker & Rayens (2003):

$$Y^* = \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} \end{pmatrix}_{n\times 2}$$

or

$$Y^{**} = \begin{pmatrix} \mathbf{1}_{n_1} \\ \mathbf{0}_{n_2} \end{pmatrix}_{n\times 1},$$

where $\mathbf{1}$ is a vector of ones, $\mathbf{0}$ is a vector zeros, n_1 and n_2 are the number of observations in each group, and $n_1 + n_2 = n$ the sample size.

Given S_{xx} of size $(p \times p)$ the variance-covariance matrix of X and S_{xy} of size $(p \times 1)$ the covariance vector between X and y.

Frank & Friedman (1993) suggested that PLS regression can be viewed as a canonical correlation analysis (CCA) where we would like to maximise

$$w^T S_{xy} S_{yx} w, \tag{2.36}$$

with constraints $w_m^T w_m = 1, \forall m$, and $w_m^T w_h$ for $m > h$.

PLS and LDA can be derived as the eigensolutions of

$$S_{xy} S_{yx} w_m = \psi w_m, \tag{2.37}$$

where S_{xy} is the sample covariance vector of X and y.

The connection between PLS and LDA can be shown as in Barker & Rayens (2003)

$$S_{xy} S_{yx} = \frac{1}{(n-1)^2} \sum_{i=1}^{g} n_i^2 (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T, \tag{2.38}$$

With one component for PLS model, maximising $w^T S_{xy} S_{yx} w$ with the constraint $w_m^T w_m = 1$ is the same as the eigensolution in Equation (2.37) Gusnanto *et al.* (2015). Therefore, PLS solution is similar to LDA solution.

Chapter 3

Filter Factors in High Dimensional Data

3.1 Overview

In the presence of nearly collinear data, ordinary least squares regression (OLS) solution fails to work due to the singularity and $X^T X$ is not invertible when ($p > n$). Therefore, other biased methods or so-called shrinkage methods have been introduced such as Lasso Tibshirani (1996), ridge regression (RR) (Hoerl & Kennard, 1970), principal components regression (PCR) (Massy, 1965) and partial least squares (PLS) regression (Wold, 1975). We will consider only those methods that have been used in practice as provided in (Frank & Friedman, 1993), (Rosipal & Krämer, 2006) and Lingjaerde & Christophersen (2000).

Shrinkage is how much the estimators of the shrinkage methods are "shrunk" from the OLS solution by some amount or a scale that shrinks the OLS estimator. This amount reduces the variance of the estimator where those methods shrink some of the parameters towards zero (Frank & Friedman, 1993).

Since the OLS solution is not applicable in high dimensional data or multicollinearity occurs, the "shrinkage factors" term could be replaced by "filter factors" more appropriately. That means the shrinkage is particularly used if there is a comparison between the estimators of biased methods and the OLS method. In this chapter, we focus on filtering in high dimensional data by comparing three popular methods of shrinkage methods: RR, PCR and PLS regression, writing their estimators in a unified approach. The relationship between these three methods has been discussed by many researchers e.g. Naes & Martens (1985), Helland (1988) and Stone & Brooks (1990).

Frank & Friedman (1993) compared the shrinkage estimators of RR, PCR and PLS regression by considering their properties. Lingjaerde & Christophersen (2000) provide some theoretical results regarding PLS shrinkage and some of their properties in a low-dimensional case. Moreover, Butler & Denham (2000) derived an alternative representation of the shrinkage of PLS regression relative to OLS. Here, we have modified those results of shrinkage factors and the properties when $p > n$.

The organisation of this chapter is as follows. Section 3.2 provides an explanation of shrinkage in some of the shrinkage methods for low dimension data with simulated examples before moving to the high dimensional data. Section 3.3 explains the filtering concept in high dimensional data for those three methods. Section 3.4 shows some examples for those filter factors in high dimensional data using two real data sets where the response variable is normal or binary. Moreover, we have deeply examined filter factors of PLS in particular by interpreting them and trying to link the eigenvalues of two different matrices as described in Section 3.5. Some general properties of the filter factors of the PLS estimator are provided in Section 3.6. Finally, some discussion regarding shrinkage in high dimensional data is found in Section 3.7.

3.2 Shrinkage in low dimensional data

In this section we focus on the low-dimensional case ($n \geq p$). Let X denote the $n \times p$ predictor matrix, and y denote the response vector, where both X and y are centred and scaled to have a unit variance. We consider the general linear regression model

$$y = X\beta + \epsilon, \tag{3.1}$$

where β is an unknown $p \times 1$ parameter vector and ϵ is an $n \times 1$ vector of errors which we assume are independently distributed with $\epsilon \sim N_p(\mu, \sigma^2 I)$. The OLS estimator for $\hat{\beta}$ in Equation (3.1) is

$$\hat{\beta}^{\text{ols}} = (X^T X)^{-1} X^T y. \tag{3.2}$$

Assuming that X has a full rank p, the singular value decomposition (SVD) of X will be used mainly in showing the amount of shrinkage of the shrinkage estimators of

$\hat{\beta}$. The X matrix can be written using SVD as

$$\underset{n\times p}{X} = \underset{n\times p}{U} \ \underset{p\times p}{D} \ \underset{p\times p}{V^T}. \tag{3.3}$$

where $U^TU = V^TV = VV^T = I_p$ (the $p\times p$ identity matrix) and where D is a diagonal matrix with the singular values $d_1 \geq d_2 \geq \cdots \geq d_p$ on the diagonal. The columns $u_1, \ldots, u_p$ of matrix U are denoted left singular vectors, and the columns $v_1, \ldots, v_p$ of matrix V are denoted the right singular vectors. Using Equation (3.2) and Equation (3.3), the ordinary least square estimator for the parameter vector β in Equation (3.1) can be written in a matrix notation using SVD as

$$\hat{\beta}^{\text{ols}} = VD^{-1}U^Ty, \tag{3.4}$$

which can be written as a sum of column vectors of the matrices in Equation (3.4) as

$$\hat{\beta}^{\text{ols}} = \sum_{i=1}^{p} \frac{u_i^T y}{d_i} v_i. \tag{3.5}$$

3.2.1 Ridge regression

One popular estimator of shrinkage estimators for the vector β in Equation (3.1) is the ridge regression estimator,

$$\hat{\beta}^{\text{rr}} = (X^TX + \delta I_p)^{-1}X^Ty, \tag{3.6}$$

where $\delta \geq 0$ is known as the ridge parameter. Writing Equation (3.6) as a linear combination of the right and left singular vectors of X, V and U respectively, using Equation (3.3), $\hat{\beta}^{\text{rr}}$ can be written as

$$\hat{\beta}^{\text{rr}} = (VDU^TUDV^T + \delta I_p)^{-1}VDU^Ty.$$

Since $U^TU = V^TV = VV^T = I_p$ and D is a diagonal matrix of singular values, we have that

$$\hat{\beta}^{\text{rr}} = V(D^2 + \delta I_p)^{-1}V^TVDU^Ty.$$

Therefore,

$$\hat{\beta}^{\text{rr}} = V(D^2 + \delta I_p)^{-1}DU^Ty, \tag{3.7}$$

where D^2 is a diagonal matrix of the eigenvalues of $X^T X$ or the square values of the singular values (d_i) in Equation (3.5). We can rewrite equation (3.7) as

$$\hat{\beta}^{\text{rr}} = V D^2 (D^2 + \delta I_p)^{-1} D^{-1} U^T y. \tag{3.8}$$

Also, Equation (3.8) can be written in sum of vectors notation as

$$\hat{\beta}^{\text{rr}} = \sum_{i=1}^{p} \frac{d_i^2}{d_i^2 + \delta} \frac{u_i^T y}{d_i} v_i. \tag{3.9}$$

By comparing Equation (3.9) to the OLS solution in Equation (3.5), we can identify the shrinkage factors in RR estimator as

$$\omega_i^{\text{rr}} = \frac{d_i^2}{d_i^2 + \delta}, \qquad i = 1, 2, \ldots, p, \tag{3.10}$$

where ω_i^{rr} is the shrinkage factor of the RR estimator.

3.2.2 Principal components regression

Principal component analysis (PCA) is a method that can write the data matrix X as outer products of two matrices called scores (T) and loadings (P) Geladi & Kowalski (1986) as

$$X = T P^T, \tag{3.11}$$

where T is a matrix of size $n \times m$, and P of size $p \times m$, where $m = 1, 2, \ldots, M$ is the number of components used in both matrices. Since the columns of P is an orthogonal matrix, Equation (3.11) can be written as

$$T = XP. \tag{3.12}$$

Another popular shrinkage estimators for $\hat{\beta}$ in Equation (3.1) is the principal components regression estimator ($\hat{\beta}^{\text{pcr}}$). It can be written using the latent variables as a linear combination of T and P matrices as

$$\hat{\beta}^{\text{pcr}} = P (T^T T)^{-1} T^T y. \tag{3.13}$$

Using the fact that the loadings in PCR estimator (P) are the eigenvectors of $X^T X$ which are represented by V in Equation (3.14). P and V have the same number of columns which is equivalent to m. If $m = M$ is equal to the rank of X. Therefore,

$$T^* = X V^*. \tag{3.14}$$

The superscript $*$ on V and T is to clarify that these matrices do not use all components. If all components are included, V^* becomes V including all eigenvalues and the corresponding eigenvectors, and T^* becomes T with all components which means the number of components is equal to p $(M = p)$. Using Equation (3.3), Equation (3.14), and the property that ${V^*}^T V^* = {U^*}^T U^* = I_m$, we have

$$T^* = U^* D^*, \tag{3.15}$$

where U^* is a $n \times m$ matrix, D^* is the a diagonal matrix of $m \times m$ of the singular values of X in a descending order and V^* is the $p \times m$ matrix of right singular vectors of X. These matrices are called reduced rank matrices.

Substituting Equation (3.15) in Equation (3.12) and using that ${U^*}^T U^* = I_m$ and $D^* = {D^*}^T$, Equation (3.13) can be expressed using the eigenvectors of $X^T X$ $(P = V^*)$ as

$$\hat{\beta}^{\text{pcr}} = V^* \Omega^{\text{pcr}} {D^*}^{-1} {U^*}^T y, \tag{3.16}$$

where Ω^{pcr} is an $m \times m$ diagonal matrix of values either zero or one as defined in Equation (3.18). It should be noted that in PCR, the T^*, V^*, U^* and D^* are the reduced sized matrices of the above ones in the OLS solution depending on the number of components, (m). If $m = M = p$, then the above matrices with $*$ become exactly as the ones in the OLS solution in section 3.2.

Also, we can write the estimator of PCR in Equation (3.16) in the univariate notation as

$$\hat{\beta}^{\text{pcr}} = \sum_{i=1}^{m} \omega_i^{\text{pcr}} \frac{u_i^T y}{d_i} v_i. \tag{3.17}$$

From Equation (3.17), the shrinkage factor in the PCR estimator is

$$\omega_i^{\text{pcr}} = \begin{cases} 1, & \text{if} \quad i\text{th component is included} \\ 0, & \text{otherwise,} \end{cases} \tag{3.18}$$

where ω_i^{pcr} is the shrinkage factor of the estimator of PCR.

3.2.3 Partial least squares regression

The last estimate we consider from the shrinkage methods discussed in this chapter for the parameter $\hat{\beta}$ in Equation (3.1) is the partial least square estimator ($\hat{\beta}^{\text{pls}}$). Recall that W is regressing the data matrix X on y which s called X-weights. From NIPALS algorithm in Chapter 2, one can write the PLS estimator in terms of the X weights (W) as

$$\hat{\beta}^{\text{pls}} = W(W^T X^T X W)^{-1} W^T X^T y. \tag{3.19}$$

The PLS regression estimator in terms of the eigenvectors of $X^T X$ can be written also in terms of the eigenvectors of $X^T X$ as (Lingjaerde & Christophersen, 2000)

$$\hat{\beta}^{\text{pls}} = W(W^T V D^2 V^T W)^{-1} W^T V D U^T y, \tag{3.20}$$

where V, D, and U have the dimensions of $p \times p$, $p \times p$, and $n \times p$ respectively, and W is the X-weights $p \times m$ matrix, where m is the number of component $m \in 1, 2, \ldots, p$. Equation (3.20) is the same as Equation (3.19).

The PLS regression estimator can have another expansion of Equation (3.20) as in Equation (3.21) which is written in the same format of the OLS solution in matrix notation.

$$\hat{\beta}^{\text{pls}} = V \Omega^{\text{pls}} D^{-1} U^T y, \tag{3.21}$$

where Ω^{pls} a diagonal matrix with dimension $(p \times p)$ and it is called the shrinkage factor of PLS estimator.

Equation (3.22) is an element notation of $\hat{\beta}^{\text{pls}}$ as

$$\hat{\beta}^{\text{pls}} = \sum_{i=1}^{p} \omega_i^{\text{pls}} \frac{u_i^T y}{d_i} v_i, \tag{3.22}$$

where ω_i^{pls} are the diagonal elements of Ω^{pls}.

Looking at Equation (3.22) and comparing it to Equation (3.5), Phatak & de Hoog (2002), Butler & Denham (2000) and Lingjaerde & Christophersen (2000) have showed that the shrinkage factors of PLS regression estimator can be written as

$$\omega_i^{\text{pls}} = 1 - \prod_{j=1}^{m} \left(1 - \frac{d_i^2}{\mu_j^2}\right), \qquad i = 1, 2, \ldots, p, \tag{3.23}$$

where $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m$ are the eigenvalues of $W_m^T X^T X W_m$ where W_m is the X weight matrix with orthonormal columns, and m is the number of components (Lingjaerde & Christophersen, 2000). The proof can be found in Lingjaerde & Christophersen (2000), Frank & Friedman (1993) and Butler & Denham (2000).

3.2.4 Common structure of shrinkage estimators in RR, PCR and PLS

All the previous shrinkage estimators in Equations (3.10), (3.17) and (3.22) can be written in a common approach as a linear combination of the singular vectors and values of X as Lingjaerde & Christophersen (2000)

$$\hat{\beta}^{\text{shrink}} = \sum_{i=1}^{p} \omega_i \frac{u_i^T y}{d_i} v_i, \tag{3.24}$$

where ω_i in Equation (3.24) are called the shrinkage factors whereas $\omega_i = 1$ in $\hat{\beta}^{\text{ols}}$. The shrinkage factor in the PCR and PLS regression estimator will be equal to one if we include all the m components which then yields to the OLS estimator, where $m = p$. Also, if the RR parameter $\delta = 0$, the RR estimator is equal to the OLS estimator.

If the shrinkage factors (ω_i) are not equal to one, the estimator is biased. However, when $\omega_i < 1$, the variance of $\hat{\beta}$ will be small. Therefore, shrinkage is useful if the bias is small comparing to the reduction in the variance of the estimator, so the mean square error (MSE) of $\hat{\beta}$ is reduced. On the other hand, if $\omega_i > 1$, the MSE of $\hat{\beta}$ is increased because the bias and the variance will be increased simultaneously (Butler & Denham, 2000).

3.2.5 Examples

To illustrate the shrinkage behaviour of all methods (RR, PCR and PLS) and when they become equal to one where their estimated is equivalent to the OLS solution, we use two artificial example. In the first example we use a normal response whereas in the second example we use a binary response. We have generated X matrix from normal distribution with mean zero and variance one, with the number of samples $n = 80$ and the number of predictors $p = 10$. β is assigned to be for $\beta_1 = \beta_2 = 2$ and the $\beta_6 = \beta_7 = -2$ and for the other $\beta_j = 0, j = 3, 4, 5, 8, 9, 10$. The errors ($\epsilon$) are generated from normal distribution with mean zero and variance equals to 1.

In the first simulation the response is on a continuous scale as in Figure 3.1 while Figure 3.2 for the second simulation when the response is binary. The shrinkage factors $\omega_i^{(m)}$ for all values of i are plotted for the first six components for the PLS model since the shrinkage factor is equal to one within the first six components. Also, the shrinkage factors for RR and PCR plotted along with the PLS shrinkage factors where the shrinkage should be the same which is calculated as ($s = \frac{||\hat{\beta}^{\text{shr}}||}{||\hat{\beta}^{\text{ols}}||}$). For choosing the value of the ridge parameter (δ) which is described in Section 3.2.1, we use this equality ($||\hat{\beta}^{\text{rr}}|| = ||\hat{\beta}^{\text{pls}}||$). The number of components in PCR was chosen when the lengths were approximately close to the PLS solution ($||\hat{\beta}^{\text{pcr}}|| \cong ||\hat{\beta}^{\text{pls}}||$). In each plot there are three values for showing the ridge parameter (δ), the number of components using PCR (m_{pcr}) and the number of PLS components (m_{pls}), which are the number of the latent variables as described in Sections 3.2.2 and 3.2.3 respectively, and the shrinkage value using this formula ($s = \frac{||\hat{\beta}^{\text{shr}}||}{||\hat{\beta}^{\text{ols}}||}$) Frank & Friedman (1993).

For the binary response, we generate two groups of X with $n = 40$ and $p = 10$ for each group. The first group is generated from normal with zero mean and $\sigma_1^2 = 1$, and the second group is generated with mean equals 3 and $\sigma_2^2 = 1$. The response vector (y) is generated by making the first 40 values equal to one, and the second 40 values equal to zero. Also, X and y are centred, so there is no intercept.

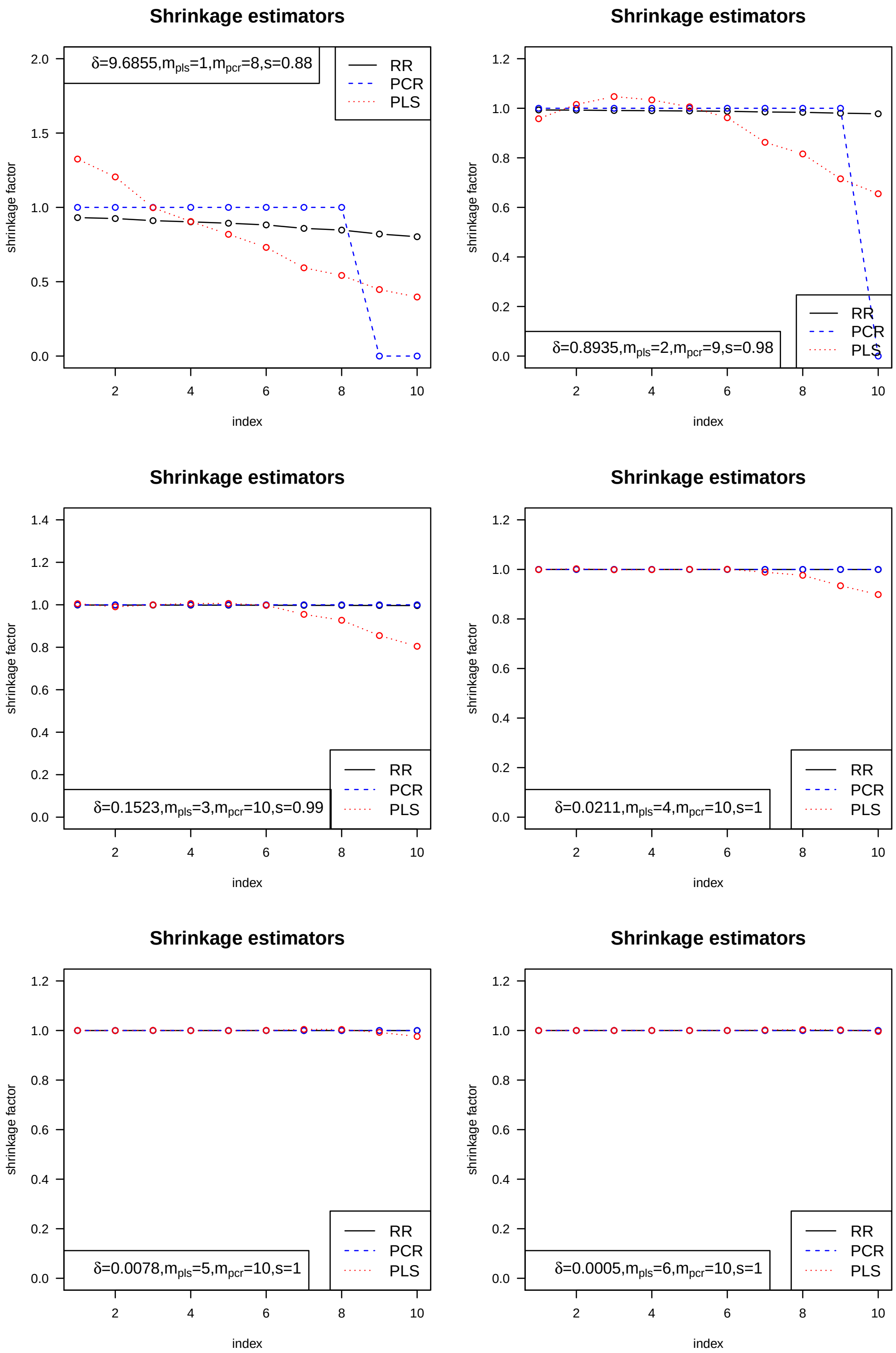


Figure 3.1: Shrinkage factors ω_i for RR, PCR, PLS for simulated data where $n \geq p$ with a normal response for the first six components.

In Figure 3.1, first row left panel, we can see that the first shrinkage factor (ω_1) for the PLS regression estimator using the first component (m is odd) is expanded and larger than one, and that is because the largest eigenvalue of $X^T X$ is larger than all eigenvalues of $W^T X^T XW$ which produces a negative product in Equation (3.23), so ω_1 will be larger than one in this case which is why we use "filter" in place of "shrinkage". In the first row right panel of Figure 3.1, it can be seen that the ω_1 is less than one for $m = 2$ Lingjaerde & Christophersen (2000), and the reason is the product in Equation (3.22) will be positive. Hence, ω_1 will be less than one. For RR, the shrinkage is always between 0 and 1 Lingjaerde & Christophersen (2000) and for PCR is either zero or one. Also, we need a large number of components for PCR in order to make the norm of both estimators as close as possible. Using 1 component of PLS results in a shrinkage s = 0.88 from the OLS solution where the data has low correlation. Looking at the shrinkage factors in the second row left panel of Figure 3.1 for RR almost equals to one because the ridge parameter (δ) is 0.1523 which is small, and the norm of the PCR using 10 components roughly equals to the norm of PLS using three components ($m_{pls} \cong 3m_{pcr}$). In the second row right panel of Figure 3.1, the shrinkage (s) equals to 1 which means that using 4 components of the PLS estimator is very close to the OLS solution while 10 components for the PCR and δ = 0.0211 for RR are needed to have the same overall shrinkage. Moreover, it can be seen in the last row of Figure 3.1 that all three shrinkage methods are the same and equal to one meaning that using 6 components for the PLS regression, 10 components for PCR and δ =0.0005 in order to have the norm of $\hat{\beta}^{\text{pls}} \cong \hat{\beta}^{\text{pcr}} = \hat{\beta}^{\text{rr}}$ also those all equals to the OLS solution. Therefore, the shrinkage to the OLS solution is decreased as the number of PLS and PCR components is increased until the number of components equals to the rank of X.

We show the shrinkage in PCR, which depends on the number of components, in Table 3.1 for simulated data when the response is on a continuous scale.

Number of components	1	2	5	10	OLS estimator
Norm of vector	0.0424	0.3871	0.6361	0.9799	0.9799

Table 3.1: the norm of PCR using some components for $n \geq p$ when the response is on a continuous scale.

Table 3.2 shows the shrinkage in PLS, which depends on the number of components that are included, for simulated data when the response is on a continuous scale.

Number of components	1	2	5	10	OLS estimator
Norm of vector	0.8627	0.9675	0.9798	0.9797	0.9799

Table 3.2: the norm of PLS using some components for $n \geq p$ when the response is on a continuous scale.

From Table 3.1, we can see that as the number of components increased, the norm is increased until the norm of the $\hat{\beta}^{\text{pcr}}$ using all components and $\hat{\beta}^{\text{ols}}$ are equal. Also, from Table 3.2, the norm of the vector of $\hat{\beta}^{\text{pls}}$ gets larger as we add more components. De Jong (1995) has showed theoretically that the PLS solution shrinks from the OLS solution and as the number of components increased the norm of the PLS solution increased till the rank of X is reached. From a geometrical point of view, Goutis *et al.* (1996) give a geometric proof to show that the coefficients derived by the OLS are shrunk by the estimates of PLS regression.

In the case where the response variable (y) is binary with values 0 and 1, the overall results are similar in terms of the behaviour of the shrinkage factors in all three methods to the normal response case. Figure 3.2 shows the shrinkage of all three methods where the response in binary.

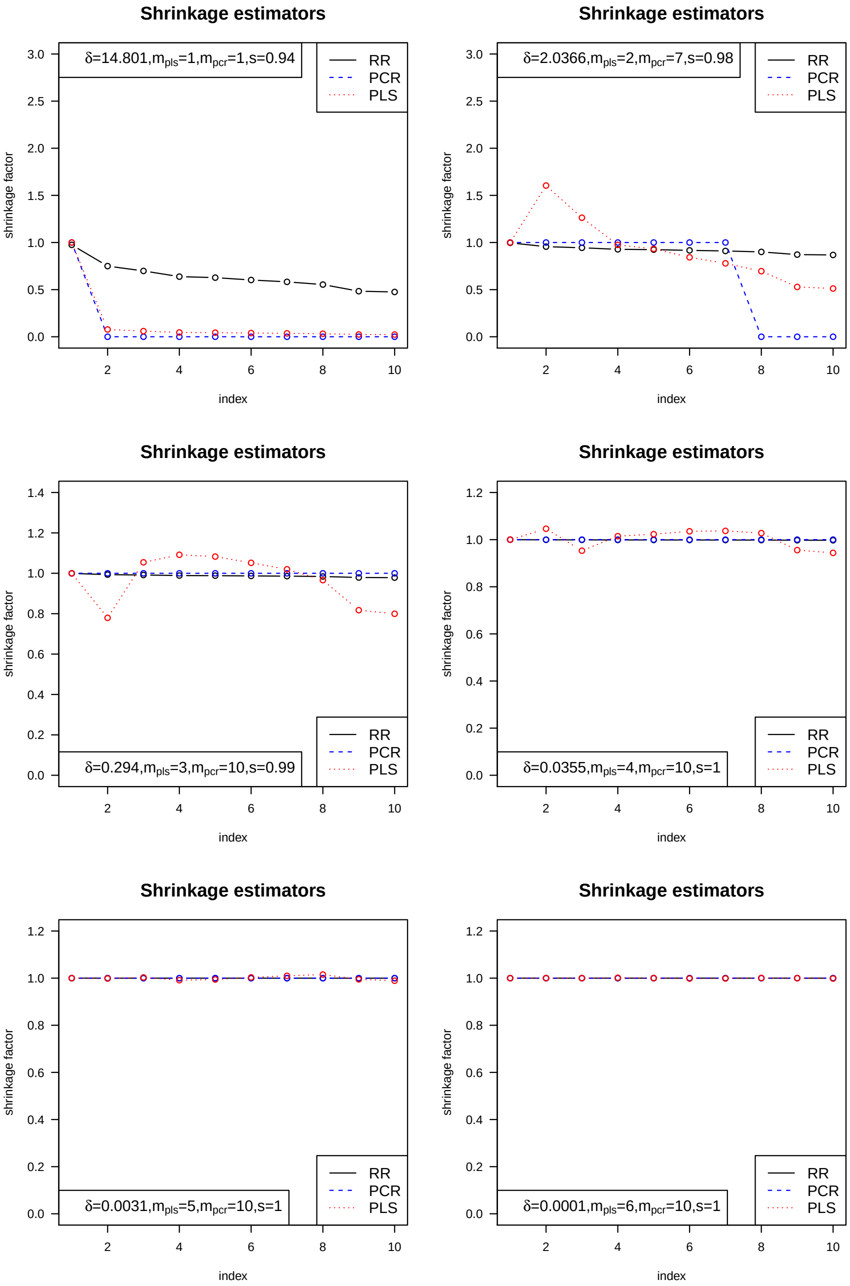


Figure 3.2: Scaled shrinkage factors for RR, PCR, PLS for simulated data where $n \geq p$ with a binary response using 1-6 components.

Figure 3.2 first row left panel provides a comparison of the shrinkage factors for three methods, and it can be seen that the overall shrinkage is 0.94 with δ = 14.801

and one component for PCR and PLS. In the first row right panel of Figure 3.2, the shrinkage factor of PLS is expanded larger than one for some ω_i^{pls} using two components while for PCR using 7 components, and the ridge parameter δ = 2.0366 and the overall shrinkage is 0.98. In the second row of Figure 3.2 right panel, the overall shrinkage is 1 which means that RR, PCR and PLS solutions are getting closer to the OLS solution for m_{pls} = 4, m_{pcr} = 10 and δ = 0.0355. Figure 3.2 last row left panel shows that roughly 5 components of PLS is almost no shrinkage from the OLS solution with overall shrinkage equals to 1 and a very small value for δ = 0.0001 and using all component for PCR, m_{pcr} = 10.

The shrinkage in PCR which depends on the number of components in Equation (3.16), is in Table 3.3 for the simulated data when the response is binary.

Number of components	1	2	5	10	OLS estimator
Norm of vector	0.1829	0.1865	0.1875	0.1948	0.1948

Table 3.3: the norm of PCR using some components for $n \geq p$ when the response is binary.

Also, showing how the shrinkage in PLS depends on the number of components is found in Table 3.4 for the simulated data when the response is binary.

Number of components	1	2	5	10	OLS estimator
Norm of vector	0.1830	0.1923	0.1948	0.1948	0.1948

Table 3.4: the norm of PLS using some components for $n \geq p$ when the response is binary.

3.3 Filtering in high dimensional data

Assuming that X is a high-dimensional data set where $p > n$ and it is centred to mean zero and scaled to have variance equals to one for all variables, the rank of X is $n - 1$ in this case. Due to failure of the OLS for these data sets, the SVD of X can be used instead. The SVD of X is different in terms of dimensions for each right and left singular vectors and singular values of X from the low dimension and is given in Equation (3.25) as

$$\underset{n\times p}{X} = \underset{n\times n-1}{U}\ \underset{n-1\times n-1}{D}\ \underset{n-1\times p}{V^T}. \tag{3.25}$$

Where $U^TU = V^TV = I_{n-1}$, and D is the diagonal with the singular values $d_1 \geq d_2 \geq \cdots \geq d_{n-1}$ on the diagonal. The columns $u_1, \ldots, u_{n-1}$ of U are denoted left singular vectors, and the columns $v_1, \ldots, v_{n-1}$ of V are denoted the right singular vectors.

3.3.1 Ridge regression

For RR estimator we can modify Equation (3.9) for the high dimensional case with different dimension sizes for all matrices as shown in Equation (3.26).

$$\hat{\beta}^{\text{rr}} = V\Omega^{\text{rr}}D^{-1}U^Ty, \tag{3.26}$$

where $\Omega^{\text{rr}} = D^2(D^2 + \delta I_p)^{-1}$ is a diagonal matrix of size $((n-1)\times(n-1))$ and $\delta \geq 0$.

The filter factors are the diagonal values of the Ω^{rr} in Equation (3.26) which are given by

$$\omega_i^{\text{rr}} = \frac{d_i^2}{d_i^2 + \delta} \qquad i = 1, 2, \ldots, n-1. \tag{3.27}$$

It should be noted that D^2 in this case is different from the case where $p < n$ is a diagonal matrix of the $n-1$ eigenvalues of X^TX, and D is a diagonal matrix of $n-1$ singular values of X in a descending order. Also, that V and U with $n-1$ singular vectors columns of X, and d_i^2 are the eigenvalues of X^TX. The difference between this case and the case $p < n$ is the dimension of matrices and apart from that everything remains the same for the filtering factors.

3.3.2 Principal components regression

Writing the PCR estimator for the first components in a high dimensional case as in Equation (3.28):

$$\hat{\beta}^{\text{pcr}} = V^*\Omega^{\text{pcr}}{D^*}^{-1}{U^*}^Ty, \tag{3.28}$$

where D^* is a diagonal matrix of singular values of SVD of X in descending order with size $m \times m$, and the columns $u_1, \ldots, u_m$ of U are denoted left singular vectors of X, and the columns $v_1, \ldots, v_m$ of V are denoted the right singular vectors.

Note that in PCR, these V^*, U^*, and D^* are the first m columns of V, U, and the first m values of the diagonal values of D respectively. Moreover, these matrices are the reduced sized matrices of the original ones in the SVD solutions of the original data matrix, X, depending the number of components (m). If $m = n - 1$, then the above matrices with $*$ become exactly as the ones in the SVD solution (Lingjaerde & Christophersen, 2000).

The filter factors in PCR are diagonal values of Ω^{pcr} in Equation (3.28), which has values of one if the ith principal component is included, or zero if it is not included (Lingjaerde & Christophersen, 2000). Mathematically:

$$\omega_i^{\text{pcr}} = \begin{cases} 1, & \text{if} \quad i \leq m \\ 0, & \text{if} \quad i > m. \end{cases} \tag{3.29}$$

Therefore, the filtering in the PCR estimator in Equation (3.29) depends on the number of components, and the number of components depends on the number of eigenvectors and eigenvalues that are included in the solution (Lingjaerde & Christophersen, 2000).

3.3.3 Finding the filter factors for PLS estimator

Let Θ denote the $(n-1) \times (n-1)$ diagonal matrix with elements of $U^T y$ on the diagonal, and define $\Omega^{\text{pls}} = (\omega_1, \omega_2, \ldots, \omega_{n-1})^T$. The parameter vector β in Equation (3.1) can be expressed as a shrinkage estimator for many estimators as

$$\hat{\beta}^{\text{pls}} = VD^{-1}\Theta\Omega^{\text{pls}}.$$

Given that $U^T U = V^T V = I_{n-1}$, and using Equation (3.21), we have

$$\Theta\Omega^{\text{pls}} = DV^T\hat{\beta}^{\text{pls}} = U^T(UDV^T\hat{\beta}^{\text{pls}}) = U^T\Lambda y,$$

where $\Lambda = XW_m(W_m^T X^T X W_m)^{-1} W_m^T X^T$, and Λ is an orthogonal projection into the subspace XW_m. These filter factors Ω depend non linearly on y which results in a difficult interpretation for them (Lingjaerde & Christophersen, 2000).

3.3.4 Partial least squares regression

Since the OLS solution is not applicable for high dimensional data, we will try to write the PLS regression estimator in terms of eigenvectors of X in another representation of Equation (3.21) in order to have similar form to RR and PCR as

$$\hat{\beta}^{\text{pls}} = V\Omega^{\text{pls}}D^{-1}U^{T}y. \tag{3.30}$$

The Ω^{pls} in Equation (3.30) is a diagonal matrix of ω_i^{pls} with dimension $((n-1)\times(n-1))$, and it is called the filter factors of PLS regression estimator as below

$$\omega_i^{\text{pls}} = 1 - \prod_{j=1}^{m}(1-\frac{d_i^2}{\mu_j^2}), \qquad i = 1, 2, \ldots, n-1, \tag{3.31}$$

where $d_1^2 \geq d_2^2 \geq \cdots \geq d_{n-1}^2$ are the eigenvalues of X^TX, and $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m$ are the eigenvalues of $W_m^TX^TXW_m$, where W_m is the X weight matrix with orthonormal columns, and m is the number of components (Lingjaerde & Christophersen, 2000).
Because of the n-th eigenvalue of X after centring and scaling is zero, we need that V, U and D with $(n-1)$ singular vectors and singular values of X. The filtering in the PLS regression estimator depends on the number of components of $W_m^TX^TXW_m$. The eigenvalues of $W_m^TX^TXW_m$ and X^TX are identical when m=rank(X).

3.3.5 Common structure of filtering in RR, PCR and PLS

We can write all these three methods in a uniform formula as in Equation (3.32) (Frank & Friedman, 1993). All of the previous filtering estimators in Equations (3.27), (3.29) and (3.31) can have a common form using the singular vectors and values of X as in Equation (3.24) for low dimensional case as

$$\hat{\beta}^{\text{filter}} = \sum_{i=1}^{n-1}\omega_i\frac{u_i^Ty}{d_i}v_i, \tag{3.32}$$

where ω_i is called the filter factors, and the rank of the X is $n-1$ (Lingjaerde & Christophersen, 2000). The filter factors for those three methods in Equations (3.27), (3.29) and (3.31) remains the same whether the response variable (y) is normal or binary and in high or low-dimensional case with a difference in the upper number of filter factors we can have.

3.4 Examples

3.4.1 Results using NIR data

To illustrate the filter factors behaviour of all three methods in high dimensional data, we use two different examples based on real data. The first example is where the response variable is normal as (NIR data) while the second example is when the response variable is binary as (CNA data). Using one of the real data sets from chemistry field which is NIR spectropscopic data where the response is on a continuous scale and here we use only the first variable of the four response variables. The filter factors ($\omega_i^{(m)}$) where $i = 1, 2, \ldots, n-1$ are plotted for the first six components in Figure 3.3, also for m = 9 and 26 components in Figure 3.4 to show the some interesting behaviour of the filter factors for the PLS estimator. The filter factors for RR and PCR are also plotted along with the PLS filter factors where the overall shrinkage should be the same ($f = \frac{||\hat{\beta}^{\text{filter}}||}{||\hat{\beta}^{\text{pls}^*}||}$). For choosing the value of the ridge parameter (δ), we use this equality ($||\hat{\beta}^{\text{rr}}|| = ||\hat{\beta}^{\text{pls}}||$). The number of components in PCR was chosen when the lengths were approximately close to the PLS solution ($||\hat{\beta}^{\text{pcr}}|| \cong ||\hat{\beta}^{\text{pls}}||$) because we would like to make the shrinkage for all methods are equivalent. In each plot there are three values for showing the ridge parameter (δ) the number of PLS components, and lastly the overall shrinkage value using this formula ($f = \frac{||\hat{\beta}^{\text{filter}}||}{||\hat{\beta}^{\text{pls}^*}||}$). The $||\hat{\beta}^{\text{pls}^*}||$ is the $||\hat{\beta}^{\text{pls}}||$ using the rank of X, which is $(n-1)$, in this case where $p > n$. Thus, in this data (NIR data) the rank will be ($n - 1 = 79$) which means that $||\hat{\beta}^{\text{pls}^*}||$ is the same as the $||\hat{\beta}^{\text{pls}}||$ using $m = 79$ components.

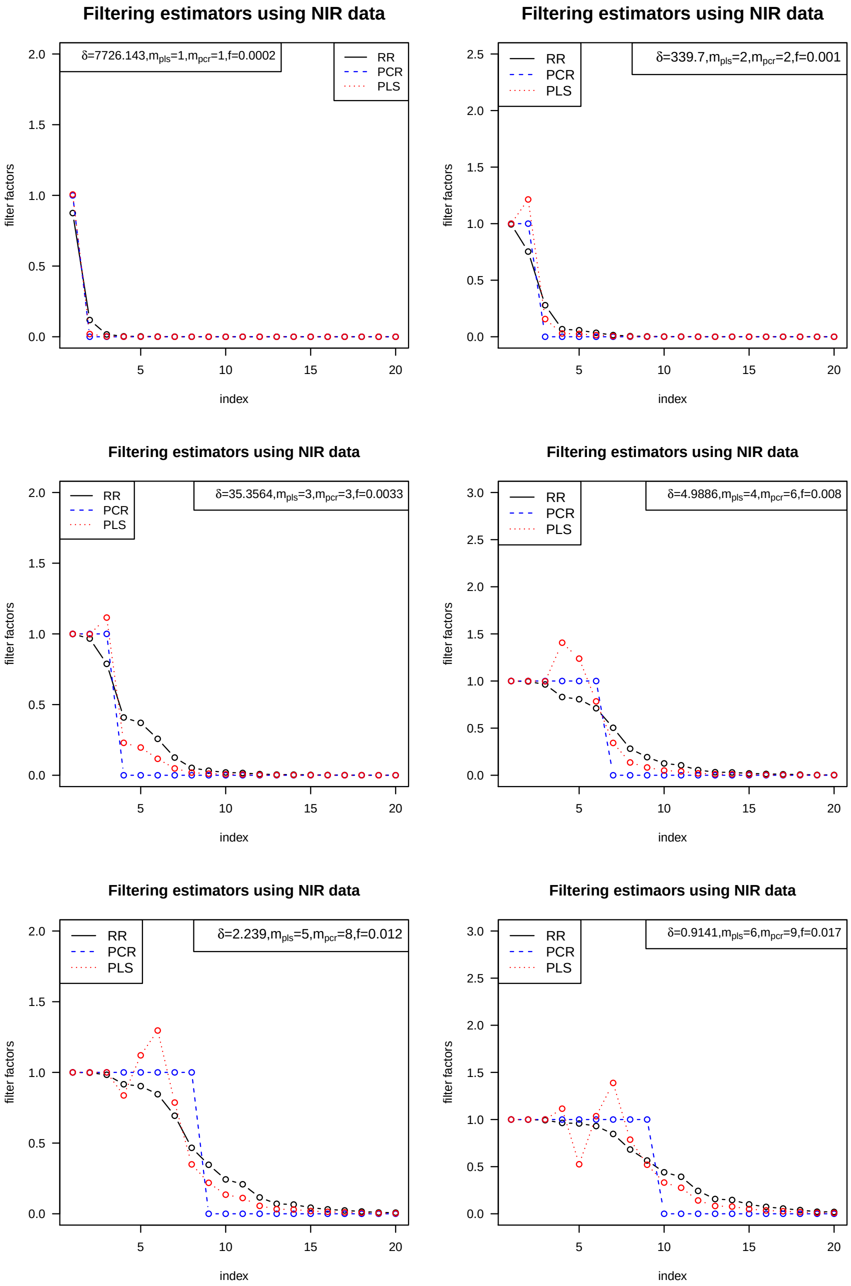


Figure 3.3: filter factors for RR, PCR, PLS for NIR data where $p > n$ with a normal using 1-6 components.

In the first row left panel of Figure 3.3 where the number of components is one using PLS and PCR estimator, we can see that the overall filtering is very small which is because the X data is highly-correlated and we need more components for PCR and PLS and a huge value of $\delta = 7726.143$. In the first row right panel of Figure 3.3, we still have a small value of the overall filtering with an expansion of the filter factors of PLS estimator while for PCR and RR the filter factors are between zero and one.

In the second row left panel, the number of components of PCR and PLS are the same. With 3 components, the norm of the estimator using PCR and PLS approximately is equal to each other. But, the filter factor of PLS are expanded sometimes and get below one. For the right panel of the same figure, we can see that some of the filter factors of PLS is larger than one and some below as well where the overall filtering is 0.008 and $\delta = 4.9886$.

The last left panel shows the filtering between RR, PCR and PLS where the number of components is $m_{pls} = 5$, $m_{pcr} = 8$, $\delta = 2.239$, and the overall filtering equals to 0.012. Again, this number of components is enough to make the overall filtering close to one, but in this data is not because of high collinearity. It can be seen that the filter factors of the PLS estimator are oscillating around one in all panels of Figure 3.3.

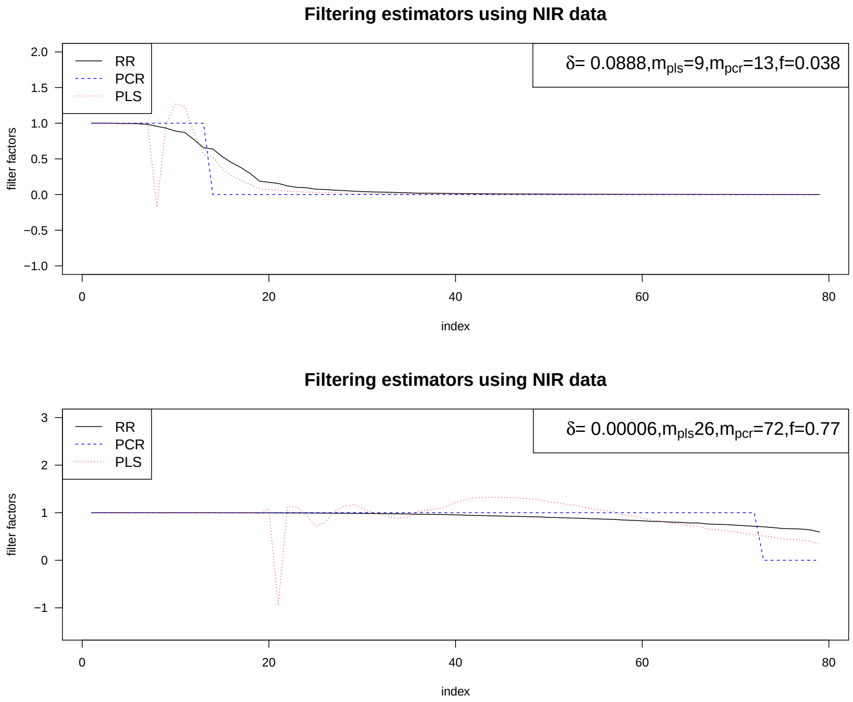


Figure 3.4: filter factors for RR, PCR, PLS for NIR data where $p > n$ with a normal response using 9 and 26 components.

In Figure 3.4, top plot, we can see something interesting where some filter factors take negative values. The number of components of PCR is 13 and for PLS is 9 and δ is 0.0888. It is really difficult to tell how the filter factors behave because of the estimator depends non linearly on the response variable (y) (Lingjaerde & Christophersen, 2000). It can be seen also in the bottom plot of Figure 3.4 that filter factors of RR and PCR are always between 0 and 1 while in PLS, they are varying around one either larger or smaller than one and even they become negative in some cases. This finding will be discussed in more detail in Chapter 4.

3.4.2 Results using CNA lung cancer data

In Figures 3.5 and 3.6 using the CNA lung cancer data when the response is binary, the filter factors $\omega_i^{(m)}$ are plotted for the first six components and the 8th component for the PLS model. The filter factors for RR and PCR also plotted along with the PLS filter factors where the overall filtering should be the same and it is calculated as in Section 3.4.1. The value of the ridge parameter (δ) and the number of components in PCR were chosen as the same criteria also in Section 3.4.1. In each plot there are three values as the ones in Figure 3.5 as well. The rank is ($n - 1$ = 75) for this data, so $||\hat{\beta}^{\text{pls}^*}||$ is the same as the norm of the $||\hat{\beta}^{\text{pls}}||$ using $m = 75$ components.

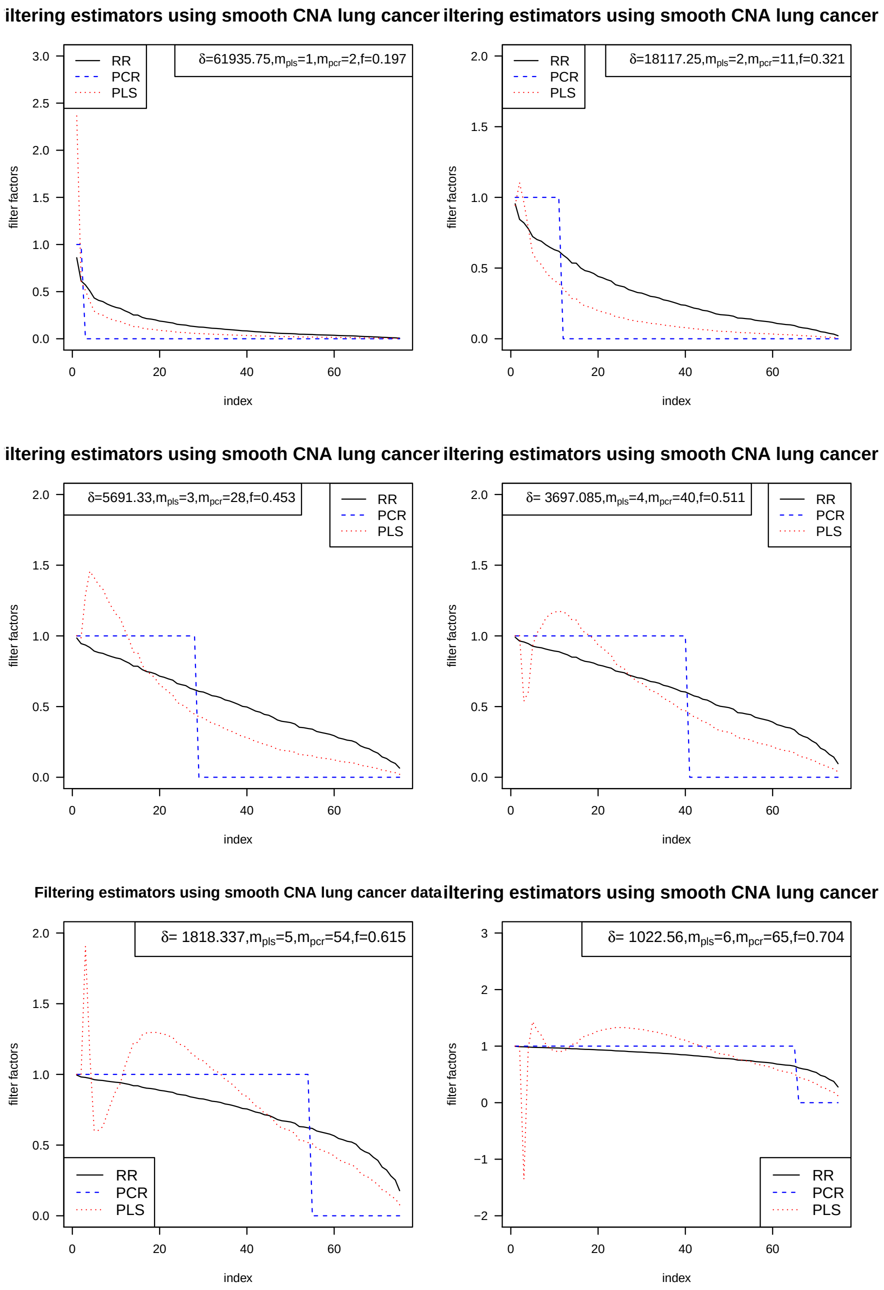


Figure 3.5: filter factors for RR, PCR, PLS for smooth CNA lung cancer data where $p > n$ with a binary response for the first six components.

Looking at the left panel of the first row of Figure 3.5, it can be seen that δ has a very large value (61935.75), $m_{pcr} = 2$ and $m_{pls} = 1$ with the overall filtering equals to

0.197. Lingjaerde & Christophersen (2000) showed that $\omega_1^{\text{pls}} > 1$ if m is odd and it is less than one if m is even. According to this property of the filter factors for the PLS estimator, we can see that ω_1^{pls} is larger than one because the number of components is odd ($m = 1$). In the same row right panel, we can see that ω_1^{pls} is less than 1 since the m_{pls} is 2 which is even.

In the second row left and right panels of Figure 3.5 show the filter factor (ω_i^{pls}) of the PLS estimator where they are expanded in the first largest eigenvalues of $X^T X$ (for large values of d_i^2), and they are contracted for small eigenvalues of $X^T X$ as d_i^2 gets smaller.

It can be seen in the third row left panel of Figure 3.5, ω_i^{pls} are oscillating around one with expanding in some filter factors for PLS and shrinking in others. Moreover, in the right panel of Figure 3.5 there is an interesting event where one of the filter factors of the PLS estimator is negative which does not affect the PLS estimator but it assures that the filter factors of PLS are very strange and complicated to be interpreted because the PLS estimator is a non linear estimator of the response variable.

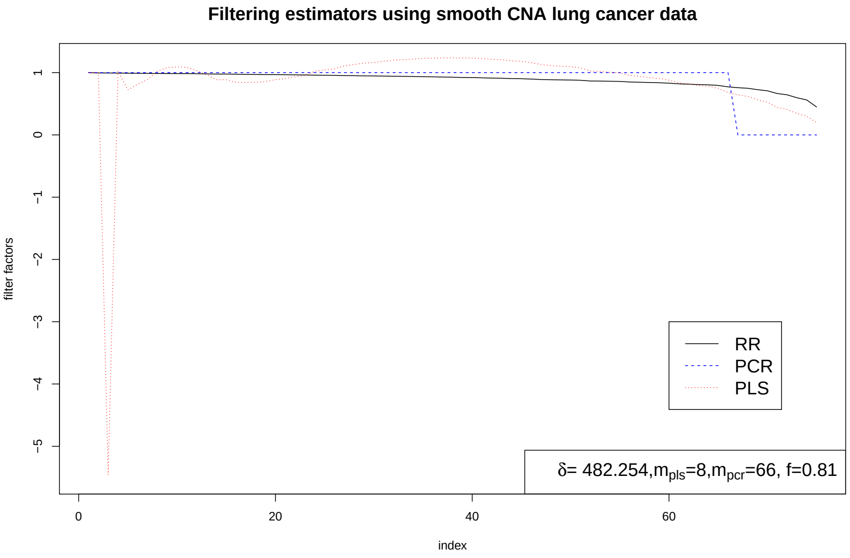


Figure 3.6: filter factors for RR, PCR, PLS for smooth CNA lung cancer data where $p > n$ with a binary response using 8 components.

Figure 3.6 shows another case where ω_6^{pls} using 8 components of the PLS estimator is negative. This is also an interesting phenomena in the filter factors in the PLS estimator that needs more investigation deeply as we will do in Chapter 4.

3.5 Interpretation of the filter factors in the PLS estimator

In order to start thinking about the filter factors in the PLS estimator, we will try to interpret them by writing each component of the filter factor in terms of each others. In particular, we try to find the connection between d_i^2 and μ_j^2. After centring X, we can write the X^TX using eigenvalue decomposition as shown in Equation (3.33).

$$X^TX = VD^2V^T. \tag{3.33}$$

Also, since X is centred, we can write X^TX in terms of the variance of X as

$$X^TX = (n-1)S_{xx}, \tag{3.34}$$

where S_{xx} is the variance-covariance matrix for X variables. By combining Equations (3.33) and (3.34), we will have that the eigenvalues of X^TX can be calculated in another way as a function of the variance-covariance matrix of the X variables, and since the eigenvectors (the columns of matrix V) are orthogonal, then $V^TV = I_{n-1}$. Thus, the eigenvalues, of X^TX, can be written as

$$D^2 = (n-1)V^TS_{xx}V, \tag{3.35}$$

On the other hand, we can write $W_m^TX^TXW_m$ using eigenvalue decomposition by putting $XW_m = B$, so $W^TX^TXW = B^TB$. By applying the eigenvalue decomposition on B^TB, we will have

$$W_m^TX^TXW_m = B^TB = K_mN_{(m)}^2K_m^T, \tag{3.36}$$

where $K_m^TK_m = K_mK_m^T = I_m$, and $N_{(m)}^2$ is a diagonal matrix with the eigenvalues $\mu_1^2 \geq \mu_2^2 \geq \cdots \geq \mu_m^2$ on the diagonal. The columns $k_1, \ldots, k_m$ of K_m are denoted the left and right eigenvectors. Using Equation (3.36), we can write $W_m^TX^TXW_m$ as a function of the eigenvalues of X^TX as

$$W_m^TX^TXW_m = W_m^TVD^2V^TW_m. \tag{3.37}$$

By combining Equations (3.36) and (3.37), we can see that $N^2_{(m)}$ is a function of D^2, which means that the eigenvalues of $W_m^T X^T X W_m$ are just functions of the eigenvalues of $X^T X$ as in Equation (3.34), and since columns of K_m are orthogonal, then $K_m^{-1} = K_m^T$, and

$$N^2_{(m)} = K_m^T W_m^T V D^2 V^T W_m K_m. \tag{3.38}$$

Therefore, since that $D^2 = (n-1)V^T S_{xx} V$, we can write the eigenvalues of $W_m^T X^T X W_m$, $N^2_{(m)}$, in terms of the variance of X, as in Equation (3.39).

$$N^2_{(m)} = (n-1) K_m^T W_m^T V V^T S_{xx} V V^T W_m K_m, \tag{3.39}$$

since $VV^T = I_p$, then we can rewrite Equation (3.39) as

$$N^2_{(m)} = (n-1) K_m^T W_m^T S_{xx} W_m K_m. \tag{3.40}$$

Hence, we can see that rank$(W_m^T X^T X W_m) \leq$ rank$(X^T X) \equiv m \leq n-1$.

3.6 Filter factors of PLS estimator

The shrinkage structure of PLS estimator has been investigated and proved by researchers such as Lingjaerde & Christophersen (2000) and Butler & Denham (2000) in low-dimensional data ($n \geq p$). Here, we investigate the filter structure of PLS in high dimensional data ($p > n$). We add one more structure which introduces negative filter factors (NFF). Denote the filter factors as before by ω_i (Lingjaerde & Christophersen, 2000). Below are some results that Lingjaerde & Christophersen (2000) showed in the low-dimensional case and these properties are applied for the high-dimensional case as well.

- $\omega_r^{(m)} \leq 1$ for all number of components m, where r is the rank of the X matrix, which is $r = (n-1)$ in the high dimensional case.

- $\omega_1^{(m)} \geq 1$ for $m = 1, 3, 5, \ldots$ odd number because it is like a telescoping series or alternatively.

- $\omega_1^{(m)} \leq 1$ for $m = 2, 4, 6, \ldots$ even number because it is like a telescoping series or alternatively.

- For the other filter factors $\omega_i^{(m)}$, $\quad i = 2, 3, \ldots, m-1$, one may have either $\omega_i^{(m)} \leq 1$ or $\omega_i^{(m)} \geq 1$. Looking at Equation (3.31), we can see that if $d_i^2 \leq \mu_j^{2(m)}$, then $0 \leq \omega_i^{(m)} \leq 1$ (Lingjaerde & Christophersen, 2000).

- Some filter factors can be negative ($\omega_i^{(m)} < 0$). This occurs when

$$\prod_{j=1}^{m}(1 - \frac{d_i^2}{\mu_j^{2(m)}}) > 1, \qquad i = 1, 2, \ldots, (n-1).$$

The filter factors of PLS estimator are negative and can be seen in the NIR data in some components and some ω_i as follows: $\omega_8^{(9)}$, $\omega_{15}^{(15)}$, $\omega_{15}^{(16)}$, $\omega_{(16)}^{(16)}$, $\omega_{17}^{(16)}$, $\omega_{16}^{(19)}$, $\omega_{17}^{(20)}$, $\omega_{21}^{(26)}$. Also, in CNA lung cancer data, we have $\omega_3^{(6)}$, $\omega_3^{(8)}$ are negative.

As can be seen from Figures 3.4 and 3.6, some of the filter factors can be negative for the PLS filter factors for specific components in both data sets. We will investigate more deeply about the negative filter factors of the PLS estimator (NFF) along with the reasons in Chapter 4.

3.7 Discussion

In this chapter, we investigate the shrinkage factors for three popular methods RR, PCR and PLS where the response variable is normal and binary. We have shown these shrinkage factors for all three methods depend on the number of components (m) for PCR and PLS estimators, and the ridge parameter (δ) for RR estimator. We also showed that based on two artificial examples for low-dimensional data by comparing their shrinkage factors to each other.

We have been able to modify the results of the "shrinkage factors" in the low-dimensional case to "filter factors" in the high-dimensional case. We write all three methods (RR, PCR and PLS) in a common form, and showed how they are derived in every method. We have also investigated the key changes in the filtering comparing to the shrinkage for the high dimensional data. Then, we interpret the filter factors of the PLS regression estimator particularly. We write the eigenvalues of X^TX (d_i^2) and $W_m^TX^TXW_m$ (μ_j^2) for each component (m) in terms of each others.

Finally, we have modified some of the shrinkage structure of PLS to the filter factors of PLS regression estimator. We provided the filter structure of PLS estimator for high dimensional data and showed that from a real data with different types of the response variable y. Most of the properties for the shrinkage factors for PLS in low-dimensional case are still available in the high-dimensional case. The filter factors in the high-dimensional case for RR and PCR are still between zero and one whereas in PLS oscillate around one and may have negative values. It is interesting that filter factors of PLS regression estimator can be negative and we will discuss and explore that in detail more in Chapter 4.

Chapter 4

Negative Filter or Shrinkage Factors in the PLS Estimator

4.1 Overview

In Chapter 3, we introduced filter factors for three popular methods of shrinkage methods. For each of the low and high dimensional data cases, we provided a common formula. Looking at the filter factors of high dimensional data with real data examples, some of the filter factors of the PLS estimator have negative values. Therefore, it is very important to study the filter factors of the PLS estimator in more depth, especially trying to determine or describe when this may occur.

Let X be the data matrix, n be the number of samples, p be the number of variables, y be the response vector, W be the matrix representing the covariance between X and y, and let m be the number of components. Lingjaerde & Christophersen (2000) have proposed theoretical results to describe the shrinkage structure of the PLS estimator where $n \geq p$.

So far there has been little attention given to explaining the circumstances leading to negative filter factors. Our attention is drawn to more investigation on the NFF. The causes of having NFF in each component for different settings of the structure of the data, will be of focus. The shrinkage structure of PLS was discussed in detail more by Lingjaerde & Christophersen (2000), Butler & Denham (2000) Rosipal & Krämer (2006) with some properties of the shrinkage factors of PLS estimator. Moreover, it has been mentioned that NFF may exist without any further investigation e.g Lingjaerde & Christophersen (2000) and Butler & Denham (2000).

In this chapter we investigate the potential occurrence of NFF. Furthermore, we illustrate NFF using simulations for both high and low-dimensional data with consideration of various conditions on the eigenvalues of X^TX. These simulations confirmed some of the potential causes of NFF. Moreover, we use a small example for different structures of X^TX, and different settings of the eigenvalues of X^TX to show when NFF may occur.

This chapter is organised as follows. Section 4.2 presents the relationship between the eigenvalues of $W_m^TX^TXW_m$ and X^TX, where W_m is the X-weight matrix used in the NIPALS algorithm in Chapter 2, and some properties of these eigenvalues. Explaining when NFF of PLS estimator occur in some components is discussed in Section 4.3. Giving some simulation to show the reasons of having NFF will be in Section 4.4 for the low and high dimensional cases. In Section 4.5 there are some examples to give wider understanding where the NFF might occur, for different structures of the variance-covariance matrix of X. Finally, in Section 4.6 some discussion on this chapter is summarised with highlighted points for the NFF.

4.2 Relationship between the eigenvalues of two matrices

Let d_i^2 be the eigenvalues of X^TX, and $\mu_j^{2(m)}$ be the eigenvalues of $W_m^TX^TXW_m$. Note that $\mu_j^{2(m)}$ represents the j-th eigenvalue of $W_m^TX^TXW_m$ for $m = 1, 2, \dots M$, where $M \leq r$, and $r = \min(n-1, p)$. It is important to know the type of the both matrices where X^TX is a symmetric matrix, and $W_m^TX^TXW_m$ is a tridiagonal matrix (Krämer, 2007). The vectors in the matrix W_m are orthogonal and every vector in W_m has magnitude of one which means they are orthonormal. This matrix is tridiagonal because it is the transformation of a symmetric matrix with orthogonal vectors, as $W_m^TW_m = I_m$. We can see that each d_i^2 is just a value on the diagonal of the matrix in Equation (3.31) as

$$d_i^2 = (n-1)V_i^TS_{xx}V_i, \tag{4.1}$$

where i is the i-th column eigenvector of matrix V. Also, for that each μ_j^2 is a single value of the diagonal matrix, $N_{(m)}^2$, in Equation (3.32):

$$\mu_j^{2(m)} = K_j^T W_m^T S_{xx} W_m K_j, \tag{4.2}$$

where j represents the j-th column of matrix K. Hence, $\omega_i^{(m)}$ is just the relation between Equations (4.1) and (4.2). The eigenvalues, μ_j^2, are going to be the same as the eigenvalues, d_i^2, if all the subspaces of the space are spanned by W_m. Furthermore, looking at Equations (4.1) and (4.2), it can be seen that if $m = \min(n-1, p)$, then $d_i^2 = \mu_j^2$ for all $i, j = 1, 2, \ldots, r$, where r is the rank of X. That means $V_i^T S_{xx} V_i = K_j^T W_m^T S_{xx} W_m K_j$. The investigation for the NFF of the PLS estimator is based on the relation between the covariance vector between X and y, and the variance-covariance matrix of X variables.

4.2.1 Simulation study

To see the relation between the eigenvalues of $W_m^T X^T X W_m$ (μ_j^2) and the eigenvalues $X^T X$ (d_i^2), we simulate a data set with $n = 20$ and $p = 50$. We use the general regression model for the simulation.

X is generated independently from a normal distribution with zero mean and a constant variance is equal to one, then X and y are centred in order to not have an intercept, $\beta \sim \mathcal{N}(0, \sigma^2 = 1)$, and $\epsilon \sim \mathcal{N}(0, \sigma^2 = 1)$.

Figure 4.1 shows that the μ_j^2 and d_i^2 values are getting close to each other as the number of components of PLS estimator is increased, and as the rank of X reaches m = $n - 1$ = 19 components. Looking at the top left panel, for 5 components there are only 5 eigenvalues of $W_m^T X^T X W_m$. Furthermore, only the first eigenvalue in both matrices are the same and as the number of components is increased, the number of equivalent eigenvalues of those matrices is increased. Reaching the rank of X as in the bottom right panel we can see that all the μ_j^2 are equal to d_i^2.

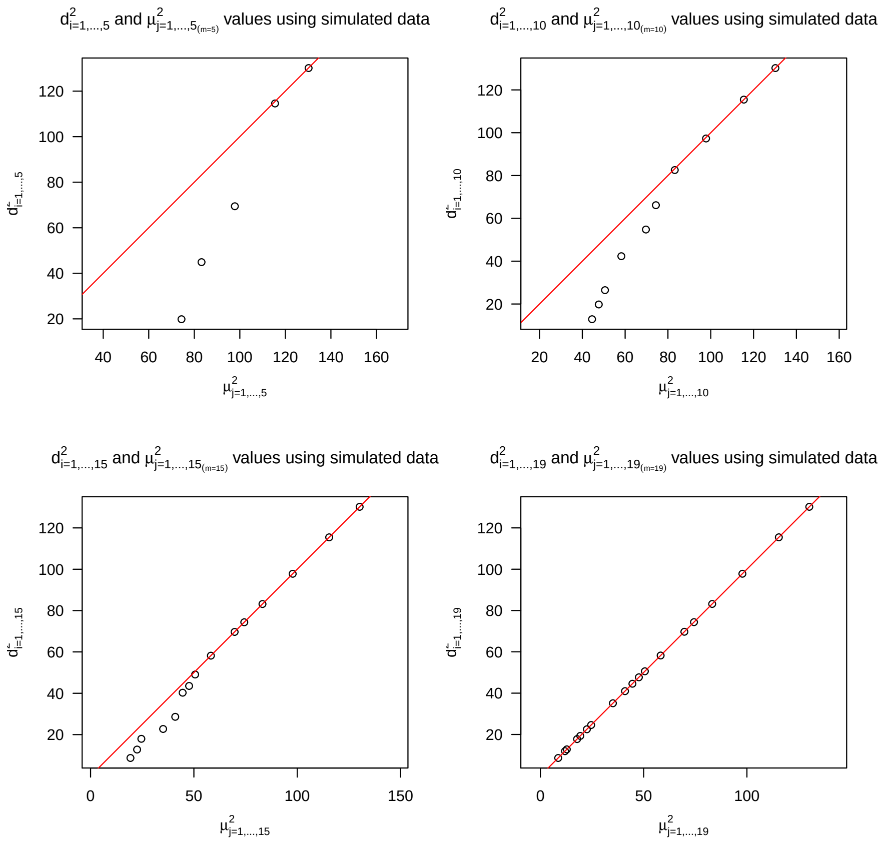


Figure 4.1: The relation between d_i^2 in Equation (4.1) and μ_j^2 in Equation (4.2) using a simulated data where $p > n$.

4.2.2 Some properties of the $\mu_j^{2(m)}$ eigenvalues

Let $m = 1, 2, \ldots, M$ represent the number of components. The eigenvalues of $W_m^T X^T X W_m$, $\mu_1^2 \geq \mu_2^2 \geq \cdots \geq \mu_M^2$, are commonly called Ritz values (Saad, 1992). Some important properties for those Ritz values are given by Lingjaerde & Christophersen (2000):

1. $d_1^2 \geq \mu_1^{2(m)} > \mu_2^{2(m)} > \cdots > \mu_m^{2(m)} \geq d_{(n-1)}^2$.

   In other words, the largest eigenvalue of $X^T X$ is larger than the largest eigenvalue if m is not equal to the rank of X, but if the rank of X equals m then those eigenvalues are equivalent, and the same for the last smallest eigenvalues. $\mu_m^{2(m)}$ means the smallest eigenvalue of the matrix $W_m^T X^T X W_m$.

2. $d^2_{p-m+i} \leq \mu_i^{2(m)} < d_i^2, i = 1, 2, \ldots, m.$

3. Between any two of the eigenvalues $\mu_{j+1}^{2(m)}$, $\mu_j^{2(m)}$, there is at least one of d_i^2.

4. The Ritz values $\{\mu_j^{2(m)}\}_{j=1}^m$ and $\{\mu_j^{2(m+1)}\}_{j=1}^{m+1}$ separate each other.
   $\mu_1^{2(m+1)} > \mu_1^{2(m)} > \mu_2^{2(m+1)} > \mu_2^{2(m)} > \cdots > \mu_m^{2(m)} > \mu_{m+1}^{2(m+1)}$

4.3 Example showing NFF in the PLS estimator

Recall that Equation (3.30), Ω^{pls} is a diagonal matrix of ω_i^{pls}, with dimension $((n-1) \times (n-1))$, and the values on the diagonal are called filter factors, ω_i^{pls}, as given in Equation (3.31). For a given number of component say m, we may have negative values of the filter factors $\omega_i^{(m)}$, because $\prod_{j=1}^m (1 - \frac{d_i^2}{\mu_j^{2(m)}}) > 1, \qquad i = 1, 2, \ldots, n-1.$ Thus, we have $\omega_i^{(m)} < 0$, where $m = 2, 3, \ldots, n-1$, if and only if $\sum_{j=1}^m I[d_i^2 > \mu_j^{2(m)}]$ for even number of the $j = 1, \ldots, m$ occurrences, and that $d_i^2 \not\approx \mu_j^{2(m)}$, or $d_i^2 \gg \mu_m^{2(m)}$. From the figures of the filter factors for the NIR and CNA data sets in Chapter 3, it was found that there are some cases where NFF exist. Some plots of NFF for NIR data and CNA data are illustrated to show the potential causes as in the Figures 4.2 and 4.3.

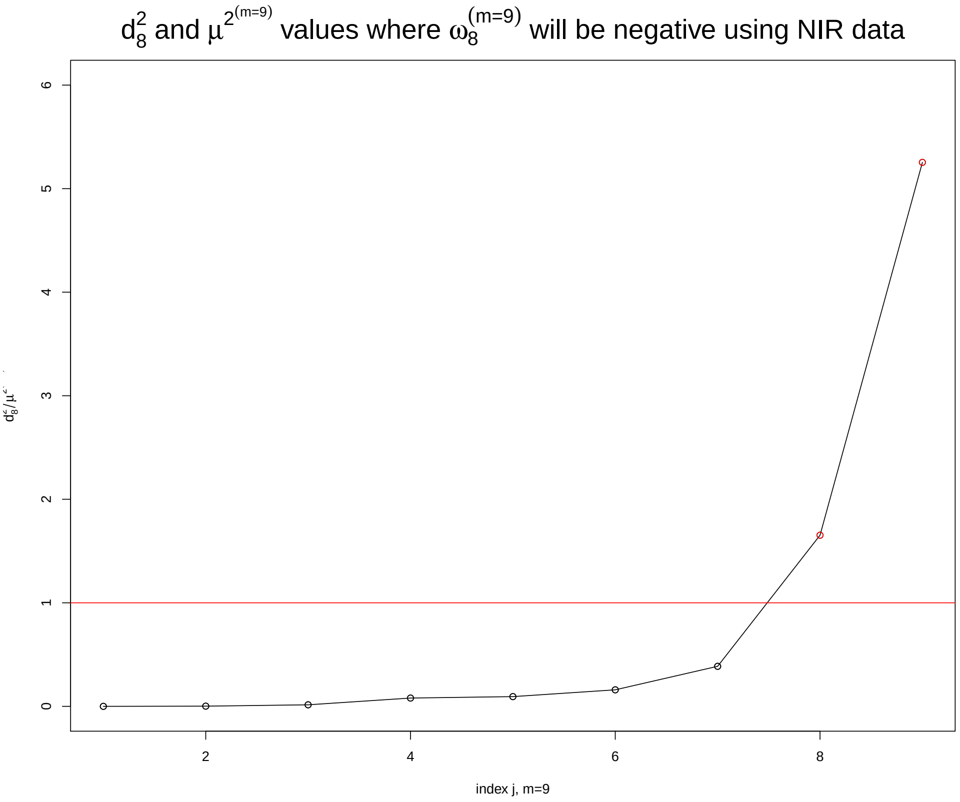


Figure 4.2: The ratio between d_8^2 and $\mu_{j=1,\dots,9}^{2(9)}$ values using (NIR data), using 9 components, $(m = 9)$.

It can be seen from Figure 4.2 that there are two ratios of the $d_8^2/\mu_j^{2(9)}$, for $j = 8, 9$ are greater than one, and none of these ratios is close to one or on the red horizontal line. Thus, NFF occurs, since the product is greater than one because of the number of those ratios is even (and none of them are close to one).

Figure 4.3 shows another example using 20 components in which $\omega_{17}^{(m=20)} < 0$, because there are four ratios, between d_{17}^2 and $\mu_{j=1,\dots,20}^{2(20)}$ which are greater than one. That means, there is even number of occurrences of $j = 17, 18, 19, 20$, where they are less than d_{17}^2.

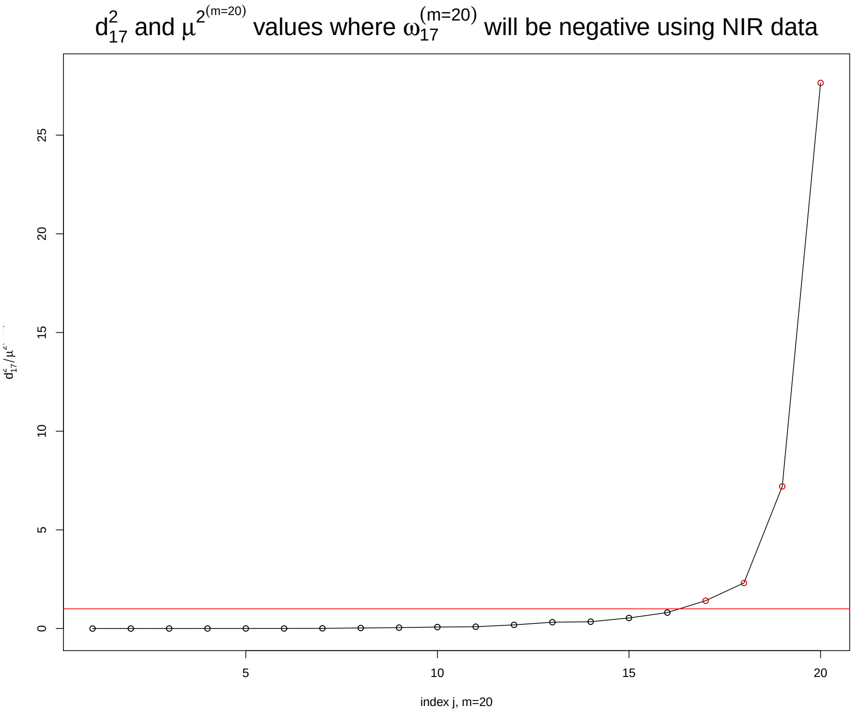


Figure 4.3: The ratio between d^2_{17} and $\mu^{2(20)}_{j=1,\ldots,20}$ values using (NIR data), using 20 components, $(m = 20)$.

To illustrate this is one of the cases (that was found when calculating the filter factors for the NIR data) has NFF. The idea behind this can be seen from Figure 4.4.

It has been observed from the investigation of NFF for different cases that the d^2_i is larger than $\mu^{2(m)}_j$ for even number of occurrences of j is enough to have NFF. There are some cases in the filter factors that have even number of occurrences of j, but they are not NFF. In these cases, we do not have NFF because they do not satisfy this $d^2_i \not\approx \mu^{2(m)}_j$, or $d^2_i \gg \mu^{2(m)}_m$ for that i, j and for a given an m component. In the following figures we utilise an example to explain this observation.

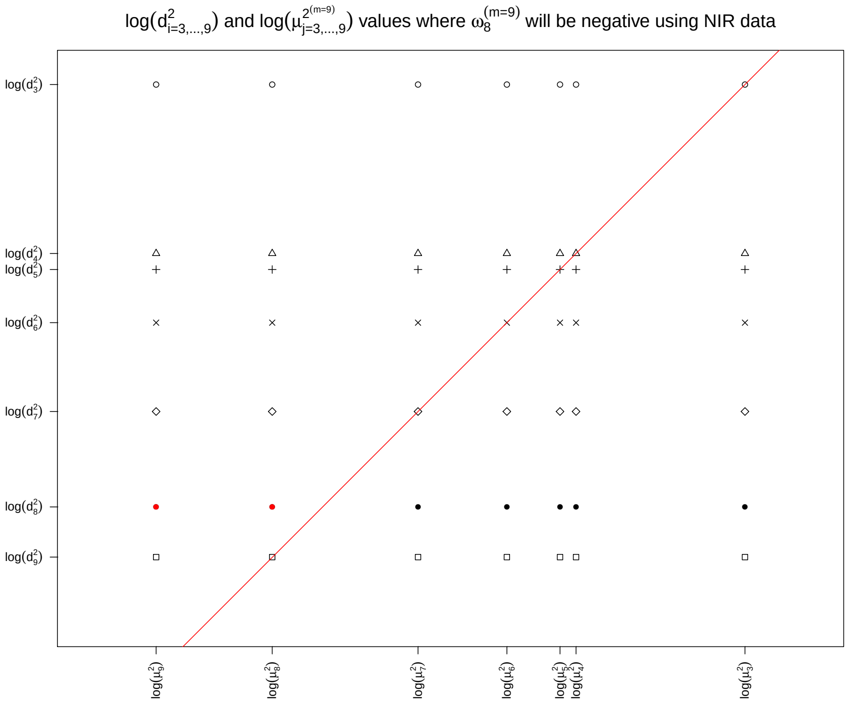


Figure 4.4: $\log(d_i^2)$ and $\log(\mu_j^{2(9)})$ values using (NIR data), using 9 components, $(m = 9)$.

Figure 4.4 shows the log of d_i^2 and $\mu_j^{2(9)}$ values for $i = j = 3, \ldots, 9$ using the (NIR data) where the number of components $m = 9$. Plotted on the X axis are the $\mu_j^{2(9)}$ for $j = 4, 5, 6, 7, 8, 9$, just to clarify the picture, and on the Y axis are the d_i^2 for $i = 4, 5, 6, 7, 8, 9$, again in order to give a clear picture of the target. It can be seen that the two red dots are corresponding to the d_8^2, $\mu_8^{2(9)}$ and $\mu_9^{2(9)}$, where they are located above the identity line which indicates $d_8^2 > \mu_8^{2(9)}$ and $d_8^2 > \mu_9^{2(9)}$. None of the rest of $\mu_j^{2(9)}$ for $j = 4, 5, 6, 7$ is equal to d_8^2. Thus, $\omega_8^{(9)} < 0$.

On the other hand, looking at the other d_i^2 for $i = 4, 5, 6, 7$ there are some cases where the $d_i^2 > \mu_j^{2(9)}$ for even number of the j occurrences, but there is at least one of the d_i^2 is equal to $\mu_j^{2(9)}$ as i and j vary. Therefore, in this case the filter factors are positive.

4.3.1 Conditions for NFF occurrence

Let r br the rank of X, so $r = n - 1$. If $m = 1$, there is only one component and since the number of components must be even, it is not possible to have any NFF. For components larger than one, there are some conditions which are based on the relation between the eigenvalues of $W_m^T X^T X W_m$ and $X^T X$ matrices. We have proposed those conditions to show when NFF occurs in each component. We report the conditions for NFF for small components, and for larger components, it can be shown, but the calculations are long.

- for m=2, $\omega_i < 0$ if and only if

$$d_i^2 > \sum_{j=1}^{2} \mu_j^2 = \mu_1^2 + \mu_2^2, \quad \text{for any} \quad i = 1, 2, \ldots, r.$$

- for m=3, $\omega_i < 0$ if and only if

$$d_i^{2(3-1)} < d_i^{2(3-2)} \sum_{j=1}^{3} \mu_j^2 - \sum_{1 \le j < s \le 3} \mu_j^2 \mu_s^2, \quad \text{for any} \quad i = 1, 2, \ldots, r.$$

- for m=4, $\omega_i < 0$ if and only if

$$d_i^{2(4-1)} > d_i^{2(4-2)} \sum_{j=1}^{4} \mu_j^2 - d_i^{2(4-3)} \sum_{1 \le j < s \le 4} \mu_j^2 \mu_s^2 + \sum_{1 \le j < s < l \le 4} \mu_j^2 \mu_s^2 \mu_l^2, \quad \text{for any} \quad i = 1, 2, \ldots, r.$$

- for m=5, $\omega_i < 0$ if and only if

$$d_i^{2(5-1)} < d_i^{2(5-2)} \sum_{j=1}^{5} \mu_j^2 - d_i^{2(5-3)} \sum_{1 \le j < s \le 5} \mu_j^2 \mu_s^2 + d_i^{2(5-4)} \sum_{1 \le j < s < l \le 5} \mu_j^2 \mu_s^2 \mu_l^2 - \sum_{1 \le j < s < l < q \le 5} \mu_j^2 \mu_s^2 \mu_l^2 \mu_q^2, \quad \text{for any} \quad i = 1, 2, \ldots, r.$$

- for m=6, $\omega_i < 0$ if and only if

$$
\begin{aligned}
d_i^{2(6-1)} > d_i^{2(6-2)} \sum_{j=1}^{6} \mu_j^2 - d_i^{2(6-3)} \sum_{1\le j<s\le 6} \mu_j^2\mu_s^2 + d_i^{2(6-4)} \sum_{1\le j<s<l\le 6} \mu_j^2\mu_s^2\mu_l^2 \\
-d_i^{2(6-5)} \sum_{1\le j<s<l<q\le 6} \mu_j^2\mu_s^2\mu_l^2\mu_q^2 \\
+ \sum_{1\le j<s<l<q<r\le 6} \mu_j^2\mu_s^2\mu_l^2\mu_q^2\mu_r^2, \\
\text{for any} \quad i = 1, 2, \ldots, r.
\end{aligned}
$$

From the above itemisation and the condition in each component, we can say that if m is even, then $d_i^{2(m-1)}$ has the greater sign of the inequality. When m is odd $d_i^{2(m-1)}$ occurs as a lower bound which means it is smaller than the right hand side of the inequality. Also, we can see that the condition is based on $d_i^{2(m-1)}$ and some combinations of μ_j^2. They are separated with the inequality, which represents the difficulty of the PLS regression filter factors to be interpreted. Otherwise, we could be able to say something about the relation between the covariance between X and y vector, and the variance-covariance matrix of X, in which NFF will (may) occur.

For $m = 2$, and any $i = 1, 2, \ldots, r$, $\omega_i < 0$ if and only if $d_i^2 > \sum_{j=1}^{2} \mu_j^2 = \mu_1^{2(2)} + \mu_2^{2(2)}$.

To show the proof of the condition for two components, $m = 2$, recall the filter factors in the PLS estimator $\omega_i^{(m)}$ in Chapter 3. Substituting $m = 2$ in $\omega_i^{(m)} = 1 - \prod_{j=1}^{m}(1 - \frac{d_i^2}{\mu_j^{2(m)}})$, we have

$$1 - [1 - \frac{d_i^2}{\mu_1^2} - \frac{d_i^2}{\mu_2^2} + \frac{d_i^4}{\mu_1^2\mu_2^2}]. \tag{4.3}$$

Solving and simplifying Equation (4.3) and using the inequality for $\omega_i^{(m)} < 0$, we get

$$\frac{d_i^2}{\mu_1^2} + \frac{d_i^2}{\mu_2^2} - \frac{d_i^4}{\mu_1^2\mu_2^2} < 0.$$

After simplifying the inequality above with respect to the power of d_i, we get

$$\frac{d_i^2}{\mu_1^2\mu_2^2} > \frac{\mu_1^2 + \mu_2^2}{\mu_1^2\mu_2^2}.$$

Then,

$$d_i^2 > \mu_1^2 + \mu_2^2$$

Also, for three components we use the same approach and starting point. For the condition and the inequality to have NFF, we substitute $m = 3$ in $\omega_i^{(m)} = 1 - \prod_{j=1}^{m}(1 - \frac{d_i^2}{\mu_j^{2(m)}})$. Thus, we have

$$\omega_i^{(2)} = 1 - [1 - \frac{d_i^2}{\mu_1^2} - \frac{d_i^2}{\mu_2^2} - \frac{d_i^2}{\mu_3^2} + \frac{d_i^4}{\mu_1^2\mu_2^2} + \frac{d_i^4}{\mu_1^2\mu_3^2} + \frac{d_i^4}{\mu_2^2\mu_3^2} - \frac{d_i^6}{\mu_1^2\mu_2^2\mu_3^2}]. \tag{4.4}$$

Solving Equation (4.4). Say, if $\omega_i^{(m)} < 0$, we obtain

$$\frac{d_i^2}{\mu_1^2} + \frac{d_i^2}{\mu_2^2} + \frac{d_i^2}{\mu_3^2} - \frac{d_i^4}{\mu_1^2\mu_2^2} - \frac{d_i^4}{\mu_1^2\mu_3^2} - \frac{d_i^4}{\mu_2^2\mu_3^2} + \frac{d_i^6}{\mu_1^2\mu_2^2\mu_3^2} < 0.$$

With some simplification of the inequality above, we get

$$d_i^4\left(\frac{\mu_1^2 + \mu_2^2 + \mu_3^2}{\mu_1^2\mu_2^2\mu_3^2}\right) > d_i^2\left(\frac{\mu_1^2\mu_2^2 + \mu_1^2\mu_3^2 + \mu_2^2\mu_3^2}{\mu_1^2\mu_2^2\mu_3^2}\right) + d_i^6\left(\frac{1}{\mu_1^2\mu_2^2\mu_3^2}\right).$$

Finally with some arrangements in the inequality and some terms cancel out, we shall have

$$d_i^4 < d_i^2\left(\mu_1^2 + \mu_2^2 + \mu_3^2\right) - \left(\mu_1^2\mu_2^2 + \mu_1^2\mu_3^2 + \mu_2^2\mu_3^2\right).$$

Therefore, we can simplify that as

$$d_i^{2(3-1)} < d_i^{2(3-2)} \sum_{j=1}^{3} \mu_j^2 - \sum_{1 \le j < s \le 3} \mu_j^2\mu_s^2$$

4.4 Some examples with a normal response

Although the motivation of investigating NFF more deeply comes from high dimensional and real data sets as NIR and CNA lung cancer, it is worth to look at the NFF in the low dimension through a simulation and the results are shown in Section 4.4.1. Also, for the high dimensional case, some results are presented in Section 4.4.2.

4.4.1 Simulations for low dimensional data

The general model for this simulation can be seen as a univariate multiple linear regression model as

$$y = X\beta + \epsilon_y. \tag{4.5}$$

We generate the X data from a multivariate normal distribution with mean zero, and a variance-covariance matrix Σ_x, with $\rho = 0.999$ between all variables, with $n = 100$ and $p = 6$. The X matrix is then decomposed using singular values decomposition (SVD) into $X = UDV^T$, then the singular values in the diagonal of the matrix D are replaced by the square roots of new sets of eigenvalues, such that the sum of the eigenvalues are equal to the sum of the eigenvalues of (NIR data) since NFF was seen firstly in that data as in Chapter 3. The new sets of the eigenvalues are utilised to create 5 different sets, called settings. The first setting of $d^2 = (27649, 27648, 0.9, 0.8, 0.7, 0.6)$, the second setting of $d^2 = (28649, 26648, 0.9, 0.8, 0.7, 0.6)$, the third setting of $d^2 = (47649, 7648, 0.9, 0.8, 0.7, 0.6)$, the fourth setting of $d^2 = (55296, 1, 0.9, 0.8, 0.7, 0.6)$, and the fifth setting of $d^2 = (11716.67, 10716.67, 9716.67, 8716.67, 7716.66, 6716.66)$. Figure 4.5 shows the replaced eigenvalues that will be used to get the X matrix. The five settings of d^2 where divided based on the average correlation between the x_i and x_j for $i \neq j$, where x_i and x_j are variables in X. The average correlation is very low as in the fifth setting, which means the data is independent as can be seen in the eigenvalues plot Figure 4.5, and it is around 0.6 as in the first and the second settings. The third setting has a high average correlation, and a very high average correlation equals to 0.9999 in the fourth setting.

Replacing the eigenvalues by all five different settings allow us to control the eigenvalues of X^TX, d_i^2. Since $\omega_i^{(m)}$ is a function of both d_i^2 and $\mu_j^{2(m)}$, then NFF depends on those eigenvalues. The eigenvalues of $W_m^T X^T X W_m$, $\mu_j^{2(m)}$, are calculated based on W_m which is the covariance of X and y for m components. The covariance of X and y can be controlled from the σ_β^2. We generate β from a normal distribution with mean zero and σ_β^2 which can take any value of different ranges of σ_β^2 as $(100, 5, 1, 10^{-1}, 10^{-3}, 10^{-5})$. We investigate how σ_β^2 can affect the proportion of datasets which have at least one NFF over all components over the simulation settings since the covariance between X and y can be controlled by σ_β^2. As a result, σ_β^2 have been chosen to have different range of values in order to vary the covariance of X and

y. The error term ϵ_y is generated from a normal distribution with mean zero and variance equals to 1. We simulate X data matrix 10000 times where the singular values, on the diagonal of the matrix D are fixed over all simulation, but the left and right singular vectors, U and V respectively, are changed as X is simulated.

The NIPALS algorithm is applied to calculate the W_m. These are used to calculate $\mu_j^{2(m)}$ as in Equation (4.2). The filter factors of the PLS regression estimator, $\omega_i^{(m)}$, are calculated using Equation (3.20).

Figures 4.6, 4.7, and 4.8 are reported for our approach to show when NFF may happen more and highlight a key finding for having more proportion of NFF in simulations. The proportion of NFF is the number of simulated data that have at least one NFF in all components divided by the total number of simulated data. NFF may occur starting from the second component, at least two components needed ($m \geq 2$), and NFF cannot happen if $m = i = \min(n-1, p)$. This is because the eigenvalues d_i^2 and $\mu_j^{2(m)}$ are the same since $m = \min(n-1, p)$ Section 4.2. Under some specific circumstances and some settings of d^2 and σ_β^2 values, NFF exists over all components and in some components, especially, at the largest eigenvalues of $X^T X$. This can be seen in Figures 4.7 and 4.8 for small and large σ_β^2 respectively.

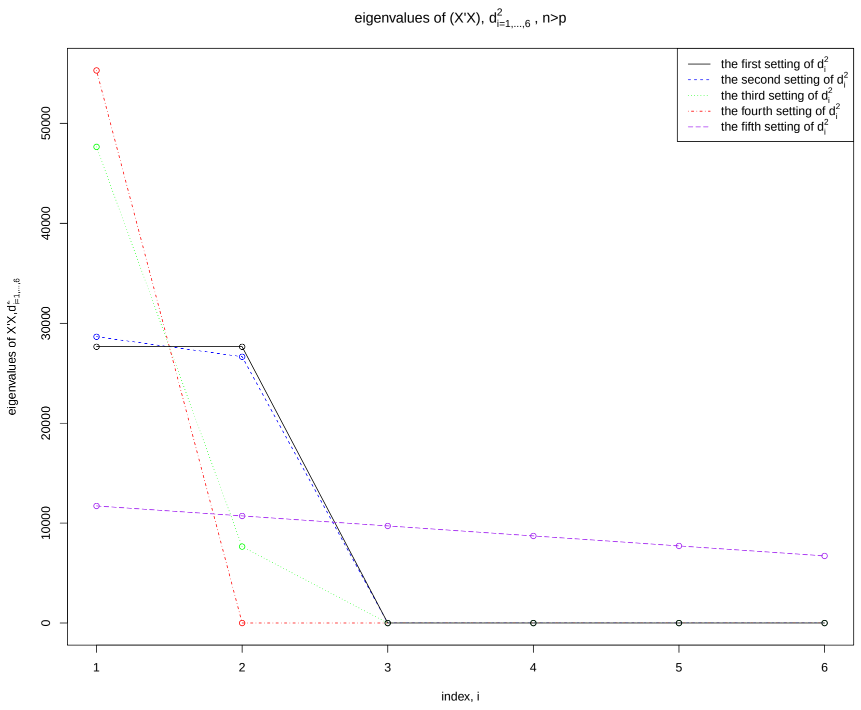


Figure 4.5: The replaced nonzero eigenvalues of $X^T X$ for the different settings.

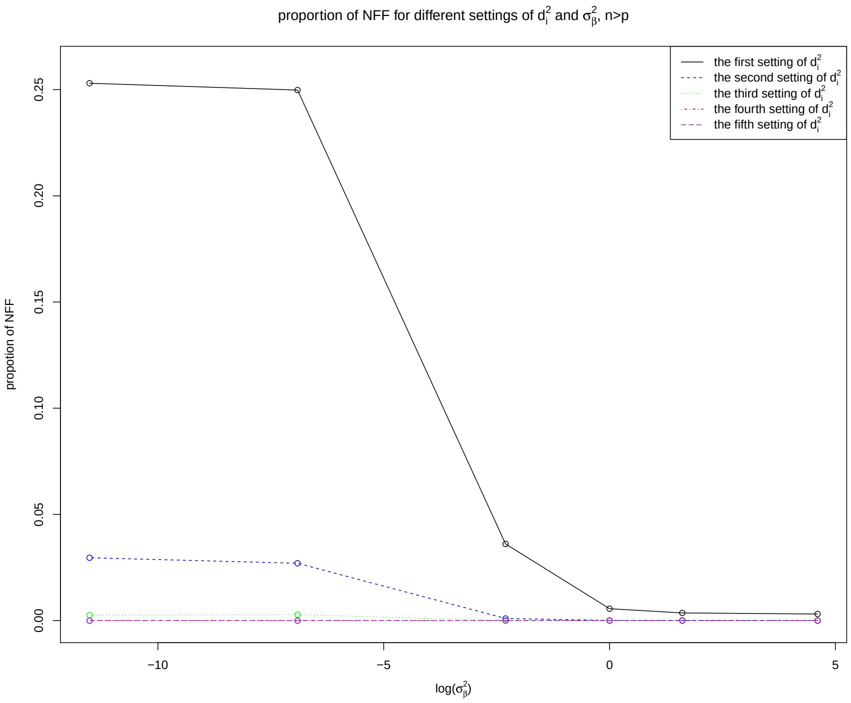


Figure 4.6: The proportion of datasets which have at least one NFF over all components having NFF for overall components for different σ_β^2 for different datasets.

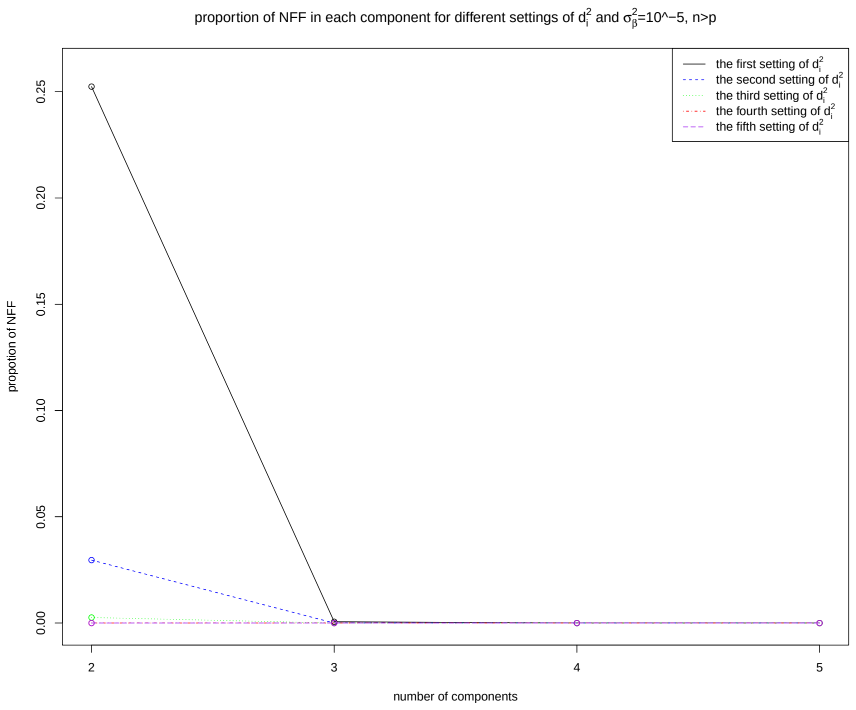


Figure 4.7: The proportion of datasets which have at least one NFF in every component for different datasets using small value of $\sigma_\beta^2 = 10^{-5}$.

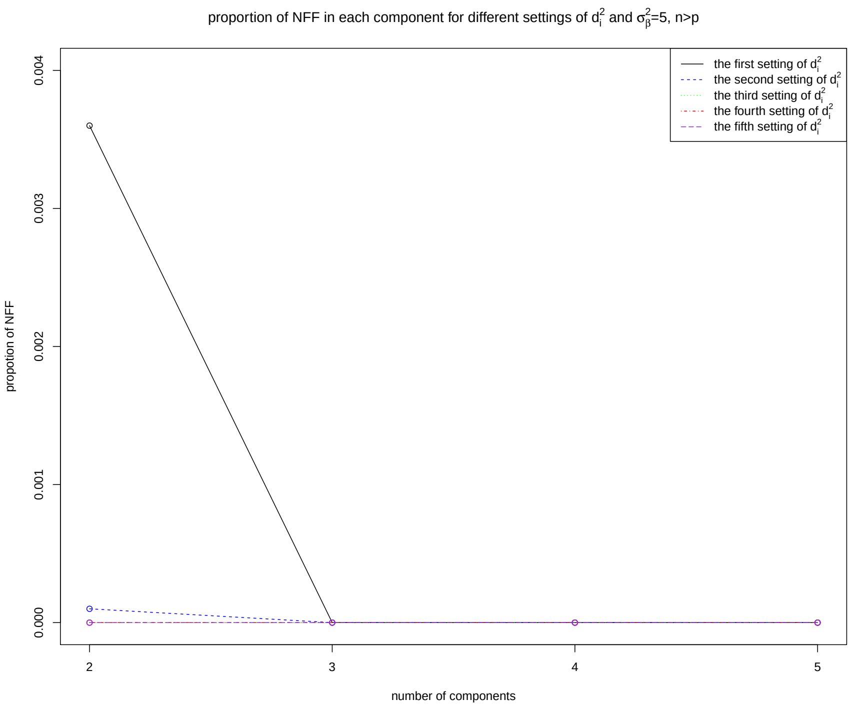


Figure 4.8: The proportion of datasets which have at least one NFF in every component for different data sets after replacing the eigenvalues by by the first, second, third, fourth, and fifth setting of d^2, using a large value of $\sigma_\beta^2 = 5$.

Looking at Figure 4.6, it can be seen that as the log of σ_β^2 gets smaller, the proportion of NFF in all components gets larger. Moreover, the average of all the values below the diagonal of the correlation matrix of X is around 0.6 when using the first setting of d^2. It is very small when using the fifth setting of d^2 which means the variables are independent as can be seen from the eigenvalues plot in Figure 4.5. From the correlation point of view, It is more likely to have NFF for those data sets that are correlated.

Furthermore, in the first four settings of the d^2, where only the first and second largest eigenvalues are the only different and the rest are the same, the first setting has a higher proportion of NFF comparing to the others. As a result, since the first eigenvalue must be larger than the second eigenvalue, the ratio between the first and the second eigenvalues is very small close to one, then this setting of d^2 is associated to have higher proportion of having at least one NFF over all components.

Moreover, Figure 4.7 shows the proportion in each possible component that may have NFF for small $\sigma^2_\beta = 10^{-5}$. Figure 4.8 shows the proportion of NFF in every possible component for large $\sigma^2_\beta = 5$. Although these two figures do not have clear and enough information to identify the reason in each component why the we have high proportion of NFF, most of the occurrences of NFF over all components comes from the second component. This can be seen in Figure 4.7 and Figure 4.8 where the first setting is used to construct X matrix.

In summary, there is a connection between the structure of the data and the proportion of datasets which have at least one NFF over all components as seen in the fifth setting (where all predictors are independent) compared to other settings (where the predictors are correlated). If the data is correlated, then there is a potential occurrence of NFF as seen above in Figure 4.6. In contrast, if the data is independent as given by the fifth setting, there is no chance to have NFF at all regardless of the value of σ^2_β as can be seen in Figure 4.6.

4.4.2 Simulations for high dimensional data

In this subsection, we are investigating NFF using the general model for the simulation as in Equation (4.5). The generation for the X, β, y and the error term ϵ_y is also the same as in Section 4.4.1 with changes in the dimension of X where n = 20 and p = 50. The X matrix is also decomposed here using SVD and the singular values were changed to the square roots of new different settings of d^2_i. They were chosen so that the sum of each of them is equal to the sum of the eigenvalues of the (NIR data) just to have similar structure of the eigenvalues of the (NIR data) where NFF occurs in some components using the (NIR data). Since $p > n$, the rank of X is $n-1$ which is equal to the number of those proposed eigenvalues. These different settings of d^2 were divided into five sets where the first four settings are the same except for the first two largest eigenvalues are different in order to follow Section 4.4.1. The first setting of $d^2 = (27648.03, 27647.03, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.09, 0.08, 0.07, 0.06, 0.05, 0.04, 0.03, 0.02)$, the second setting of $d^2 = (28648.03, 26647.03, \dots)$, the third setting of $d^2 = (47648.03, 7647.03, \dots)$, the fourth setting of $d^2(55294.06, 1, \dots)$, and the fifth setting of $d^2_i = (3810.6, 3710.6, 3610.6, 3510.6, 3410.6, 3310.5, 3210.5,$ $3110.5, 3010.5, 2910.5, 2810.5, 2710.5, 2610.5, 2510.5, 2410.5, 2310.5, 2210.5, 2110.5,$

2010.5).

Figure 4.9 shows the five different settings of d_i^2, where $i = 1, \ldots, 7$, the rest were omitted because they are the same for the first four settings. From the correlation point of view using these different settings, the average of the values that are below the diagonal of the correlation matrix of X are varied to have high around 0.9, medium around 0.6, low around 0.3 and independent. They are varied from moderate to high respectively for the first four settings, and the fifth setting of d^2 has a very small average correlation.

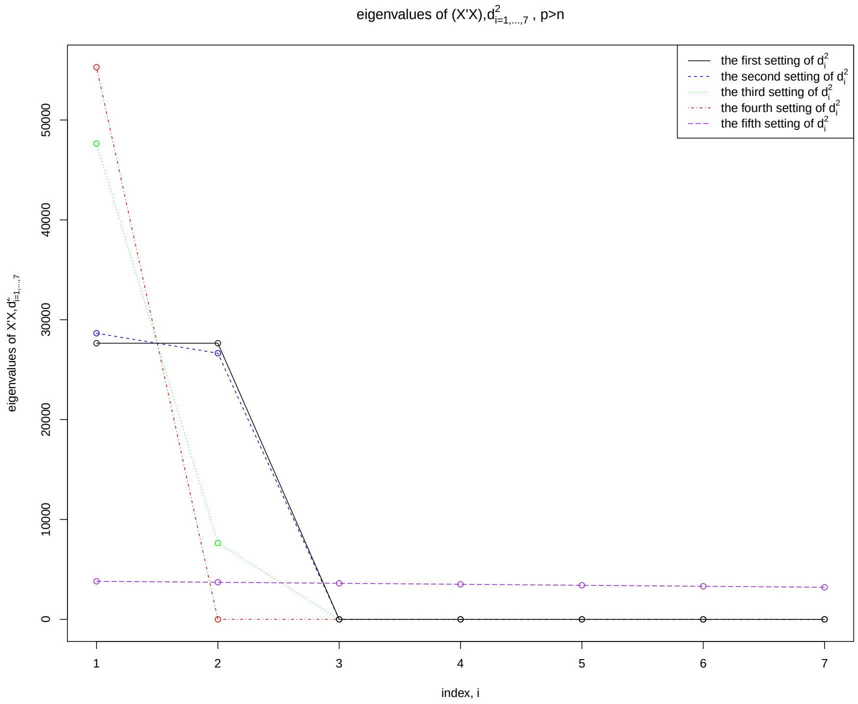


Figure 4.9: The first seven eigenvalues of the replaced nonzero eigenvalues of $X^T X$ for the different settings.

Figure 4.10 illustrates the proportion of datasets which have at least one NFF over all components for each dataset after replacing their singular values by the new setting of d^2 using the simulated data for high dimensional case.

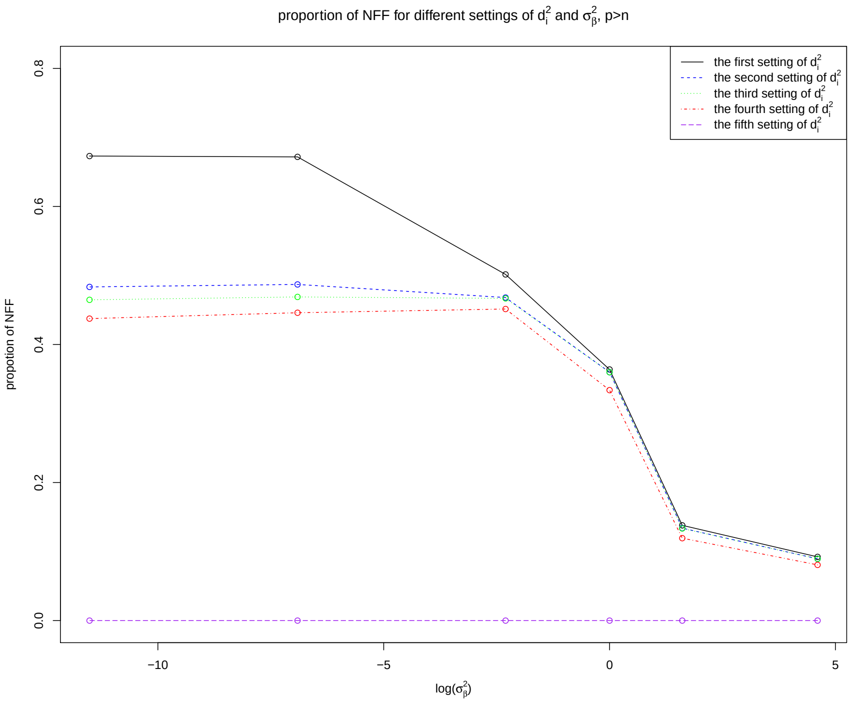


Figure 4.10: The proportion of datasets which have at least one NFF over all components for different σ_β^2 for different datasets.

The proportion of NFF occurrence for all different settings of d^2 is decreased as the log of σ_β^2 is increased as seen in Figure 4.10. The first setting of d^2 has the largest proportion of NFF as the σ_β^2 gets smaller comparing to the other settings and that confirms the same finding in the low-dimensional case. The second, third and fourth settings of d^2 have similar proportion of NFF as σ_β^2 gets smaller. All settings of d^2 have quite similar of the proportion of NFF for large values of σ_β^2. In terms of whether the data matrix X is correlated or independent as in the fifth setting of the d^2, it is more likely to have NFF in the correlated data set but not for the independent one.

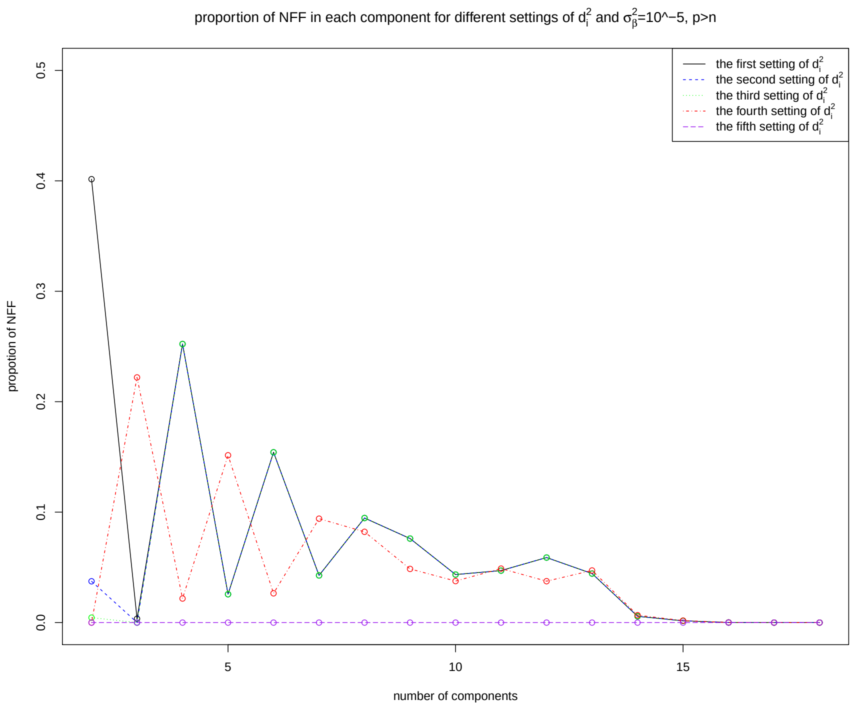


Figure 4.11: The proportion of datasets which have at least one NFF in every component for different datasets using a small value of $\sigma_\beta^2 = 10^{-5}$.

Figure 4.11 shows the proportion of datasets which have at least one NFF in every possible component for small $\sigma_\beta^2 = 10^{-5}$. The proportion of the NFF in the first, second and third settings of d^2 is high in the even components and low in the odd components and that is because the difference between the first two largest eigenvalues is not very huge, but it is very large between the second and the third eigenvalues. However, in the fourth setting, the difference between the first two eigenvalues is large but it is not large between the second and the third eigenvalues. The proportion of NFF is the same in each component for the data using the first, second and third settings of d^2. The proportion of NFF becomes more similar for the data using all settings of d^2 at the latest components. Once the rank of X is reached, there are no NFF for any setting of d^2 even for small $\sigma_\beta^2 = 10^{-5}$ because the eigenvalues of X^TX and $W_m^TX^TXW_m$ are the same, as seen in Section 4.2.

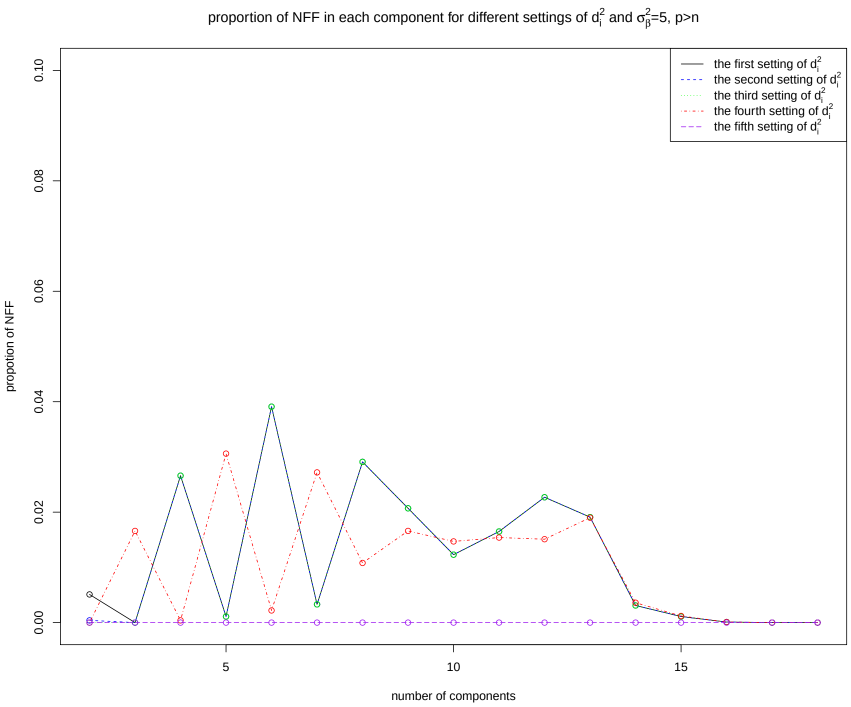


Figure 4.12: The proportion of datasets which have at least one NFF in every component for different datasets using a large value of $\sigma_\beta^2 = 5$.

On the other hand, Figure 4.12 shows that the proportion of datasets which have at least one NFF in every possible component for large $\sigma_\beta^2 = 5$. The proportion of NFF in the first, second and third settings of d^2 is very low in all components though it is small in the even components, and in the odd components is very low in the first seven components. However, the proportion of NFF in the fourth setting is low in the odd components and it is very low in the even components in the first seven components. In the last components where $m = n - 1$, the proportion of NFF is exactly zero.

Based on the simulation results above. It was seen that the occurrence of NFF depends on the relation between d_i^2 and $\mu_j^{2(m)}$ for m components. We have been trying to have a control and deal with $\mu_j^{2(m)}$ and get some insights by doing simulations. It is also important to look at the structure of the X^TX matrix since d_i^2 are the eigenvalues of this matrix. In the following Section 4.5 we have done more through investigation when the NFF may happened based on some examples to show and get some more insights of the causes of the NFF using a small matrix of X and only 2 components

($m = 2$). In these sections, we have been trying to find any link between the initial suspected reasons for causing the NFF which simply are the covariance between X and y and the variance-covariance matrix of X.

To summarise, the structure of the data plays a role to have NFF which can be seen when comparing the proportion of the datasets that have at least one NFF over all components using the fifth setting to other settings of d^2. It is more likely to have NFF if the data is correlated which can be seen in Figure 4.6. By contrast, it is almost impossible to have NFF with data that has structure as in the fifth setting where the variables are independent, regardless of the value of σ_β^2.

4.5 Investigating NFF based on different structures of variance-covariance matrix of X

In the previous sections we were investigating NFF based on the relationship between the eigenvalues of X^TX and $W_m^TX^TXW_m$ matrices. In this section we are going to deal with the elements of those matrices by combining between the condition of NFF and (the eigenvalues d_i^2 and $\mu_j^{2(m)}$) for two components ($m = 2$). The simulation setting for all subsections below are the same except for the structure of the variance-covariance matrix of X. Using the general regression model for the simulation as in Equation (4.5).

X is a matrix of size ($n \times p$), with $n = 100$ and $p = 3$. The response variable (y) is a univariate vector of length $n = 100$. Setting $\beta = (\beta_1, \beta_2, 0)$ where the values of β_1 and β_2 are being chosen from a sequence ranges from -1000 to 1000 for 100 values of each of β_1 and β_2, but β_3 is set to be equal to zero for simplicity and to concentrate on the first two values of the vector β. The error term is a vector of length n generated from a normal distribution with mean zero and variance equals to one. Generating X from a multivariate normal distribution with mean zero, and a variance-covariance matrix to be different in each structure of X^TX. Given only 2 components, $m = 2$, denote

$$W_2 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix}.$$

4.5.1 Two parameter model of X^TX

In this structure of X^TX, the goal is to understand when NFF occurs by investigating a direct relation between the eigenvalues of X^TX and the eigenvalues of $W_2^TX^TXW_2$ through a simulation. Having the structure of the variance-covariance matrix of X as

$$X^TX = \begin{bmatrix} a & b & b \\ b & a & b \\ b & b & a \end{bmatrix},$$

where the values on the diagonal are equal and the off-diagonal values are all the same. Because W matrix is the covariance vector of X and y, and from the linear regression model, β vector can be considered as a cause of having NFF. This might give an intuitive reason for having NFF for some values of β. Considering two components and recall that NFF occurs if and only if $d_i^2 > \sum\limits_{j=1}^{2} \mu_j^2 = \mu_1^2 + \mu_2^2$ for any $i = 1, 2, \ldots, r$. The sum of μ_j^2 is equal to the sum of the diagonal of $W_2^TX^TXW_2$. Substituting the condition for having NFF using two components by $\sum\limits_{j=1}^{2} \mu_j^2$, and the largest eigenvalue of X^TX using this structure. Mostly for two components NFF occurs at $\omega_1^{(2)}$ which corresponds to the largest eigenvalue of X^TX, d_1^2. Therefore, the largest eigenvalue of X^TX is $a + 2b$ if $a > b > 0$. Thus,

$$\omega_i < 0 \;\Leftrightarrow\; a + 2b > 2a + 2b((w_{11}w_{21}) + (w_{11}w_{31}) + (w_{21}w_{31}) + (w_{12}w_{22}) + (w_{12}w_{32}) \\ +(w_{22}w_{32})).$$

Since the columns in W_2 are orthonormal, the right hand side of the inequality can be written as

$$((w_{11}w_{21}) + (w_{11}w_{31}) + (w_{21}w_{31}) + (w_{12}w_{22}) + (w_{12}w_{32}) + (w_{22}w_{32})) \\ = \frac{((w_{11} + w_{21} + w_{31})^2 + (w_{12} + w_{22} + w_{32})^2 - 2)}{2}.$$

Hence, the condition can be rewritten as

$$\omega_i < 0 \;\Leftrightarrow\; a + 2b > 2a + b((w_{11} + w_{21} + w_{31})^2 + (w_{12} + w_{22} + w_{32})^2 - 2).$$

Therefore,

$$\omega_i < 0 \;\Leftrightarrow\; ((w_{11} + w_{21} + w_{31})^2 + (w_{12} + w_{22} + w_{32})^2 - 2) < 2 - \frac{a}{b}. \tag{4.6}$$

As an example of this two parameter model structure of X^TX, it is created where the diagonal values are the same. So, $a = 10$ and $b = 9$. Now, β_1 and β_2 are in a grid search between -1000 and 1000 such that $((w_{11}w_{21}) + (w_{11}w_{31}) + (w_{21}w_{31}) + (w_{12}w_{22}) + (w_{12}w_{32}) + (w_{22}w_{32})) < 2 - \frac{a}{b}$. However, the condition is not satisfied in this structure of X^TX due to the left hand side of the inequality of Equation(4.6) is equal to 1. But, the maximum value that $2 - \frac{a}{b}$ can have is 0.88, for $a > b > 0$ where a, b are positive integers. That means, it is impossible to have NFF in this two parameter model structure of X^TX with the constraints on a, b.

4.5.2 Three parameter diagonal structure of X^TX

In this section, the structure of X^TX as follow

$$X^TX = \begin{bmatrix} a & b & c \\ b & a & b \\ c & b & a \end{bmatrix}.$$

The same approach is going to be used here in order to link the largest eigenvalue of X^TX to the sum of $\mu^2_{j_{(2)}}$. If $\min(b, c) > 0$, the largest eigenvalue of X^TX is $\frac{1}{2}(2a + \sqrt{8b^2 + c^2} + c)$. The trace of $W_2^T X^T X W_2$ is equivalent to the sum of μ_j^2 which are the eigenvalues of $W_2^T X^T X W_2$. Recall, the condition for two components to have NFF is $d_i^2 > \sum\limits_{j=1}^{a} \mu_j^2$, for any $i = 1, 2, \ldots, r$. This condition can be rewritten in terms of the eigenvalues of X^TX and the trace of $W_2^T X^T X W_2$ as

$$\begin{aligned}\omega_i < 0 \;\Leftrightarrow\; \frac{1}{2}(2a + \sqrt{8b^2 + c^2} + c) > 2a + 2b((w_{11}w_{21}) + (w_{21}w_{31}) + (w_{12}w_{22}) \\ +(w_{22}w_{32})) + 2c((w_{11}w_{31}) + (w_{12}w_{32})).\end{aligned}$$

which can be written as

$$\begin{aligned}\omega_i < 0 \;\Leftrightarrow\; ((w_{11}w_{21}) + (w_{21}w_{31}) + (w_{12}w_{22}) + (w_{22}w_{32})) + \frac{c}{b}((w_{11}w_{31}) \\ +(w_{12}w_{32})) < \frac{-a}{2b} + \frac{\sqrt{8b^2 + c^2}}{4b} + \frac{c}{4b}. \qquad (4.7)\end{aligned}$$

We investigate three examples with different values for the variance-covariance matrix of X.

First example

Recall the simulation in Section 4.5 for ϵ_y, β, and the variance-covariance matrix of X is structured as in Section 4.5.2 where $a = 10$, $b = 1$, $c = 9$. β_1 and β_2 are chosen from a sequence as in Section 4.5. Optimising β_1 and β_2 such that the left hand side of the condition in Equation (4.7) is satisfied. All β values including those satisfied the condition in Equation (4.7) for $m = 2$ are shown in Figure 4.13.

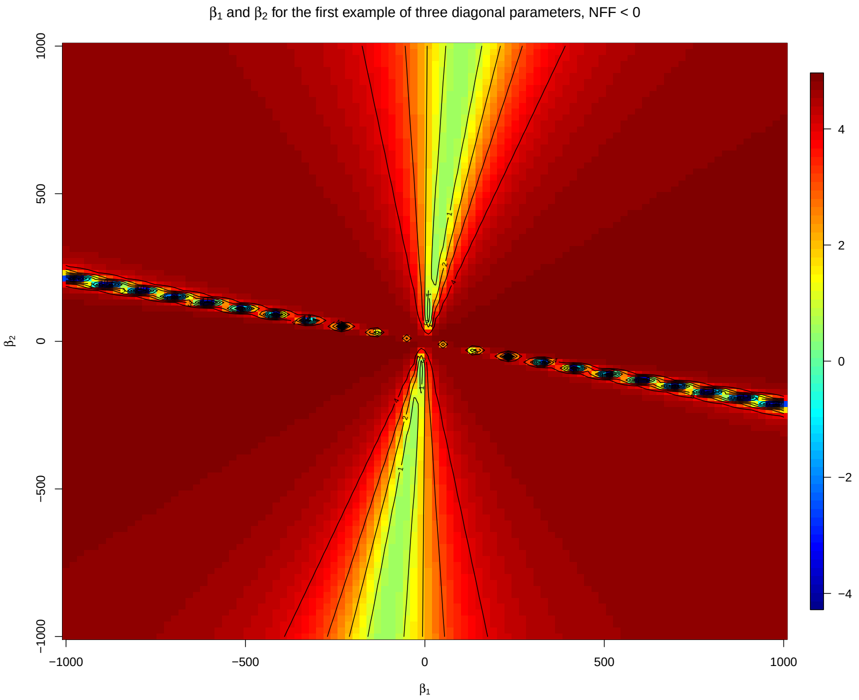


Figure 4.13: The values of the left hand side of the inequality in Equation (4.7). By subtracting the right hand side in Equation (4.7) from both sides, so the threshold here is zero. If those values are less than zero, the condition is satisfied, and we consider the combinations between (β_1 and β_2) to contribute to result in NFF. The colours show that when the colour gets red, the value of (β_1 and β_2) are away from contributing to the occurrence of NFF whereas as when the colour gets blue, those values of (β_1 and β_2) contribute more for NFF.

Figure 4.13 shows a large range of both β_1 and β_2. The optimal values of β_1 and β_2 that are potentially to result in NFF are those which their corresponding value of the left hand side subtracted from the right hand side of the inequality in Equation (4.7) is below zero as it is being the threshold.

It can be seen from Figure 4.13 that the NFF occurs for a specific combination of β_1 and β_2. For a combination of β_1 and β_2, β_1 takes some positive values while β_2 take negative values that are less than the values of β_1. Also, for the other direction of the sign between β_1 and β_2. As both β_1 and β_2 go to zero, NFF does not occur. Moreover, the NFF lay on a straight line all the way from top left to the end right as a diagonal, and there are some others that lay on another line coming from the other side, top middle, going down to the middle where both values for β_1 and β_2 are zeroes, but not crossing the first line. The occurrence of NFF is stopped before the two regression lines (between β_1 and β_2) reach zero. To illustrate, it could be imagined that those two lines are like a valley where there are some humps and areas that NFF appear while in some other areas do not. The NFF are roughly lay within that valley passing through the zero.

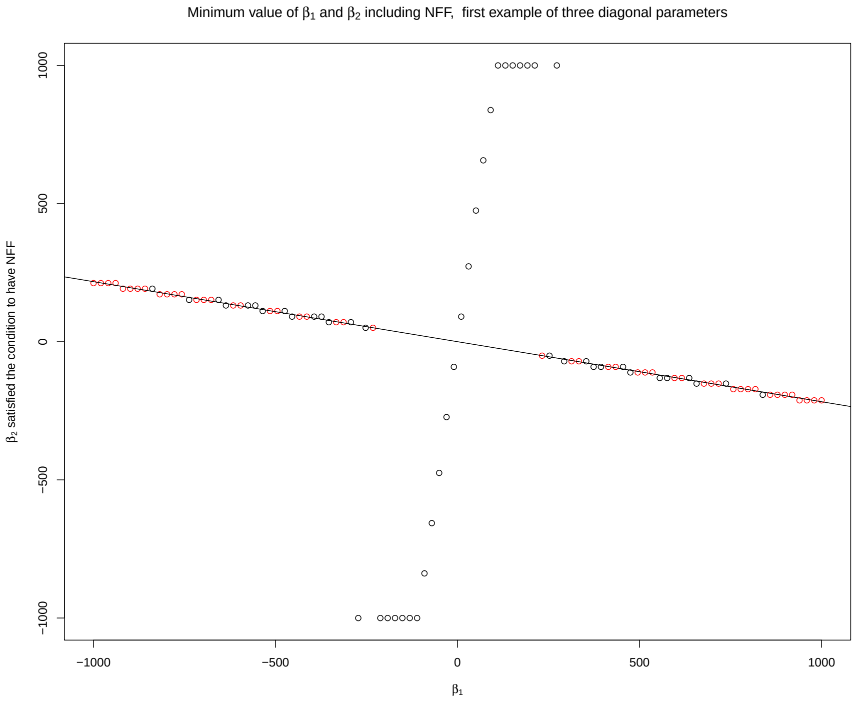


Figure 4.14: The slope between β_1 and β_2, and the minimum values of the left hand side of Equation (4.7) that correspond to β_2, where the one satisfied the condition is coloured by red for any combination of β_1 and β_2.

Figure 4.14 shows the equation line between (β_1 and β_2) or the slope of all the minimum values of the left hand side of Equation (4.7) including the optimal β values that satisfied the condition using the first example of the three parameter diagonal structure of X^TX. Looking at Figure 4.14 where the slope between β_1 and β_2 where it is discontinuous along that line, and there is a specific combination of β_1 and β_2 are more likely to results in NFF which are coloured by red. It is clear that there are some areas do not have NFF though the areas next to them have NFF, and that is because when making the sequence for β_1 is made to be like a grid and that specific value of β_1 may not locate on the grid that produces NFF. Because of time limitation , we consider only 100 values of the sequence of β_1 and β_2.

Second example

In this example, the variance-covariance matrix of X is changed to have a strong two blocks diagonal matrices in one matrix to see how that correlation might affect on having NFF as in Section 4.5.2 where $a = 10$, $b = 7$, $c = 1$.
In this example, we have done the same simulation settings as in Section 4.5.2 for ϵ_y, β, and X with another variance-covariance matrix of X as above. In order to find at what values of β the NFF is more promised to happen, β_1 and β_2 are being optimised and the condition in Equation (4.7) has been checked. All of the candidate values for β_1 and β_2 including the optimal combinations of them are shown in Figure 4.15

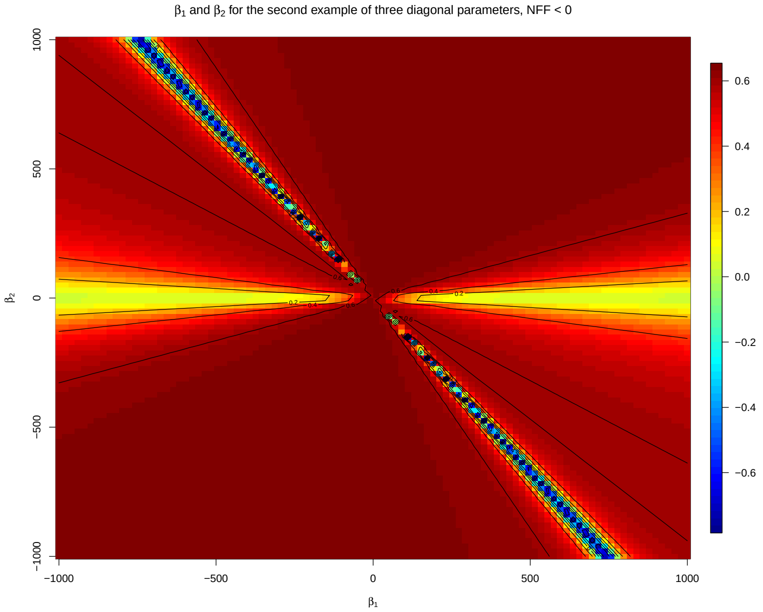


Figure 4.15: The values of the left hand side of the inequality in Equation (4.7). By subtracting the right hand side in Equation (4.7) from both sides, so the threshold here is zero. If those values are less than zero, the condition is satisfied, and we consider the combinations between (β_1 and β_2) to contribute to result in NFF. The colours show that when the colour gets red, the value of (β_1 and β_2) are away from contributing to the occurrence of NFF whereas as when the colour gets blue, those values of (β_1 and β_2) contribute more for NFF.

It can be seen that on the along the line from the top left going to the origin of β_1

and β_2 where the line is discontinuous. The values of β_1 are all negative and β_2 are positive in order to have NFF where the colour is blue and light blue. NFF is more likely to occur if β_1 and β_2 are alternating the sign between them negative and positive for different values. As β_1 and β_2 have very small values, NFF does not occur and that confirms the findings in Section 4.4 where if σ^2_β is small, NFF is more likely to happen.

Figure 4.16 illustrates the slope between β_1 and β_2 using the second example of three parameter diagonal structure of X^TX including the one satisfied the condition.

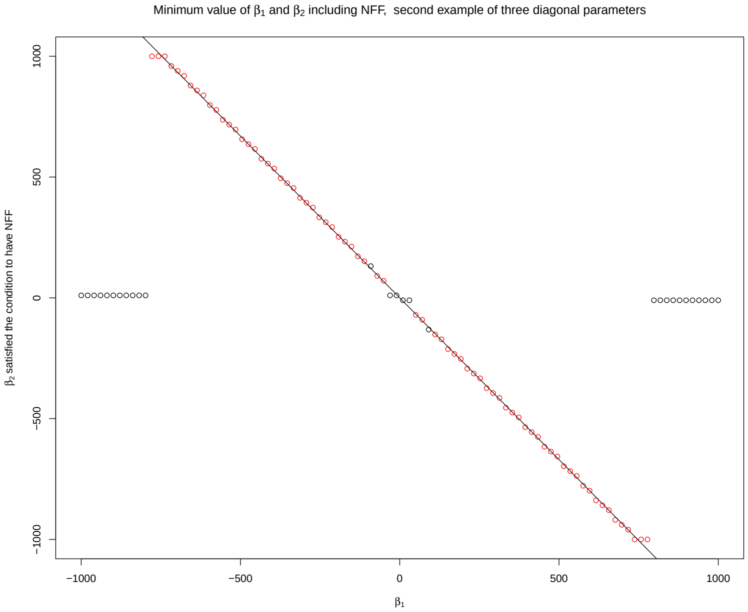


Figure 4.16: The slope between β_1 and β_2, and the minimum values of the left hand side of Equation (4.7) that correspond to β_2, where the one satisfied the condition is coloured by red for any combination of β_1 and β_2.

Looking at Figure 4.16 where the slope between β_1 and β_2 is plotted along all of the candidate values for both β_1 and β_1. The minimum values of the left hand side of Equation (4.7) corresponding to β_2 is being plotted. The red circles are those satisfied the condition out of the minimum values. The number of NFF cases in this example is more and the reason probably is the structure of X^TX where there are two strong block diagonal matrices in X^TX matrix.

Third example

In this example, the variance-covariance matrix of X is structured as in Section 4.5.2 where $a = 10$, $b = 9$, $c = 8$.
The simulation settings in this section have followed those in Section 4.5.2 for ϵ_y and β, but the variance-covariance matrix of X is different where the covariance between all variables is high. Figure 4.17 shows 100 values of the left hand side of Equation (4.7) based on 100 values of β_1 and β_2 including the optimal values of β_1 and β_2 those satisfied the condition in Equation (4.7) for $m = 2$.

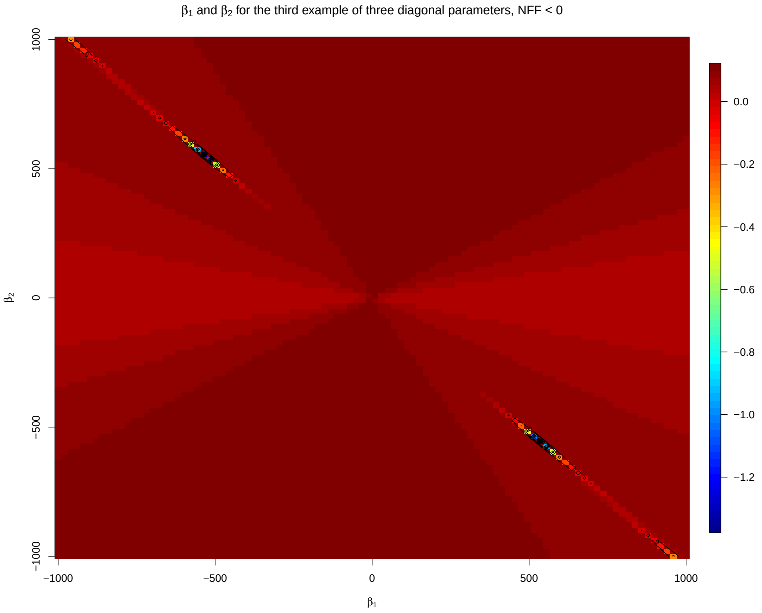


Figure 4.17: The values of the left hand side of the inequality in Equation (4.7). By subtracting the right hand side in Equation (4.7) from both sides, so the threshold here is zero. If those values are less than zero, the condition is satisfied, and we consider the combinations between (β_1 and β_2) to contribute to result in NFF. The colours show that when the colour gets red, the value of (β_1 and β_2) are away from contributing to the occurrence of NFF whereas as when the colour gets blue, those values of (β_1 and β_2) contribute more for NFF.

As can be seen from Figure 4.17, the cases of NFF are less where the colours starting from the reddish is considered as NFF. The combination of β_1 and β_2 as seen from the first and second example the sign of β_1 is the opposite sign of β_2 regardless to the value.

Figure 4.18 shows the slope between β_1 and β_2 for those minimum values of the left hand side of the inequality in Equation (4.7) using the third example of three diagonal parameter structure of X^TX. The minimum values of the amount of the left hand side in Equation (4.7) including the optimal β values that satisfied the condition in order to have NFF.

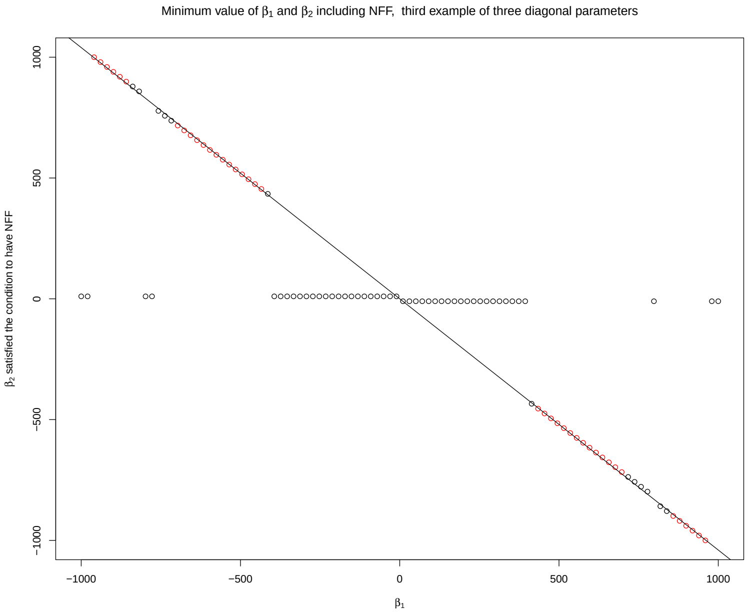


Figure 4.18: The slope between β_1 and β_2, and the minimum values of the left hand side of Equation (4.7) that correspond to β_2, where the one satisfied the condition is coloured by red for any combination of β_1 and β_2.

The optimal values of β, which their combination produces NFF are those where their corresponding amount of the left hand side of Equation (4.7) is below zero in Figure 4.18. To illustrate, both sides of the inequality in Equation (4.7) are subtracted by the right hand side amount in order to make the threshold equals to zero instead of the amount of the right hand side of the inequality. The amount of the left hand side of the inequality after subtracting the amount of the right hand side is coloured by red where those have met the condition in Equation (4.7), which means NFF is appeared. Those corresponding β_1 and β_2 are being used to calculate the amount of the left hand side since it depends on a complicated mathematical operations of multiplication and additional between all elements in the w_2 matrix for $m = 2$.

In summary, from Section 4.4, we found that there is a connection between NFF and the σ^2_β. This connection can be also seen here if we consider that as the radius squared is increased as the σ^2_β increased. This is why we can still have NFF at large values of β.

4.5.3 Finding the connection between the correlation of data matrices

Fitting the slope between (β_1 and β_2) over (b and c)

Having investigated the connection between the eigenvalues of X^TX and $W_2^TX^TXW_2$ matrices in order to find the causes of NFF, it has been found that there is a complicated relationship between those eigenvalues. It is very difficult to get the insights of NFF and how (the covariance between X and y and the variance-covariance matrix of X) contribute to likely result in NFF. Thus, the direct connection between the variance covariance of X matrix and the covariance between X and y vector should be investigated. One way is to see how the relation between the slope between (β_1 and β_2, γ) and (the values of b and c) in the variance-covariance matrix of X. This can be done using the same simulation in Section 4.5.1 for β, ϵ_y, and X where for the variance-covariance matrix of X, we let the elements, b, c in the variance-covariance matrix vary and take different values such that the variance-covariance matrix is semi-positive definite. Letting both b and c vary, this might give more insights for the relation between this covariance matrix of X and γ.
The slope between β_1 and β_2 represented by γ which is the minimum values of the left hand side of Equation (4.7) including those satisfied the condition for the inequality in Equation (4.7).

Figure 4.19 shows the relation between γ over possible different values of b for each value of c of the variance-covariance matrix of X.

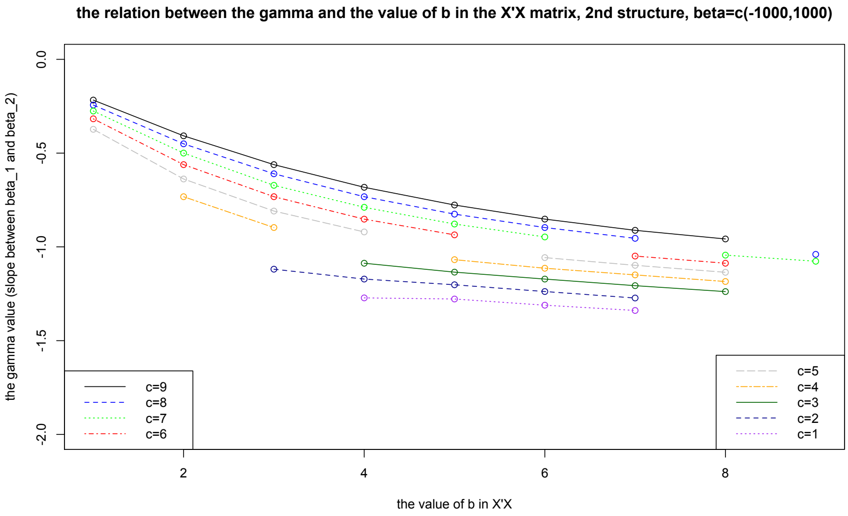


Figure 4.19: The slope between β_1 and β_2, (γ), and (b and c) of the variance-covariance matrix of X.

It can be seen that from Figure 4.19 as the $cov(x_1, x_3)$, c, gets larger, γ gets larger in magnitude. Also, as the $cov(x_1, x_2)$ and $cov(x_2, x_3)$, b, gets larger as γ gets smaller in magnitude for a given fixed value of the $cov(x_1, x_3)$, c.

Looking at Figure 4.19, it can be seen that the relation between γ and (b, c) is not linear. Thus, we use the transformation by squaring the γ in order to get better model because it does not look linear. However, the other transformed models did not improve the fitted values, and the more simpler model, the better is, so we will use the linear model.

Assuming the linear model for the relation between γ and (b, c) as follows:

$$\gamma = \kappa_0 + \kappa_1 b + \kappa_2 c + \epsilon_\gamma, \tag{4.8}$$

where $b = cov(x_1, x_2) = cov(x_2, x_3)$ and $c = cov(x_1, x_3)$, and ϵ_γ is the error term of γ. The fitted values are $\kappa_0 = -0.838$, $\kappa_1 = -0.091$, $\kappa_2 = 0.068$. For some diagnostics on the fitted model, the residuals plot for the chosen fitted model along with QQ normal plot are shown in Figure 4.20.

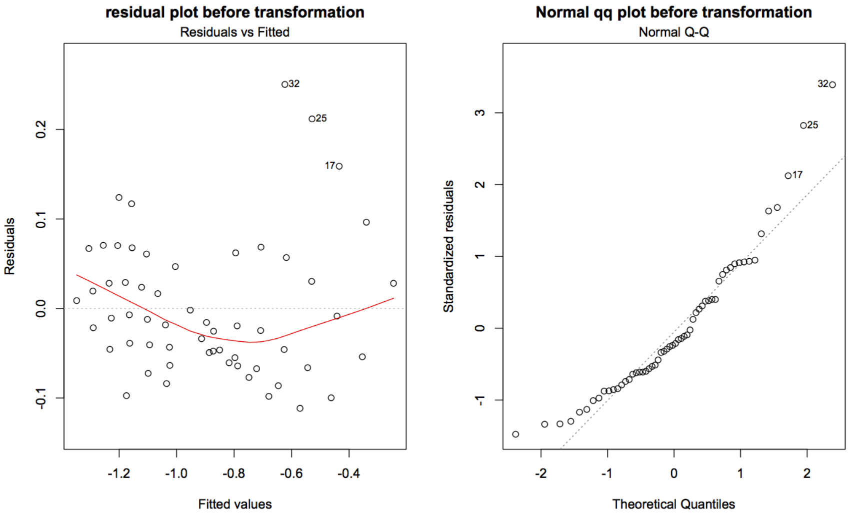


Figure 4.20: Residuals plot and QQ plot, without transformation.

Finding relationship between (the correlation matrix of X) and (the correlation between X and y)

Instead of using γ, the correlation vector between X and y is used to give a clearer picture and direct connection between the correlation of X and y and the covariance between X variables in order to seek for the reasons of having NFF from the variance-covariance of X matrix and the correlation vector between X and y. Here, we are going to find any pattern that gives any connection between the correlation of X matrix and the values of γ. It has been observed from the simulation that the values of γ does not change across every single simulation and that is because of β_1 and β_2 are chosen from a wide sequence ranges from -1000 to 1000, and for small σ_e^2 equals to one.

In order to find a theoretical relationship for the correlation of X and y, we assume that X and y variables are centred. We will find the connection for each x_j variable separately. We have that for a linear model using three variables

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon,$$

where y is the response vector, β_1, β_2, and β_3 are the coefficients, x_1, x_2, and x_3 are the predictors, ϵ is the residual term. We also fix $\beta_2 = \gamma\beta_1$ for some γ. Therefore,

$$y = \beta_1 x_1 + \gamma\beta_1 x_2 + \beta_3 x_3 + \epsilon.$$

The correlation between X_1 and y can be calculated as in Equation (4.9)

$$cor(x_1, y) = \frac{cov(x_1, y)}{\sqrt{var(x_1)}\sqrt{var(y)}}. \tag{4.9}$$

Since X and y are centred, E(x_1) = E(x_2) = E(x_3) = E(y) = 0. Thus, $cov(x_1, y)$ = E($x_1 y$) Calculating the E($x_1 y$), we have

$$E(x_1 y) = E(x_1(\beta_1 x_1 + \gamma\beta_1 x_2 + \beta_3 x_3 + \epsilon)).$$

Then,

$$E(x_1 y) = \beta_1 E(x_1^2) + \gamma\beta_1 E(x_1 x_2) + \beta_3 E(x_1 x_3). \tag{4.10}$$

Looking at the second structure matrix, we can find the expectation for each term in Equation (4.10). Thus,

$$cov(x_1, y) = E(x_1 y) = \beta_1[a + \gamma b] + \beta_3 c.$$

For the variance of x_1 and y, we can compute that using the property of centring for both. Therefore,

$$var(y) = E(y^2) = E((\beta_1 x_1 + \gamma\beta_1 x_2 + \beta_3 x_3 + \epsilon)^2),$$
$$var(y) = E(y^2) = \beta_1^2[a + \gamma^2 a + 2\gamma b] + \beta_3^2 a + 2\beta_1\beta_3[c + \gamma b],$$

and

$$var(x_1) = E(x_1^2) = a.$$

Going back to the formula for the correlation between x_1 and y, we have the correlation between x_1 and y in Equation (4.11).

$$cor(x_1, y) = \frac{[a + \gamma b] + \beta_3 c}{\sqrt{a}\sqrt{[a + \gamma^2 a + 2\gamma b] + \beta_3^2 a + 2\beta_1\beta_3[c + \gamma b]}}. \tag{4.11}$$

Similarly finding the cor(x_2, y) and cor(x_3, y) as in Equations (4.12) and (4.13), respectively.

$$cor(x_2, y) = \frac{[b + \gamma a] + \beta_3 b}{\sqrt{a}\sqrt{[a + \gamma^2 a + 2\gamma b] + \beta_3^2 a + 2\beta_1\beta_3[c + \gamma b]}}. \tag{4.12}$$

For the correlation between x_3 and y, we have that in Equation (4.9)

$$cor(x_3, y) = \frac{[c + \gamma b] + \beta_3 a}{\sqrt{a}\sqrt{[a + \gamma^2 a + 2\gamma b] + \beta_3^2 a + 2\beta_1\beta_3[c + \gamma b]}}. \tag{4.13}$$

Using the above theoretical calculations for the correlations in Equations (4.11), (4.12) and (4.13) between X variables and y as a function of a, b, c, and γ to see if there is a relationship between the correlation of X variables and y and the γ values. We set $\beta_3 = 0$, $a = 10$, b and $c \in \{1, 2, \ldots, 9\}$ where $(b, c < a)$.

Figure 4.21 shows the relation between $cor(x_1, y)$ and different values of b for each value of c.

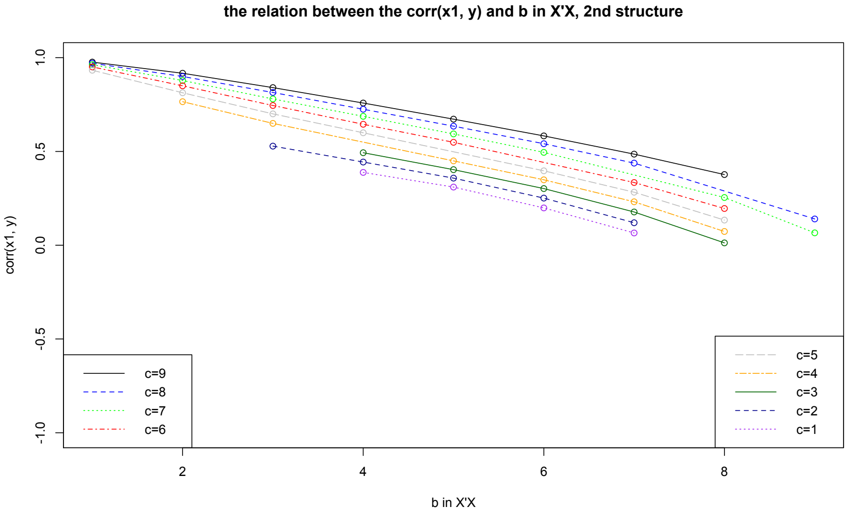


Figure 4.21: $cor(x_1, y)$, b and c (the off-diagonal values of three parameter model structure of $X^T X$).

It can be seen that in this figure as $cor(x_1, x_3)$, c, gets larger, $cor(x_1, y)$ gets larger in magnitude. As $cor(x_1, x_2)$ and $cor(x_2, x_3)$, b, gets larger $cor(x_1, y)$ gets smaller in magnitude for a given fixed value of $cor(x_1, x_3)$, c.

Figure 4.22 shows the relation between the $cor(x_2, y)$ over a different values of b and c.

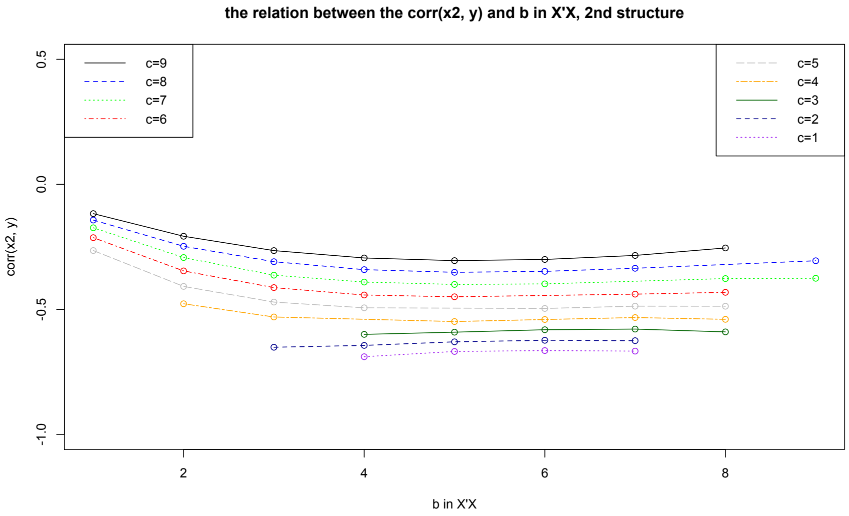


Figure 4.22: $cor(x_2, y)$, b and c (the off-diagonal values of three parameter model structure of $X^T X$).

It can be seen from Figure 4.22 that as the cor(x_1, x_3), c, gets larger, the $cor(x_2, y)$ gets smaller in magnitude. As the $cor(x_1, x_2)$ and $cor(x_2, x_3)$, b, gets larger the $cor(x_2, y)$ gets larger in magnitude up to the half of the value of cor(x_1, x_2) and $cor(x_2, x_3)$, b, then the $cor(x_2, y)$ gets smaller in magnitude for a given fixed value of the $cor(x_1, x_3)$, c. Also, as $cor(x_1, x_2)$ and $cor(x_2, x_3)$, b, gets larger, and if $(b > c)$, then the $cor(x_2, y)$ gets smaller in magnitude.

Figure 4.23 shows the relation between the cor(x_3, y) and sequence values of b for each value of c.

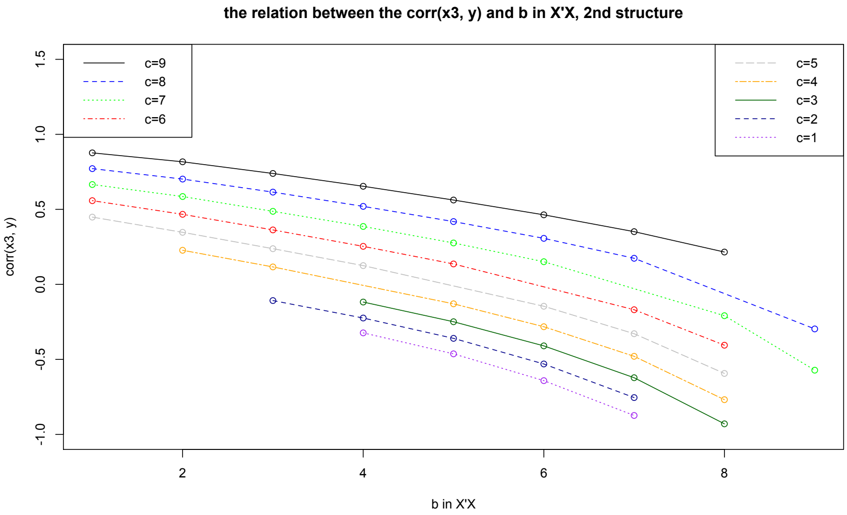


Figure 4.23: $cor(x_3, y)$, b and c (the off-diagonal values of three parameter model structure of $X^T X$).

It can be seen that in this figure as the $cor(x_1, x_3)$, c, gets larger, the $cor(x_3, y)$ gets larger in magnitude. As the $cor(x_1, x_2)$ and $cor(x_2, x_3)$, b, gets larger and if $(b < c)$, then the $cor(x_3, y)$ gets smaller in magnitude until the $(b > c)$, then the $cor(x_3, y)$ gets larger in magnitude for a given fixed value of the $cor(x_1, x_3)$, c.

Using the correlation between x_1 and y by the observed γ from the slope between (β_1 and β_2), and the estimated $\hat{\gamma}$ after using the values of κ_0, κ_1 and κ_2 from the linear model in Equation (4.8). Figure 4.24 shows $cor(x_1, y)$ from using the exact value of γ and the estimated $\hat{\gamma}$.

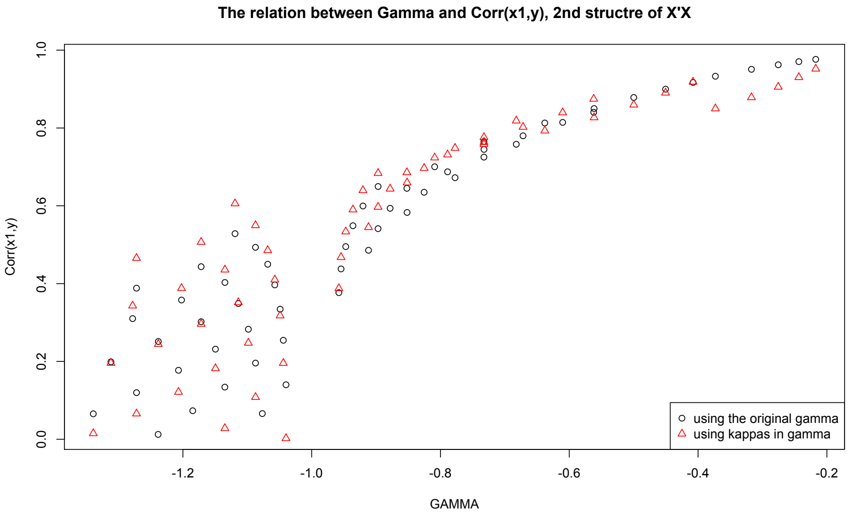


Figure 4.24: $cor(x_1, y)$ using the observed and estimated values of γ, (the observed γ is calculated for different values of b and c in the X^TX matrix and the $\hat{\gamma}$ using the fitted model in Equation (4.8)), to calculate the correlation using Equation (4.11).

Figure 4.25 shows the relation between the $cor(x_2, y)$ and the observed γ.

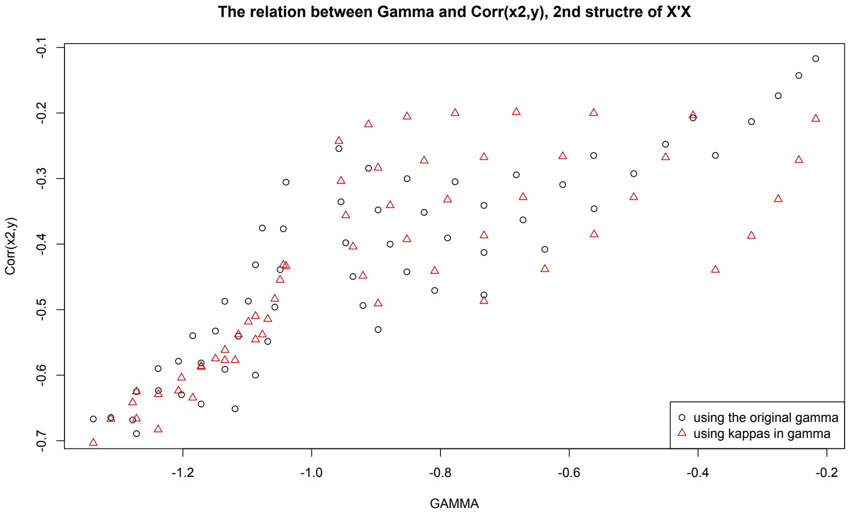


Figure 4.25: $cor(x_2, y)$ using the observed and estimated values of γ, (the observed γ is calculated for different values of b and c in the X^TX matrix and the $\hat{\gamma}$ using the fitted model in Equation (4.8)), to calculate the correlation using Equation (4.12).

Figure 4.26 shows the relation between the $cor(x_3, y)$ and the observed γ.

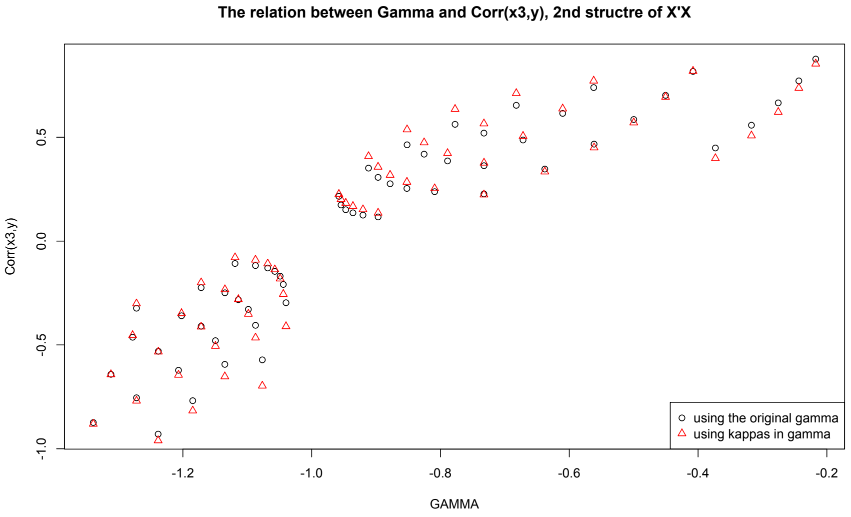


Figure 4.26: $cor(x_3, y)$ using the observed and estimated values of γ (the observed γ is calculated for different values of b and c in the X^TX matrix and the $\hat{\gamma}$ using the fitted model in Equation (4.8)), to calculate the correlation using Equation (4.13).

As it can be seen from Figure 4.24 that the correlation between x_1 and y using the exact γ and the estimated $\hat{\gamma}$ are quite similar. That indicates the linear model that has been chosen to fit the relation between γ and b and c is an appropriate model. It can be seen in Figures 4.24, 4.25 and 4.26 that even the correlation between X and y is large, we still may have NFF. The above results aim to find any connection between the covariance between X and y and the variance-covariance matrix of X.

4.5.4 Four parameter model structure of X^TX

In this section, we consider a four parameter model structure of X^TX. The main gaol of this structure is the same as in the previous structures which is to get some insights of when NFF occurs when the structure of X^TX changes. Simulating β and ϵ_y as in Section 4.5, but the variance-covariance matrix of X between variables are different as

$$X^TX = \begin{bmatrix} a & b & c \\ b & a & d \\ c & d & a \end{bmatrix}.$$

For $m = 2$ components, the sum of the eigenvalues of $W_2^T X^T X W_2$ is $2a + 2b((w_{11}w_{21}) + (w_{12}w_{22})) + 2c((w_{11}w_{31}) + (w_{12}w_{32})) + 2d((w_{21}w_{31}) + (w_{22}w_{32}))$. If $\min(b, c, d) > 0$, the largest eigenvalue for $X^T X$ is represented as d_1^2. Note that d here is an element in $X^T X$ matrix which is not the square root of first eigenvalues of $X^T X$ represented by d_1^2. The sum of the eigenvalues of $W_2^T X^T X W_2$ is equal to trace of $W_2^T X^T X W_2$. Having seen that for $m = 2$, $\omega_i < 0$ if and only if $d_i^2 > \sum_{j=1}^{m} \mu_j^2$, for any $i = 1, 2, \ldots, r$ from Section 4.3.1. Substituting the sum of the eigenvalues by the trace of $W_2^T X^T X W_2$, and the condition where NFF occurs for 2 components can be rewritten as

$$\begin{aligned}\omega_i < 0 \;\Leftrightarrow\; d_1^2 > 2a + 2b((w_{11}w_{21}) + (w_{12}w_{22})) + 2c((w_{11}w_{31}) + (w_{12}w_{32}))\\ +2d((w_{21}w_{31}) + (w_{22}w_{32})).\end{aligned}$$

Letting the covariance between X variables in one side of the inequality as much as it could be leads to

$$\begin{aligned}\omega_i < 0 \;\Leftrightarrow\; b((w_{11}w_{21}) + (w_{12}w_{22})) + c((w_{11}w_{31}) + (w_{12}w_{32})) +\\ d((w_{21}w_{31}) + (w_{22}w_{32})) < \frac{d_1^2}{2} - a. \qquad (4.14)\end{aligned}$$

Example

In this example the variance-covariance matrix of X is structured as

$$X^T X = \begin{bmatrix} 10 & 1 & 2 \\ 1 & 10 & 9 \\ 2 & 9 & 10 \end{bmatrix},$$

100 values of β_1 and β_2 are chosen from the sequence ranges from -1000 to 1000. The left hand side of the inequality in Equation (4.14) is illustrated for all combination of β_1 and β_2 in Figure 4.27.

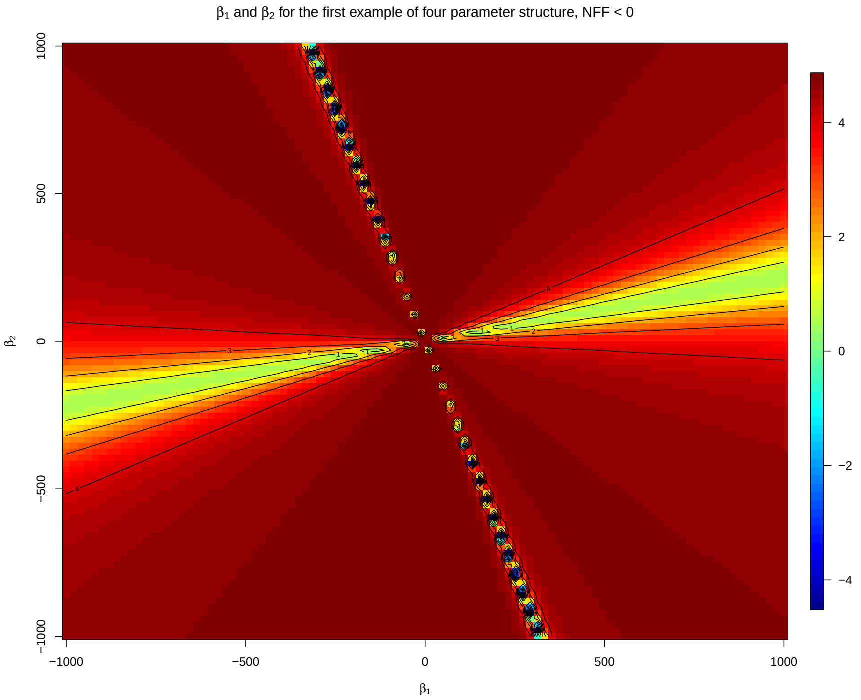


Figure 4.27: The values of the left hand side of the inequality in Equation (4.14). By subtracting the right hand side in Equation (4.7) from both sides, so the threshold here is zero. If those values are less than zero, the condition is satisfied, and we consider the combinations between (β_1 and β_2) to contribute to result in NFF. The colours show that when the colour gets red, the value of (β_1 and β_2) are away from contributing to the occurrence of NFF whereas as when the colour gets blue, those values of (β_1 and β_2) contribute more for NFF.

As it can be noticed from Figure 4.27 that NFF occurs for some combination of small positive values of β_1 and large negative values for β_2 and vice-versa. However, there is no NFF as the value of both β_1 and β_2 go to the origin, and as they both get larger as well.

Figure 4.28 illustrates the equation line which is the slope between β_1 and β_2, γ.

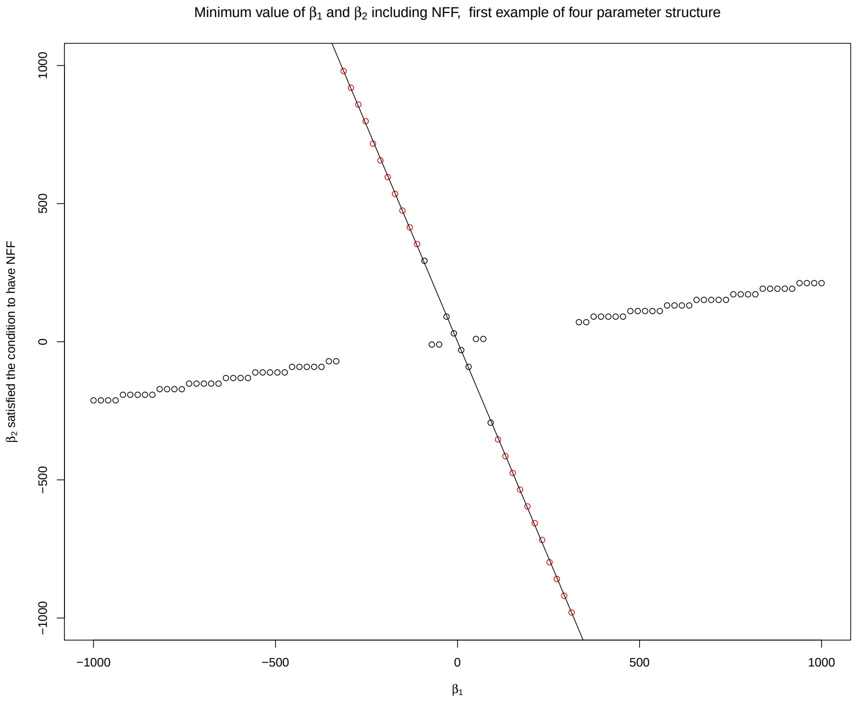


Figure 4.28: The slope between β_1 and β_2, γ.

From Figure 4.28, it can seen that the combination of β_1 and β_2 that gives the minimum value of the left hand side of Equation (4.14) including the optimal β values which satisfied the condition in Equation (4.14). These combinations between β_1 and β_2 can be seen for smaller positive values for β_1 and larger negative values of β_2. Using these values for the variance-covariance matrix of X for that specific example and letting β_1 and β_2 to have some values between -1000 and 1000. Then, the slope between β_1 and β_2, γ is equal to -3.13.

4.5.5 Six parameter model structure of $X^T X$

In this section, the structure of $X^T X$ is very complicated where all the values on the diagonal and off-diagonal are different. This structure will show the condition for 2 components in Section 4.3.1 can be written in terms of those elements of the variance-covariance matrix of X. The variance-covariance matrix is structured as

$$X^TX = \begin{bmatrix} a & b & c \\ b & e & d \\ c & d & f \end{bmatrix}.$$

It is difficult to write the largest eigenvalue of the above matrix explicitly. Thus, the largest eigenvalue of X^XX is represented as d_1^2. Note that d here is an element in X^TX matrix which is not the square root of first eigenvalues of X^TX represented by d_1^2. The sum of the eigenvalues of W^TX^TXW is $a(w_{11}^2+w_{12}^2)+e(w_{21}^2+w_{22}^2)+f(w_{31}^2+w_{32}^2)+2b((w_{11}w_{21})+(w_{12}w_{22}))+2c((w_{11}w_{31})+(w_{12}w_{32}))+2d((w_{21}w_{31})+(w_{22}w_{32}))$. Writing the condition for 2 components in Section 4.3.1 in terms of the elements of X^TX as

$$\omega_i < 0 \;\Leftrightarrow\; a(w_{11}^2+w_{12}^2)+e(w_{21}^2+w_{22}^2)+f(w_{31}^2+w_{32}^2)+2b((w_{11}w_{21})+(w_{12}w_{22})) \\ +2c((w_{11}w_{31})+(w_{12}w_{32}))+2d((w_{21}w_{31})+(w_{22}w_{32})) < d_1^2.$$

Making the covariance between X variables with the largest eigenvalue of X^TX in one side of the inequality leads to

$$\begin{aligned} \omega_i < 0 \;\Leftrightarrow & ((a-f)(w_{11}^2+w_{21}^2)) + ((e-f)(w_{12}^2+w_{22}^2)) \\ & + 2b((w_{11}w_{21})+(w_{12}w_{22})) + 2c((w_{11}w_{31})+(w_{12}w_{32})) \\ & + 2d((w_{21}w_{31})+(w_{22}w_{32})) < d_1^2 - 2f. \end{aligned} \tag{4.15}$$

Example

This example is to identify at which combination of β_1 and β_2 that produces NFF. The variance-covariance matrix of X is given by

$$X^TX = \begin{bmatrix} 10 & 3 & 1.5 \\ 3 & 2 & 1 \\ 1.5 & 1 & 9 \end{bmatrix},$$

where the variance of $x_2 = e = 2$ is smaller than the variance for the other variables x_1 and x_3 are large.

β_1 and β_2 have been optimised over 100 chosen values from a sequence from -1000 to 1000 using the left hand side of the inequality in Equation (4.15). The values of left hand side for all combination of β_1 and β_2 values are plotted as in Figure 4.29.

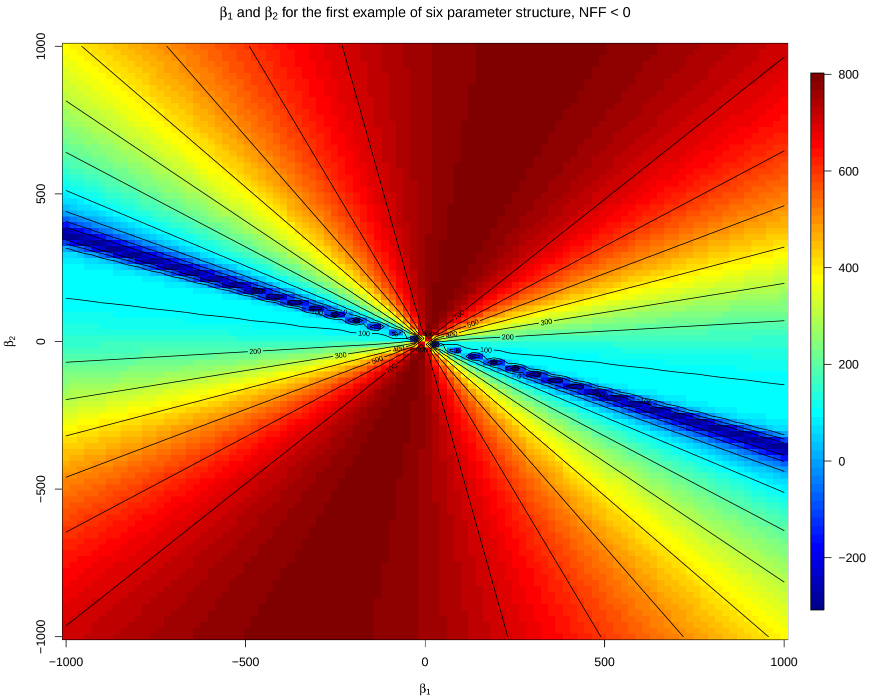


Figure 4.29: The values of the left hand side of the inequality in Equation (4.15). By subtracting the right hand side in Equation (4.7) from both sides, so the threshold here is zero. If those values are less than zero, the condition is satisfied, and we consider the combinations between (β_1 and β_2) to contribute to result in NFF. The colours show that when the colour gets red, the value of (β_1 and β_2) are away from contributing to the occurrence of NFF whereas as when the colour gets blue, those values of (β_1 and β_2) contribute more for NFF.

It can be noticed from Figure 4.29 that there are more possible combinations between β_1 and β_2 that are more likely to result in NFF. For small and large negative values of β_1 and the values of β_2 are positive and less than β_1 are more likely to result in NFF. Also, the vice-versa for the signs for both β_1 and β_2 can be observed from this figure. However, as going closer to the origin of β_1 and β_2, NFF does not exist.

Figure 4.30 shows the slope between β_1 and β_2, γ, using an example of six parameter model structure of X^TX.

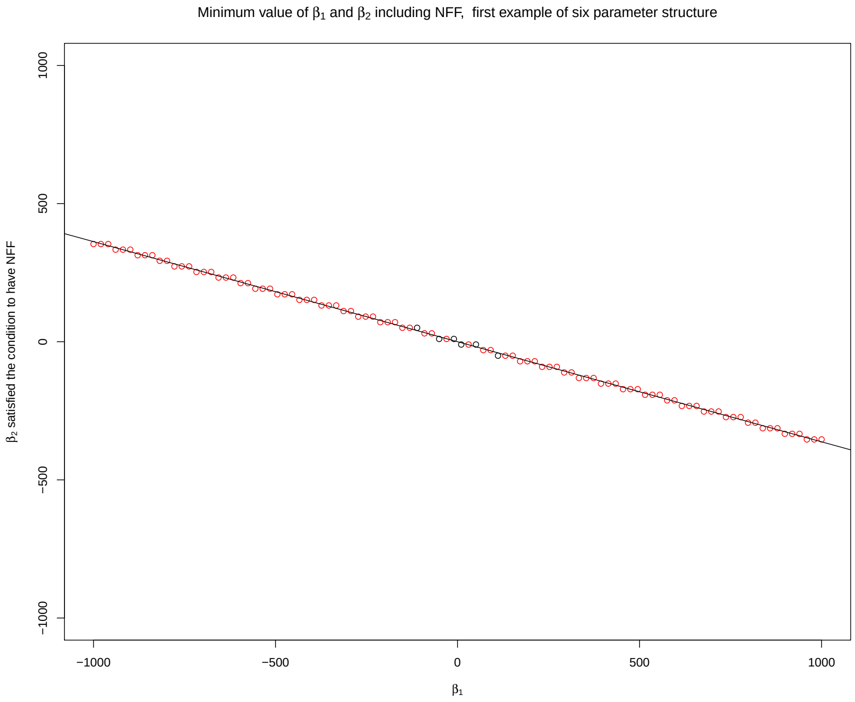


Figure 4.30: The slope between β_1 and β_2, γ.

As it can be seen from Figure 4.30 that along the slope between β_1 and β_2 the red circles represent the values of the left hand side of the inequality in Equation (4.15) that are below the right hand side in the same equation. Consequently, those red circles represent NFF using the corresponding combinations between β_1 and β_2. The slope between β_1 and β_2, γ, for this example of the six parameter model structure of $X^T X$ is equal to -0.3623.

4.6 Discussion

Having introduced the shrinkage of three popular methods in RR, PCR and PLS regression estimators in Chapter 3 for high dimensional data, we have been exploring the challenges of identifying the circumstances in which NFF are more likely (or even guaranteed) to occur in each component for the normal response data sets.

Although the results have not covered particularly the main causes of having NFF in each component and how that is related either the covariance of X and y or the variance-covariance matrix of X, we have proposed very clear conditions in each component based on the relation between the eigenvalues of X^TX, d_i^2, and the eigenvalues of $W_m^TX^TXW_m$, $\mu_j^{2(m)}$. We have also confirmed through some simulations for a normal response that as σ_β^2 gets smaller, the proportion of datasets which have at least one NFF over all components is larger for low and high dimensional cases.

Furthermore, we have showed that NFF is more likely happen using a correlated data matrix, X. However, using an independent matrix does not have NFF whatever the σ_β^2. We have been looking for some intuitive reasons by applying different investigation using small examples with different structures of X^TX. This concludes that the NFF is promised to occur at a specific combination of β_1 and β_2 along with the values of b and c in X^TX.

It should be noted that if there is a combination of β_1 and β_2 that gives NFF, there is a another combination which is very close to it, but does not give NFF. That means, the covariance of X and y only is not the key to result in NFF, but also the structure of X^TX. Since both the covariance of X and y, which is represented by W_m, and the variance-covariance matrix of X, represented by X^TX are used to calculate $\mu_j^{2(m)}$. It is almost impossible to distinguish and find an exact direct condition or connection on either the covariance of X and y or the variance-covariance matrix of X.

Finally, the connection between NFF and the structure of the data is very important. For instance, NFF cannot be seen for data with independent variables. For the connection between the structure of data and (the correlation between X and y), the results from a simulated data with limited values given for the elements in X^TX matrix. They are positive with $a < b < c$ and they ranges between 1 and 10. Although this can be seen as a limitation of our simulation, we still find that the structure of X^TX has a huge impact of NFF in the PLS estimator.

Chapter 5

Sparse Smoothed Partial Least Squares Regression

5.1 Overview

In Chapter 2, we introduced the ordinary PLS for high-dimensional and highly-correlated data sets. The results of using the ordinary PLS on (NIR data) and (CNA data) in Chapter 2, are difficult to interpret in the estimation of $\hat{\beta}$ and the subsequent components of the direction vectors W_m. This is because these types of data sets have two main problems which are: finding a method to deal with the dimension which takes account of the dependencies between covariates; avoiding the impact of irrelevant covariates (genomic regions or wavelengths) which may result in a poor prediction. Thus, a sparse solution may help to interpret the estimates of $\hat{\beta}$ and w_m.

Some sparse methods have been widely studied for the NIR data (e.g. Gusnanto & Pawitan (2015) and Lee *et al.* (2011)). Chun & Keleş (2010) proposed a sparse partial least square (SPLS) method where they impose the sparsity at the dimension reduction step. This leads to dimension reduction and variable selection simultaneously. Lee *et al.* (2011) developed a new formulation of SPLS by imposing the sparsity at the regression step using two versions of NIPALS algorithm for a normal response. Lee *et al.* (2013) proposed L_1 and hierarchical likelihood (HL) penalties on the survival data using SPLS-L1 and SPLS-HL.

However, a sparse solution can be achieved using an L_1 penalty which is known to select only one covariate out of a group of predictors that are highly-correlated. This is a problem that needs a method to tackle the high correlation between covariates. Moreover, Huang *et al.* (2009) proposed a procedure for classification of array CGH data using a smoothed logistic regression model. To our knowledge, previous methods of SPLS in the literature have not considered the dependencies between the covariates.

Since the data sets are highly-correlated, in order to tackle the dependencies between the genomic regions for (CNA data) or wavelengths for (NIR data), smoothness is needed. We extend the SPLS formulation that Lee *et al.* (2011) proposed by considering a different penalty, where we assume the penalty function to be a mixture that controls sparseness and smoothness. We consider the second differences of adjacent values of w to follow a Cauchy distribution to achieve smoothness, and to achieve sparseness we use the Laplace distribution. The model is based on assuming the penalty function to follow a mixture of two distributions which control smoothness and sparseness.

The organisation of this chapter is as follows. Section 5.2 provides the SPLS method. In Section 5.3 we develop the smoothed PLS method with the first NIPALS algorithm. Using the first NIPALS algorithm 1, enables us to develop the sparse-smoothed PLS (SSPLS) solution which is provided in Section 5.4. In Section 5.5 we provide another smoothed PLS based on the second NIPALS algorithm. In Section 5.6 we develop another SSPLS with the second NIPALS algorithm 2. Simulation results are discussed in Section 5.7 for both methods of SSPLS based on the first and second NIPALS algorithms. Applying these methods of SSPLS on NIR with real-valued response variable is provided in Section 5.8 using (NIR data). In Section 5.9, we applied both methods of SSPLS on the (CNA data) where the response variable is binary. Finally, some discussion is given in Section 5.10.

5.2 Sparse PLS

A sparse PLS (SPLS) method for wavelength selection in spectroscopic data has been proposed by Chun & Keleş (2010) who considered imposing the sparsity at the dimension reduction step using an L_1 penalty. For classification with microarray data Chung & Keles (2010) proposed two methods of SPLS to classification problems. The first is SPLS discriminant analysis (SPLSDA) and the second is sparse generalised PLS

(Chung & Keles, 2010). Both of these methods aim to improve the model by employing variable selection and dimension reduction simultaneously (Chung & Keles, 2010). Moreover, Lee *et al.* (2011) proposed a new SPLS method by the use of an unbounded penalty called hierarchical likelihood (HL) proposed by Lee & Oh (2009) to achieve sparsity.

Recall the first NIPALS algorithm (1) where in the first step, the solution vector w_m is written as

$$w_m = z_m^T y_m^T t_m.$$

Lee *et al.* (2011) considered writing the direction vector (w) for a component as the OLS estimator in the multivariate regression model. However, since our model is based on a univariate response variable, we will consider writing w for a component in the following univariate regression model

$$z = cw^T + \epsilon_z, \tag{5.1}$$

where the univariate response variable $z = y^T X$ is a raw vector of length p, the covariate $c = y^T t$ is a scalar and the regression coefficient is a p vector. Let ϵ_z represent the error term, and it is normally distributed.

To impose sparseness on the PLS direction vectors (w), Lee *et al.* (2011) considered the penalised least squares estimation of the regression model in Equation (5.1) by minimising Equation (5.2)

$$Q_\theta(w, z) = \sum_{j=1}^{p} \left\{ \frac{1}{2}(z_j - cw_j)^T (z_j - cw_j) + p_\theta(|w_j|) \right\}. \tag{5.2}$$

Where $p_\theta(.)$ is a penalty function. Here, given the tuning parameter θ, the solution of w can be obtained. With $p_\theta(w) = \frac{1}{\theta} w$, SPLS-L1 is found.

Following the random effect model approach, we assume that the random error (ϵ_z) in Equation (5.1) to follow a normal distribution with mean 0 and covariance Σ_z. We also assume that w is p vector of random effects to follow a Laplace distribution with location 0 and scale $\sqrt{\theta}$. We assume that w and ϵ_z are independent of each other. The penalised log-likelihood $L(w)$ can be written as

$$\log L(w, \theta) = -\frac{1}{2}(z - cw^T)\Sigma_z^{-1}(z - cw^T)^T - \frac{1}{\sqrt{\theta}} \sum_{j=1}^{p} |w_j|,$$

where the second term of Equation (5.3) is the penalty function. This penalty is not well behaved because it is not differentiable at $w = 0$ though it is concave and continuous. Thus, to derive an estimation of w, we could use one of the standard convex optimisers for instance convex GLM optimisations as the proposed IWLS algorithm by Lee *et al.* (2006). However they give an approximate solution. We will use gradient ascent, as proposed by Goeman (2010). Because our focus is on the mixture of smoothness and sparseness, we provide the description of the gradient ascent algorithm in Section 5.4.1 more specifically Algorithm 3.

Lee *et al.* (2011) used the unbounded penalty proposed by Lee & Oh (2009) as well as the L_1 penalty as a special case of their (HL) penalty in order to have sparsity. With $p_\lambda(|w_j|) = \lambda|w_j|$ we shall have the SPLS-L_1, where $\lambda \equiv \frac{1}{\theta}$. The HL penalty is based on the use of the random-effect models to generate penalty functions for variable selection (Lee & Oh, 2009). Using the double hierarchical generalised models (DHGLMs) proposed by Lee & Nelder (2006), Lee & Oh (2009) suggested the use of a gamma mixture for β as a penalty function for β. The normal-type (L_2) and LASSO-type (L_1) penalties are special cases of the unbounded penalty Lee & Oh (2009).

SPLS-L1 and SPLS-HL do not consider the correlation between covariates especially for highly-correlated data as NIR and CNA data sets. The correlation between covariates can be tackled by assuming the second differences of the direction vectors (w) to follow a Cauchy distribution to achieve smoothness. The smoothness considers the correlation between variables in order to interpret the estimation easily since the SPLS-L1 and SPLS-HL may improve the prediction accuracy but with sacrificing the interpretation.

5.3 Smoothed PLS with first NIPALS algorithm

SPLS methods in Section 5.2 for highly-correlated data may have difficulties in the interpretation of the estimation of $\hat{\beta}$. For example, when a group of predictors that are associated with the outcome, are highly-correlated, but SPLS identify only one or some of those predictors that are associated with outcome. Here, we consider the correlation between predictors and tackle this problem with Cauchy penalty. This penalty allows for the neighbouring predictors to be taken into account since we have dependencies between predictors as the windows in the CNA data and wavelengths in NIR data. Thus, we need to have smoothness to tackle the dependencies between covariates.

To impose smoothness on the PLS direction vectors (w), we consider the penalised least squares estimation of the regression model in Equation (5.1) by minimising Equation (5.2) where $p_\theta(.)$ is a penalty function as in Lee *et al.* (2011) for sparseness.

To impose smoothness, we assume that

$$\Delta w_j = w_j - w_{j-1},$$

second differences

$$\Delta^2 w_j = w_j - 2w_{j-1} + w_{j-2},$$

to follow a specific distribution such as normal (Pawitan, 2001). However, since we are dealing with CNA data, Huang *et al.* (2009) suggest to assume the random effects to follow Cauchy distribution instead of a normal distribution. Cauchy distribution is a heavy-tailed distribution which allows for the sudden jumps. We will consider the second differences because in CNA data has many jumps as well as strong serial correlations which needs smoothing.

Therefore, we assume the second differences of the direction vector (w) i.e.

$$\Delta^2 w \equiv \begin{pmatrix} w_3 - 2w_2 + w_1 \\ w_4 - 2w_3 + w_2 \\ \vdots \\ w_p - 2w_{p-1} + w_{p-2} \end{pmatrix},$$

to follow Cauchy distribution with location zero and scale $K(\theta_1) \equiv \theta_1 I_{p-2}$, where $\theta_1 = \sigma^2$. Assuming that $w \sim Cauchy(0, \theta_1 I_{p-2})$ is equivalent to assuming w is Cauchy distribution with location 0 and inverse scale matrix $K(\theta_1)^{-1} \equiv \theta_1^{-1} R_2^{-1}$, where

$$R_2^{-1} \equiv \Delta^{2^T} \Delta^2 = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & \dots & 0 \\ -2 & 5 & -4 & 1 & 0 & \dots & 0 \\ 1 & -4 & 6 & -4 & 1 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 1 & -4 & 6 & -4 & 1 \\ 0 & \dots & 0 & 1 & -4 & 5 & -2 \\ 0 & \dots & \dots & 0 & -1 & -2 & 1 \end{pmatrix},$$

and Δ^2 is $(p-2)\times p$ in dimension which is defined as the generalised inverse of R_2^{-1}; in practice we never need to compute it (Pawitan, 2001).

We follow Gusnanto & Pawitan (2015) when they assume β in the general linear regression model as random effects and follow a Cauchy distribution, but in our case, we assume that the second differences of the direction vectors (w) have a penalty function which is the Cauchy distribution.

Assuming that the random error, ϵ_z, in Equation (5.1) to follow a normal distribution. This means that the conditional distribution $f(z|w)$ is assumed to follow a normal distribution with mean cw^T and variance $\Sigma_z = \sigma_z^2 I_p$. Thus, the conditional distribution of z given w is

$$f(z|w) = (2\pi)^{-\frac{p}{2}}|\Sigma_z|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\Big((z-cw^T)\Sigma_z^{-1}(z-cw^T)^T\Big)\right]. \tag{5.3}$$

Let $f(w)$ be the probability density function (PDF) of the multivariate Cauchy distribution for the parameter w as

$$f(w) = \Gamma\left(\frac{1+p}{2}\right)\Gamma\left(\frac{1}{2}\right)^{-1}\pi^{-\frac{p}{2}}|K(\theta_1)|^{-\frac{1}{2}}(1+w^TK(\theta_1)^{-1}w)^{-\frac{p+1}{2}}. \tag{5.4}$$

We assume that w and ϵ_z are independent of each other. The log-likelihood of the parameters can be written as

$$\log L(w,\theta) = \log f(z|w) + \log f(w). \tag{5.5}$$

Where $\log f(w)$ is the penalty function following the second-difference of the Cauchy distribution in order to achieve the smoothness.

The first term in Equation (5.5) is given by

$$\log f(z|w) = -\frac{1}{2}\log|\Sigma_z| - \frac{1}{2}\Big((z-cw^T)\Sigma_z^{-1}(z-cw^T)^T\Big). \tag{5.6}$$

The second term of Equation (5.5) is the penalty function which assume the second differences of adjacent values of w to follow a Cauchy distribution.

Combining the conditional log-likelihood with the log of penalty ($\log f(w)$) with omitting the terms that do not depend on w we have

$$\begin{aligned}\log L(w,\theta_1) = &-\frac{1}{2}\log|\Sigma_z| - \frac{1}{2}(z-cw^T)\Sigma_z^{-1}(z-cw^T)^T \\ &-\frac{1}{2}\log|K(\theta_1)| - \frac{p+1}{2}\log\Big(1+w^TK(\theta_1)^{-1}w\Big).\end{aligned} \tag{5.7}$$

To derive an estimation of w at fixed values of θ_1 for each m components, we take the first derivative of the log-likelihood in Equation (5.7) with respect to w, and setting this to zero and solve for w. This gives the estimation of w as

$$\hat{w}_c = \left(\Sigma_z^{-1} c^T c + \left[\frac{(p+1)K(\theta_1)^{-1}}{1 + w^T K(\theta_1)^{-1} w} \right] \right)^{-1} \Sigma_z^{-1} z^T c, \tag{5.8}$$

where the estimation of the second differences of $\hat{w}_c$ under the Cauchy distribution assumption. With a starting value w_c^0, the estimation of w_c is done alternatively at a fixed value of θ_1 by first computing w in the right hand side of Equation (5.8).

Σ_z^{-1} is a diagonal matrix of σ_z^2 where the estimation of σ_z^2 can be done robustly as Gusnanto & Pawitan (2015) by

$$\hat{\sigma}_z^2 = \text{median}(z - c\hat{w}_c)^2.$$

For m and θ_1, we estimate them using five-fold cross-validation. For each θ_1 and m, we compute the mean square error of cross-validation (MSECV) as proposed by Lee *et al.* (2011), and is given by

$$\text{MSECV}(\theta_1, m) = \frac{1}{n} \sum_{f=1}^{5} ||y_{[s]} - X_{[s]} \hat{\beta}_{m,\theta_1}^{-s}||^2. \tag{5.9}$$

Where $\hat{\beta}_{m,\theta_1}^{-s}$ is the coefficient estimates using the m number of components and θ_1 the s-th test set. We use the s-th validation sets for $y_{[s]}$ and $X_{[s]}$ to calculate the prediction values of y. Then, m and θ_1 are chosen such that minimise MSECV(m, θ_1).

5.4 Sparse-smoothed PLS with first NIPALS algorithm

In this section, we consider the first NIPALS algorithm 1 as in Lee *et al.* (2011), but with different approaches or penalty functions. There are mainly two problems: dealing with the dependencies between neighbouring variables; and selecting a set of significant variables among a large number of variables. Consequently, we will impose smoothness and sparseness simultaneously on the PLS direction vectors by assuming w as random effects. In our proposed sparse smoothed partial least squares regression method SSPLS, we will be using the mixture of two distributions as a penalty term.

We initially considered a mixture of three distributions: normal, Cauchy (second-differences) and Laplace. The main goal of choosing the normal distribution is in order to shrink large values of w. However, there is no shrinking of the extreme values of w for our model since we are using the NIPALS algorithm. This algorithm maximises the covariance between X and y such that the norm of w equals to one. Therefore, although we assume that the random effects w to follow a normal distribution, the extreme values of w will not be shrunk. The proof of this penalty function normal-type (L_2) can be found in Zou *et al.* (2006) for sparse PCA, but with changing to the direction vectors w which are independent to θ. Therefore, we consider a mixture of only two penalties (Cauchy to achieve smoothness) and (Laplace to achieve sparseness).

5.4.1 Mixture of the products of Cauchy and Laplace distributions

In this section, we assume that the w follows a mixture model of two distributions: the first one is the second differences of adjacent values of w to follow a Cauchy distribution with location 0 and a scale matrix $K(\theta_1)$; the second one is a Laplace distribution with location 0 and scale $\sqrt{\theta_2}$ as seen in Equation (5.13). A similar idea was first discussed in Tibshirani *et al.* (2005) for penalised regression perspective, and later in Huang *et al.* (2009) for smoothed logistic regression.

The penalised log-likelihood based on the observation z and the random effect w is given in Equation (5.5) where the first term given in Equation 5.6 is the conditional distribution of z given w. The penalty function ($\log f(w)$) is called the sparse smoothed penalty that assumes the random effects follow a mixture of two distributions: Cauchy (for the second-order differences) and Laplace distributions. The second-difference of the Cauchy distribution in order to achieve the smoothness. The combined density is Huang *et al.* (2009)

$$\log f(w) = \tau \log f_1(w) + (1-\tau)\log f_2(w), \tag{5.10}$$

where

$$\log f_1(w) = -\frac{(p+1)}{2}\log(1 + w^T K(\theta_1)^{-1} w), \tag{5.11}$$

and

$$\log f_2(w) = -\frac{1}{\sqrt{\theta_2}}\sum_{j=1}^{p} |w_j|. \tag{5.12}$$

We introduce the tuning parameter in order to control the penalty type. For example, if the smoothness is only desired, we set τ to be equal to 1. If the sparseness is only desired, we set τ to be zero. For simplicity, we set $\theta_1 = \theta_2 = \theta$ so that we would only have to look for one tuning parameter θ instead of two different values for θs. Thus, the log-likelihood with respect to the tuning parameters is

$$\begin{aligned}\log L(w, \theta, \tau) = &-\frac{1}{2}(z - cw^T)\Sigma_z^{-1}(z - cw^T)^T \\ &- \left[\tau \frac{(p+1)}{2} \log(1 + w^T K(\theta)^{-1} w) \right. \\ &\left. + \frac{(1-\tau)}{\sqrt{\theta}} \sum_{j=1}^{p} |w_j| \right], \end{aligned} \tag{5.13}$$

where the second and third terms correspond to the penalty function. This penalty function is a mixture of second order difference Cauchy and Laplace assumptions for w. The τ is the weight for both distributions. If $\tau = 0.5$, then the penalty is divided into half for smoothness and the other half is for sparseness. If $\tau = 0$, then the penalty is an L_1 penalty sparsity. But, if $\tau = 1$, then we are considering only smoothing with no sparsity on the estimation of w.

The estimation of the model parameter w given the tuning parameters $H = (m, \theta, \tau)$ is done by estimating w at fixed value of H. We first differentiate the log-likelihood $\log L(w)$ in Equation (5.13) partially with respect to w.

To derive an estimation of w, we could use one of the standard convex optimisers as discussed earlier in Section 5.2. However they give an approximate solution. We will use gradient ascent, as proposed by Goeman (2010), with some modifications. In our model assumption, we include the penalty of the Cauchy distribution in the log-likelihood whereas Goeman (2010) chose to use only the log partial likelihood because he used only a Laplace distribution for the random effects assumption. The combination of the likelihood and the Cauchy distribution is still well behaved because it is continuous and twice differentiable everywhere.

Gradient ascent was chosen as a method for estimating our model parameter w because it can be understood by looking more closely into the penalised log-likelihood function that is to be optimised. The penalised log-likelihood in Equation (5.13) can be rewritten as a sum of two terms:

$$\log L(w,\tau) = l_{\mathrm{c}} - \frac{(1-\tau)}{\sqrt{\theta}} \sum_{j=1}^{p} |w_j|. \tag{5.14}$$

The first term of Equation (5.14), l_c, is the combination of the first two terms on the right hand side of Equation (5.13), is log-likelihood of w plus the penalty from the Cauchy distribution. This is a highly regular function: everywhere at least twice differentiable in the target model given in Equation (5.13).

The second term of Equation (5.14), the penalty of Laplace, $f(w) = \frac{(1-\tau)}{\sqrt{\theta}} \sum_{j=1}^{p} |w_j|$, is not well behaved: it is concave and continuous, but is only differentiable at points with $w_j \neq 0$ for all j.

A gradient ascent algorithm for the optimisation of one coefficient after taking the derivative at that point, it takes a step in that direction. it estimates this coefficient when the derivative is zero.

There are two issues that should be noted: the penalised likelihood function may show weak concavity near the optimum, especially if θ is large, this can be a major convergence problem for Lasso algorithms in general; the penalised likelihood is not differentiable everywhere due to the lack of differentiability of Lasso penalty (Goeman, 2010). Although of these issues, Goeman (2010) defined a directional derivative

$$l'_p(w;v) = \lim_{t\to 0} \frac{1}{t}\{l(w+tv) - l(w)\}$$

for every point w and every direction $v \in \Re^p$. The gradient can then be defined for every w as the scaled direction of steepest ascent. The algorithm follows the gradient in the direction v_{opt} which maximises $l'_p(w;v)$ among all v such that $||v|| = 1$

Therefore, we can calculate the gradient as follows:

$$g_j(w) = \begin{cases} h_j(w) - \frac{1-\tau}{\sqrt{\theta}}\text{sign}(w_j) & \text{if } w_j \neq 0 \\ h_j(w) - \frac{1-\tau}{\sqrt{\theta}}\text{sign}(h_j(w)) & \text{if } w_j = 0 \text{ and } |h_j(w)| > \frac{1-\tau}{\sqrt{\theta}} \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

where

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0, \end{cases}$$

and $h_j(w)$ is the first derivative of the log-likelihood and the Cauchy penalty in Equation (5.13).

The gradient ascent algorithm is calculated in an iterative way until convergence, using update,

$$w_{new} = w_{old} + tg(w_{old}),$$

where t is the step size.

The penalised likelihood is approximated locally at each step from w in the direction of the gradient by a directional second order Taylor approximation. There is no meaning of this approximation if it is not within a single subdomain of gradient continuity, i.e. for $0 < t < t_{\text{edge}}$, with

$$t_{\text{edge}} = \text{min}_j \left\{ -\frac{w_j}{g(w_j)} \text{:sign}(w_j) = -\text{sign}\{g(w_j)\} \neq 0 \right\},$$

where the optimum of the Taylor approximation in the subdomain is at

$$t_{\text{opt}} = -\frac{l'(w; g(w))}{l''(w; g(w))}.$$

For every w and $g(w)$, $l^{'}(w; g(w))$ and $l^{''}(w; g(w))$ are the directional first and second derivative, respectively.

$$l^{'}(w; g(w)) = g(w)^T g(w)/||g(w)||$$

$$l^{''}(w; g(w)) = g(w)^T \frac{\partial^2 l_{\text{c}}}{\partial w \partial w'} g(w).$$

It is difficult to calculate the full $p \times p$ Hessian matrix of l_{c} to obtain the directional second derivative in practice because the direction $g(w)$ of interest, which is the direction of the gradient, will have many zeros. Therefore, the algorithm is shown below

Algorithm 3 A simple modified gradient ascent algorithm for the penalised log-likelihood of SSPLS

```
1: Start with some $w^{(0)}$.
2: For steps $s = 0, 1, 2, \dots$: iterate $w^{(s+1)} = w^{(s)} + \text{min}\{t_{\text{opt}}, t_{\text{edge}}\} g(w^{(s)})$
3: End if it converges ($g(w) = 0$).
```

The modification to the first NIPALS algorithm (1) is only in the first step where the estimation of w is replaced by an iterative estimate of w as in Algorithm (3).

5.4.2 Tuning parameter selection

For $\hat{\sigma}_z^2$, we estimate it as described in Section 5.3.

There is a global model for PLS which uses the regression coefficients vector using the PLS model (β_{pls}). In this global model, we choose the optimal values for the tuning parameters, which are θ, τ and m, via five-fold cross-validation. For each θ, τ, and m, we compute the (MSECV) for a normal response as

$$\text{MSECV}(m, \theta, \tau) = \frac{1}{n} \sum_{f=1}^{5} ||y_{[s]} - X_{[s]} \hat{\beta}_{m,\theta,\tau}^{-s}||^2. \tag{5.16}$$

The optimal θ, τ and m are chosen that correspond to the minimum value of the MSECV.

For a binary response we use the same procedure as in the normal case except we calculate the misclassification error rate of cross-validation (MERCV) as proposed by Gusnanto *et al.* (2015) instead of the MSECV.

Considering that there is a local model in each component which is represented by Equation (5.1), this model used to choose the optimal θ and τ for each component. The optimal values of θ and τ can be chosen in each component by maximising the marginal of the log-likelihood in the penalised log-likelihood in Equation (5.13). We use five-fold cross-validation with the maximum of the marginal log-likelihood. After we choose the optimal θ and τ in each component, we compute using these optimal θ and τ the global model's coefficients vector (β_{pls}). Now, the MSECV is calculated using β_{pls} to choose the optimal m for the normal response as given in Equation (5.16).

For the binary response, we do the same procedure for choosing the optimal θ and τ locally, but instead of using MSECV, we use MERCV.

5.5 Smoothed PLS with second NIPALS algorithm

In this section, we consider smoothed PLS based on the second NIPALS algorithm 2. Consider SPLS2 proposed by Lee *et al.* (2011), where the direction vector w in the first step of the second NIPALS algorithm was regarded as the OLS estimator in the following regression problem

$$X = yw^T + \epsilon_x, \tag{5.17}$$

where ϵ_x is a random error matrix. Given y, the OLS estimator for w minimises

$$\text{trace}\Big((X - yw^T)^T(X - yw^T)\Big), \tag{5.18}$$

where the trace of a matrix is the sum of the diagonal values.

Lee *et al.* (2011) imposed sparseness on the PLS direction vector w, with the objective function

$$\text{trace}\Big((X - yw^T)^T(X - yw^T)\Big) + p_\theta\Big(|w|\Big), \tag{5.19}$$

where $p_\theta(.)$ is a penalty function. Lee *et al.* (2011) argued that the unbounded HL penalty is needed for more sparseness when the number of predictors is very large as in genomic data applications.

In this section, we consider again the dependencies between variables since NIR and CNA data sets are highly-correlated. The second proposed method SPLS2 with HL penalty proposed by Lee *et al.* (2011) does not take into account the dependency between the variables. First, we assume that the second differences of adjacent values of w to follow a Cauchy distribution with location 0 and inverse scale matrix $K(\theta)^{-1}$ as defined in Section 5.3.

We assume that the random error matrix, ϵ_x, in Equation (5.15) to follow a multivariate normal distribution. Thus, the conditional distribution of X given w is a multivariate normal distribution with mean yw^T and variance-covariance matrix Σ_x. The conditional distribution of X given w can be written for one component as

$$f(X|w) = (2\pi)^{-\frac{p}{2}}|\Sigma_x|^{-\frac{1}{2}} \exp\Bigg[-\frac{1}{2}\Big((X - yw^T)\Sigma_x^{-1}(X - yw^T)^T\Big)\Bigg]. \tag{5.20}$$

To impose the smoothness, we assume that w as a random effect and the second differences of adjacent values of w to follow a Cauchy distribution.

We assume also that w is independent of ϵ_x. Combining the conditional log-likelihood with $\log p(w)$, we have

$$\begin{aligned}\log L(w,\theta) = -\frac{1}{2}\log|\Sigma_x| - \frac{1}{2}\sum_{i=1}^{n}\text{trace}\Big(\Sigma_x^{-1}(X_i - y_i w^T)(X_i - y_i w^T)^T\Big)\\ -\frac{(p+1)}{2}\log(1 + w^T K(\theta)^{-1}w).\end{aligned} \tag{5.21}$$

At fixed values of θ and m, we take the first derivative of the log-likelihood with respect to w, and setting this derivative to zero and solving it for w, we will have the estimate of w as

$$\hat{w}_c = \left(\Sigma_x^{-1} y^T y + \left[\frac{(p+1)K(\theta)^{-1}}{1 + w^T K(\theta)^{-1} w} \right] \right)^{-1} \Sigma_x^{-1} X^T y, \tag{5.22}$$

where w_c, is the second difference of w estimates, are under the Cauchy distribution assumption on w, and $K(\theta)^{-1}$ is defined in Section 5.3.

The estimation of Σ_x of the model in Equation (5.16) can be done by calculating the median of the mean of each row of Σ_x. Then σ_x^2 can be estimated as proposed by the median of the mean of each row as (Gusnanto & Pawitan, 2015)

$$\hat{\sigma}_x^2 = \text{median}\{\text{mean}_i (X_i - y_i \hat{w})^2\}. \tag{5.23}$$

We estimate m and θ using five-fold cross-validation as described in Section 5.3.

5.6 Sparse-smoothed PLS with second NIPALS algorithm

In this section we follow the same characteristics of the penalty in Section 5.4 with only changing the conditional likelihood depending on the NIPALS algorithms that used here is the second NIPALS algorithm 2.

5.6.1 Mixture of smoothed Cauchy and Laplace distributions

In this section, we follow the same approach in SSPLS with first NIPALS algorithm for the penalty functions by assuming the mixture of Cauchy and Laplace distributions. We assume also that w is independent of ϵ_x. We combine the conditional log-likelihood with $\big(\tau \log f_c(w; K(\theta_1))$ and $(1-\tau) \log f_l(w; \sqrt{\theta_2})\big)$. Assuming that $\theta_1 = \theta_2 = \theta$, and omitting the terms that do not depend on w, we have the same penalty function in Section 5.4.1 but different conditional likelihood. Thus, Equation (5.13) becomes as

$$\log L(w, \theta, \tau) = -\frac{1}{2} \sum_{i=1}^{n} \text{trace}\Big(\Sigma_x^{-1} (X_i - w^T y_i)(X_i - w^T y_i)^T \Big)$$
$$- \left[\tau \left(\frac{(p+1)}{2} \log(1 + w^T K(\theta)^{-1} w) \right) + \left(\frac{(1-\tau)}{\sqrt{\theta}} \sum_{j=1}^{p} |w_j| \right) \right]. \tag{5.24}$$

Where the last two terms of the log-likelihood that correspond to the penalty function, which is a mixture of Cauchy and Laplace distributions. The τ is the weight for each distribution. The penalty function has the same characteristics as in the SSPLS with the first NIPALS algorithm described in Section 5.4.

The estimation of the model parameter w and the tuning parameter $H = (m, \theta, \tau)$ is done by estimating w at fixed H. Selecting the tuning parameter is done via five-fold cross-validation. We compute the mean squared error prediction of cross-validation (MSECV) for H.

We will use gradient ascent since the penalised likelihood in Equation (5.24) is not differentiable everywhere when $j = 0$ for $|w_j|$. Therefore, we follow the estimation of w as in the first method of SSPLS with changing the marginal log-likelihood to be as in Equation (5.24). The modified gradient ascent algorithm is the same as the Algorithm 3, but the $h_j(w)$ which is used in the calculation of the gradient $g_j(w)$ is based on the log-likelihood defined in Equation (5.24). That means $h_j(w)$ is going to be a combination of the first two terms in the right hand side of Equation (5.24) which are the log likelihood plus the penalty from the Cauchy distribution. The rest remains the same as in Section 5.4.1.

The modification to the second NIPALS algorithm (2) is only in the first step where the estimation of w is replaced by an iterative estimate of w as in Algorithm 3.

5.6.2 Tuning parameter selection

For selecting the tuning parameters $H = (m, \theta, \tau)$ we use the same measures in Section 5.4.2 for the local and global models and for the normal and binary responses via MSECV and MERCV respectively.

We compute the estimation of σ_x^2 as in Equation (5.23).

5.7 Simulation results

5.7.1 Simulation setting

To understand the working characteristics of the model, we follow the simulation setting of Bøvelstad *et al.* (2007) , Nygård *et al.* (2008), and Lee *et al.* (2011). The details and procedures are explained in the following paragraph.

We generate a matrix of predictors of size $n \times p$ where $n = 100$ and $p = 200$ as $X \sim MVN(0, \Psi)$ where MVN is the multivariate normal distribution density and Ψ is 200 × 200 block diagonal covariance matrix. For a given L, it is defined as

$$\Psi = \begin{pmatrix} \Psi_{11} & 0 & \dots & 0 \\ 0 & \Psi_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Psi_{LL} \end{pmatrix},$$

and, for $l = 1, 2, \dots, L$, Ψ_{ll} is of size $L \times L$, L^T= (200/L), and defined as

$$\Psi_{ll} = \begin{pmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \dots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \dots & 1 \end{pmatrix},$$

In this simulation, we set ρ equals to 0.9, because it is close to our real data and Lee *et al.* (2011) argued that their methods work better with higher correlation.
For $i = 1, 2, \dots, n$, we assumed

$$y_i = \sum_{j=1}^{40} x_{ij}\beta_j + \epsilon_i,$$

where y_i is the element of a response vector of length n, x_{ij} is the (i, j) element of X. β_j = exp $(-\alpha(j-1))$ and $\beta_{j+20} = \beta_j$ for $j = 1, \dots, 20$. This setting indicated that only the first 40 covariates are associated with response among 200 covariates. The error term ϵ_i is the element of the error vector and it is randomly distributed from normal distribution with mean zero and standard variation equals one.
The regression parameters were exponentially decaying, and the speed of the decay was controlled by the parameter α. We used a slow decay, where α = 0.0141, such that exp (-49α) = 0.5 as proposed by Lee *et al.* (2011).

5.7.2 Simulation results: one simulation using the first method of SSPLS

Figure 5.1 shows the estimated $\hat{\beta}$ with mixture of Cauchy and Laplace distributions using the first method of SSPLS. In this simulation, one component ($m = 1$) is used because the main reason of this figure is to show how the model changes over different values of τ. The left panel uses the first method of SSPLS with $\tau = 0$ to show the sparseness penalty only, and the middle panel uses the same method with $\tau = 0.5$ to show the effect of sparseness and smoothness together. The right panel uses the same method with $\tau = 1$ to show the smoothness effect only.

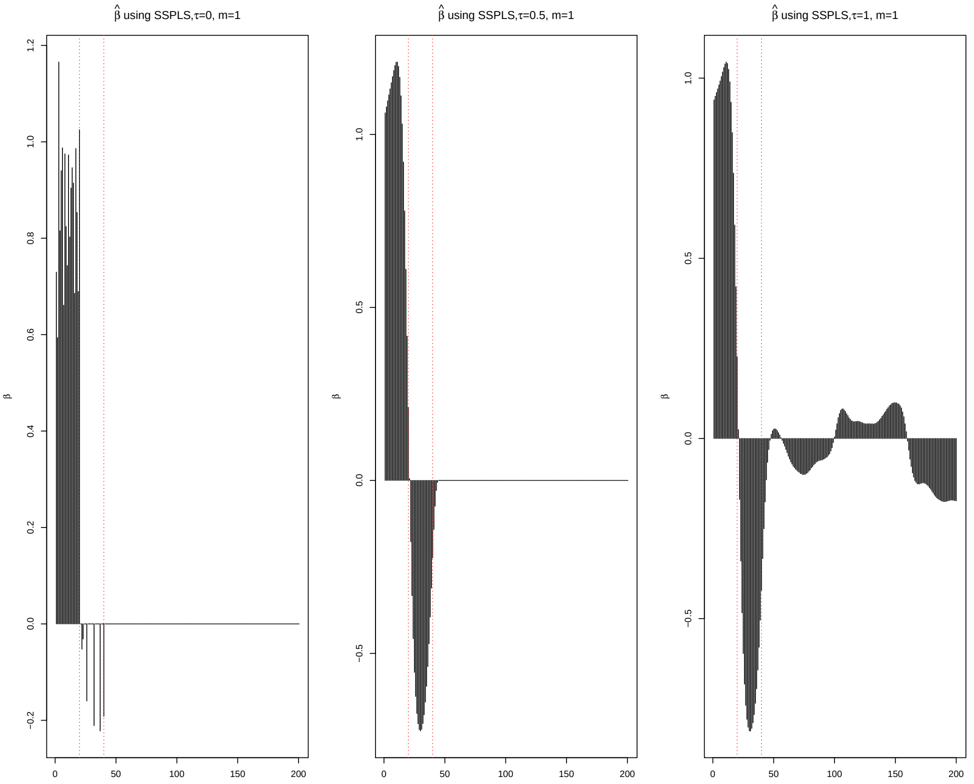


Figure 5.1: $\hat{\beta}$ with mixture of Cauchy and Laplace distributions penalty using the first method of SSPLS. The left panel uses $\tau = 0$, middle panel $\tau = 0.5$, and right panel uses $\tau = 1$, where $m = 1$, the red lines are to show the first 20 covariates are positive and the second 20 covariates are negative.

It can be seen from Figure 5.1 how the model using the first method of SSPLS is changed by changing the value of τ which is the relative weight for Cauchy (smoothness) and Laplace (sparseness). In the left panel, we can see there is no smoothing at all because the weight for Cauchy distribution is zero. It is clear that Lasso tends to detect only 6 covariates among the second 20 correlated covariates which are associated with the response. This is a well known potential problem with Lasso penalty (L_1 penalty)

In the middle panel of Figure 5.1, we set the weight of Cauchy and Laplace to be equal to 0.5 ($\tau = 0.5$). We can see the effect of smoothness and sparseness clearly. Moreover, in our simulation's setting, the first 20 covariates are positively and the second 20 covariates are negatively associated with response variable among 200 covariates. The first 26 covariates detect a signal only, and the rest covariates (41-200) are not equal to zero in the left panel.

In the right panel of Figure 5.1, $\tau = 1$, our model reduces to Cauchy only with no contribution for Laplace. Thus, none of the coefficients of β has been set to zero. This model can be called smoothed partial least squares (smoothed PLS).

5.7.3 Simulation results: one simulation using the second method of SSPLS

Figure 5.2 shows the estimated $\hat{\beta}$ with mixture of Cauchy and Laplace distributions using the second method of SSPLS. The results are based on one component since the importance of this figure is to show how the model changes over different values of τ. The left panel uses $\tau = 0$ to achieve the sparsity only, and the middle panel uses $\tau = 0.5$ to show the effect of the mixture of sparseness and smoothness with equal effect. The right panel uses $\tau = 1$ to show the smoothness effect without sparseness.

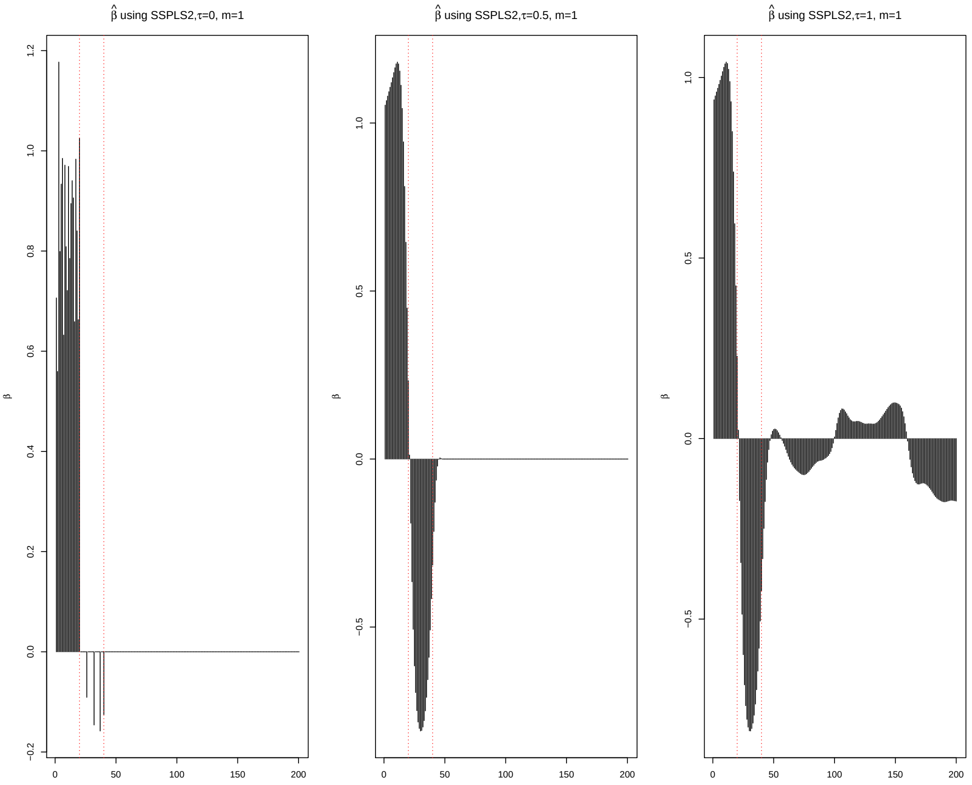


Figure 5.2: $\hat{\beta}$ with mixture of Cauchy and Laplace distributions penalty using the second method of SSPLS. The left panel uses $\tau = 0$, middle panel $\tau = 0.5$, and right panel uses $\tau = 1$, where $m = 1$.

We can see that by setting the weight to be equal to zero ($\tau = 0$), Laplace distribution is only used and our proposed second method of SSPLS reduces to Lasso solution. As we can see from the left panel of Figure 5.2, there are only 4 covariates reveal among the second 20 covariates. This is again a possible problem in Lasso penalty.

In the middle panel, setting the $\tau = 0.5$ every distribution will contribute to the model by 50 percent. This means that, we can see the sparseness and smoothness in the estimation of $\hat{\beta}$. Furthermore, among 200 covariates, the first 40 covariates are associated with the response variables in the setting of our simulation. The first 40 covariates are significant as we anticipated, and the rest covariates (41-200) are equal to zero in the left panel.

In the right panel by setting $\tau = 1$ indicates that only Cauchy distribution is being used in the second method of SSPLS. It can be seen non of the estimation of $\hat{\beta}$ has been set to be equal to zero because the weight for Laplace is zero. This model can be called the second method of smoothed PLS.

Looking at the results above in Figures 5.1 and 5.2, the first and second method of SSPLS tend to have similar results with different values of τ.

5.7.4 Simulation results: the first method of SSPLS for local and global models

In this section, we focus more on the first method of SSPLS with two different ways to select the optimal tuning parameters as discussed in Section 5.4.2. These two ways are called local and global models.

Figure 5.3 shows the plot for log of the MSECV over various number of components using five-fold cross-validation on simulated data with the first method of SSPLS for global and local models.

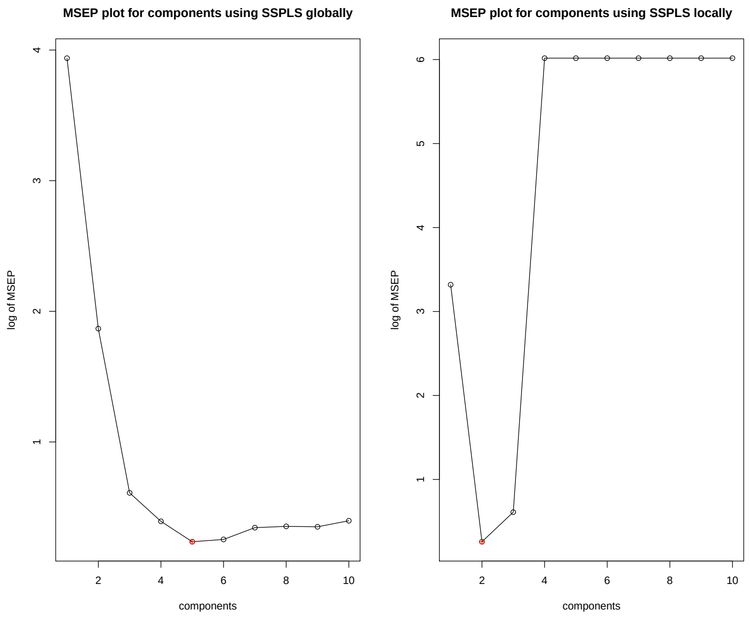


Figure 5.3: The log of MSPE over various number of components using five-fold MSECV on simulated data with the first method of SSPLS for global (left panel) and local (right panel) models. The red point gives the optimal number of components in each model.

As can be seen from Figure 5.3, left panel shows the log of the MSECV over different components and the optimal m is 5. This can be seen as large number of components compared to the optimal number of components for the local model (right panel), where $m = 2$, which can be seen from the right panel in the same figure.

Figure 5.4 shows the plot for the MSECV values using five-fold cross-validation with the first method of SSPLS globally for the optimal number of components (m) between various values of τ and θ.

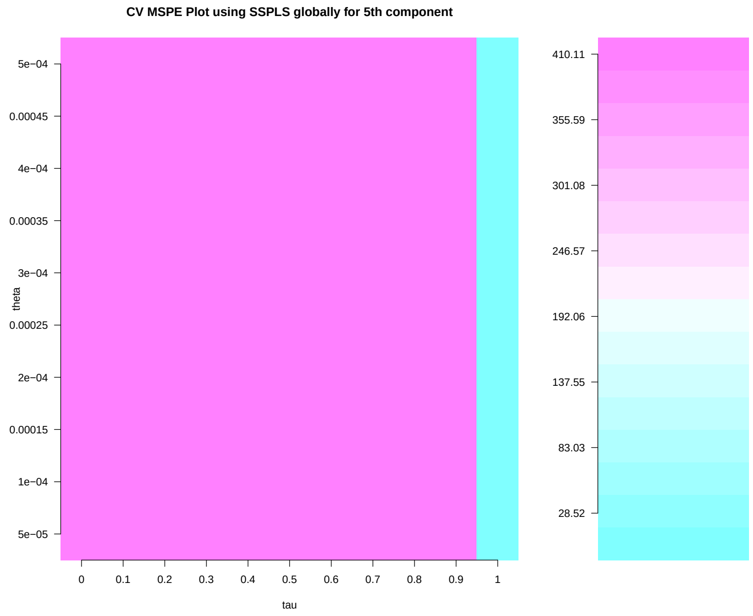


Figure 5.4: MSPE values using five-fold CV with first SSPLS globally for the optimal component between various values of τ and θ, where the optimal values for $\theta_{opt} = 0.00035$, $\tau_{opt} = 1$ and $m_{opt} = 5$.

In Figure 5.4, we consider the global model which is used the coefficients estimation β_{pls}. In this model, we select the optimal tuning parameters that give the minimum MSECV. Therefore, in all components, we use the same optimal values of τ and θ without considering each component independently. It can be seen that in this case, the optimal number of components $m_{opt} = 5$ and there is only one optimal value for each τ and θ since we use the global model.

Figure 5.5 shows plot for the maximum log-likelihood (MLL) values using five-fold CV with the first method of SSPLS locally for the first component between various values of τ and θ.

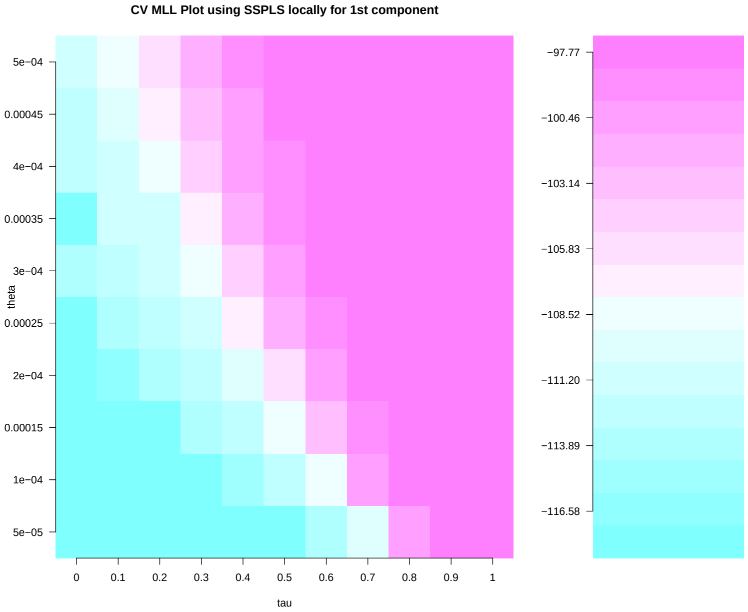


Figure 5.5: The maximum log likelihood (MLL) values using five-fold CV with first SSPLS locally for the first component between various values of τ and θ, where optimal $\theta_{opt} = 10^{-4}$ and $\tau_{opt} = 0.8$ for $m = 1$.

Figure 5.6 shows plot for the maximum log-likelihood (MLL) values using five-fold CV with the first method of SSPLS locally for the second component between various values of τ and θ.

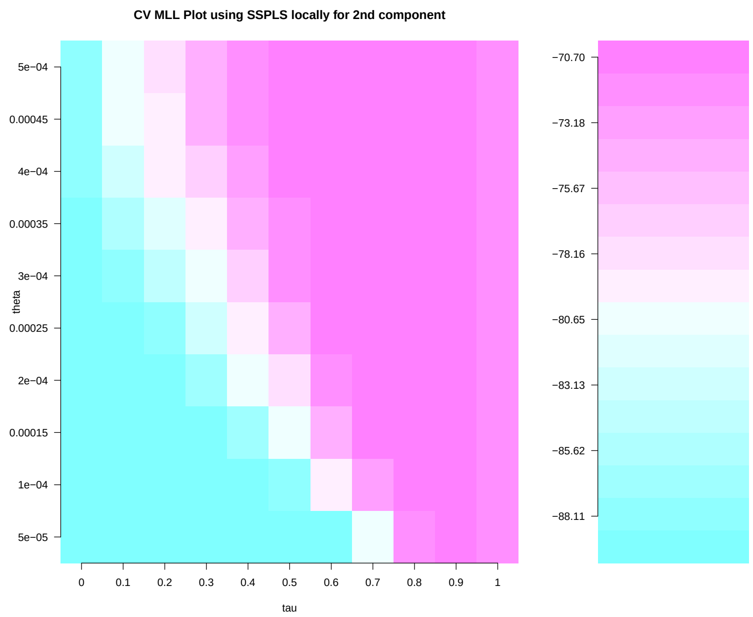


Figure 5.6: The maximum log-likelihood (MLL) values using five-fold CV with first SSPLS locally for the second component between various values of τ and θ, where optimal $\theta_{opt} = 5 \times 10^{-5}$ and $\tau_{opt} = 0.9$ for $m = 2$.

It can be seen from Figures 5.5 and 5.6 that the optimal values of the tuning parameters τ and θ in each component are different. This is because in each component, the optimal τ and θ are chosen based on the local model that maximises the marginal log-likelihood in Equation (5.13) without the penalty function.

Figure 5.7 shows the estimation of $\hat{\beta}$ using the mixture model of Cauchy and Laplace distributions with the optimal tuning parameters above. The left panel using the first SSPLS with the global model, and the right panel using the local model of the first method of SSPLS.

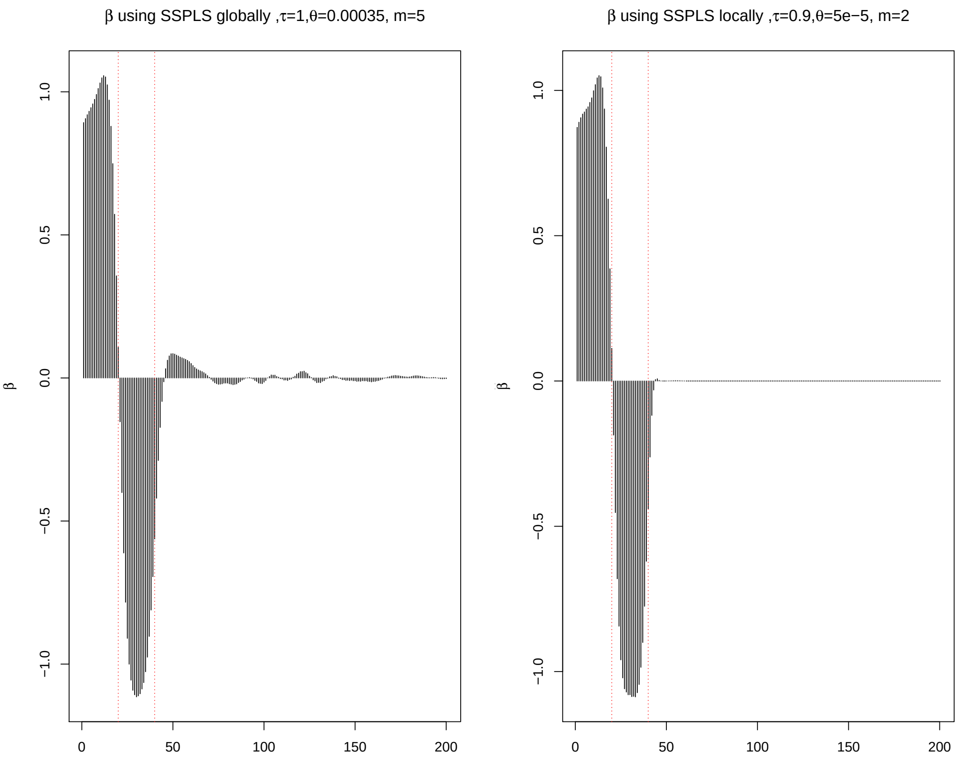


Figure 5.7: $\hat{\beta}$ with mixture of Cauchy and Laplace distributions penalty using the first SSPLS. The tuning parameters for the global model (left panel) are $\tau_{opt} = 1$, $\theta_{opt} = 0.00035$, $m_{opt} = 5$, and for the local model (right panel) are $\tau_{opt} = 0.8, 0.9$, $\theta_{opt} = 10^{-4}, 5 \times 10^{-5}$, $m_{opt} = 2$.

It can be seen from Figure 5.7 that the first 20 covariates should be positive and the covariates from 21 to 40 are negative as it has been set for the true β in the simulation. For the rest 160 covariates have been set equal to zero for true β in the simulation. These covariates are not equal to zero using the optimal tuning parameters based on the global model as seen in the left panel. Also, as the number of components is larger, none of the estimation of $\hat{\beta}$ are equal to zero. Since the optimal τ for the global model is 1, this means that there is only Cauchy effect and no sparse effect from the penalty function. However, in the right panel using the tuning parameters based on the local model, almost all of the covariates are zero for the non zero covariates. The first 40 covariates are associated with the outcome.

Figure 5.8 shows w_1 using the mixture model of Cauchy and Laplace distributions with the optimal tuning parameters above. The left panel using the first method of SSPLS with the global model, and the local model (right panel).

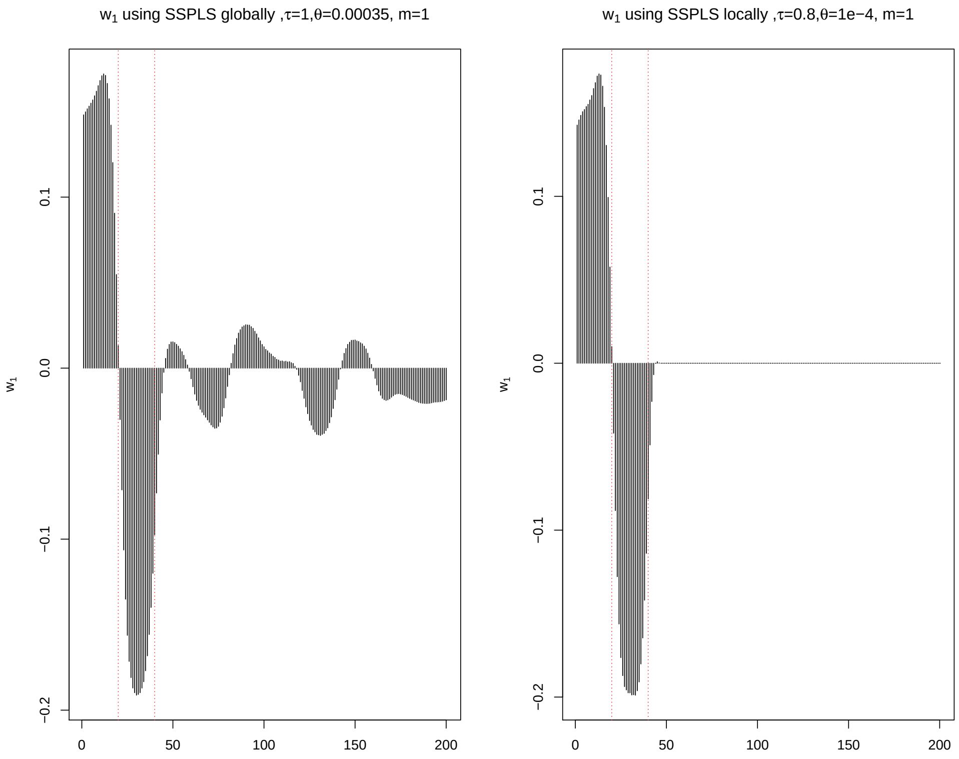


Figure 5.8: w_1 with mixture of Cauchy and Laplace distributions penalty using the first SSPLS. The tuning parameters for the global model are $\tau_{opt} = 1$, $\theta_{opt} = 0.00035$, $m = 2$, and for the local model are $\tau_{opt} = 0.8$, $\theta_{opt} = 10^{-4}$, $m = 2$.

It can be seen from the left panel of Figure 5.8 w_1 using the first method of SSPLS with the global model with no sparseness effect. This is because the $\tau_{opt} = 1$ which reduces our model to only smoothed PLS. On the other hand, it can be seen that the first 44 covariates only have signals and they are smoothed and not equal to zero while covariates from 45 to 200 are equal to zero. This indicates that our proposed model has provided sparseness and smoothness together.

Figure 5.9 shows w_2 with mixture of Cauchy and Laplace distributions using the optimal tuning parameters above. The left panel using the first SSPLS with the global model, and the right panel using the local model of the first SSPLS.

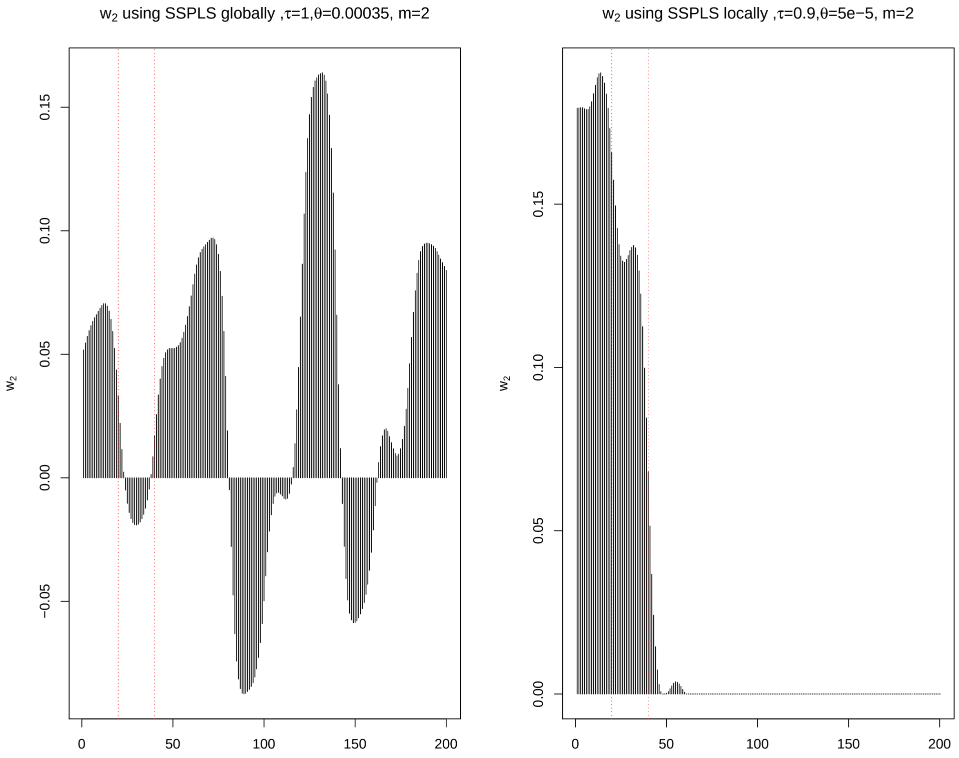


Figure 5.9: w_2 with mixture of Cauchy and Laplace distributions penalty using the first SSPLS. The tuning parameters for the global model are $\tau_{opt} = 1$, $\theta_{opt} = 0.00035$, $m = 2$, and for the local model are $\tau_{opt} = 0.9$, $\theta_{opt} = 5 \times 10^{-5}$, $m = 2$.

Looking at the left panel of Figure 5.9, it can be seen the effect of the sparseness is not appeared using the optimal tuning parameters globally because τ_{opt} is equal to 1. However, in the right panel, using the optimal tuning parameters in the second component, where $\tau_{opt} = 0.9$, the is some sparsity in the second component because weight for the Laplace distribution is 0.1 as the wight for Cauchy is 0.9. Moreover, as the number of components is large as in the global model which is 5 components, there is no sparsity effect for the first method of SSPLS.

In summary, it can be seen from this simulation and comparing the global and local models using the first method of SSPLS that there is a difference in the results of the values of the selected tuning parameters. The selection of the tuning parameters make changes in the estimation of $\hat{\beta}$ and the w_m for each component. Utilising the global model, we do not see any effect from the Laplace penalty which means our SSPLS model is reduced to only smoothed PLS model. In contrast, using the local model give more chance for the impact of Laplace distribution and Cauchy to contribute together in the model. Moreover, looking at the optimal number of components m with the global model, we need more number of components that in the local model. The reason could be that we treated all components with the same optimal values of τ and θ for the global model. However, in the local model we select the optimal τ and θ for each component independently, then we use them to select the optimal number of components m.

5.7.5 Simulation results: the second method of SSPLS for local and global models

Figure 5.10 shows the plot for the log of MSPE over various number of components using five-fold CVMSPE on a simulated data with the second method of SSPLS for global and local models.

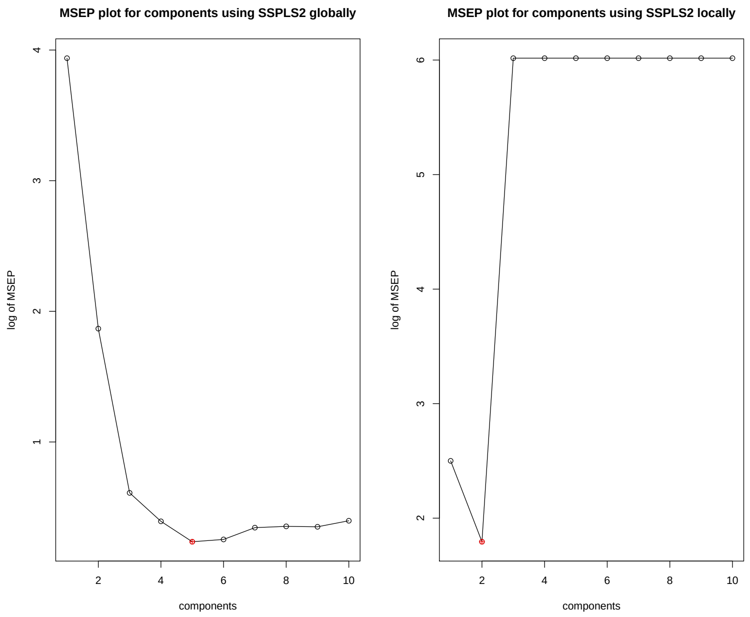


Figure 5.10: The log of MSPE over various number of components using five-fold CV on a simulated data with second method of SSPLS for global (left panel) and local (right panel) models, the red colour is referred to the optimal number of components (m).

It can be seen from the left panel of Figure 5.10, the optimal number of components is 5 using the global model for the second method of SSPLS. In the right panel, the optimal number of components is 2 using the local model for the second SSPLS.

Figure 5.11 shows plot for the log of MSPE using five-fold MSECV with second SSPLS globally for the optimal number of components ($m_{opt} = 5$) between various values of τ and θ.

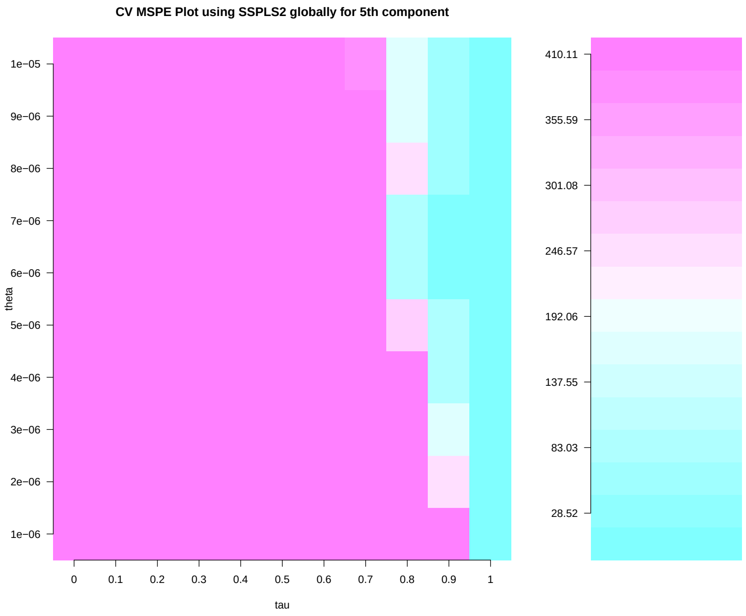


Figure 5.11: log of MSPE values using five-fold CV with second SSPLS globally for the optimal component between various values of τ and θ, where the optimal $\theta_{opt} = 4 \times 10^{-6}$, $\tau_{opt} = 1$ for $m_{opt} = 5$.

It can be seen from Figure 5.11 the optimal $\theta_{opt} = 4 \times 10^{-6}$ and $\tau_{opt} = 1$ using the global model for the second SSPLS in the optimal number of components ($m_{opt} = 5$).

Figure 5.12 shows plot for the maximum log-likelihood (MLL) values using five-fold CV with the second SSPLS locally for the first component between various values of τ and θ, where optimal $\theta_{opt} = 7 \times 10^{-6}$ and $\tau_{opt} = 0.3$ for $m = 1$.

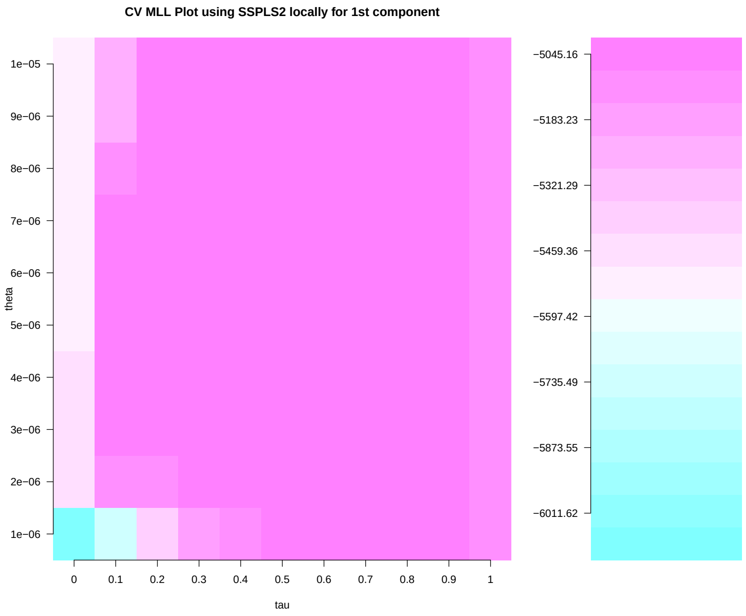


Figure 5.12: The maximum log-likelihood (MLL) values using five-fold CV with second SSPLS locally for the first component between various values of τ and θ, where optimal $\theta_{opt} = 7 \times 10^{-6}$ and $\tau_{opt} = 0.3$ for $m = 1$.

For the local model using second SSPLS, it can be seen from Figure 5.12 that the optimal $\tau_{opt} = 0.3$ and $\theta_{opt} = 7 \times 10^{-6}$ for the first component.

Figure 5.13 shows plot for the maximum log-likelihood (MLL) values using five-fold CV with the second method of SSPLS locally for the second component between various values of τ and θ, where optimal $\theta = 10^{-5}$ and $\tau_{opt} = 0.9$ for $m = 2$.

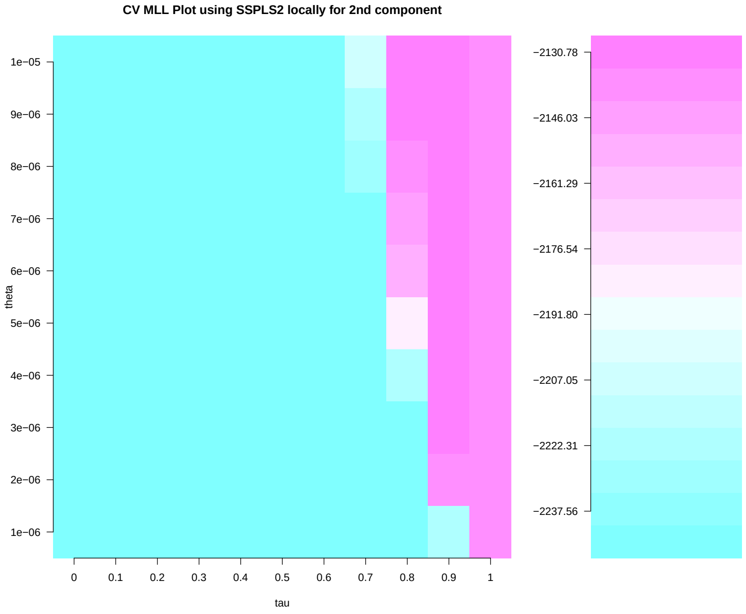


Figure 5.13: The maximum log-likelihood (MLL) values using five-fold CV with second SSPLS locally for the second component between various values of τ_{opt} and θ_{opt}, where the optimal $\theta_{opt} = 10^{-5}$ and $\tau_{opt} = 0.9$ for $m = 2$.

It can be seen from Figure 5.12 the optimal $\tau_{opt} = 0.9$ and $\theta_{opt} = 10^{-5}$ for the second component using using second SSPLS locally. It can be noticed that the optimal θ and τ are different from the optimal ones in the first component unlike the global model. In the global model the optimal θ and τ are the same for all candidate components

Figure 5.14 shows the estimation of $\hat{\beta}$ using second SSPLS with global model (left panel), and locally (right panel).

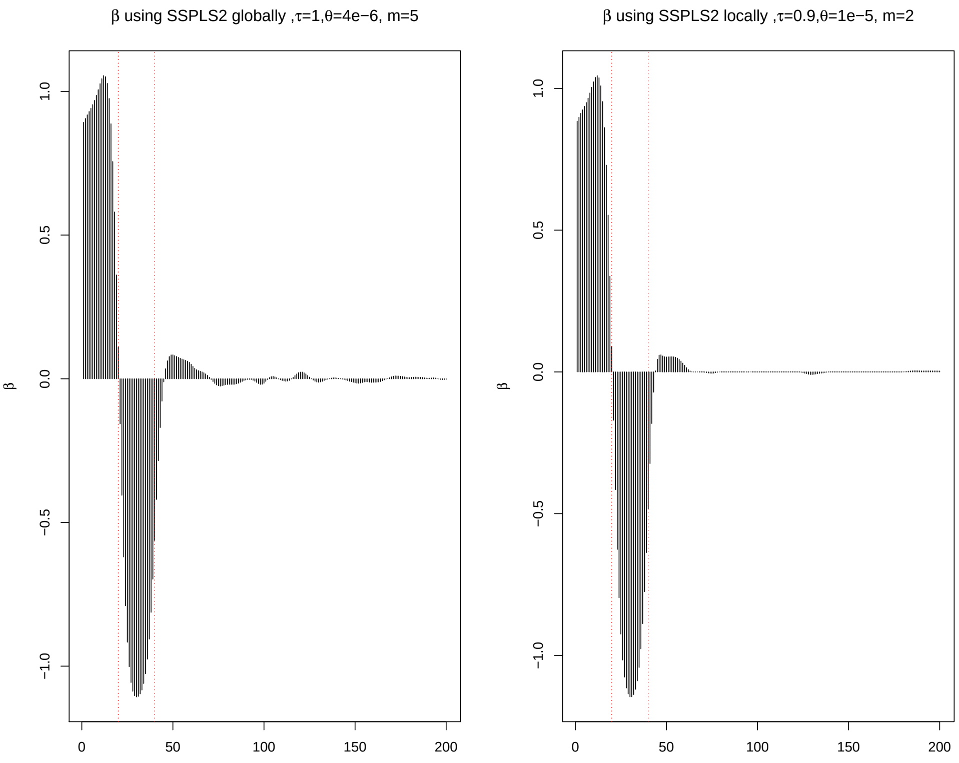


Figure 5.14: The estimation of $\hat{\beta}$ using second SSPLS globally (left panel) with ($\tau_{opt} = 1$, $\theta_{opt} = 4 \times 10^{-6}$, $m_{opt} = 5$), and locally (right panel) with ($\tau_{opt} = 0.9$, $\theta_{opt} = 1 \times 10^{-5}$, $m_{opt} = 2$).

It can be seen from the left panel of Figure 5.14, there is no sparsity effect because optimal τ (τ_{opt}) equals to one using the global model of the second SSPLS. In the right panel of Figure 5.14, the sparseness and smoothness can be seen because there is contribution from both Cauchy (smoothness) and Laplace (sparseness) distributions penalty.

Figure 5.15 shows the estimation of w_1 using second SSPLS when using the global model (left panel), and the estimation of w_1 using second SSPLS locally (right panel) for the first component.

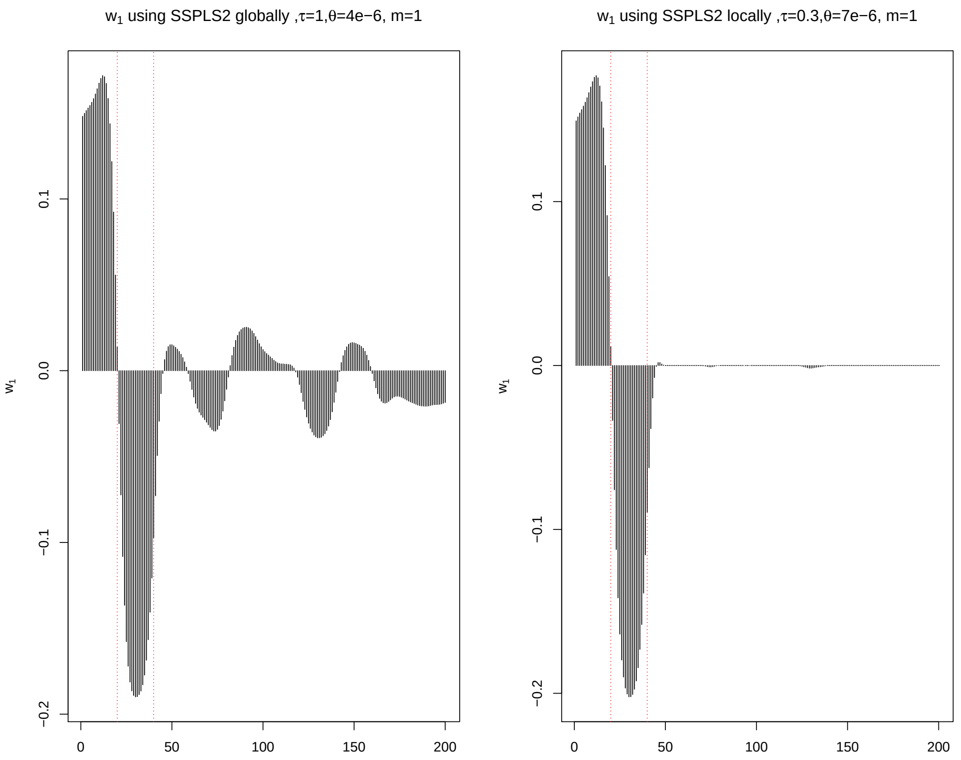


Figure 5.15: The estimation of w_1 using the second method of SSPLS globally with ($\tau_{opt} = 1$, $\theta_{opt} = 4 \times 10^{-6}$, $m = 1$), and locally with ($\tau_{opt} = 0.3$, $\theta_{opt} = 7 \times 10^{-6}$, $m = 1$).

We can see from the left panel of Figure 5.15, w_1 in the first component using the optimal tuning parameters of the global model. Because of $\tau_{opt} = 1$ meaning the whole model is only Cauchy with no Laplace contribution, it can be seen there is non of the estimation of w_1 are equal to zero. Using the tuning parameters of the local model with $\tau_{opt} = 0.3$, the model has 0.3 of Cauchy distribution to achieve smoothing and 0.7 for sparseness to achieve sparsity for w_1 in the first component as can be seen in the right panel of the same figure.

Figure 5.16 shows the estimation of w_2 using the second proposed SSPLS globally (left panel), and local model (right panel) for the second component.

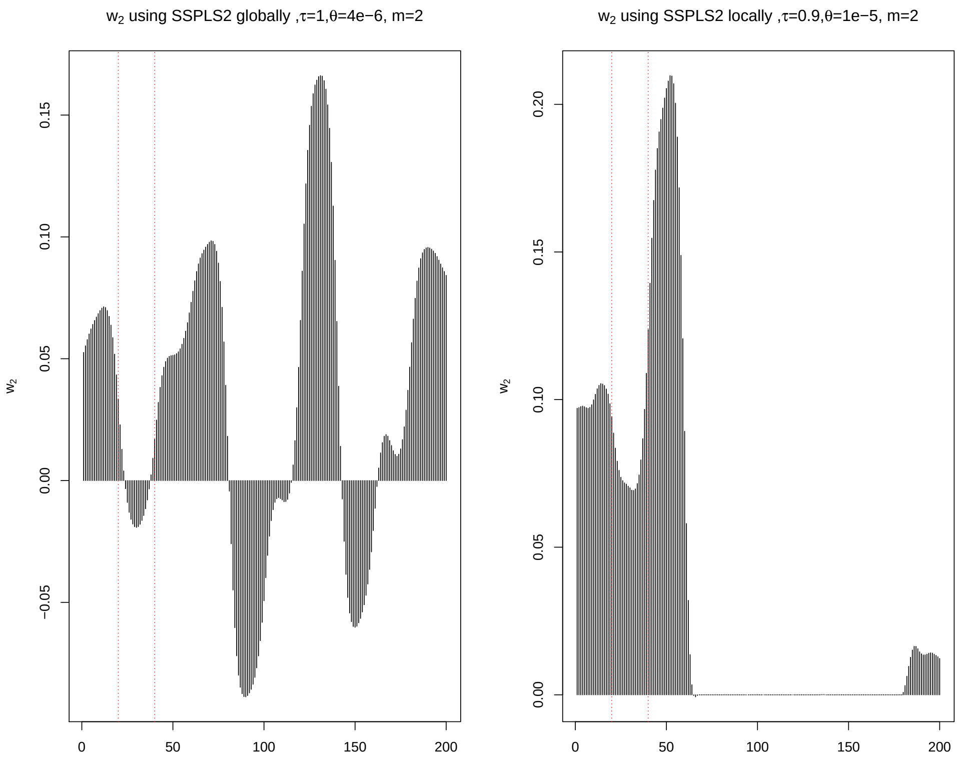


Figure 5.16: The estimation of w_2 for the second component using the second method of SSPLS globally with ($\tau_{opt} = 1$, $\theta_{opt} = 4 \times 10^{-6}$, $m = 2$), and second SSPLS locally with ($\tau_{opt} = 0.9$, $\theta_{opt} = 1 \times 10^{-5}$, $m = 2$).

Looking at the second component of w (w_2), the left panel used the optimal $\tau = 1$ and $\theta = 4 \times 10^{-6}$ that have been chosen for the first component. These tuning parameters uses the weight for Cauchy distribution equals to 1 with zero weight for Laplace distribution. Hence, no of the estimation of w_2 in the left panel are zeroes. In contrast, looking at the right panel where the optimal tuning parameters are used with considering every component is treated independently. Since $\tau_{opt} = 0.9$, this means that weight on the Laplace distribution is 0.1 and this still gives some sparsity on the estimation of w_2.

To summarise, we have investigated two methods of SSPLS and their results do not show critical difference on the simulated data in terms of the estimation of $\hat{\beta}$ and w_m for different components. Both methods of SSPLS have two ways of selecting their tuning parameters which are globally and locally. In the global model we lose the sparsity due to the choice of τ_{opt}, and for the local model we can see the effect of both penalties which are Laplace (sparseness) and Cauchy (smoothness). Furthermore, using the global model needs more number of components than using the local model for the first and second methods of SSPLS.

5.7.6 Simulation results: comparative study

For each data set, we evaluated the following methods:

- SPLS-HL: sparse PLS with HL penalty Lee *et al.* (2011)
- SPLS2-HL: sparse PLS2 with HL penalty Lee *et al.* (2011)
- Our proposed first method SSPLS: first sparse-smoothed PLS with mixture of Cauchy and Laplace distributions.
- Our proposed second method SSPLS: second sparse-smoothed PLS with mixture of Cauchy and Laplace distributions.

These methods have been evaluated with respect to the square root of the mean square prediction error ($\sqrt{\text{MSPE}}$) after the tuning parameters have been chosen. The $\sqrt{\text{MSPE}}$ is done using the formula in Equation (5.15), and it is calculated using an unseen simulated data sets with the same simulation settings in Section 5.7.1.

It is important to see how the estimation of $\hat{\beta}$ looks like for one simulation for the candidates methods. Figure 5.17 shows β from one simulated data set with the optimal tuning parameters λ and m for SPLS-HL, and $H = (\tau, \theta, m)$ for the first SSPLS using the local model.

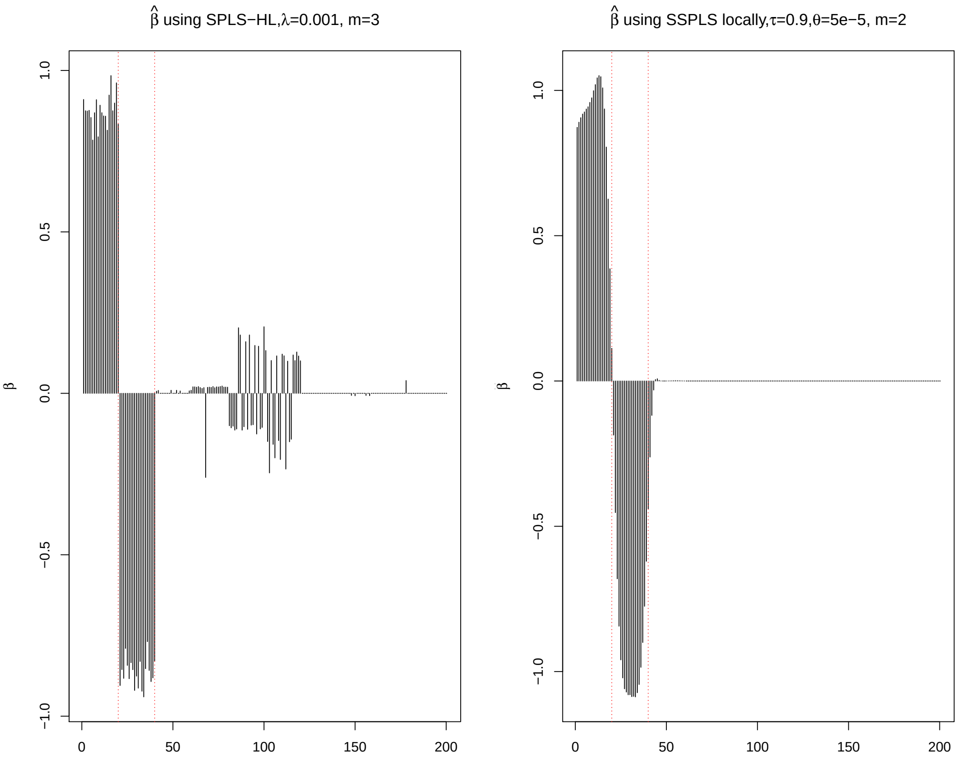


Figure 5.17: The estimation of $\hat{\beta}$ using SPLS-HL method (left panel) with optimal tuning parameters ($\lambda = 0.001$, $m_{opt} = 3$). The estimation of $\hat{\beta}$ using the first SSPLS with local model (right panel) with optimal parameters ($\tau_{opt} = 0.9$, $\theta_{opt} = 5 \times 10^{-5}$, $m_{opt} = 2$).

It is clear from looking to Figure 5.17 that the SPLS-HL (left panel) does not have smoothing and there are sudden spikes without considering the correlation between the variables. There is no sparsity also using the optimal (m and λ). On the other hand, our proposed SSPLS using the local model (right panel) has smoothness and sparseness to deal with the dependencies between variables and the irrelevant variables respectively. The red lines are for the first 40 covariates that are associated with the response. In our simulations settings, the fist 20 are positively associated and covariates from 21-40 are negatively associated with the response variable. Using the SPLS1-HL, we can see the first 40 covariates reveal the signals in the estimation of $\hat{\beta}$. However, the smoothness is not found and there are are other covariates reveal signals which should be equal to zero. In contrast, the estimation of $\hat{\beta}$ using the first SSPLS reveal signals as we expected and they are smoothed. The rest of covariates are set to be equal to zero.

Figure 5.18 shows $\hat{\beta}$ from one simulated data set with the optimal tuning parameters λ and m for SPLS-HL, and $H = (\tau, \theta, m)$ for the first SSPLS using the local model.

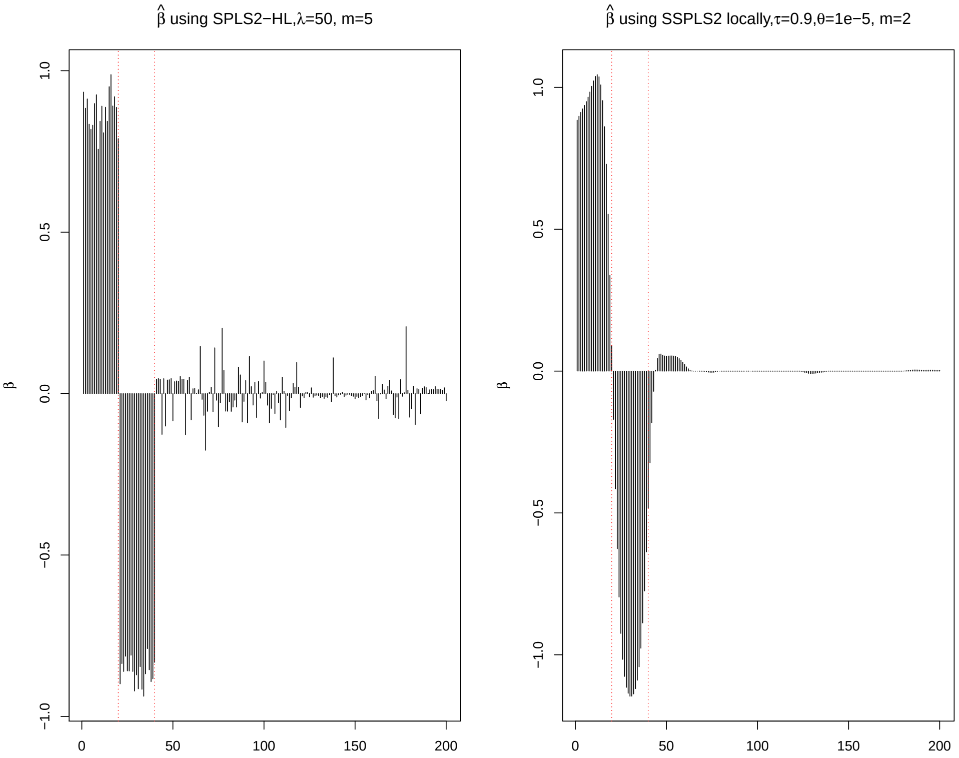


Figure 5.18: The estimation of $\hat{\beta}$ using SPLS2-HL method (left panel) with optimal tuning parameters ($\lambda = 50$, $m_{opt} = 5$). The estimation of $\hat{\beta}$ based on the second method of SSPLS with local model (right panel) with tuning parameters ($\tau_{opt} = 0.9$, $\theta_{opt} = 10^{-5}$, $m_{opt} = 2$).

As can be seen from Figure 5.18 that the SPLS2-HL (left panel) does not have smoothness which does not consider the dependencies between the variables. There is sparsity based on the optimal m and λ. Our proposed second method of SSPLS based on the optimal tuning parameters using the local model (right panel) shows the estimation of $\hat{\beta}$ has obvious smoothness and sparsity. The red lines to show that in our setting for the simulation, we set the first 20 covariates to be positively correlated to the response variable and the second 20 covariates are negatively associated. The estimation of $\hat{\beta}$ using SPLS2-HL reveal the signals and more other covariates with almost no sparsity. On the other hand, the estimation of $\hat{\beta}$ using the second SSPLS reveal the first 40 covariates and very little number of the rest covariates with a very clear smoothness.

Figure 5.19 shows a box plot of the ($\sqrt{\text{MSPE}}$) using five-fold cross-validation for six methods which are SPLS-HL, SPLS2-HL (from Lee *et al.* (2011)), first SSPLS global, second SSPLS global, first SSPLS local and second SSPLS local.

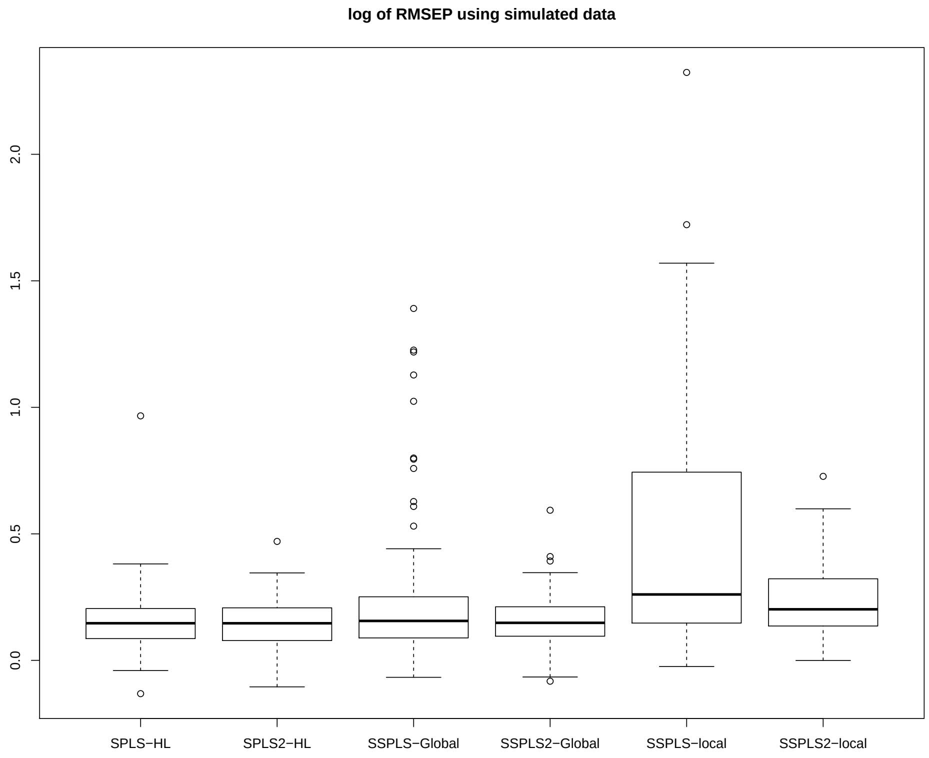


Figure 5.19: Box plot of the log of $\sqrt{\text{MSPE}}$ using five-fold cross-validation for six methods which are SPLS-HL, SPLS2-HL (from Lee *et al.* (2011)), first SSPLS using global model, second SSPLS using global model, first SSPLS using local model and second SSPLS using local model.

Although that the difference of $\sqrt{\text{MSPE}}$ between our proposed first and second methods of SSPLS using the optimal tuning parameters based on the global model and the local model is quite small, the achievements of the sparseness may not be likely happened since the optimal τ might be one. Having said that, the user can change the weight of each distribution as needed for more sparseness or smoothness. It can be said that the difference of $\sqrt{\text{MSPE}}$ between (SPLS-HL, SPLS2-HL) and (first SSPLS, second SSPLS) might be quite small as well with an outperformance of SPLS-HL and SPLS2-HL, but these two methods do not consider the dependencies between variables which first SSPLS and second SSPLS do.

5.8 NIR data results

5.8.1 SSPLS with first NIPALS algorithm

The optimal θ and τ here for the local model are ($\tau_{opt} = 0.1$ and $\theta_{opt} = 26.1$) in the first component $m = 1$. Using the global model, the optimal values are ($\tau_{opt} = 1$ and $\theta_{opt} = 26$).

Figure 5.20 shows the plot for the MSPE over various number of components using five-fold MSECV using NIR data with the first SSPLS for global and local models.

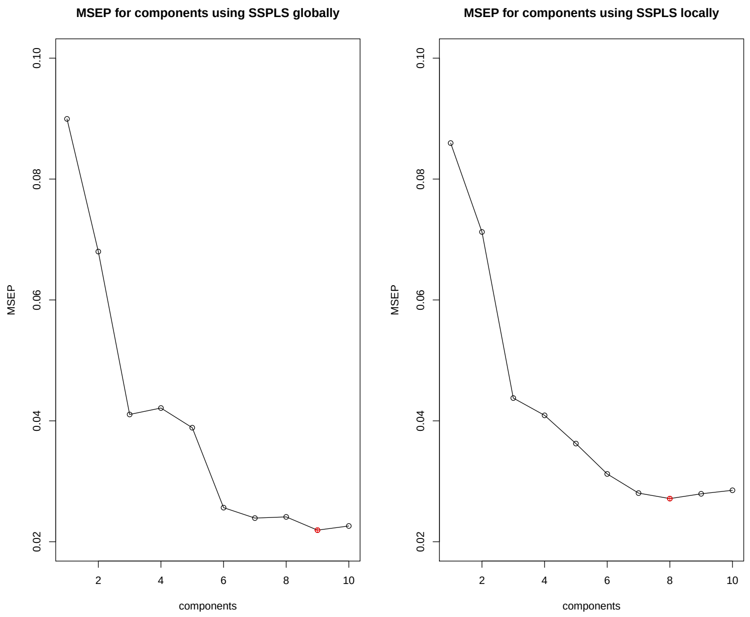


Figure 5.20: MSPE using five-fold CV with NIR data using the first SSPLS for global (left panel) and local (right panel) models.

It can be seen from Figure 5.20 that the optimal number of components using the global model for the first method of SSPLS (left panel) is 9 components. Using the local model for the first method of SSPLS for the θ and τ for each component then compute the optimal number of components based on those chosen θ_{opt} and τ_{opt} (right panel), the optimal number of components is 8.

Figure 5.21 shows the estimation of $\hat{\beta}$ using the first SSPLS based on the global model (left panel), and locally (right panel).

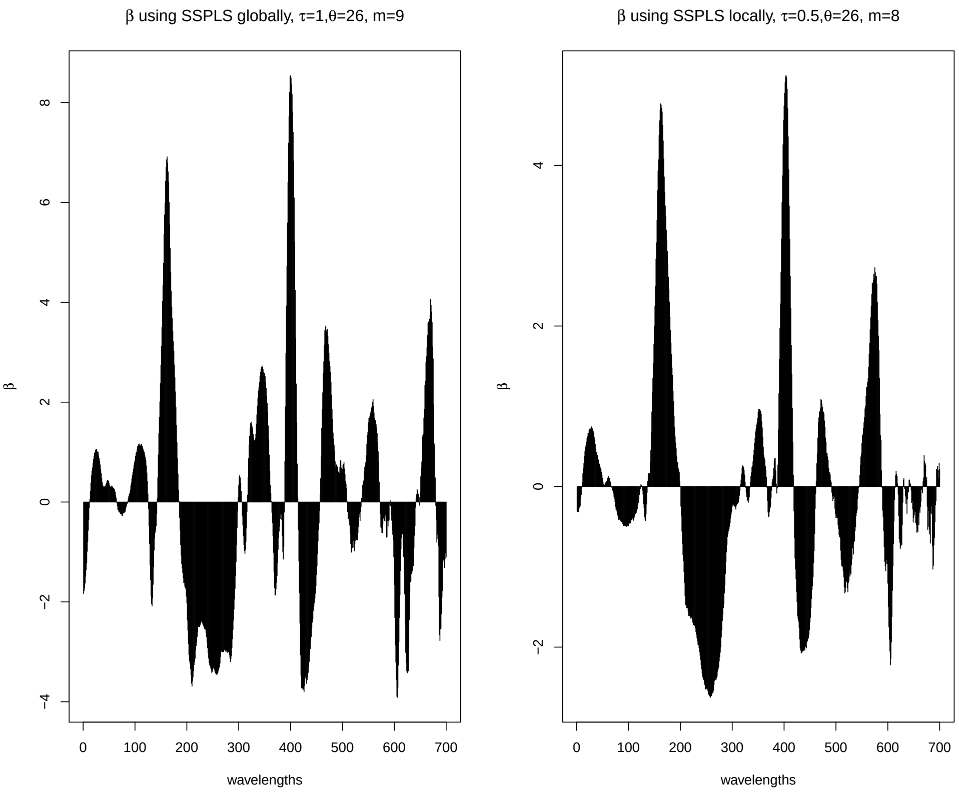


Figure 5.21: The estimation of $\hat{\beta}$ using the first SSPLS with global model (left panel) with ($\tau_{opt} = 1$, $\theta_{opt} = 26$, $m_{opt} = 9$), and the first SSPLS with local model (right panel) with ($\tau_{opt} = 0.5$, $\theta_{opt} = 26$, $m_{opt} = 8$).

Looking at Figure 5.21, the estimation of $\hat{\beta}$ using the optimal tuning parameters using the global model for the first method of SSPLS (left panel) does not show any sparsity. This is because the number of components is large ($m_{opt} = 9$) and the optimal $\tau = 1$ which means the weight for Laplace distribution is zero. In other words, our first proposed model of SSPLS is reduced to the smoothed PLS model. In the right panel of Figure 5.21, β is plotted using the optimal tuning parameters based on the local model for the first method of SSPLS for each component for θ and τ. Although $\tau_{opt} = 0.5$, there is also no sparsity using the local model for the estimation of $\hat{\beta}$. The reason is because the value of θ_{opt} is too large for the model to have sparseness. There might be some sparsity on the estimation of $\hat{\beta}$ with small θ_{opt}. However, using the small values of θ from a range of different θ cannot be used because the smallest value of θ that

NIR data may have is $\theta = 26$ for the first component. If θ gets smaller value less than 26, the estimation of $\hat{\beta}$ (all covariates) in the first component will be zero (over sparseness). The other components could have had some sparseness if the θ_{opt} was less than 26 which means there is no residuals from the first component to be used to calculate the second component and the rest of the components. Positive estimates of $\hat{\beta}$ indicate that the relevant wavelengths are significant with increases in the moisture. On the other hand, the negative estimates of $\hat{\beta}$ indicate that the relevant wavelengths covariates are associated with decreases in the moisture.

Figure 5.22 shows the estimation of w_1 using the first SSPLS based on the global model (left panel), and local model (right panel).

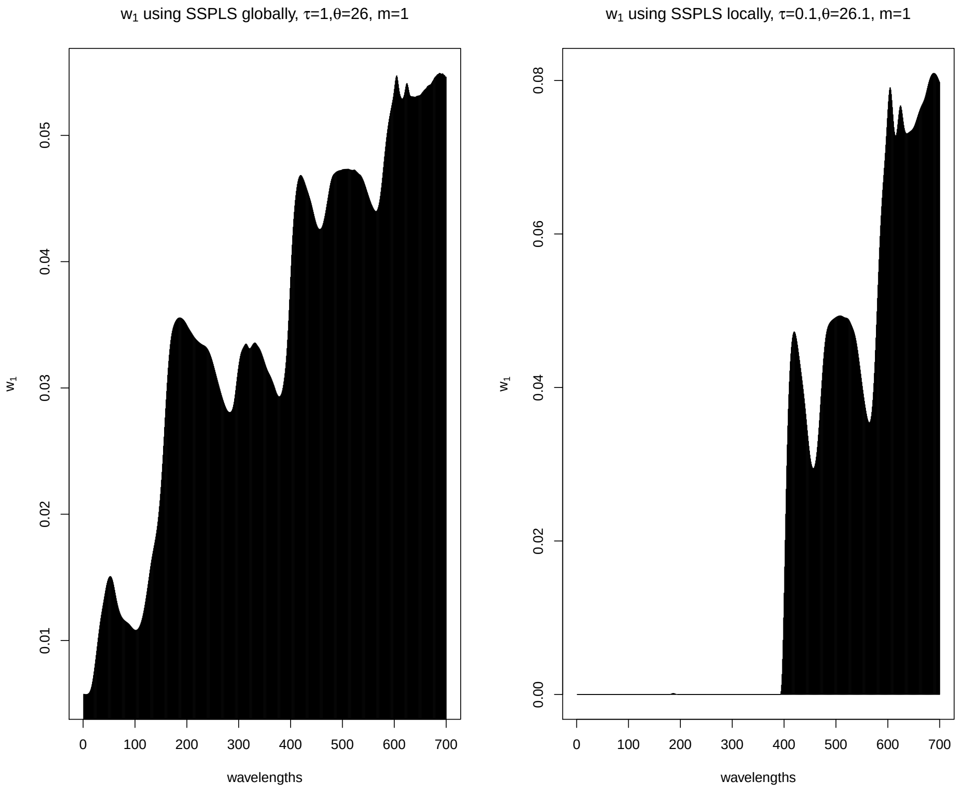


Figure 5.22: The estimation of w_1 in the first component using the first SSPLS with global model (left panel) with ($\tau_{opt} = 1$, $\theta_{opt} = 26$, $m = 1$), and the first SSPLS with local model with ($\tau_{opt} = 0.1$, $\theta_{opt} = 26.1$, $m = 1$).

Because $\tau_{opt} = 1$, there is no sparsity in the first component. Looking at the right panel of Figure 5.22 using the local model for the first SSPLS, the estimation of w_1 has some variables are set to be equal to zero since $\tau_{opt} = 0.5$. Having $\tau_{opt} = 0.5$ balances the weight on both distributions (Cauchy for smoothness and Laplace for sparseness) which results in smoothness and sparseness.

In deed, the estimation of $\hat{\beta}$ using the first method of SSPLS with global model and local model does not have sparseness due to large number of components as an optimal ($m_{opt} = 9$ or 8) for the NIR data. Because of the optimal number of components is large, we lost the effect of Laplace distribution. Although the local model might have some sparseness on some components because the selected θ_{opt} and τ_{opt} differ in each component unlike the global model we do not select θ_{opt} and τ_{opt} over each component. Both models with the first method of SSPLS do not guarantee sparseness for the NIR data.

5.8.2 SSPLS with the second NIPALS algorithm

Here, the optimal θ and τ here for the local model are ($\tau_{opt} = 0.5$ and $\theta_{opt} = 6 \times 10^{-7}$) in the first component $m = 1$. Using the global model, the optimal values are ($\tau_{opt} = 0.7$ and $\theta_{opt} = 6 \times 10^{-7}$).

Figure 5.23 shows the plot for the MSPE over various number of components using 5 fold MSECV using NIR data with the second SSPLS for global and local models.

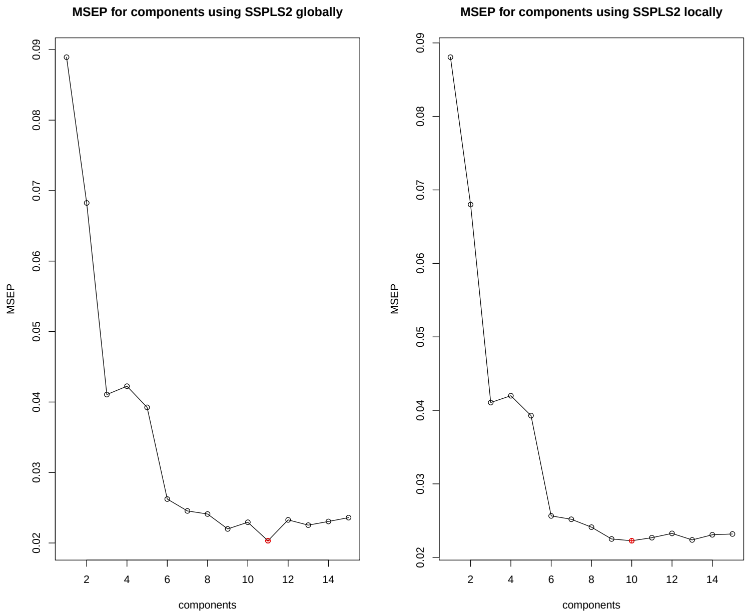


Figure 5.23: MSPE using five-fold CV with NIR data using the second SSPLS for global (left panel) with 11 components and local (right panel) models with 10 components.

It can be seen from Figure 5.23 the optimal number of components using the global model for the second SSPLS is 11 components based on five-fold cross-validation with minimum of MSECV. The optimal number of components using local model for the second SSPLS can be seen in the right panel of Figure 5.23 which is 10 components based on five-fold cross-validation with the minimum of MSECV.

Figure 5.24 shows the estimation of $\hat{\beta}$ using the second SSPLS based on the global model (left panel), and locally (right panel).

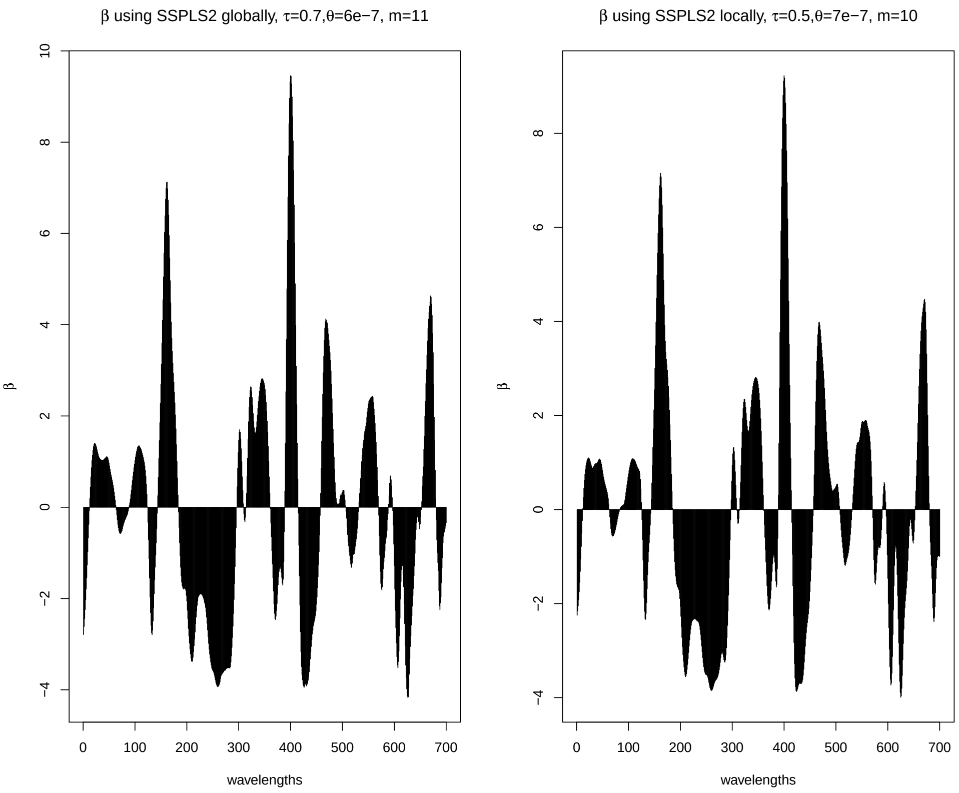


Figure 5.24: The estimation of $\hat{\beta}$ using the second method of SSPLS with the global model ($\tau_{opt} = 0.7$, $\theta_{opt} = 6 \times 10^{-7}$, $m_{opt} = 11$), and local model with ($\tau_{opt} = 0.5$, $\theta_{opt} = 7 \times 10^{-7}$, $m_{opt} = 10$).

Estimation of $\hat{\beta}$ can be seen in Figure 5.24 using the second SSPLS based on global model (left panel) with optimal tuning parameters ($\tau_{opt} = 0.7$, $\theta_{opt} = 6 \times 10^{-7}$, $m_{opt} = 11$). Looking at the estimation of $\hat{\beta}$, there is no sparsity though $\tau_{opt} = 0.7$ because the optimal number of components is 11 which is large. As the number of component is increased, sparseness is less common. Although if some covariates of w_m in each component are set to be equal zero, w_m and β become dense. Hence, with small number of components, the chance is more to have sparseness of β Lee *et al.* (2011). Similarly, looking at the right panel of Figure 5.24, the estimation of $\hat{\beta}$ using the second SSPLS with local model does not have sparsity though $\tau_{opt} = 0.5$, since ($m_{opt} = 10$) is large. We interpret the estimates of $\hat{\beta}$ as the positive estimates are associated with increase in the moisture. In contrast, the negative estimates are associated with decrease in the moisture.

Figure 5.25 shows the estimation of w_1 using the second SSPLS based on the global model (left panel), and locally (right panel).

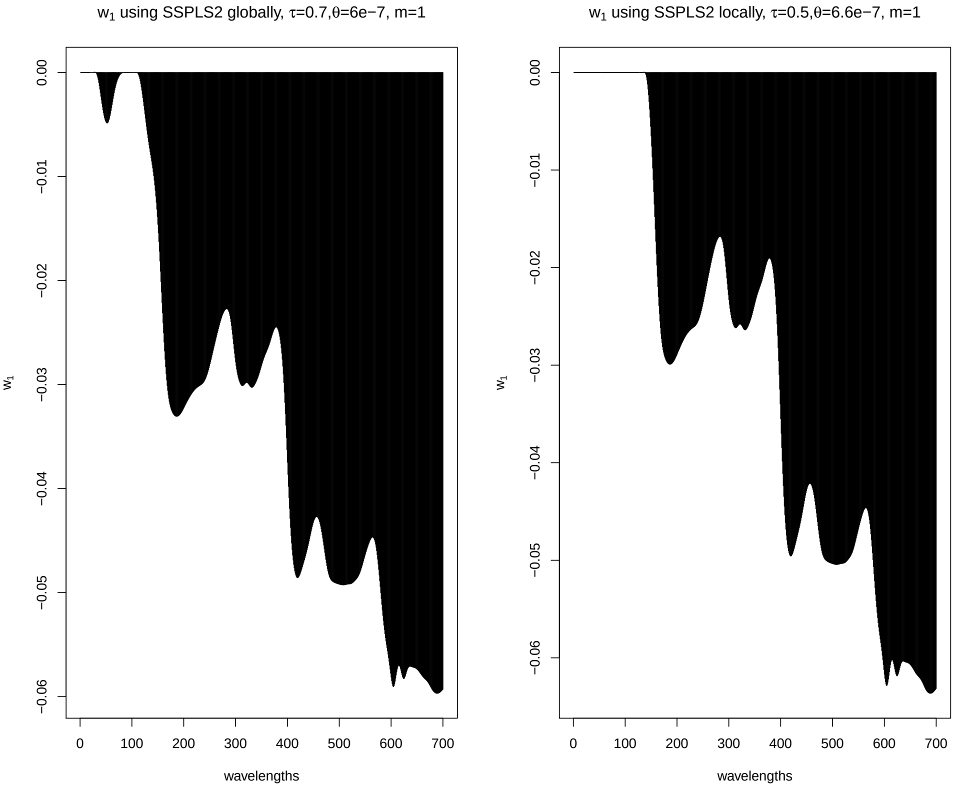


Figure 5.25: The estimation of w_1 using the second SSPLS globally with ($\tau_{opt} = 0.7$, $\theta_{opt} = 6 \times 10^{-7}$, $m = 1$), and locally with ($\tau_{opt} = 0.5$, $\theta_{opt} = 6 \times 10^{-7}$, $m = 1$).

It can be seen from Figure 5.25 that there is some sparseness on the estimation of w_1 using the second SSPLS based on the global (left panel) and local models (right panel). We can see that more elements are set to be equal to zero using the local model because the optimal $\tau_{opt} = 0.5$ whereas for the global model $\tau_{opt} = 0.7$. This means that the weight for Laplace distribution is 0.3 for the global model while it is 0.5 for the local model.

In summary, both methods of SSPLS with local and global models are similar with a little difference in the optimal number of components where the second methods need more number of components than the first method of SSPLS here. However, in the second method, we can see some sparseness in each component w_m when using the local model.

5.8.3 Comparisons

Figure 5.26 shows a box plot of the root mean squares prediction error ($\sqrt{\text{MSPE}}$) for six methods which are SPLS-HL, SPLS2-HL (from Lee *et al.* (2011)), first SSPLS global, second SSPLS global, first SSPLS local and second SSPLS local.

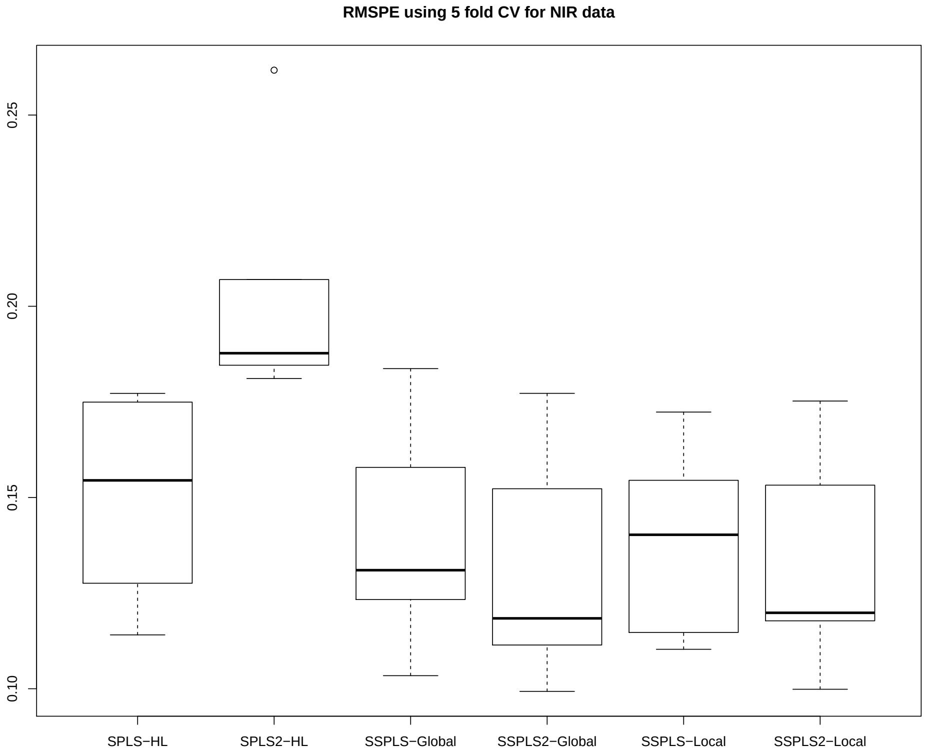


Figure 5.26: Box plot of the root mean squares prediction error $\sqrt{\text{MSPE}}$ for six methods which are SPLS-HL, SPLS2-HL (from Lee *et al.* (2011)), first SSPLS global, second SSPLS global, first SSPLS local and second SSPLS local.

Looking at Figure 5.26, we can see that the second SSPLS using the global model has the lowest median equals to 0.1332 of the five-fold cross-validation. Because the response variable of the NIR data is normal, we use RMSPE in the calculation. The second SSPLS using the local model has the second lowest median equals to 0.1334 of five-fold cross-validation of RMSPE. SSPLS using the global is the third lowest median equals to 0.1392. Then, the first SSPLS using the local model is the fourth lowest among the six methods. SPLS-HL and SPLS2-HL proposed by Lee *et al.* (2011) have median of RMSPE equal to 0.1577 and 0.1573 respectively. Although the first and second methods of SSPLS using the local model have RMSPE larger than using the global model, there is no sparseness in the first component using the global model for both methods of SSPLS as can be seen in Figures 5.22 and 5.25.

5.9 CNA data results

5.9.1 SSPLS with first NIPALS algorithm

Here, we use the same criteria for choosing θ and τ as in Section 5.4.2 with the first SSPLS for local and global model.

Applying sparse PLS solutions on high dimensional data ($n \ll p$) where the response variable is binary, have been widely applied as in Chung & Keles (2010). Figure 5.27 shows the plot for the (MERCV) using five-fold cross-validation over various number of components for CNA data with the first SSPLS for global and local models.

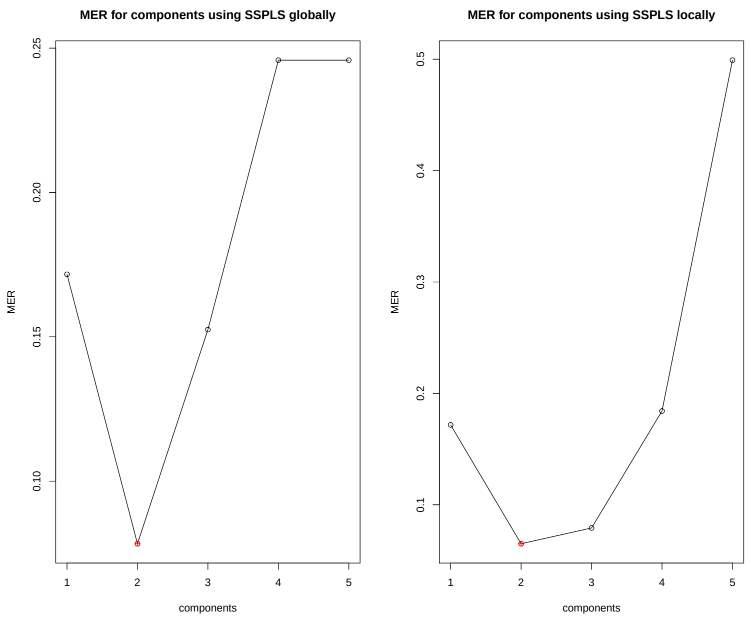


Figure 5.27: MERCV over various number of components using five-fold cross-validation for CNA data with the first SSPLS for global and local models.

It can be seen that from Figure 5.27 left panel using the global model where all components have the same optimal θ and τ with optimal tuning parameters ($m_{opt} = 2$, $\theta_{opt} = 4 \times 10^{-4}$ and $\tau_{opt} = 0.9$). When the local model is used as can be seen in the right panel with $m_{opt} = 2$, every component will have different optimal values of θ and τ. These values using the local model with the first SSPLS for the first component are $\theta_{opt} = 0.01$ and $\tau_{opt} = 0.4$. For the second component, the optimal values are $\theta_{opt} = 0.007$ and $\tau_{opt} = 0.7$.

Figure 5.28 shows the estimation of $\hat{\beta}$ using the first SSPLS based on 2 components.

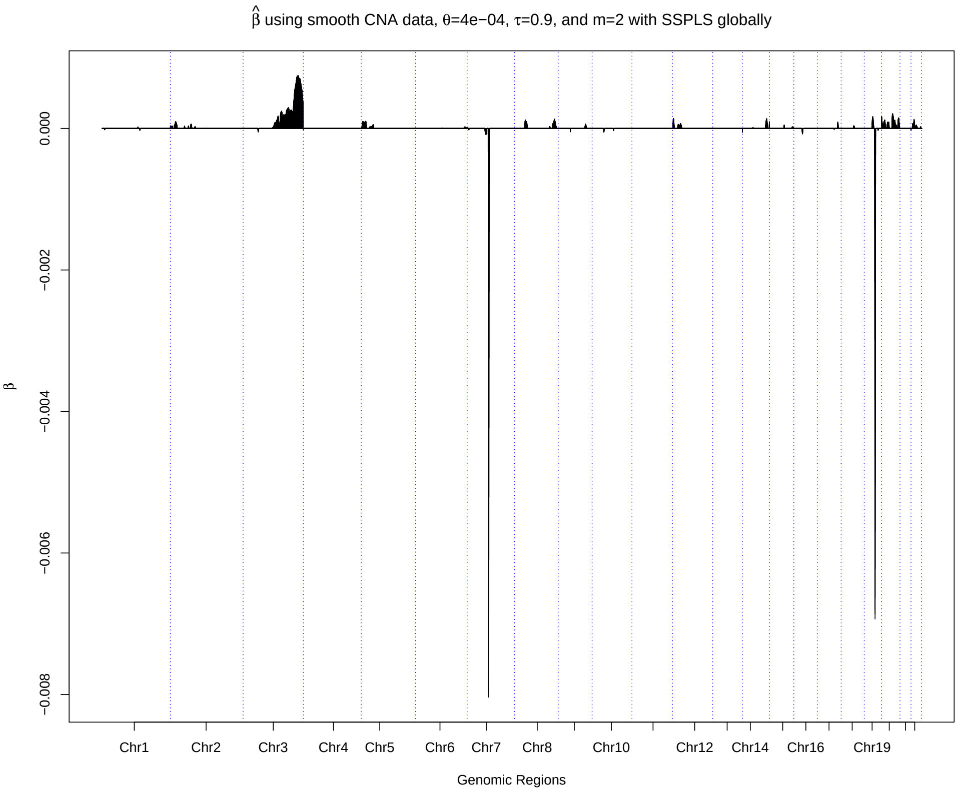
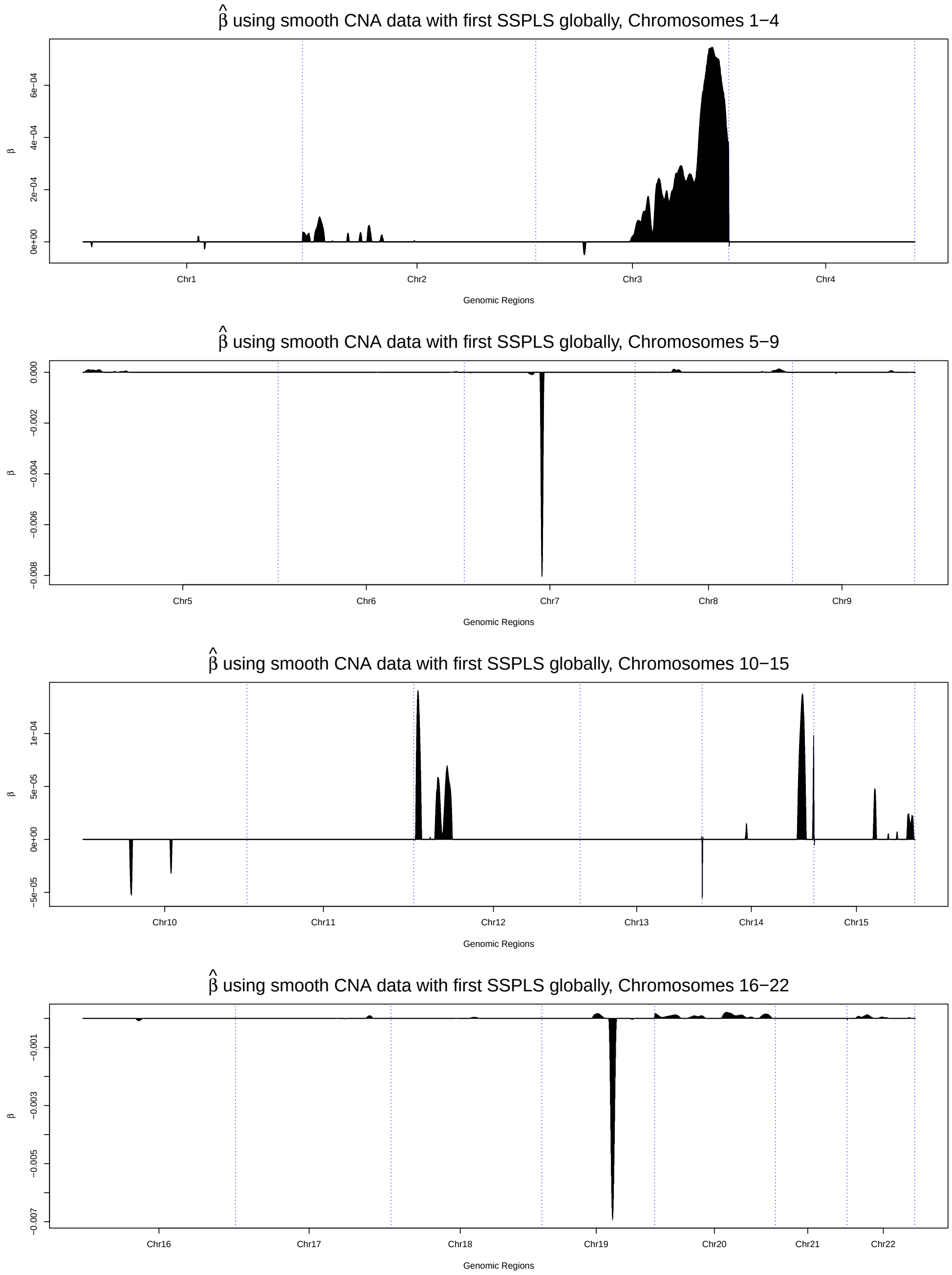


Figure 5.28: The estimation of $\hat{\beta}$ using the first SSPLS with the global model for CNA profiles. Those genomic windows with missing values (for example in the centromere regions) were not plotted since they were not used in the analysis. A more detailed view of the random effects estimates in each chromosome is presented in Figure 5.29.

Figure 5.29 shows a more detailed view of estimation of $\hat{\beta}$.



$\hat{\beta}$ using smooth CNA data with first SSPLS globally, Chromosomes 1−4

$\hat{\beta}$ using smooth CNA data with first SSPLS globally, Chromosomes 5−9

$\hat{\beta}$ using smooth CNA data with first SSPLS globally, Chromosomes 10−15

$\hat{\beta}$ using smooth CNA data with first SSPLS globally, Chromosomes 16−22

Figure 5.29: The estimation of $\hat{\beta}$ using the first SSPLS with the global model for CNA profiles. Those genomic windows with missing values (for example in the centromere regions) were not plotted since they were not used in the analysis.

It can be seen from Figure 5.29 in chromosomes 2, 3, 12, 14 and 15 have positive signals in the genomic regions. Because the response variable is binary and centred, so the large positive values contribute to squamous carcinoma (SCC), and the large absolute values of the negative estimates contribute more to adenocarcinoma (ADC). The negative estimates appeared in chromosomes 7, 10 and 19.

Figure 5.30 shows the X-weights in the first component w_1 using the first SSPLS with global model.

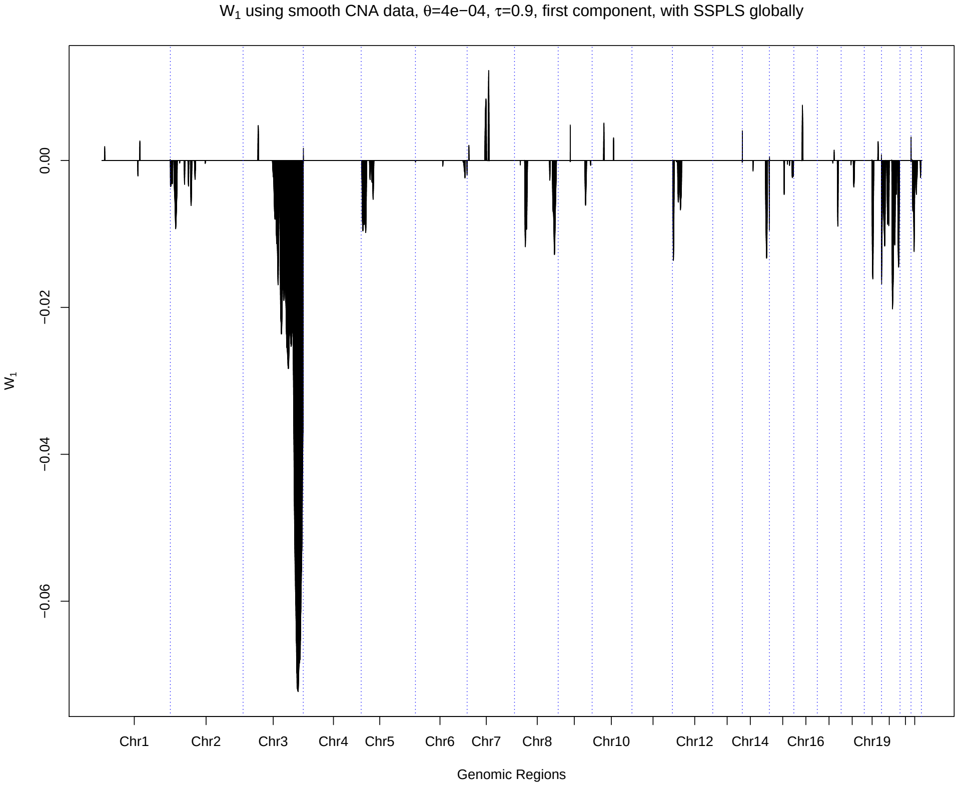


Figure 5.30: w_1 with mixture of smoothed Cauchy and Laplace distributions penalty using first SSPLS globally ($\tau_{opt} = 0.9$, $\theta_{opt} = 4 \times 10^{-4}$, $m = 1$).

It can be seen from Figure 5.30 there are some chromosomes (e.g. 4, 11 and 13) that do not have significant signals for the genomic regions. This means that these chromosomes have an association with cancer type either SCC or ADC.

Figure 5.31 shows the X-weights in the second component w_2 using the first SSPLS with global model.

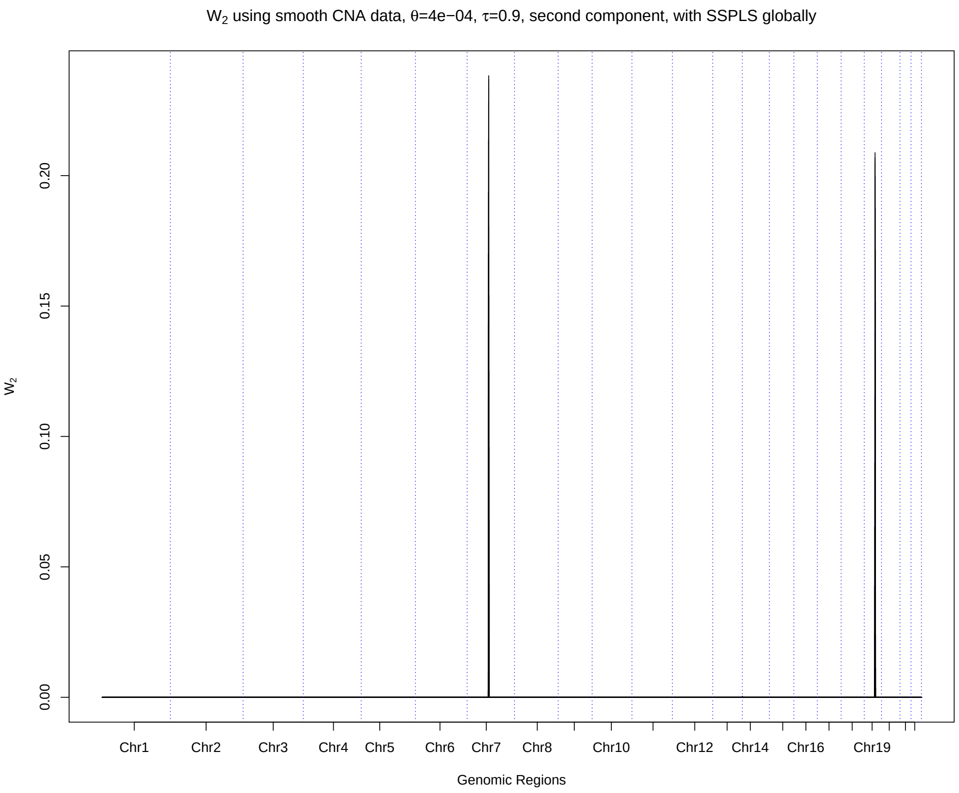


Figure 5.31: w_2 with mixture of smoothed Cauchy and Laplace distributions penalty using SSPLS globally ($\tau_{opt} = 0.9$, $\theta_{opt} = 4 \times 10^{-4}$, $m = 2$).

Looking at the estimation of w_2 in Figure 5.31, we can see that some genomic regions in chromosomes 7 and 19 have signals. These signals are associated with the cancer type.

Figure 5.32 shows the estimation of $\hat{\beta}$ using the first SSPLS based on 2 components.

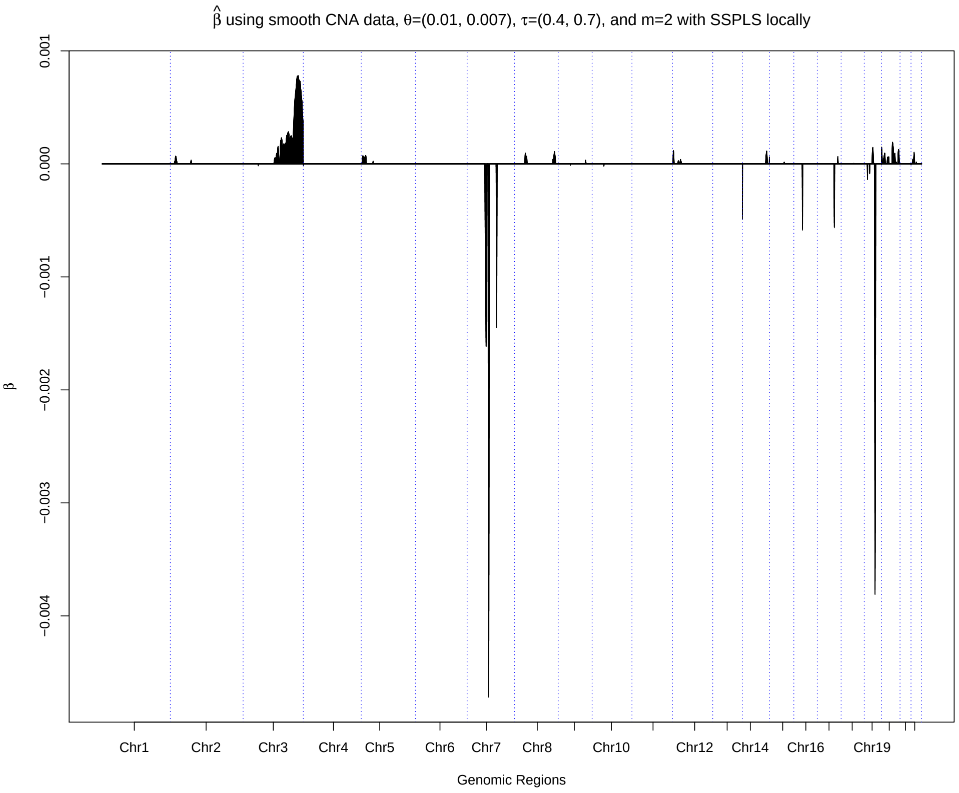


Figure 5.32: The estimation of $\hat{\beta}$ using the first SSPLS with the local model for CNA profiles. A more detailed view of the random effects estimates in each chromosome is presented in Figure 5.33.

Figure 5.33 shows a more detailed view of estimation of $\hat{\beta}$ using the first SSPLS based on 2 components.

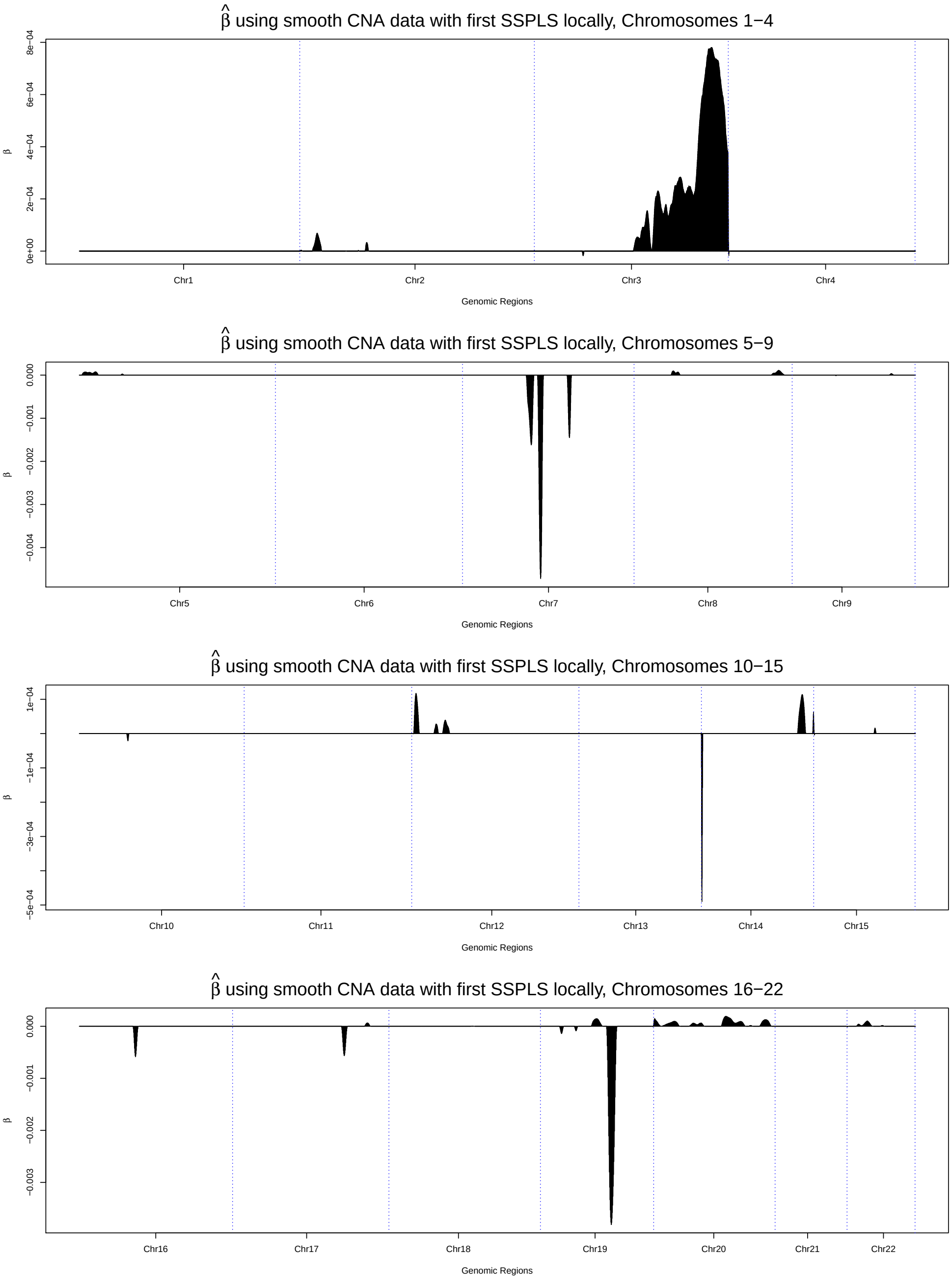


Figure 5.33: The estimation of $\hat{\beta}$ using the first SSPLS with the local model for CNA profiles. Those genomic windows with missing values (for example in the centromere regions) were not plotted since they were not used in the analysis.

Looking at Figure 5.33, we can see the genomic regions in chromosomes 3, 12 and

14 which have large positive estimates of $\hat{\beta}$ contribute more to SCC class. Genomic regions in chromosomes 7, 14 and 19, which have large absolute value of the negative estimates of $\hat{\beta}$, contribute to ADC class.

Figure 5.34 shows w_1 with mixture of smoothed Cauchy and Laplace distributions using the first SSPLS locally.

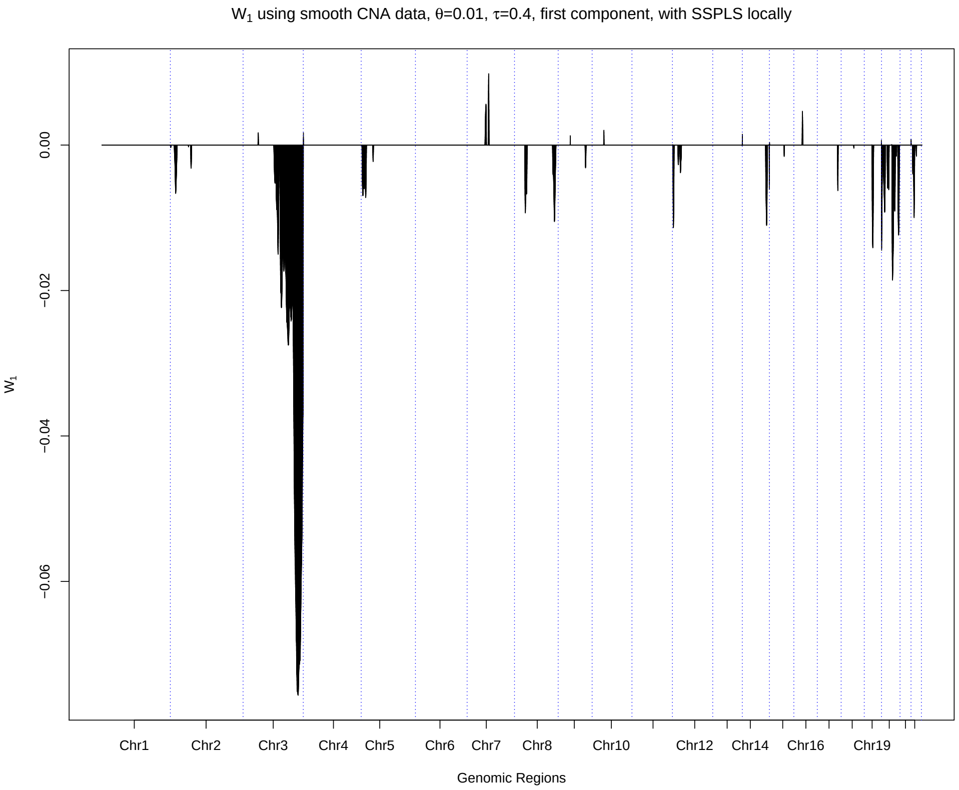


Figure 5.34: w_1 with mixture of smoothed Cauchy and Laplace distributions penalty using the first SSPLS globally ($\tau_{opt} = 0.4$, $\theta_{opt} = 0.01$, $m = 1$).

It can be seen from Figure 5.34 that the genomic regions in chromosomes (3,19 and 20) have signals which indicate that they have an association with the cancer type.

Figure 5.35 shows w_2 with mixture of smoothed Cauchy and Laplace distributions using the first SSPLS locally.

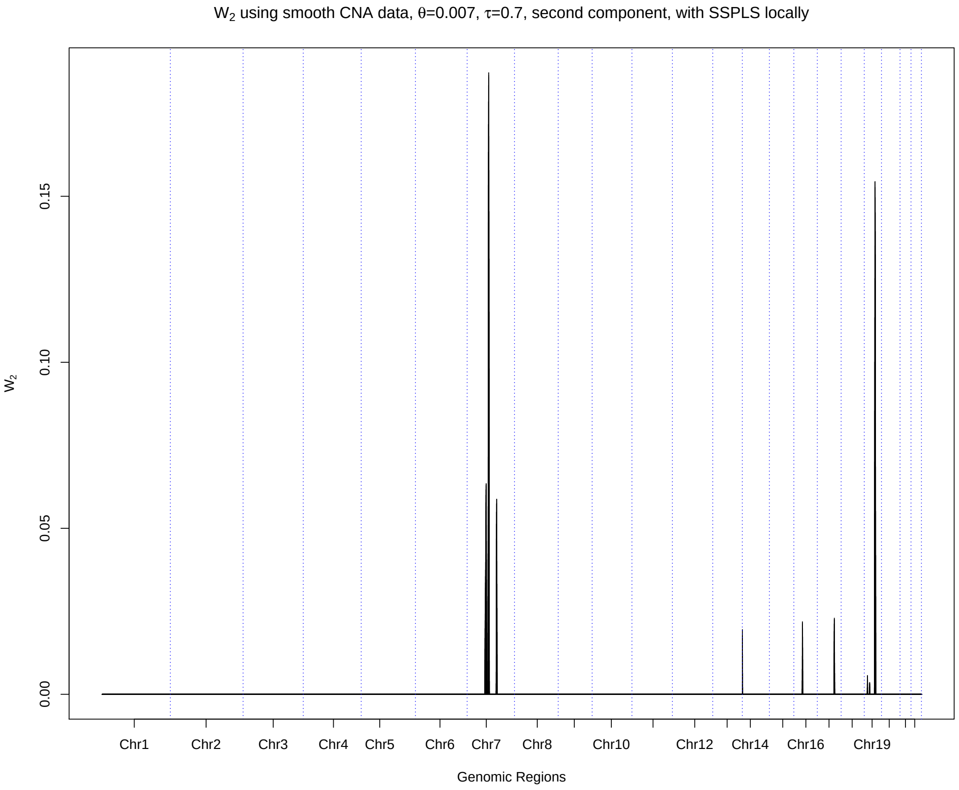


Figure 5.35: w_2 with mixture of smoothed Cauchy and Laplace distributions penalty using the first SSPLS locally ($\tau_{opt} = 0.7$, $\theta_{opt} = 0.007$, $m = 2$).

Positive estimates of w_2 of the genomic regions in the chromosomes (7, 14, 16, 17, 19) which can be seen in Figure 5.35. This means that these regions have an association with the cancer type.

To compare the estimation of $\hat{\beta}$ using the global and local models, we can see that both models are similar in the estimation of $\hat{\beta}$ for some chromosomes. However, in some chromosomes (16 and 17), they are associated with the ADC using the local model, but they are not with the global model. Moreover, from time point of view the local model is faster than global model to select the optimal tuning parameters via five-fold cross-validation.

5.9.2 SSPLS with second NIPALS algorithm

Here, we use the same criteria for choosing the optimal θ and τ in Section 5.4.2 for the global and local model.

Figure 5.36 shows the plot for the misclassification error rate (MERCV) using five-fold cross-validation over various number of components for CNA data with the second SSPLS for global and local models.

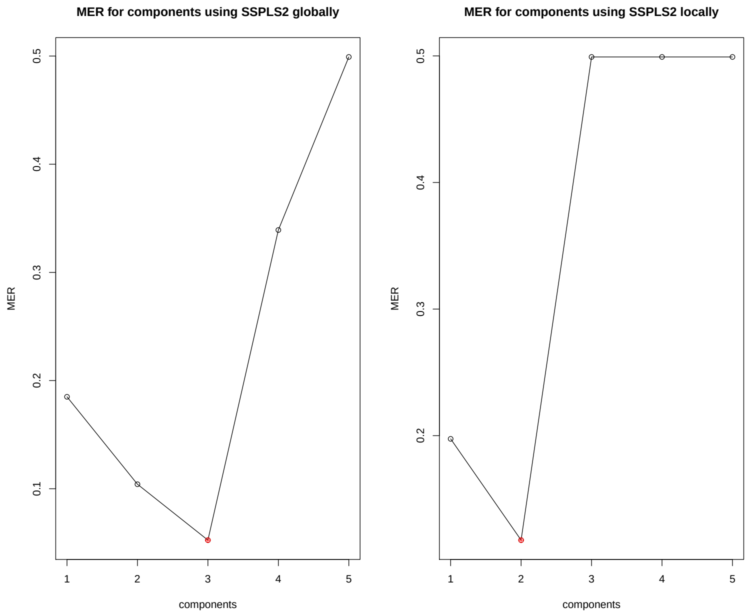


Figure 5.36: MERCV over various number of components using five-fold cross-validation for CNA data with the second SSPLS for global (left panel) and local (right panel) models.

It can be seen that from Figure 5.36 left panel using the global model where all components have the same optimal θ and τ with optimal tuning parameters ($m_{opt} = 3$, $\theta_{opt} = 6 \times 10^{-5}$ and $\tau_{opt} = 0.8$). When the local model is used as can be seen in the right panel, every component will have different optimal values of θ and τ. Having identified the optimal $m_{opt} = 2$ using the local model of the second SSPLS, the optimal $\theta_{opt} = 9 \times 10^{-5}$ and $\tau_{opt} = 0.4$ in the first component. In the second component, the optimal $\theta_{opt} = 3 \times 10^{-4}$ and $\tau_{opt} = 0.3$ using the same method and model.

Figure 5.37 shows the estimation of $\hat{\beta}$ using the second SSPLS with global model optimal values for CNA data based on 3 components.

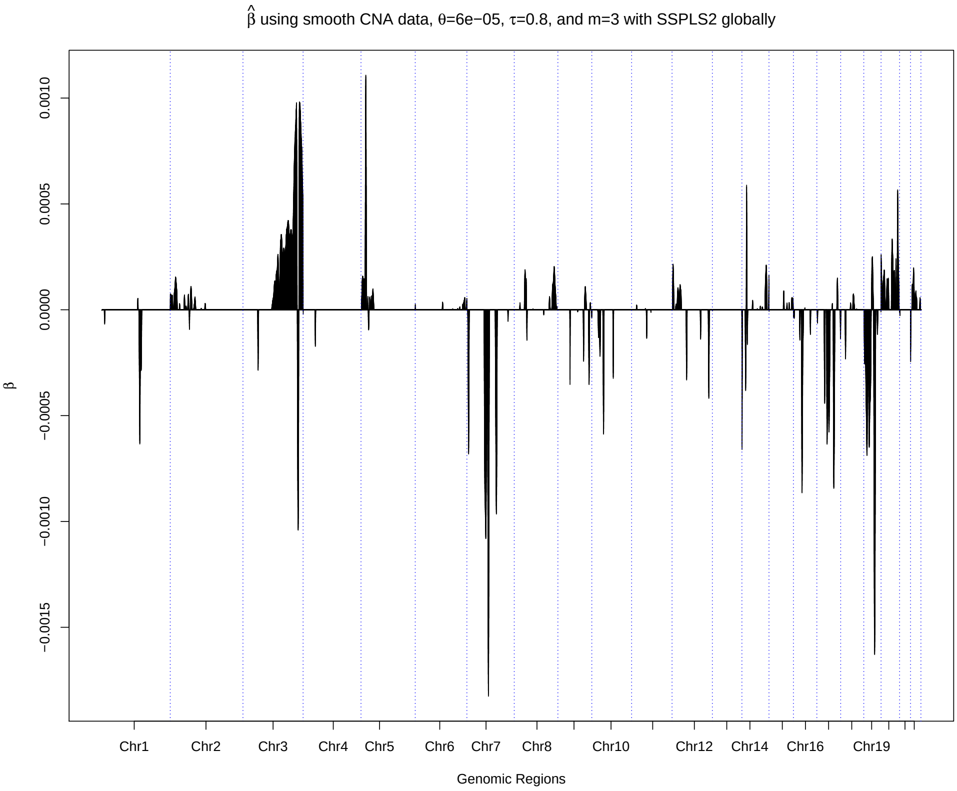


Figure 5.37: The estimation of $\hat{\beta}$ using the second SSPLS with the global model for CNA profiles. A more detailed view of the random effects estimates in each chromosome is presented in Figure 5.38.

Figure 5.38 shows a more detailed view of estimation of $\hat{\beta}$ using SSPLS method with optimal values of the tuning parameters using the global model for CNA data based on 2 components.

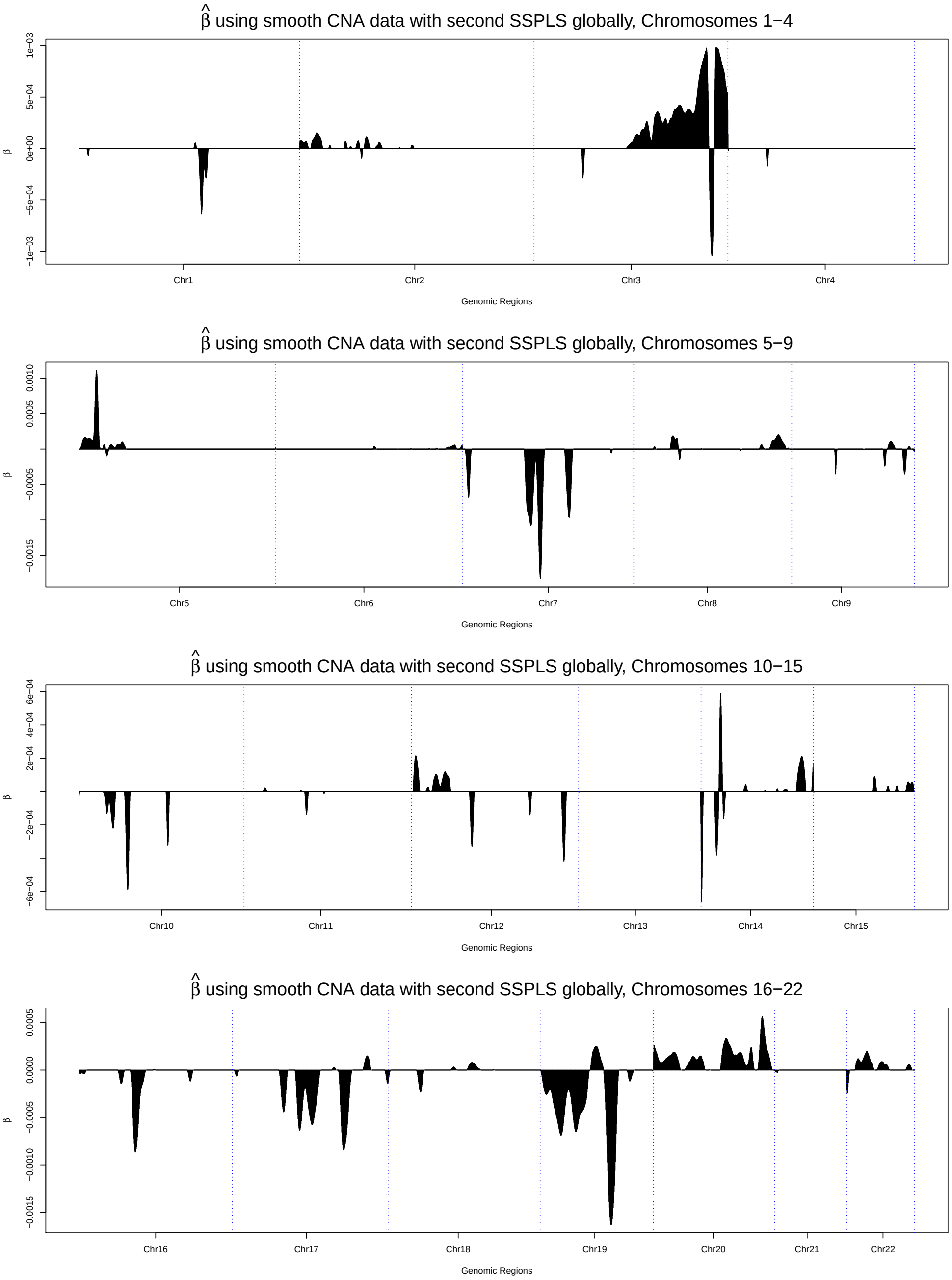


Figure 5.38: The estimation of $\hat{\beta}$ using the second SSPLS with the global model for CNA profiles. Those genomic windows with missing values (for example in the centromere regions) were not plotted since they were not used in the analysis.

It can be seen from Figure 5.38 that the large positive estimates of $\hat{\beta}$ which are in chromosomes 3, 5, 20 and 22 contribute more to SCC class. Large absolute value of the negative estimates of $\hat{\beta}$ in chromosomes 1, 7, 9, 10, 12, 14, 16, 17 and 19 contribute to ADC class.

Figure 5.39 shows w_1 with mixture of smoothed Cauchy and Laplace distributions using the second SSPLS globally.

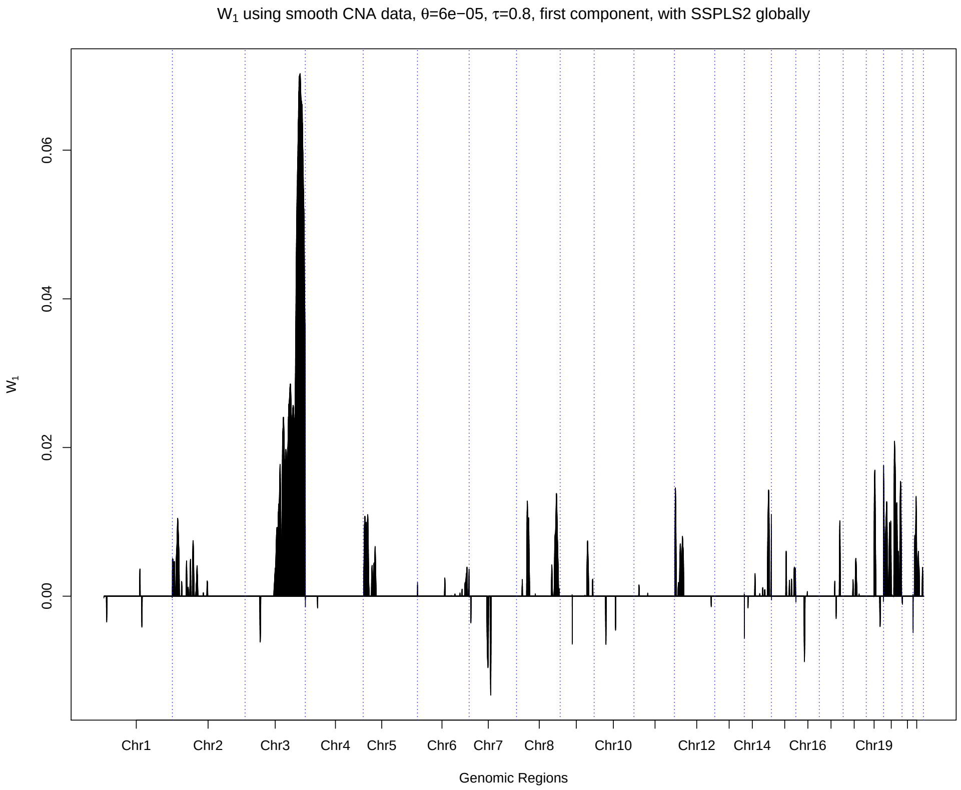


Figure 5.39: w_1 with mixture of smoothed Cauchy and Laplace distributions penalty using the second SSPLS globally ($\tau_{opt} = 0.8$, $\theta_{opt} = 6 \times 10^{-5}$, $m = 1$).

Looking at the estimation of w_1, we can see that the genomic regions that have signals indicate that they have an association to the cancer type.

Figure 5.40 shows w_2 with mixture of smoothed Cauchy and Laplace distributions using the second SSPLS globally.

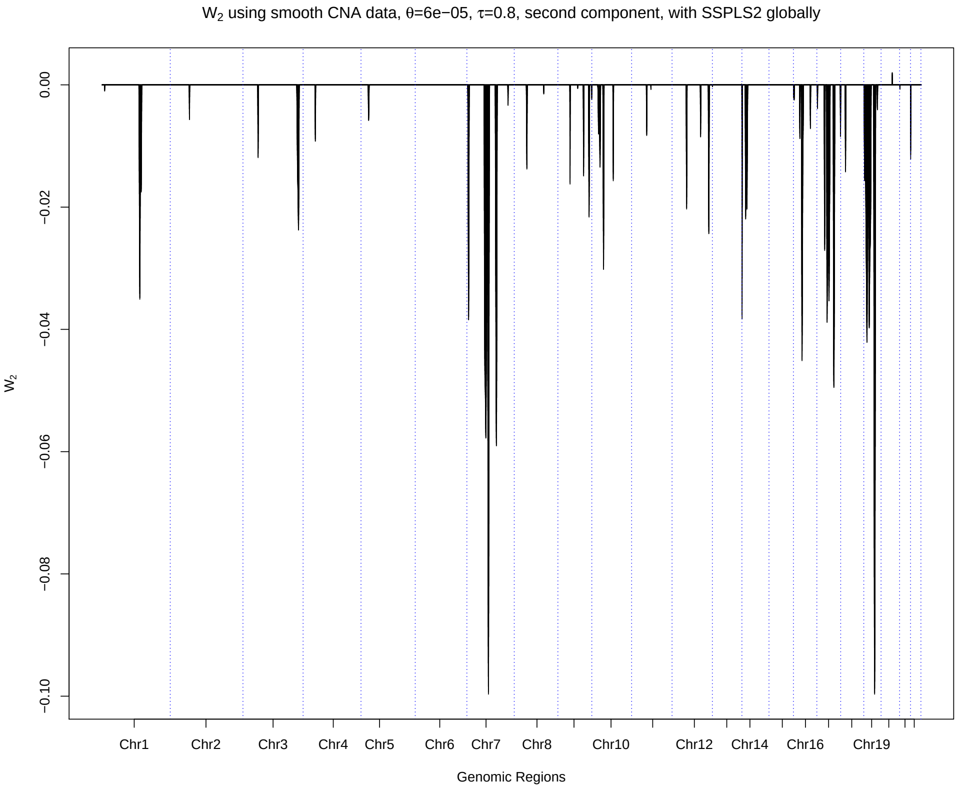


Figure 5.40: w_2 with mixture of smoothed Cauchy and Laplace distributions penalty using the second SSPLS globally ($\tau_{opt} = 0.8$, $\theta_{opt} = 6 \times 10^{-5}$, $m = 2$).

It can be seen from Figure 5.40 that all the chromosomes that have signals in their genomic regions are associated with the cancer type.

Figure 5.41 shows w_3 with mixture of smoothed Cauchy and Laplace distributions using the second SSPLS globally.

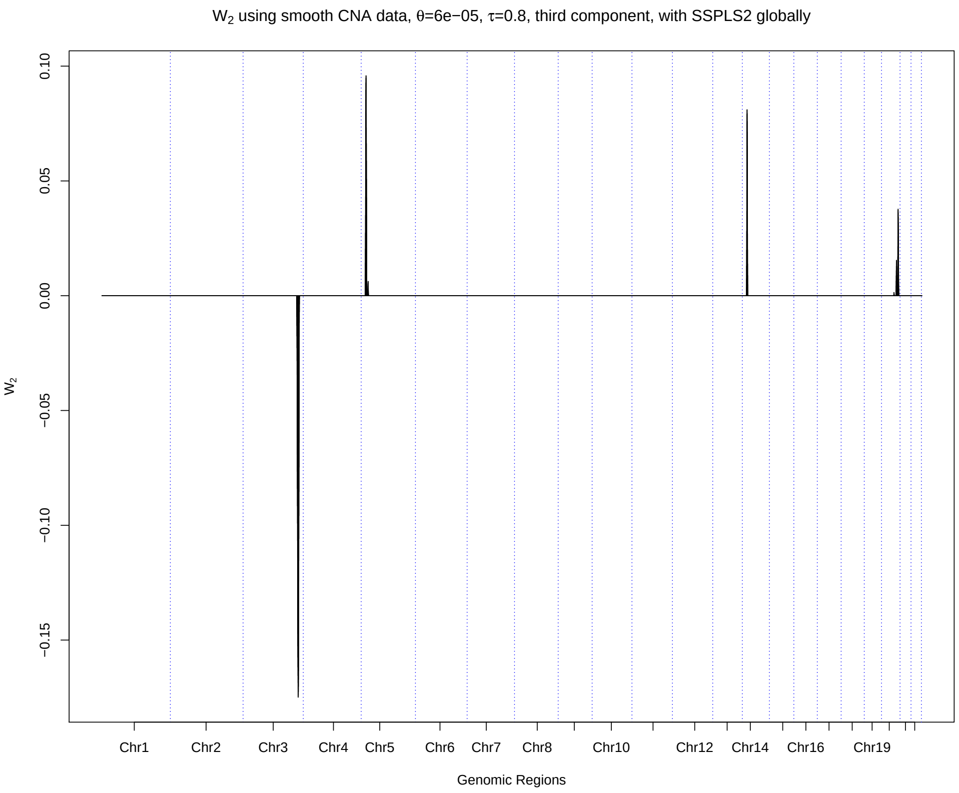


Figure 5.41: w_3 with mixture of smoothed Cauchy and Laplace distributions penalty using the second SSPLS globally ($\tau_{opt} = 0.8$, $\theta_{opt} = 6 \times 10^{-5}$, $m = 3$).

We can see from Figure 5.41, the signals that chromosomes that are associated with the type of the cancer.

Figure 5.42 shows the estimation of $\hat{\beta}$ using the second SSPLS with global model optimal values for CNA data based on 2 components.

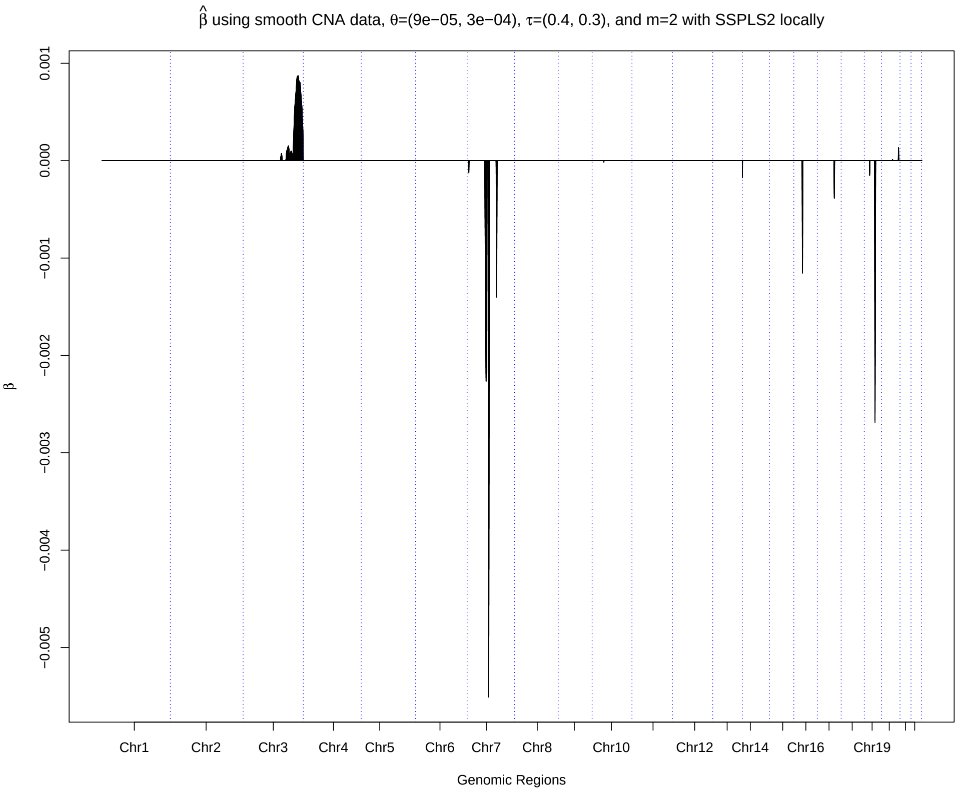
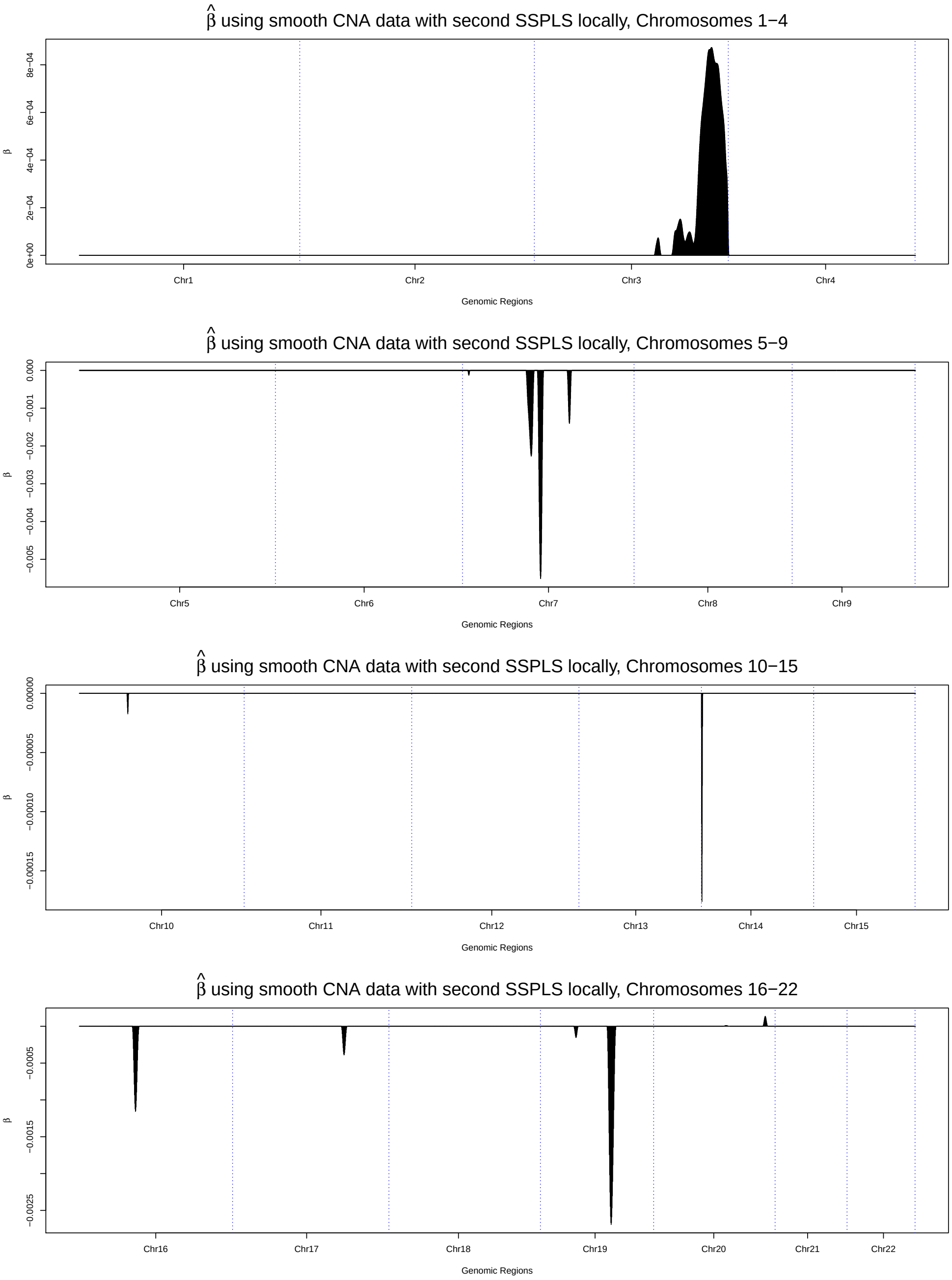


Figure 5.42: The estimation of $\hat{\beta}$ using the second SSPLS with the full local model for CNA profiles. A more detailed view of the random effects estimates in each chromosome is presented in Figure 5.43.

Figure 5.43 shows a more detailed view of estimation of $\hat{\beta}$ using the second SSPLS with optimal values of the tuning parameters using the local model for CNA data based on 2 components.



$\hat{\beta}$ using smooth CNA data with second SSPLS locally, Chromosomes 1−4

$\hat{\beta}$ using smooth CNA data with second SSPLS locally, Chromosomes 5−9

$\hat{\beta}$ using smooth CNA data with second SSPLS locally, Chromosomes 10−15

$\hat{\beta}$ using smooth CNA data with second SSPLS locally, Chromosomes 16−22

Figure 5.43: The estimation of $\hat{\beta}$ using the second SSPLS with the full local model for CNA profiles. Those genomic windows with missing values (for example in the centromere regions) were not plotted since they were not used in the analysis.

We can see from Figure 5.43 that some of the genomic regions in chromosome

3 have large positive estimates of $\hat{\beta}$. This indicate that they contribute to SCC class. In contrast, large absolute values of negative estimates of $\hat{\beta}$ as can be seen in some genomic regions in chromosomes 7, 14, 16, 17 and 19 contribute more to ADC class.

Figure 5.44 shows w_1 with mixture of smoothed Cauchy and Laplace distributions using the second proposed SSPLS locally.

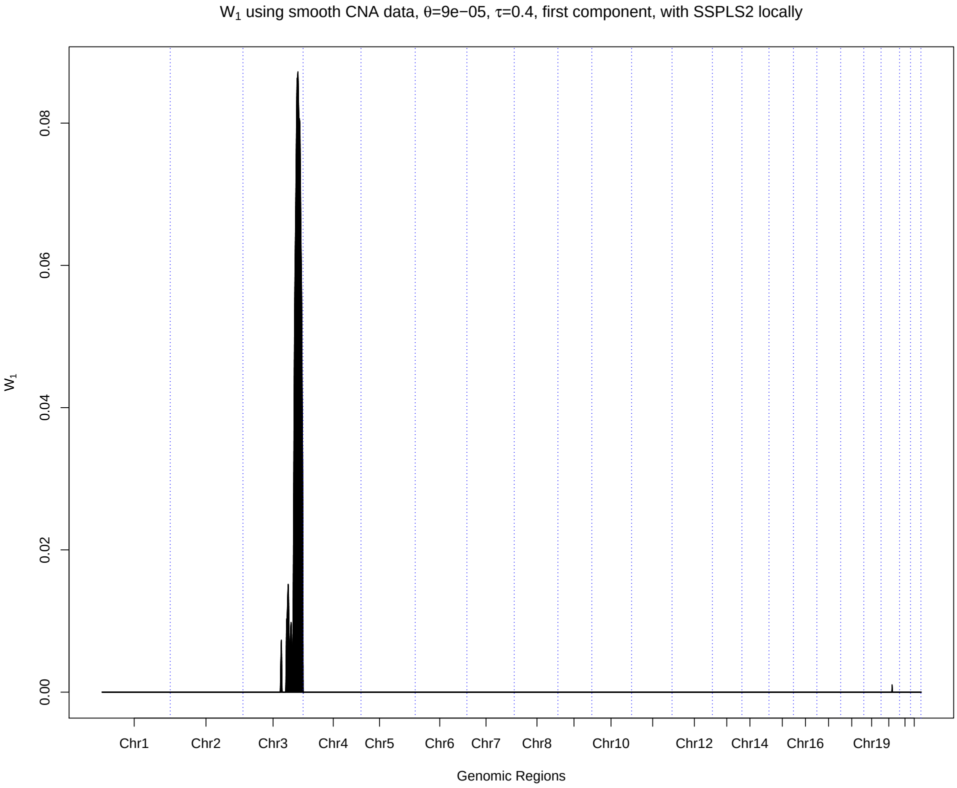


Figure 5.44: w_1 with mixture of smoothed Cauchy and Laplace distributions penalty using the second SSPLS locally ($\tau_{opt} = 0.4$, $\theta_{opt} = 9 \times 10^{-5}$, $m = 1$).

It can be seen from Figure 5.44 the genomic regions that have signals are associated with the cancer type.

Figure 5.45 shows w_2 with mixture of smoothed Cauchy and Laplace distributions using the second proposed SSPLS locally.

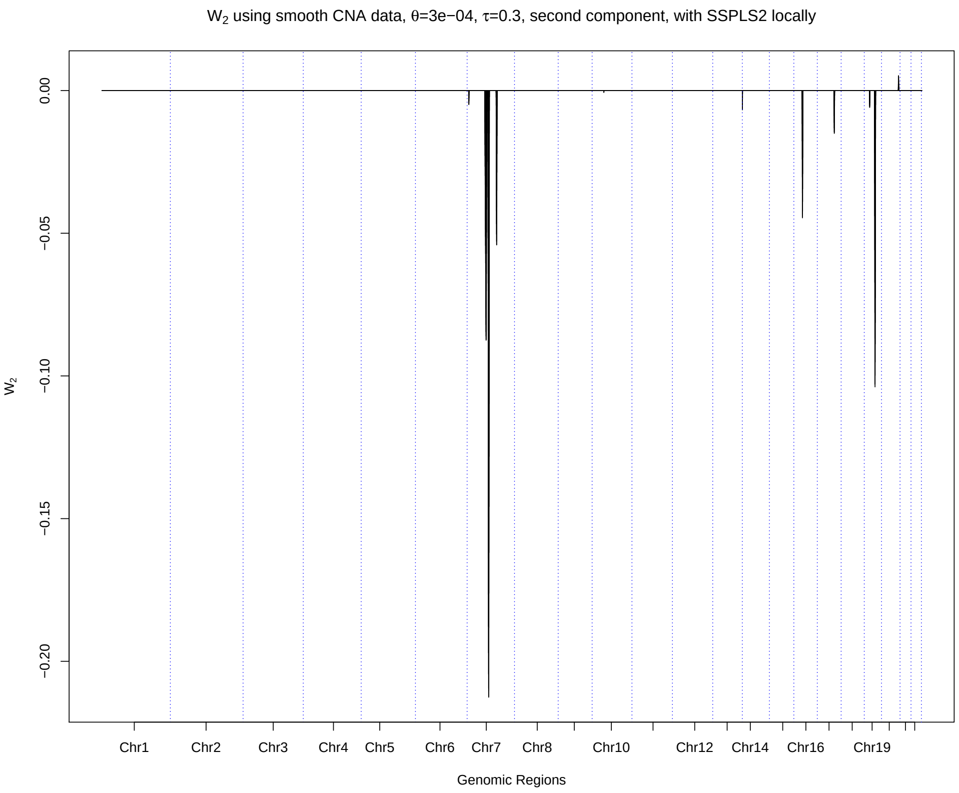


Figure 5.45: w_2 with mixture of smoothed Cauchy and Laplace distributions penalty using the second SSPLS locally ($\tau_{opt} = 0.3$, $\theta_{opt} = 3 \times 10^{-4}$, $m = 2$).

Looking at the estimates of w_2, we can see from Figure 5.45 that the genomic regions in chromosomes (7, 14, 16, 17, 19) have signals associated with cancer type.

Over all, to compare between the first and second methods of SSPLS, we can see that they are similar in detecting the genomic regions and chromosomes that are associated with either SCC or ADC. Both methods have provided sparseness and smoothness. From time computation point of view using the second method is recommended with local model.

5.9.3 Comparisons

Figure 5.46 shows a box plot of the root misclassification error rate (RMERCV) using five-fold cross-validation for six methods which are SPLS-HL, SPLS2-HL proposed by Lee *et al.* (2011), first SSPLS global, second SSPLS global, first SSPLS local and second SSPLS local using CNA data. We split the data into five folds, then we applied the candidates methods above using the optimal tuning parameters for each method on the training sets then we calculate the RMERCV for each fold. Therefore, the result of this procedure is being plotted as in Figure 5.46.

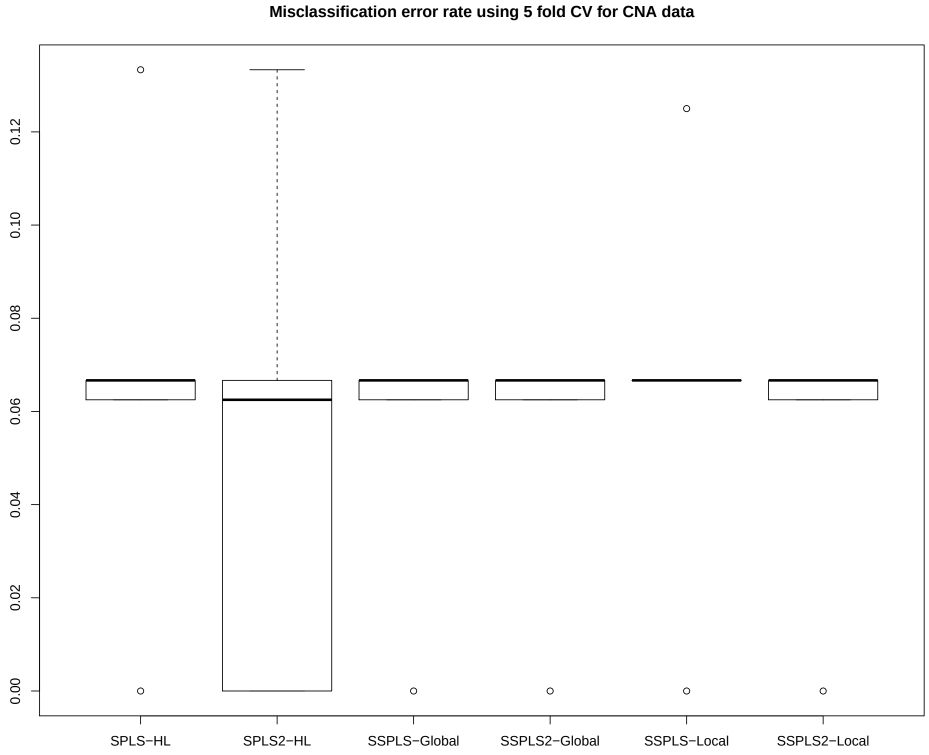


Figure 5.46: Box plot of MER for six methods which are SPLS-HL, SPLS2-HL, first SSPLS global, second SSPLS global, first SSPLS local and second SSPLS local using CNA data.

Looking at Figure 5.46, we can see that all six methods have a comparable misclassification error rate (MER) using five-fold cross-validation. However, SPLS2-HL has the lowest median of MER compared to other five methods.

5.10 Discussion

In this chapter, we have proposed two methods with the same penalty which is a mixture of two distributions (Cauchy and Laplace). In terms of choosing the tuning parameters, we propose two different solutions using the global PLS model and local model for each method (first SSPLS and second SSPLS). These two methods are based on two different versions of NIPALS algorithm for calculating the direction vectors. Those direction vectors were used to impose smoothness and sparseness. Applying both methods with global and local models using the simulated data where the response is on a continuous scale. The results show that the first SSPLS and the second SSPLS with local model give more sparseness since the weight for Laplace distribution (1 - τ) is not equal to zero.

Comparing all candidate methods (first SSPLS and second SSPLS) with global and local models to SPLS-HL and SPLS2-HL on simulation data, we found that SPLS-HL and SPLS2-HL slightly outperform our proposed methods. However, first SSPLS and second SSPLS might be preferred when dealing with highly-correlated data where SPLS-HL and SPLS2-HL would be difficult to interpret the results. If there are some sudden jumps for one variable without considering the correlation to the neighbouring variables.

Applying the same methods on the real data as NIR for a real-valued response, it can be seen that the first SSPLS and the second SSPLS globally and locally outperform over SPLS-HL and SPLS2-HL. For genomic profiles data (CNA data), we can see that all six methods are close to each other in terms of MER. However, using the first SSPLS and the second SSPLS local model give more sparse solutions. In contrast, first SSPLS and second SSPLS using the global model do not give sparsity as much as the local model. We think it might be appropriate to use the local model for both methods of SSPLS because the residuals from the component before is like a new data which should be treated independently with it optimal tuning parameters of θ and τ. We recommend the user to apply all methods and then choose one for the analysis.

Finally, our new method automatically selects relevant variables without sacrificing prediction performance. Not only that, we also imposed smoothness to deal with the spatial structure of CNA genomic regions.

Chapter 6

Graphical Modelling and Partial Least Square Regression

6.1 Overview

In Chapter 2 we introduced the ordinary PLS and from both NIPALS algorithms 1 and 2, we can see that the direction vectors w_m have a direct connection between the (X) predictors and the response variable (y). In this chapter we focus more on the w in each component m by trying to interpret them. To interpret w in component m, we combine the idea of the graphical modelling and PLS first. After that, the latent variables of PLS regression model can be used to interpret w in component m.

Wold (1966) developed the PLS approach to structural equation modelling (PLS path modelling) which is used as an alternative to the covariance based model (CB-SEM) proposed by Jöreskog (1978). PLS path models maximise the explained variance of the latent variables whereas CBSEM estimates model parameters so that the discrepancy between the estimated and sample covariance is minimised. PLS path models in path analysis have been discussed by many researchers e.g. Monecke & Leisch (2012) and F. Hair Jr *et al.* (2014). In PLS path models, the most interest of the latent variables is only the coefficient parameters without interpreting the w_m where m is the number of components that are used in the PLS model.

In graphical modelling we are interested in interpreting w_m such that we can find the relationships between the predictors before affecting y. Since graphical modelling depends on the principle of conditional independence, it might be helpful to interpret

w in component m in order to find the layers before reaching the response. For example, we are interested in interpreting the subsequent components by identifying the variables that are conditionally independent. Moreover, we are investigating if there are some variables that are affecting the response variable through other variables.

As a result, combining graphical modelling with PLS might help us to interpret the subsequent components (w_m). More specifically, using graphical Gaussian model to model the data and applying PLS methods to interpret the output components and get some insight of how the explanatory variables are connected to the response either directly or indirectly through each other.

This chapter is organised as follows. In Section 6.2, we introduce the graphical Gaussian model and the role of the inverse covariance matrix (S^{-1}). In Section 6.3, there is some discussion about the principle of the connection between PLS and graphical Gaussian model. Section 6.4 provides some examples of different graphs to get insight of the effect of PLS model. In Section 6.5, we discuss the idea of a graphical Gaussian model using PLS regression.

6.2 Graphical Gaussian models

Recall the general univariate regression model has an equation of the form

$$y = X\beta + \epsilon, \tag{6.1}$$

where y is the response variable, X is the predictors matrix, β is the coefficients vector of length p, and ϵ is the error term normally distributed with mean 0 and constant variance equals to one. X and y are centred to have no intercept in the model 6.1.

Graphical Gaussian models are based on the multivariate distribution. Whittaker (2009) was the first who introduces continuous models and called them graphical Gaussian models. Suppose that $X = (x_1, \ldots, x_p)$ is a p-dimensional random variable, with a multivariate normal distribution with mean

$$\mu = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

and covariance matrix

$$S = \begin{pmatrix} \sigma_{11} & \dots & \sigma_{1p} \\ \vdots & \ddots & \vdots \\ \sigma_{p1} & \dots & \sigma_{pp} \end{pmatrix},$$

The importance of the graphical Gaussian model comes from the inverse covariance matrix, S^{-1}, written as

$$S^{-1} = \begin{pmatrix} \sigma_{11}^{-1} & \dots & \sigma_{1p}^{-1} \\ \vdots & \ddots & \vdots \\ \sigma_{p1}^{-1} & \dots & \sigma_{pp}^{-1} \end{pmatrix}.$$

This is sometimes called the precision matrix.
For $p = 3$, we say that X_1 and X_2 are independent given X_3 if the $\sigma_{12}^{-1} = 0$ and $\sigma_{13}^{-1} \neq 0$ and $\sigma_{23}^{-1} \neq 0$. In other words, X_1 and X_2 are conditionally independent given X_3. For more details about graphical Gaussian models see e.g. Edwards (2012).

We simulate the data from a multivariate normal distribution where the number of samples (n) and the number of explanatory variables (p) are changed for each graph as in Section 6.4. The regression vector β is set to have different values where the covariance between X and the response variable y to be high around 0.7 and some of them medium around 0.5 and low around 0.1. The response y is created according to the general regression model in Equation (6.1). The mean vector for predictors is set to be a vector of zero values. The inverse covariance matrix is generated where that those variables are connected with each other, their $\sigma_{ij}^{-1} \neq 0$ for ($i \neq j$). Those are not linked with each other, their $\sigma_{ij}^{-1} = 0$ for ($i \neq j$).

6.3 Connection between PLS latent variables and the covariance matrix

From the second NIPALS algorithm 2, we have that $w = X^T y / y^T y$ for component m. It is important to look at the $w_1, \dots, w_m$ as they span the same space of Krylov sequence (Helland, 2001):

$$X^T y, (X^T X) X^T y, \dots, (X^T X)^{m-1} X^T y.$$

In order to see the connection between Gaussian graphical model and PLS, we can find there is a connection in terms of the covariance matrix of X and the covariance vector between X and y. In graphical Gaussian modelling, we create the inverse covariance matrix where $\sigma_{ij}^{-1} = 0$, for $i \neq j$, for those which are not connected between each other. Since w_1 is just the covariance vector between X and y ($S_{xy} = \frac{1}{n-1}X^Ty$), the first component will show those variables that are connected directly to y.

By looking at w, we can get some information regarding the covariances between X and y. Moreover, if we need to find some information and relations within X variables (predictors) only, it can be found in the multiplication of some of the latent variables. We have from the PLS model building and NIPALS algorithm that:

$$X = TP^T \tag{6.2}$$

Multiplying X by its transpose, we will have:

$$X^TX = PT^TTP^T, \tag{6.3}$$

and if X is a column centred, we have that

$$X^TX = (n-1)S_{xx}. \tag{6.4}$$

Since the left hand side of Equation (6.3) and (6.4) are the same, we can equate the right hand sides i.e.

$$S_{xx} = \frac{1}{(n-1)}PT^TTP^T. \tag{6.5}$$

In order to identify the explanatory variables (X variables) that are connected or affecting the response variable (y) directly, we may get them by looking at the significant variables of the first component of X-weights, w_1. Also, in order to identify the connection between the explanatory variables to each other, it might be done by looking at the combination of some latent variables. This can be seen in Equation (6.5) PT^TTP^T where T is the matrix of the X-scores, and P is the matrix of the X-loadings. The principle is that if there are two explanatory variables say x_1 and x_2 are dependent and x_1 and y are dependent, x_1 is expected to be significant in w_1, and x_2 may appear in the second component (w_2) indicating that x_2 and y are independent given that x_1. That means, x_2 is connected to y through x_1. From W matrix of PLS, we expect that identifying the covariates that are connected to y indirect by looking at the significant covariates in the second component w_2. To gain some insight about how PLS and graphical Gaussian model are related, some examples and graphs are provided below with discussion.

6.4 Applying PLS models on different simulated graphs

6.4.1 First graph

Let the regression model can be written as

$$y = X\beta + \epsilon, \tag{6.6}$$

where y is the response variable with size $n \times 1$, n is the sample size which can take three different values as 100, 1000 and 10000. The coefficient β is of length $p = 8$, and ϵ is the error term normally distributed with mean 0 and constant variance equals to one. X matrix of size $n \times p$ is generated from a multivariate normal distribution with mean zero, and an inverse covariance matrix as

$$S^{-1} = \begin{pmatrix} \sigma_{11}^{-1} & \dots & \sigma_{1p}^{-1} \\ \vdots & \ddots & \vdots \\ \sigma_{p1}^{-1} & \dots & \sigma_{pp}^{-1} \end{pmatrix},$$

where $\sigma_{ij}^{-1} = 0$ for those are not connected with each other, and $\sigma_{ij}^{-1} = -0.1$ for $i \neq j$ as in the first graph 6.1.

In this graph, we simulate 100 data sets based on the covariance matrix as in the first graph Figure 6.1. The coefficients vector β is set where x_1 is connected directly to y with a large value of β_1 equals to 0.7, x_2 and x_3 are connected directly to y with a medium value for β_2 and β_3 equal 0.5 while x_4 is connected to y directly with a low value for β_4 equals to 0.1. For the other variables x_5, x_6, x_7 and x_8 have β values equal to zero.

We set the connection between the covariates and the response variable as above in order to check if the results of the PLS methods still can identify the variables that are connected to the response variable or not.

We use the general regression model Equation (6.6) to generate y.

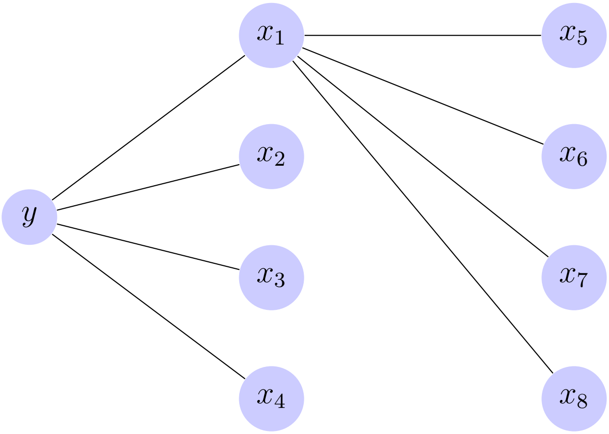


Figure 6.1: The first graph where for the nodes that have an edge, the $\sigma^{-1}_{x_i x_j} = -0.1$ for $i \neq j$, and for the nodes that do not an edge, the $\sigma^{-1}_{x_i x_j} = 0$. For $i = j$, $\sigma^{-1}_{x_i x_i} = 1$. The coefficients are set as $\beta_1 = 0.7, \beta_2 = 0.5, \beta_3 = 0.5, \beta_4 = 0.1, \beta_j = 0$, for $j = 5, 6, 7, 8$.

Figure 6.2 shows the box plot of 100 simulated data sets of $\hat{\beta}$ using the standard PLS1 model as described in Chapter 2 for the generated data using the first graph Figure 6.1.

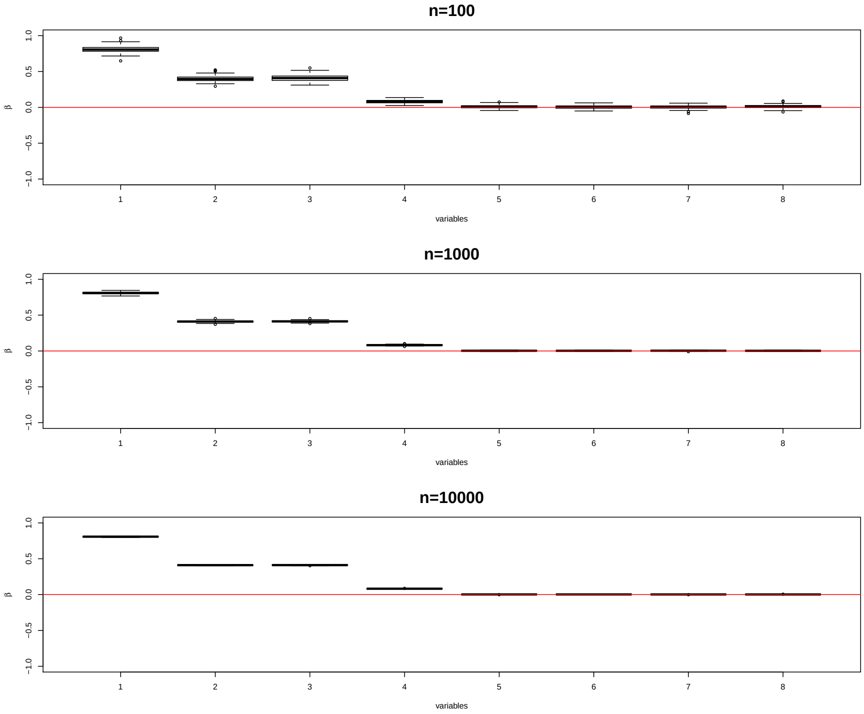


Figure 6.2: Box plot of 100 simulated data sets of $\hat{\beta}$ using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where ($p = 8$) for first graph. The red horizontal line is to identify significant variables that are above the red line and non-significant if they are below the red line. The number of components used in the PLS1 model for this graph graph is 2.

It can be seen from top panel of Figure 6.2 the first four variables from x_1 to x_4 are above the red line. This means they are not equal to zero which indicates that these variables are associated to the response variable (outcome) (y) as we anticipated where the number of samples ($n = 100$). Although the correlation between x_4 and the y is very low and close to the correlation between x_5 and y, we still can see that x_4 is not exactly equal to zero because it is connected directly to y. The rest variables from x_5 to x_8 are not significant since they are not connected to y in the constructed first graph Figure 6.1. In the middle panel of Figure 6.2 where the number of samples is 1000, we can see the first four variables from x_1 to x_4 are clearly, over 100 simulations, their estimations of $\hat{\beta}$ are not equal to zero. The estimation of $\hat{\beta}$ for variables from x_5 to x_8 are equal to zero because they are not connected to y directly. In the bottom panel of Figure 6.2, it can be seen when the number of samples increased to 10000, the estimation of $\hat{\beta}$ for covariates from x_1 to x_4 are clearly not equal to zero over all 100 simulated data sets. The estimation of $\hat{\beta}$ for the covariates from x_5 to x_8 are set to be equal zero which indicates that they are not associated with y. Indeed, it can be seen from Figure 6.2 the estimation of $\hat{\beta}$ using PLS method can help to identify the covariates that are directly connected to the response variable (y) and as we increase the number of samples, the estimation of $\hat{\beta}$ for the covariates that are connected to y are not equal to zero.

Figure 6.3 shows the box plot of 100 simulated data sets of w_1 using the standard PLS1 model as described in Chapter 2 for the generated data using the first graph Figure 6.1.

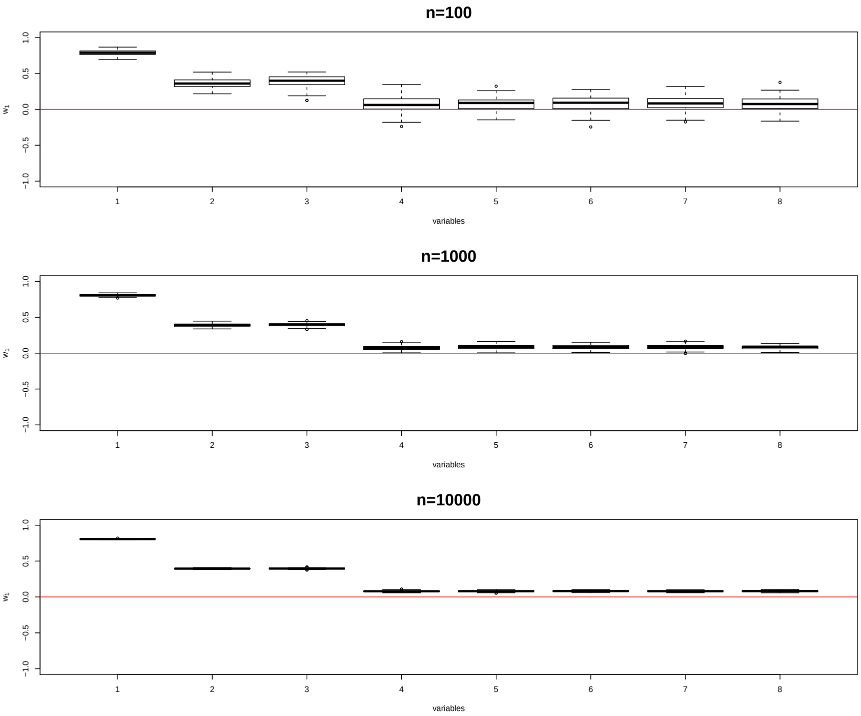


Figure 6.3: Box plot of 100 simulated data sets of the first component w_1 using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where ($p = 8$) for first graph. The red horizontal line is to identify significant variables that are above the red line and non-significant if they are below the red line.

It can be seen from the top panel of Figure 6.3 the X-weights for the first component (w_1) when the number of samples is 100 overall 100 simulated datasets. In this component we can see that the first three variables (x_1, x_2 and x_3) are not equal to zero. The median of 100 simulated data sets for the estimation of w_1 of the x_4 is above the red line but some simulated data sets have the estimation of w_1 equal to zero because the covariance between x_4 and y is very small. The estimation of w_1 of the covariates from x_5 to x_8 is equal to zero because they are not connected to y directly as we anticipated. We expect the estimation of w_1 for x_4 may not be equal to zero, but that does not hold because the covariance between x_4 and y is very small. Looking at the middle panel of Figure 6.3, we can see the estimation of w_1 for the covariates from x_1 to x_4 is not equal to zero when the number of samples is increased to be 1000. The estimation of w_1 for the covariates from x_5 to x_8 is close to zero (red horizontal line) which indicates that they are not equal to zero. In the bottom panel of Figure 6.3, we can see the estimation of w_1 for the first four variables from x_1 to x_4 is not equal to zero when the number of samples is 10000. The estimation of w_1 for the covariates from x_5 to x_8 is less than 0.1 because these covariates are not connected directly to y. It can be seen from Figure 6.3 as the number of samples increased the estimation of 100 simulated data sets for x_4 becomes more clear to be not equal to zero in the first component. This means that the first component has x_1, x_2, x_3 and x_4 not to be equal zero since they are connected to y directly as can be seen in the first graph 6.1.

Figure 6.4 shows the box plot of 100 simulated data sets of w_2 using the standard PLS1 model as described in Chapter 2 for the generated data using the first graph Figure 6.1.

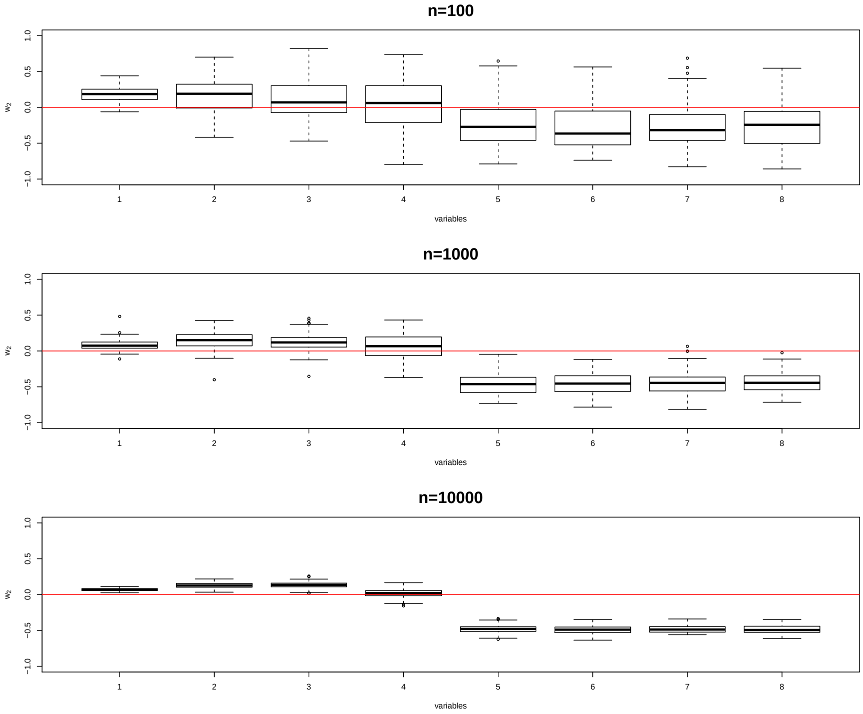


Figure 6.4: Box plot of 100 simulated data sets of the second component w_2 using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where ($p = 8$) for first graph. The red horizontal line is to identify significant variables that are above the red line and non-significant if they are below the red line.

It can be seen from the top panel of Figure 6.4 the X-weights for the second component (w_2) when the number of samples is 100. The estimation of w_2 for the covariates from x_5 to x_8 for some simulated data sets is not equal to zero in absolute values, but for the other simulated data sets they are equal to zero. Looking at the middle panel of Figure 6.4, we can see that when the number of samples is increased to be 10000, the estimation of w_2 for covariates from x_5 to x_8 are not equal to zero as we anticipated since they are connected to y through x_1. The bottom panel of Figure 6.4 illustrates that as the number of samples to be 10000, we can see that the estimation of w_2 for covariates from x_5 to x_8 are not equal to zero over all 100 simulated data sets. In short, we could conclude that as the number of samples increased, the estimation of w_2 for the covariates that are connected to y indirectly or conditionally independent to y are not equal zero. We are anticipating this to be achieved from combining the graphical modelling with w_2 which is the second component of W from the latent variables constructed from using the standard PLS regression method.

6.4.2 Second graph

In this graph, we use the same simulation settings as in Section 6.4.1 to generate y and ϵ. For X matrix of predictors, we simulate it from multivariate normal distribution with mean 0 and inverse covariance matrix as in Section 6.4.1. However, we use the second graph as shown in Figure 6.5 to construct the inverse-covariance matrix where $p = 20$.
The coefficients vector β is set where x_1 is connected to y with a high value of β_1 equals to 0.7, x_2 and x_3 are connected to y with medium values for β_2 and β_3 respectively with values equal to 0.5. x_4 is connected to y directly with a low value equals to 0.1 for β_4. For the variables that are not connected with y directly as in the second graph Figure 6.5, β is set to be zero.
Any two nodes that have an edge as in the second graph Figure 6.5, their inverse covariance values have $\sigma_{ij}^{-1} = -0.1$ for $i \neq j$. For any two nodes that do not have an edge, they have $\sigma_{ij}^{-1} = 0$ for $i \neq j$. We simulate 100 data sets based on the graph below as shown in Figure 6.5.

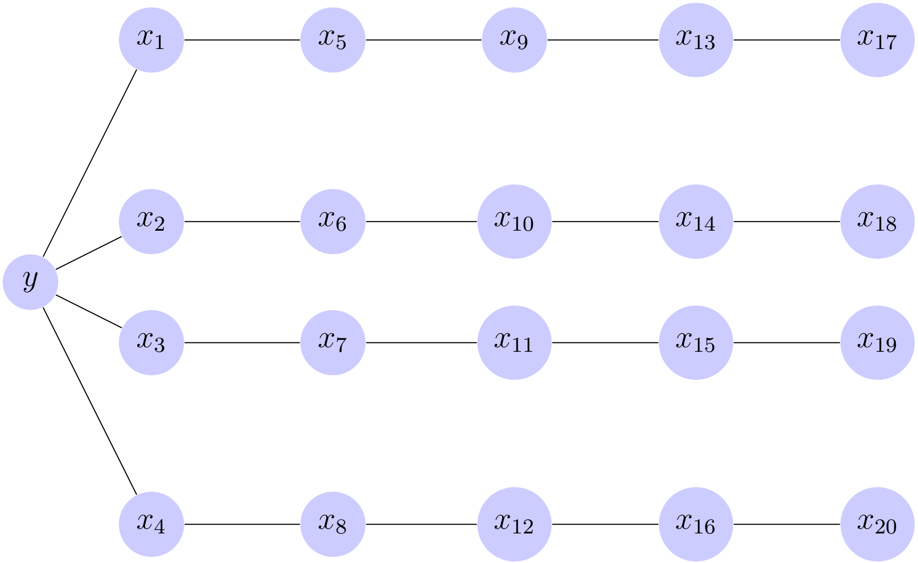


Figure 6.5: The first graph where for the nodes that have an edge, the $\sigma^{-1}_{x_i x_j} = -0.1$ for $i \neq j$, and for the nodes that do not an edge, the $\sigma^{-1}_{x_i x_j} = 0$. For $i = j$, $\sigma^{-1}_{x_i x_i} = 1$. The coefficients are set as $\beta_1 = 0.7, \beta_2 = 0.5, \beta_3 = 0.5, \beta_4 = 0.1, \beta_j = 0$, for $j = 5, 6, \ldots, 20$.

Figure 6.6 shows the box plot of 100 simulated data sets of $\hat{\beta}$ using the standard PLS1 model as described in Chapter 2 for the generated data using the second graph Figure 6.5.

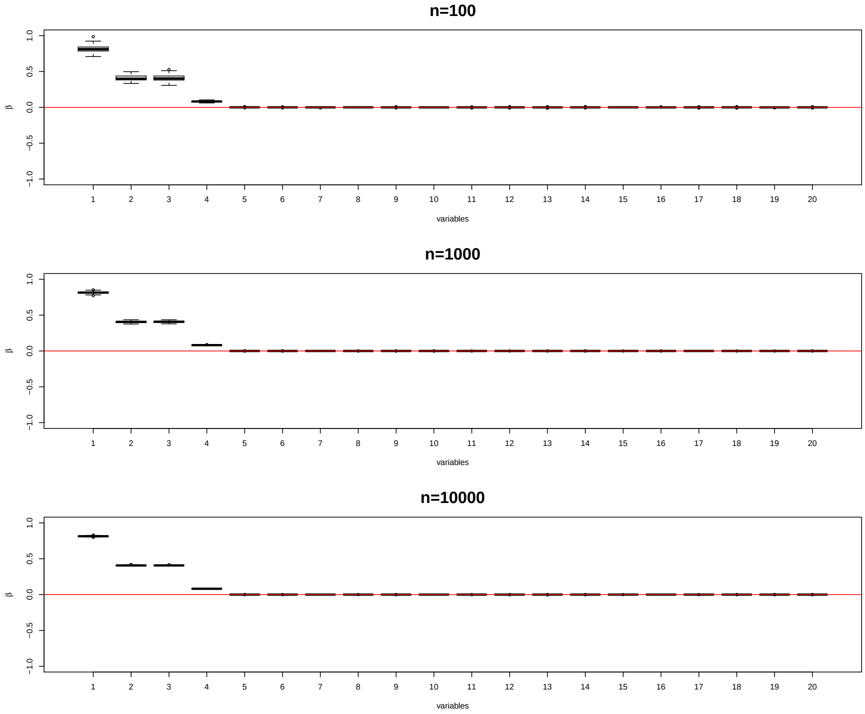


Figure 6.6: Box plot of 100 simulated data sets of $\hat{\beta}$ using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where ($p = 20$) for the second graph. The red horizontal line is to identify significant variables that are above the red line and not significant if they are below the red line.The number of components used in the PLS1 model is 5.

It can be seen from the top panel of Figure 6.6 using $n = 100$ that the estimation of $\hat{\beta}$ for the covariates x_1, x_2, x_3 and x_4 is not equal to zero which indicates that those variables are associated with the response variable. The estimation of $\hat{\beta}$ for the covariates from x_5 to x_{20} is equal to zero since they are on the red horizontal line. This indicates that they have no association with the outcome directly. In the middle panel of Figure 6.6 we can see that by increasing the number of samples to be 1000, the estimation of $\hat{\beta}$ for the covariates that are connected to the outcome directly more clearer than with $n = 100$. It can be seen from the bottom panel of Figure 6.6 the estimation of $\hat{\beta}$ where the first four covariates (x_1, x_2, x_3 and x_4) are not equal to zero whereas for the covariates from x_5 to x_{20} are equal to zero. The estimation of $\hat{\beta}$ for the covariates that are not equal to zero indicates that they have an association with the outcome directly as seen in the second graph 6.5. As the number of samples increased the estimation of $\hat{\beta}$ for the covariates that is associated with the outcome (y) become more clearer as it can be seen from the estimation plot of $\hat{\beta}$ in Figure 6.6.

Figure 6.7 shows the box plot of 100 simulated data sets of w_1 using the standard PLS1 model as described in Chapter 2 using the generated data using the second graph Figure 6.5.

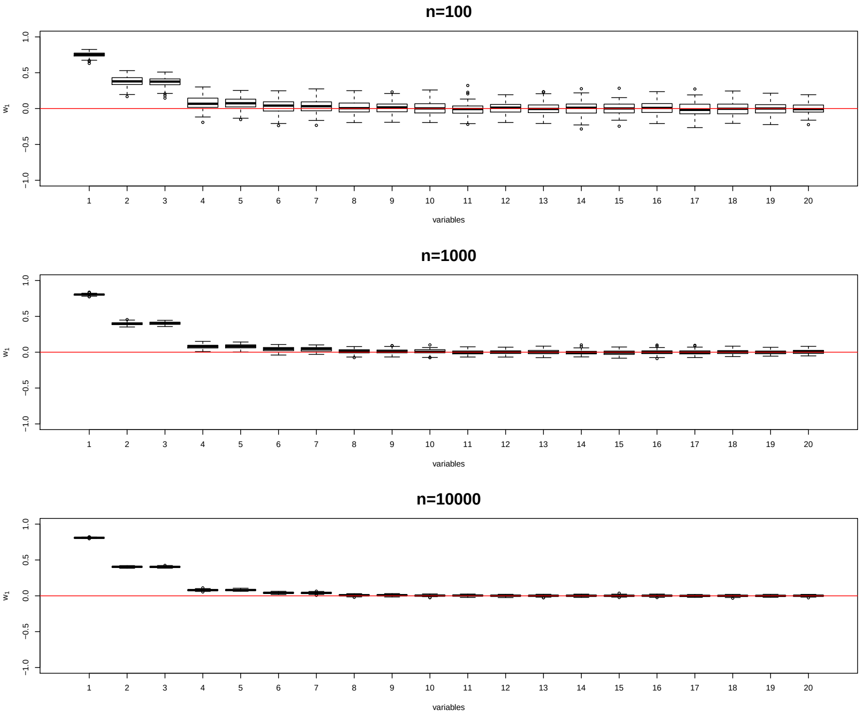


Figure 6.7: Box plot of 100 simulated data sets of the first component w_1 using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where ($p = 20$) for the second graph. The red horizontal line is to identify significant variables that are above the red line and not significant if they are below the red line.

It can be seen from the top panel of Figure 6.7 the estimation of w_1 for the covariates x_1, x_2, x_3 are not equal to zero when the number of samples is 100. This means they are associated with y as anticipated from the second graph 6.5. However, the estimation of w_1 for x_4 in some of the 100 simulated data sets are equal to zero. We would expect that the estimation of the first four covariates from x_1 to x_4 is not equal to zero. This indicates the from the estimation of w_1 (the first component of w), only the covariates that are related to y directly to have their estimation not equal to zero. The estimation of the covariates from x_5 to x_{20} is not equal to zero. In the middle panel of Figure 6.7 we can see that by increasing the number of samples to be 1000, x_4 becomes significant for all simulated 100 data sets. Looking at the bottom panel of Figure 6.7, it can be seen the estimation of w_1 when the number samples is increased, x_1, x_2, x_3 and x_4 are directly connected to y as constructed in the second graph. Furthermore, x_5, x_6 and x_7 have values of the 100 simulated data sets less than 0.1 and close to zero. These covariates are being identified in the first component which we did not expect that though their estimations of w_1 are close to zero. We think the reason is that these covariates are connected to other x covariates which have large covariances with the outcome (y). In short, we can still get some insights from interpreting the estimation of w_1 which may be used to identify the covariates that are connected directly to y.

Figure 6.8 shows the box plot of 100 simulated data sets of w_2 using the standard PLS1 model as described in Chapter 2 for the generated data using the second graph Figure 6.5.

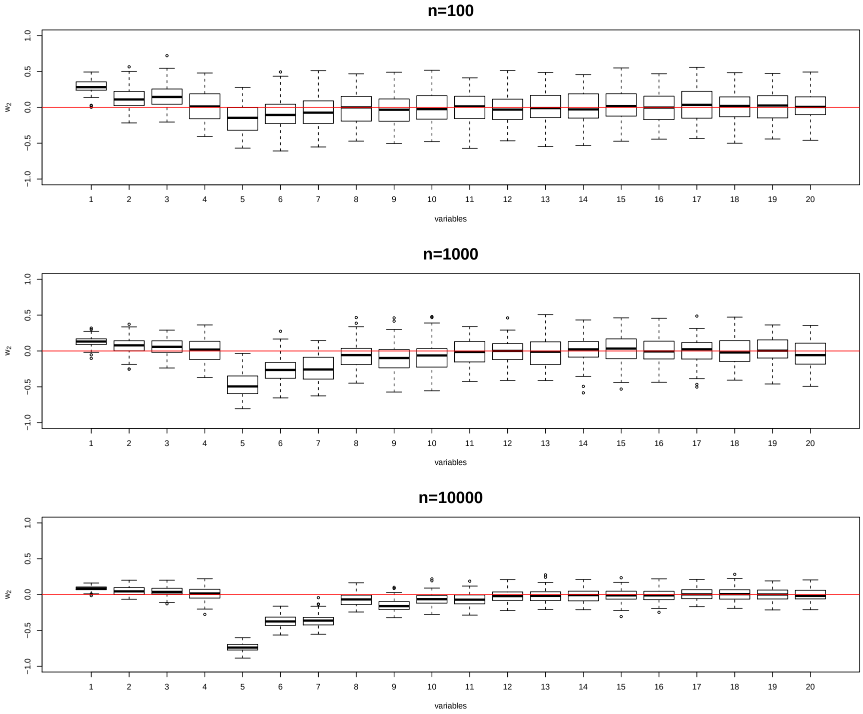


Figure 6.8: Box plot of 100 simulated data sets of the second component w_2 using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where ($p = 20$) for the second graph. The red horizontal line is to identify significant variables that are above the red line and not significant if they are below the red line.

It can be seen from the top panel of Figure 6.8 that it is difficult to identify which covariates that have the estimation of w_2 not equal zero easily because the number of samples is 100. In the middle panel of Figure 6.8 where the number of samples is 1000, we can see the median of the estimation of w_2 for the covariates x_5, x_6 and x_7 is below the red line which we anticipate to see in the second component w_2. This means that the second component w_2 may identify the covariates that are connected to the response through another covariate as constructed in the second graph Figure 6.5. The median of the estimation of x_8, x_9 and x_{10} is below the red line, but there are some of the simulated data sets above the red line. Therefore, it is difficult to tell which of these covariates may are connected to y within two steps. In the bottom panel of Figure 6.8, it can be seen that we the number of samples is increased it becomes more clear for all simulated data sets of the estimation of w_2 for x_5, x_6, x_7 are not equal to zero as we constructed in the second graph Figure 6.5. Indeed, our hypothesis is that the covariates x_5, x_6, x_7 and x_8 are located in what called second layer which means they are linked to y indirectly. We expect that the second component w_2 can be able to identify their signals.

Figure 6.9 shows the box plot of 100 simulated data sets of w_3 using the standard PLS1 model as described in Chapter 2 for the generated data from the second graph Figure 6.5.

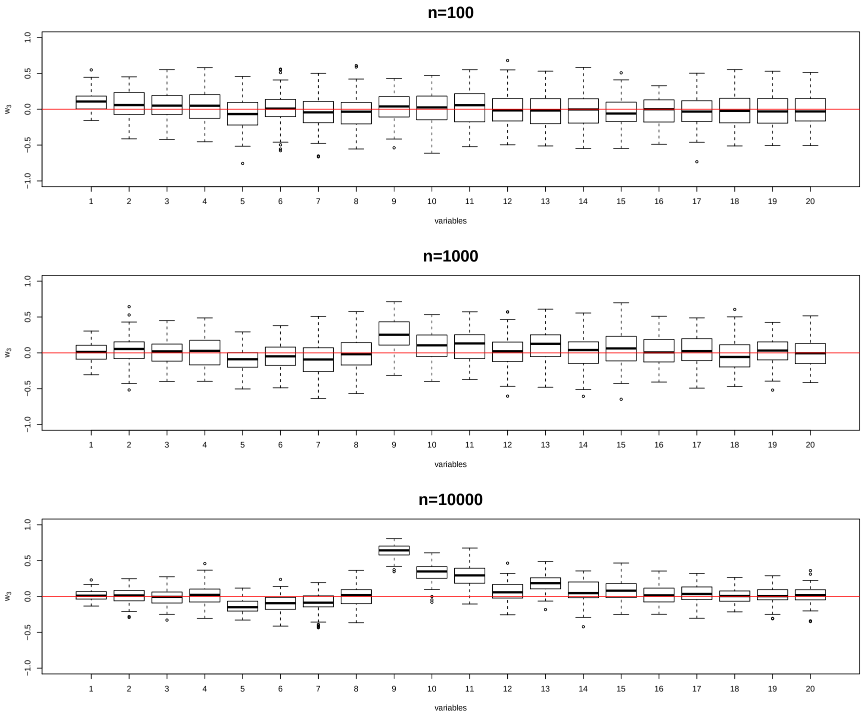


Figure 6.9: Box plot of 100 simulated data sets of the third component w_3 using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where ($p = 20$) for the second graph. The red horizontal line is to identify significant variables that are above the red line and not significant if they are below the red line.

It can be illustrated from the top panel of Figure 6.9 that the median of the estimation of w_3 for all covariates are close to zero which indicates it is difficult to have some insights with small number of sample with 100. Looking at the middle panel of

Figure 6.9, we can see the median of 100 simulated data sets of the estimation of w_3 is not equal to zero for covariates x_9, x_{10} and x_{11} when the number of samples is 1000. This indicates that w_3 might be able to identify the covariates that are connected in three steps to y (third layer). In other words, connected to y through other covariates as constructed in the second graph 6.5. It can be seen very clearly from the bottom panel of Figure 6.9 as the number of samples is increased, the estimation of w_3 for covariates x_9, x_{10}, x_{11} are not equal to zero for most of the 100 simulations. x_{12} is not very clear that it can be considered as in the third layer because it is connected to y through x_4 and x_8. But x_4 has a very small covariance with y ($\beta_4 = 0.1$).

Figure 6.10 shows the box plot of 100 simulated data sets of w_4 using the standard PLS1 model as described in Chapter 2 for the generated data from the second graph Figure 6.5.

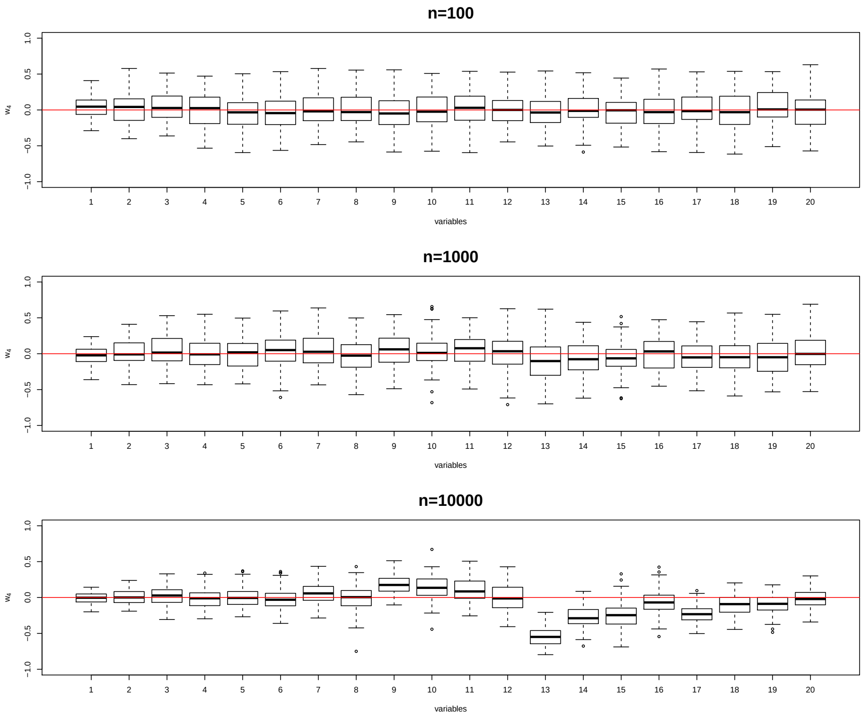


Figure 6.10: Box plot of 100 simulated data sets of the 4-th component w_4 using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where ($p = 20$) for the second graph. The red horizontal line is to identify significant variables that are above the red line and not significant if they are below the red line.

It can be seen from the top panel of Figure 6.10 when the number of samples is 100 and when the number of components is increased, it is very difficult to get some insights from the estimation of w_4. In the middle panel of Figure 6.10 when the number of samples is increased to be 1000, the estimation of w_4 for all covariates is close to zero. In the bottom panel of Figure 6.10 when the number of samples is increased to be 10000, we can see that the median of 100 simulated data sets for x_{13}, x_{14} and x_{15} is not equal to zero. This indicates that w_4 may have some information about the covariates that are connected to y in four steps (fourth layer). The covariates from x_{17} to x_{20} have a very small value of the median. In short, as the number of components is increased, the signals of the covariates become weak. Thus, it may be difficult to see the covariates that have signals in w_4.

Figure 6.11 shows the box plot of 100 simulated data sets of w_5 using the standard PLS1 model as described in Chapter 2 for the generated data from the second graph Figure 6.5.

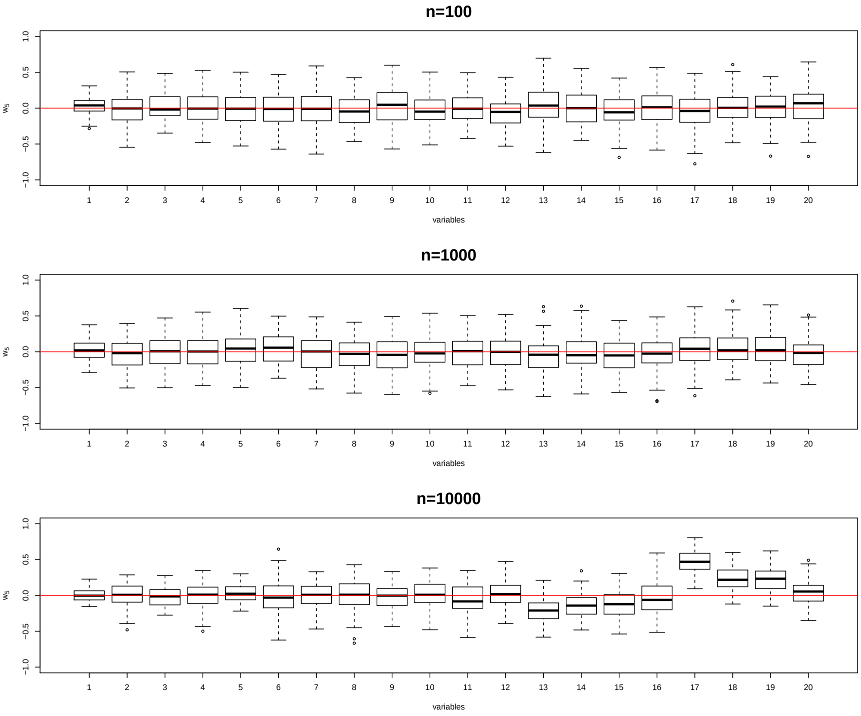


Figure 6.11: Box plot of 100 simulated data sets of the 5-th component w_5 using PLS1 model with different number of samples $n = 100$ (top panel) $n = 1000$ (middle panel) $n = 10000$ (bottom panel) where $(p = 20)$ for the second graph. The red horizontal line is to identify significant variables that are above the red line and not significant if they are below the red line.

It can be seen that from the top panel of Figure 6.11 when the number of samples in 100, the estimation of w_5 for all covariates is close to zero. In the middle panel of Figure 6.11 when the number of samples is 1000, the estimation of w_5 also close to zero for all covariates. From the bottom panel of Figure 6.11 we can see that by increasing the number of samples to be 10000, the estimation of x_{17} is not equal to zero for all 100 simulations. The median of the estimation of w_5 for x_{18}, x_{19} and x_{20} is not equal to zero. This may indicate that w_5 can show the covariates that have signals and connected to y in five steps (fifth layer) as constructed in the second graph Figure 6.5.

6.5 Discussion

In this chapter, we have investigated the graphical Gaussian modelling using PLS where we focus more on the latent variables specifically the X-weights matrix W_m. It was hoped that the information in w would correspond to that of the graphical model. Unfortunately, for large number of components (m), w_m failed to detect which covariates that lay in the m layer and they are connected to y indirectly through other covariates. However, it could be seen from the simulated graphs results that w_1 in the first component for sure will show the explanatory variables that strongly correlated or directly connected to y while in the second component w_2 would have covariates that are less connected to y. To interpret w_m, we might see that as the number of components is increased, there is no more strong relationship between the covariates and the response variable y. We anticipate that the estimation of w_m may have the covariates that have signals in each component m as constructed in the graph. Finally, as the number of components is increased, small sample size, and for complicated graphs, it is very difficult to interpret w_m in each component as we anticipated.

Chapter 7

Conclusion and Future Work

7.1 Overview

In this short chapter we provide a summary of the work done in this thesis in Section 7.2 and discuss some open problems in Section 7.3.

7.2 Summary of work done

Recall that the objective of our thesis is to deal with high-dimensional and highly-correlated data by various applications of the PLS method.

In Chapter 2 we provided an overview of the standard partial least squares regression for univariate response variable (PLS1) and for multivariate response variables (PLS2) with an investigation of three apparently different versions of the NIPALS algorithm for PLS2. We provided a proof of the equivalent of the estimation parameter β_{pls2} of using any of the three different versions of the NIPALS algorithms for PLS2 when the regression between X-scores and Y-scores (α) is included in the calculation of β_{pls2}. Using the NIR data with univariate and multivariate response variables, the standard PLS1 was applied as in Chapter 2. Although PLS was not designed for classification problems, we applied the standard PLS1 using CNA data where the response variable is binary.

In Chapter 3 we reviewed the shrinkage factors of the estimation of $\hat{\beta}$ for three common shrinkage methods in the literature which are RR, PCR and PLS when $n > p$.

We modified the idea of the estimation of $\hat{\beta}$ of these methods in the high-dimension case where $p > n$ using the reduced rank of the singular value decomposition. Since the OLS solution is not applicable in the high-dimension case, we replace the shrinkage factors term by filter factors. Moreover, we investigated the filter factors of the estimator of the three methods using NIR and CNA data sets. From the results of the filter factors, we observed that some of the filter factors values are negative with few references which have explored this more in detail.

In Chapter 4 we investigated more deeply the negative filter factors (NFF) using simulations. We provided conditions to show when the NFF occurs in each component (m) based on the relationship between the eigenvalues of two matrices (X^TX and $W_m^TX^TXW_m$), where W_m is the X-weights matrix with m components. We found that when the covariance between X and y is small, we may have more chances to have NFF. Further, for data that has independent variables, there is no chance to have NFF. We investigated more by simulating a small example to control the coefficients and explore the β values that are not likely to result in NFF and a more discussion is provided in Chapter 4. Looking at the results for the standard PLS1 on NIR and CNA data sets, we noted two problems: first, the length of the estimation of $\hat{\beta}$ which may results in a poor prediction if the irrelevant variables are included in the prediction model. Second, the dependencies between wavelengths in NIR data and genomic regions in CNA data.

To tackle these two problems, in Chapter 5 we proposed two methods of sparse smoothed partial least squares (SSPLS) (based on two different of the NIPALS algorithms) to achieve variable selection and smoothness. We achieve the sparseness by assuming the direction vectors w as random effects that follow a Laplace distribution, and to achieve smoothness we assume the second differences of adjacent values of w to follow a Cauchy distribution. We combined both distributions to solve the two problems simultaneously using a mixture of penalties derived from the two distributions, which are Laplace and Cauchy. Moreover, to estimate the tuning parameters, we considered two approaches which are based on the local model maximising the log-likelihood and a global model which minimises the regression model when β of PLS is used.

In Chapter 6 we explored a new view of interpreting the direction vectors w in each component by combining graphical Gaussian model and PLS. It was a hope that the information in w would correspond to that of the graphical model. In other words, we were trying to find the covariates that are correlated to each other before they are affecting the response variable. Unfortunately, we found that this idea is not valid and it may need more consideration in the future.

7.3 Future work

In this thesis we consider two types of response variable which are real-valued as in NIR data and binary as in CNA data. We mainly consider only a univariate response in the analysis except Chapter 2. We can extend the use of SSPLS method to deal with multivariate response variables by considering the response variable y which is a matrix of size $n \times q$ where q is the number of response variables. SSPLS is one of the most interested methods to deal with multivariate response variables since PLS2 is designed for that.

In some biological data sets we can include clinical information about patients such as age at surgery, sex, stage of disease, and grade of cancer by adding them in a matrix as fixed covariates. Inclusion of these variables would be straightforward, but they would be subject to different penalties (for example, no smoothness) to the genomic data. In this thesis we consider only the genomic regions (CNA profiles) as random effects in the original model.

In epidemiological research, the response variable (outcome) can be a non-normal distribution such as time to event data (survival data) or the Poisson distribution. For instance, it is important to extract the relevant information in the genomic regions in the CNA data (ultra-high dimensional data) in the prediction of cancer patients survival.

In order to deal with survival data, it could be interesting to adapt the idea of assuming the direction vectors w as random effects as in Lee *et al.* (2013) but we use our sparse smoothed penalty which is a mixture of two distributions (Laplace for sparseness and Cauchy for smoothness). The only part will change from this thesis is the way that w is calculated in the iterative reweighed partial least squares algorithm for Cox regression instead of the first step in both NIPALS algorithms (1 and 2) provided in this thesis earlier.

Appendix A

Additional of NIPALS Algorithms for PCA and PLS2

The NIPALS algorithm for PCA

This algorithm is given in (Geladi & Kowalski, 1986). Say X is an $(n \times p)$ data matrix, X is mean centred. The X-loading vectors, p_m form an orthonormal set, and the X-score vectors, t_m are orthogonal to each other.
t will be the scores vector for X
p will be the loadings vector for X
Initialisation: Set $X_1 = X$, t_1 is the first column of X, and m=1, where $m = 1, 2, \ldots, M$ and $M \leq \min(n-1, p)$.

Algorithm 4 The NIPALS algorithm for PCA

```
Take $t_m$ vector is set to a column in $X$, $t_m = X_1$
$p_m = (X^T t_m)/(t_m^T t_m)$
$p_m = p_m / \sqrt{p_m^T p_m}$        (normalisation)
$t_m = X_m p_m$
Check for convergence by comparing between $t_m$ used in step 2 with the one obtained in step 4. If they are the same, stop iterations and got to step 6, otherwise, go to step 2.
Update $X_m = X_m - (t_m p^T)$
```

After the first component is calculated, X in steps 2 and 4 has to be replaced by its residual, $X_m = X_{m-1} - (t_{m-1}p_{m-1}^T)$.

The NIPALS algorithm for PLS2

There are three versions of the NIPALS algorithm for PLS2 where they differ from each other based on the normalisation step for the loadings of X (P) and Y (Q)

The first version of the NIPALS algorithm for PLS2

This algorithm (5) is given in (Wold *et al.*, 1984). In this algorithm X-loadings (P) and Y-loadings (Q) are not normalised. The inner relation, α, is equal to one. It is called the simple NIPALS algorithm. Also, the X-weight vectors, w_m form an orthonormal set, and the X-score vectors, t_m are orthogonal to each other.

X is an $(n \times p)$ data matrix, Y is an $(n \times q)$ matrix of response variables, X and Y are mean centred.

t_m will be the scores vector for X

p_m will be the loadings vector for X

w_m will be the weights vector for X

u_m will be the scores vector for Y

q_m will be the loadings vector for Y

α_m will be the inner relation, which is a regression of u_m on t_m with no intercept.

Initialisation: Set $X_1 = X$, $Y_1 = Y$, and m=1, where $m = 1, 2, \ldots, M$ and $M \leq \min(n-1, p)$.

Algorithm 5 The first version of the NIPALS algorithm for PLS2

```
1: Take $u$ vector is set to a column in $Y$, $u_m = Y_1$
2: $w_m = X_m^T u_m^T / \sqrt{u_m^T u_m}$
3: $w_m = w_m / \sqrt{w_m^T w_m}$        (normalisation)
4: $t_m = X_m w_m$
5: $q_m = Y_m^T t_m / (t_m^T t_m)$
6: $u_m = Y_m q_m / (q_m^T q_m)$
7: Check for convergence by comparing between current $t_m$ with the previous $t_m$. If
   they are the same, stop iterations, then got to step 8, otherwise go to step 2.
8: $p_m = X_m^T t_m / (t_m^T t_m)$
9: The inner relation: $\alpha_m = u_m^T t_m / (t_m^T t_m)$
10: Save: X-loadings: $p_m$, X-weights: $w_m$, X-scores: $t_m$, Y-loadings: $q_m$, Y-scores:
    $u_m$, The inner relation: $\alpha_m$
11: Update: $X_m = X_{m-1} - (t_{m-1} p_{m-1}^T)$
12: Update: $Y_m = Y_{m-1} - (\alpha_{m-1} t_{m-1} q_{m-1}^T)$
```

The regression parameters: $\hat{\beta}_{\text{pls}} = W(P^T W)^{-1} A Q^T$.
After first component is calculated, X in steps 2, 4, and 8 has to be replaced by its residual. Also, Y in steps 5 and 6 has to be replaced by its residual, $\varepsilon_0 = X$, $F_0 = Y$.

The second version of the NIPALS algorithm for PLS2

This algorithm (6) is given (Höskuldsson, 1988). In this algorithm Y-loadings, Q is normalised, but X-loadings, P is not. The inner relation, α, is not equal to one. Also, the X-weight vectors, w_m form an orthonormal set, and the X-score vectors, t_m are orthogonal to each other.

X is an $(n \times p)$ data matrix, Y is an $(n \times q)$ matrix of response variables, X and Y are mean centred.

t_m will be the scores vector for X
p_m will be the loadings vector for X
w_m will be the weights vector for X
u_m will be the scores vector for Y
q_m will be the loadings vector for Y
α_m will be the inner relation, which is a regression of u_m on t_m with no intercept.

Initialisation: Set $X_1 = X$, $Y_1 = Y$, and m=1, where $m = 1, 2, \ldots, M$ and $M \leq \min(n-1, p)$.

Algorithm 6 The second version of the NIPALS algorithm for PLS2

```
Take $u_m$ vector is set to a column in $Y$, $u_m = Y_1$
$w_m = X_m^T u_m / (u_m^T u_m)$
$w_m = w_m / \sqrt{w_m^T w_m}$        (normalisation)
$t_m = X_m w_m$
$q_m = Y_m^T t_m / (t_m^T t_m)$
$q_m = q_m / \sqrt{q_m^T q_m}$        (normalisation)
$u_m = Y_m q_m$
Check for convergence by comparing between current $t_m$ with the previous $t_m$. If they are the same, stop iterations, then got to step 9, otherwise go to step 2.
$p_m = X_m^T t_m / (t_m^T t_m)$
The inner relation: $\alpha_m = u_m^T t_m / (t_m^T t_m)$
Save: X-loadings: $p_m$, X-weights: $w_m$, X-scores: $t_m$, Y-loadings: $q_m$, Y-scores: $u_m$, The inner relation: $\alpha_m$
Update: $X_m = X_{m-1} - (t_{m-1} p_{m-1}^T)$
Update: $Y_m = Y_{m-1} - (\alpha_{m-1} t_{m-1} q_{m-1}^T)$
```

The regression parameters: $\hat{\beta}_{pls} = W(P^T W)^{-1} A Q^T$.
After first component is calculated, X in steps 2, 4, and 9 has to be replaced by its residual. Also, Y in steps 5 and 7 has to be replaced by its residual, $\varepsilon_0 = X$, $F_0 = Y$

The third version of the NIPALS algorithm for PLS2

This algorithm (7) is given in (Geladi & Kowalski, 1986). In this algorithm X-loadings, P and Y-loadings, Q are normalised. The inner relation, α_m, is not equal to one. Also, the X-weight vectors, w_m are not form an orthonormal set, but they are orthogonal, and the X-score vectors, t_m are orthogonal to each other.

X is an $(n \times p)$ data matrix, Y is an $(n \times q)$ matrix of response variables, X and Y are mean centred.
t_m will be the scores vector for X
p_m will be the loadings vector for X

w_m will be the weights vector for X
u_m will be the scores vector for Y
q_m will be the loadings vector for Y
α_m will be the inner relation, which is a regression of u_m on t_m with no intercept. Initialisation: Set $X_1 = X$, $Y_1 = Y$, and m=1, where $m = 1, 2, \ldots, M$ and $M \leq \min(n-1, p)$.

Algorithm 7 The third version of the NIPALS algorithm for PLS2

```
1: Take $u_m$ vector is set to a column in $Y$, $u_m = Y_1$
2: $w_m = X_m^T u_m/(u_m^T u_m)$
3: $w_m = w_m/\sqrt{w_m^T w_m}$        (normalisation)
4: $t_m = X_m w_m$
5: $q_m = Y_m^T t_m/(t_m^T t_m)$
6: $q_m = q_m/\sqrt{q_m^T q_m}$        (normalisation)
7: $u_m = Y_m q_m$
8: Check for convergence by comparing between current $t_m$ with the previous $t_m$. If
   they are the same, stop iterations, then got to step 9, otherwise go to step 2.
9: $p_m = X_m^T t_m/(t_m^T t_m)$
10: $p_m = p_m/\sqrt{p_m^T p_m}$        (normalisation)
11: $t_m = t_m(\sqrt{p_m^T p_m})$
12: $w_m = w_m(\sqrt{p_m^T p_m})$
13: The inner relation: $\alpha_m = u_m^T t_m/(t_m^T t_m)$
14: Save: X-loadings: $p_m$, X-weights: $w_m$, X-scores: $t_m$, Y-loadings: $q_m$, Y-scores:
    $u_m$, The inner relation: $\alpha_m$
15: Update: $X_m = X_{m-1} - (t_{m-1} p_{m-1}^T)$
16: Update: $Y_m = Y_{m-1} - (\alpha_{m-1} t_{m-1} q_{m-1}^T)$
```

The regression parameters: $\hat{\beta}_{pls} = W(P^T W)^{-1} A Q^T$.
After first component is calculated, X in steps 2, 4, and 9 has to be replaced by its residual. Also, Y in steps 5 and 7 has to be replaced by its residual, $\varepsilon_0 = X$, $F_0 = Y$

Appendix B

Theoretical proof of parameters estimation for all three versions of the NIPALS algorithm for PLS2

In the first version of the second NIPALS algorithm where X-loadings (P) and Y-loadings (Q) are not normalised as given in S. Wold (1984). In the second version of the second NIPALS algorithm algorithm where X-loadings (P) are normalised, but Y-loadings (Q) are not as given in Hoskuldsson (1988). In the third version of the second NIPALS algorithm where X-loadings (P) and Y-loadings (Q) are normalised as given in Geladi and Kowalski (1986).

$$\hat{\beta}^{(1)}_{\text{pls2}} = W^{(1)}(P^{(1)^T}W^{(1)})^{-1}A^{(1)}Q^{(1)^T},$$

$$\hat{\beta}^{(2)}_{\text{pls2}} = W^{(2)}(P^{(2)^T}W^{(2)})^{-1}A^{(2)}Q^{(2)^T}.$$

From the above NIPALS algorithms, we can see that $W^{(1)} = W^{(2)}$, which means that the W matrix in the first and the second algorithms are the same. Also, $P^{(1)^T} = P^{(2)^T}$, which means that the P matrix in the first and the second algorithms are the same. However, $A^{(1)} \neq A^{(2)}$, which means that the A matrix in the first and the second algorithms are not the same. And, $Q^{(1)} \neq Q^{(2)}$, which means that the Q matrix in the first and the second algorithms are not the same. Thus, we need to show that Equation

(B.1) is true for the first component, then we can prove that $\hat{\beta}^{(1)}_{\text{pls2}} = \hat{\beta}^{(2)}_{\text{pls2}}$.

$$\alpha^{(1)} q^{(1)^T} = \alpha^{(2)} q^{(2)^T}. \tag{B.1}$$

From the second algorithm step 6, we can write

$$q^{(2)^T} = \frac{q^{(1)^T}}{\sqrt{q^{(1)^1} q^{(1)}}}. \tag{B.2}$$

By combining Equation (B.1) and Equation (B.2), we have

$$\alpha^{(2)} = \alpha^{(1)} \sqrt{q^{(1)^T} q^{(1)}}.$$

Also, using the values of $a^{(2)}$ and $q^{(2)^T}$, we have

$$\alpha^{(2)} q^{(2)^T} = \alpha^{(1)} \sqrt{q^{(1)^T} q^{(1)}} \frac{q^{(1)^T}}{\sqrt{q^{(1)^T} q^{(1)}}}.$$

Some terms will be canceled out. Therefore, we will have that:

$$\alpha^{(2)} q^{(2)^T} = \alpha^{(1)} q^{(1)^T}. \tag{B.3}$$

Nevertheless, $\hat{\beta}^{(1)}_{\text{pls2}} = \hat{\beta}^{(2)}_{\text{pls2}}$ since $\alpha^{(1)} q^{(1)^T} = \alpha^{(2)} q^{(2)T}$ as shown in Equation (B.3) for one component, and for m components.

$$\hat{\beta}^{(2)}_{\text{pls2}} = W^{(2)} (P^{(2)^T} W^{(2)})^{-1} A^{(2)} Q^{(2)^T},$$

$$\hat{\beta}^{(3)}_{\text{pls2}} = W^{(3)} (P^{(3)^T} W^{(3)})^{-1} A^{(3)} Q^{(3)^T}.$$

From the above NIPALS algorithms, we can see that $W^{(2)} \neq W^{(3)}$, which means that the W matrix in the second and the third algorithms are not the same. Also, $P^{(2)^T} \neq P^{(3)^T}$, which means that the P matrix in the second and the third algorithms are the not same. However, $A^{(2)} \neq A^{(3)}$, which means that the A matrix in the second and the third algorithm are not the same. And, $Q^{(2)} = Q^{(3)}$, which means that the Q matrix in the second and the third algorithm are the same. Thus, if we can show that $w^{(2)} (p^{(2)^T} w^{(2)})^{-1} \alpha^{(2)} = w^{(3)} (p^{(3)^T} w^{(3)})^{-1} \alpha^{(3)}$ for one component, we can prove that $\hat{\beta}^{(2)}_{\text{pls2}} = \hat{\beta}^{(3)}_{\text{pls2}}$.

To show that $w^{(2)}(p^{(2)^T}w^{(2)})^{-1}\alpha^{(2)} = w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)}$, we need to write $w^{(3)}$ in terms of $w^{(2)}$, and the same for $p^{(3)}$ and $\alpha^{(3)}$ for one component.

$$w^{(2)} = \frac{X^T u^{(2)}}{\sqrt{(X^T u^{(2)})^T (X^T u^{(2)})}}.$$

Since $u^{(3)} = u^{(2)}$, we can write $w^{(3)}$ as

$$w^{(3)} = X^T u_2 \sqrt{p_2^T p_2}.$$

We can write that $w^{(3)}$ as an equation of $w^{(2)}$ as in Equation (B.4)

$$w^{(3)} = w^{(2)} \sqrt{(X^T u^{(2)})^T (X^T u^{(2)})} \sqrt{p^{(2)^T} p^{(2)}}. \tag{B.4}$$

Also, we can write $p^{(3)}$ as an equation of $p^{(2)}$ as in Equation (B.5)

$$p^{(3)^T} = \frac{p^{(2)^T}}{\sqrt{p^{(2)^T} p^{(2)}}}. \tag{B.5}$$

Also, we can write $t^{(3)}$ as an equation of $t^{(2)}$ in order to calculate $\alpha^{(3)}$ as in Equation (B.6)

$$t^{(3)} = t^{(2)} \sqrt{p^{(2)^T} p^{(2)}}. \tag{B.6}$$

Then, $\alpha^{(3)}$ can be written as an equation of $\alpha^{(2)}$ as in Equation (B.7)

$$\alpha^{(3)} = \frac{\alpha^{(2)}}{\sqrt{p^{(2)^T} p^{(2)}}}. \tag{B.7}$$

By combining the equations (B.4), (B.5), and (B.7) and after some calculations, we have:

$$w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)} = w^{(2)} \sqrt{(X^T u^{(2)})^T (X^T u^{(2)})} \sqrt{p^{(2)^T} p^{(2)}}$$
$$\left(\frac{p^{(2)^T}}{\sqrt{p^{(2)^T} p^{(2)}}} w^{(2)} \sqrt{(X^T u^{(2)})^T (X^T u^{(2)})} \sqrt{p^{(2)^T} p^{(2)}} \right)^{-1} \frac{\alpha^{(2)}}{\sqrt{p^{(2)^T} p^{(2)}}}. \tag{B.8}$$

Some terms will be canceled out. Therefore, we will have that:

$$w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)} = w^{(2)}(p^{(2)^T}w^{(2)})^{-1}\alpha^{(2)}. \tag{B.9}$$

Nonetheless, $\hat{\beta}^{(2)}_{\text{pls2}} = \hat{\beta}^{(3)}_{\text{pls2}}$ since $w^{(2)}(p^{(2)^T}w^{(2)})^{-1}\alpha^{(2)} = w^{(3)}(p^{(3)^T}w^{(3)})^{-1}\alpha^{(3)}$ as shown in Equation (B.9) for one component and for m components.

Since $\hat{\beta}^{(1)}_{\text{pls2}} = \hat{\beta}^{(2)}_{\text{pls2}}$ and $\hat{\beta}^{(2)}_{\text{pls2}} = \hat{\beta}^{(3)}_{\text{pls2}}$, then $\hat{\beta}^{(1)}_{\text{pls2}} = \hat{\beta}^{(2)}_{\text{pls2}} = \hat{\beta}^{(3)}_{\text{pls2}}$. Therefore, all $\hat{\beta}_{\text{pls2}}$ for all three versions of the NIPALS algorithm for multivariate responses (for PLS2) are the same even if they are different in terms of normalisations or not.

References

ABDI, H. (2010). Partial least squares regression and projection on latent structure regression (pls regression). *Wiley Interdisciplinary Reviews: Computational Statistics*, **2**, 97–106. 19, 21

AJANA, S., ACAR, N., BRETILLON, L., HEJBLUM, B.P., JACQMIN-GADDA, H. & DELCOURT, C. (2019). Benefits of dimension reduction in penalized regression methods for high-dimensional grouped data: a case study in low sample size. *Bioinformatics*. 6, 10

ALAIYA, A.A., FRANZÉN, B., HAGMAN, A., SILFVERSWÄRD, C., MOBERGER, B., LINDER, S. & AUER, G. (2000). Classification of human ovarian tumors using multivariate data analysis of polypeptide expression patterns. *International Journal of Cancer*, **86**, 731–736. 45

ALBERTS, B. (2008). Molecular biology of the cell. 8

ALGAMAL, Z.Y. & LEE, M.H. (2015). Regularized logistic regression with adjusted adaptive elastic net for gene selection in high dimensional cancer classification. *Computers in Biology and Medicine*, **67**, 136–145. 4

ANDERSSON, M. (2009). A comparison of nine pls1 algorithms. *Journal of Chemometrics*, **23**, 518–529. 22, 23

BARKER, M. & RAYENS, W. (2003). Partial least squares for discrimination. *Journal of Chemometrics*, **17**, 166–173. 16, 46, 47

BENJAMINI, Y. & HOCHBERG, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, **57**, 289–300. 4

BLACKBURN, M. (2005). NIR of corn samples for standardization benchmarking. http://eigenvector.com/data/Corn/index.html. 7

BOULESTEIX, A.L. (2004). PLS dimension reduction for classification with microarray data. *Statistical Applications in Genetics and Molecular Biology*, **3**, 1–30. 5, 15, 46

BØVELSTAD, H.M., NYGÅRD, S., STØRVOLD, H.L., ALDRIN, M., BORGAN, Ø., FRIGESSI, A. & LINGJÆRDE, O.C. (2007). Predicting survival from microarray dataa comparative study. *Bioinformatics*, **23**, 2080–2087. 137

BREIMAN, L. *et al.* (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics*, **24**, 2350–2383. 4

BUTLER, N.A. & DENHAM, M.C. (2000). The peculiar shrinkage properties of partial least squares regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **62**, 585–593. 6, 10, 49, 53, 54, 72, 75, 76

CHIAROMONTE, F. & MARTINELLI, J. (2002). Dimension reduction strategies for analyzing global gene expression data with a response. *Mathematical Biosciences*, **176**, 123–144. 5

CHIN, W.W. & FRYE, T. (2003). PLS graph–version 3.0. *Soft Modeling Inc. URL http://www. plsgraph. com*. 21

CHO, J.H., LEE, D., PARK, J.H., KIM, K. & LEE, I.B. (2002). Optimal approach for classification of acute leukemia subtypes based on gene expression data. *Biotechnology Progress*, **18**, 847–854. 45

CHUN, H. & KELEŞ, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**, 3–25. 6, 10, 123, 124

CHUNG, D. & KELES, S. (2010). Sparse partial least squares classification for high dimensional data. *Statistical Applications in Genetics and Molecular Biology*, **9**. 10, 124, 125, 171

COLOMBANI, C., CROISEAU, P., FRITZ, S., GUILLAUME, F., LEGARRA, A., DUCROCQ, V. & ROBERT-GRANIÉ, C. (2012). A comparison of partial least squares (pls) and sparse pls regressions in genomic selection in french dairy cattle. *Journal of Dairy Science*, **95**, 2120–2131. 6

DE JONG, S. (1995). PLS shrinks. *Journal of Chemometrics*, **9**, 323–326. 5, 58

DENHAM, M.C. (1997). Prediction intervals in partial least squares. *Journal of Chemometrics*, **11**, 39–52. 45

DETTLING, M. & BÜHLMANN, P. (2003). Boosting for tumor classification with gene expression data. *Bioinformatics*, **19**, 1061–1069. 3

DRAPER, N. & SMITH, H. (1981). Applied regression analysis, 709 pp. 17

DURIF, G., MODOLO, L., MICHAELSSON, J., MOLD, J.E., LAMBERT-LACROIX, S. & PICARD, F. (2017). High dimensional classification with combined adaptive sparse pls and logistic regression. *Bioinformatics*, **34**, 485–493. 6

EDWARDS, D. (2012). *Introduction to Graphical Modelling*. Springer Science & Business Media. 195

F. HAIR JR, J., SARSTEDT, M., HOPKINS, L. & G. KUPPELWIESER, V. (2014). Partial least squares structural equation modeling (pls-sem) an emerging tool in business research. *European Business Review*, **26**, 106–121. 193

FORT, G. & LAMBERT-LACROIX, S. (2005). Classification using partial least squares with penalized logistic regression. *Bioinformatics*, **21**, 1104–1111. 5, 15, 46

FRANK, L.E. & FRIEDMAN, J.H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, **35**, 109–135. 2, 6, 10, 13, 18, 47, 48, 49, 53, 55, 63

FU, G.H., XU, Q.S., LI, H.D., CAO, D.S. & LIANG, Y.Z. (2011). Elastic net grouping variable selection combined with partial least squares regression (en-plsr) for the analysis of strongly multi-collinear spectroscopic data. *Applied Spectroscopy*, **65**, 402–408. 7, 8, 10

GARTHWAITE, P.H. (1994). An interpretation of partial least squares. *Journal of the American Statistical Association*, **89**, 122–127. 6, 21

GELADI, P. & KOWALSKI, B.R. (1986). Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, **185**, 1–17. 2, 5, 15, 17, 18, 19, 20, 27, 28, 29, 33, 51, 217, 220

GHOSH, D. (2003). Penalized discriminant methods for the classification of tumors from gene expression data. *Biometrics*, **59**, 992–1000. 4

GOEMAN, J.J. (2010). L1 penalized estimation in the cox proportional hazards model. *Biometrical Journal*, **52**, 70–84. 11, 14, 126, 131, 132

GOLUB, T.R., SLONIM, D.K., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J.P., COLLER, H., LOH, M.L., DOWNING, J.R., CALIGIURI, M.A. *et al.* (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537. 45

GOUTIS, C. *et al.* (1996). Partial least squares algorithm yields shrinkage estimators. *The Annals of Statistics*, **24**, 816–824. 58

GUNST, R.F. & MASON, R.L. (1980). *Regression Analysis and its Application: A data-oriented Approach*, vol. 34. CRC Press. 17

GUSNANTO, A. & PAWITAN, Y. (2015). Sparse alternatives to ridge regression: a random effects approach. *Journal of Applied Statistics*, **42**, 12–26. 7, 123, 128, 129, 136

GUSNANTO, A., WOOD, H.M., PAWITAN, Y., RABBITTS, P. & BERRI, S. (2011). Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data. *Bioinformatics*, **28**, 40–47. 9

GUSNANTO, A., TAYLOR, C.C., NAFISAH, I., WOOD, H.M., RABBITTS, P. & BERRI, S. (2014). Estimating optimal window size for analysis of low-coverage next-generation sequence data. *Bioinformatics*, **30**, 1823–1829. 8

GUSNANTO, A., TCHERVENIAKOV, P., SHUWEIHDI, F., SAMMAN, M., RABBITTS, P. & WOOD, H.M. (2015). Stratifying tumour subtypes based on copy number alteration profiles using next-generation sequence data. *Bioinformatics*, **31**, 2713–2720. 9, 47, 134

HEDENFALK, I., DUGGAN, D., CHEN, Y., RADMACHER, M., BITTNER, M., SIMON, R., MELTZER, P., GUSTERSON, B., ESTELLER, M., RAFFELD, M. *et al.* (2001). Gene-expression profiles in hereditary breast cancer. *New England Journal of Medicine*, **344**, 539–548. 3

HELLAND, I.S. (1988). On the structure of partial least squares regression. *Communications in Statistics-Simulation and Computation*, **17**, 581–607. 23, 49

HELLAND, I.S. (2001). Some theoretical aspects of partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, **58**, 97–107. 23, 24, 195

HOERL, A.E. & KENNARD, R.W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67. 48

HÖSKULDSSON, A. (1988). PLS regression methods. *Journal of Chemometrics*, 211–228. 2, 3, 5, 28, 31, 219

HUANG, J., GUSNANTO, A., O'SULLIVAN, K., STAAF, J., BORG, Å. & PAWITAN, Y. (2007). Robust smooth segmentation approach for array cgh data analysis. *Bioinformatics*, **23**, 2463–2469. 9

HUANG, J., SALIM, A., LEI, K., O'SULLIVAN, K. & PAWITAN, Y. (2009). Classification of array cgh data using smoothed logistic regression model. *Statistics in Medicine*, **28**, 3798–3810. 7, 10, 124, 127, 130

HUANG, X. & PAN, W. (2003). Linear regression and two-class classification with gene expression data. *Bioinformatics*, **19**, 2072–2078. 45

HUANG, X., PAN, W., GRINDLE, S., HAN, X., CHEN, Y., PARK, S.J., MILLER, L.W. & HALL, J. (2005). A comparative study of discriminating human heart failure etiology using gene expression profiles. *BMC Bioinformatics*, **6**, 205. 4, 45

HULLAND, J. (1999). Use of partial least squares (pls) in strategic management research: a review of four recent studies. *Strategic Management Journal*, **20**, 195–204. 2

JÖRESKOG, K.G. (1978). Structural analysis of covariance and correlation matrices. *Psychometrika*, **43**, 443–477. 193

KALINA, J. (2014). Classification methods for high-dimensional genetic data. *Biocybernetics and Biomedical Engineering*, **34**, 10–18. 4

KRÄMER, N. (2007). An overview on the shrinkage properties of partial least squares regression. *Computational Statistics*, **22**, 249–273. 76

LÊ CAO, K.A., ROSSOUW, D., ROBERT-GRANIÉ, C. & BESSE, P. (2008). A sparse PLS for variable selection when integrating omics data. *Statistical Applications in Genetics and Molecular Biology*, **7**. 10

LEE, D., LEE, W., LEE, Y. & PAWITAN, Y. (2011). Sparse partial least-squares regression and its applications to high-throughput data analysis. *Chemometrics and Intelligent Laboratory Systems*, **109**, 1–8. xix, 5, 6, 7, 8, 9, 10, 13, 14, 24, 123, 124, 125, 126, 127, 129, 134, 135, 137, 138, 158, 161, 168, 170, 171, 190

LEE, D., LEE, Y., PAWITAN, Y. & LEE, W. (2013). Sparse partial least-squares regression for high-throughput survival data analysis. *Statistics in Medicine*, **32**, 5340–5352. 10, 123, 216

LEE, Y. & NELDER, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **55**, 139–185. 126

LEE, Y. & OH, H.S. (2009). *Random-effect Models for Variable Selection*. Department of Statistics, Stanford University. 125, 126

LEE, Y., NELDER, J.A. & PAWITAN, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall/CRC. 126

LI, H., LIANG, Y., XU, Q. & CAO, D. (2009). Key wavelengths screening using competitive adaptive reweighted sampling method for multivariate calibration. *Analytica Chimica Acta*, **648**, 77–84. 8

LINGJAERDE, O.C. & CHRISTOPHERSEN, N. (2000). Shrinkage structure of partial least squares. *Scandinavian Journal of Statistics*, **27**, 459–473. 6, 10, 13, 48, 49, 53, 54, 57, 62, 63, 67, 70, 72, 73, 75, 76, 78

MAN, M.Z., DYSON, G., JOHNSON, K. & LIAO, B. (2004). Evaluating methods for classifying expression data. *Journal of Biopharmaceutical Statistics*, **14**, 1065–1084. 46

MANNE, R. (1987). Analysis of two partial-least-squares algorithms for multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, **2**, 187–197. 28

MARDIA, K.V., KENT, J.T. & BIBBY, J.M. (1979). *Multivariate Analysis*. Academic press. 17

MASSY, W.F. (1965). Principal components regression in exploratory statistical research. *Journal of the American Statistical Association*, **60**, 234–256. 2, 5, 48

MONECKE, A. & LEISCH, F. (2012). sempls: Structural equation modeling using partial least squares. 193

MUSUMARRA, G., BARRESI, V., CONDORELLI, D., FORTUNA, C. & SCIRE, S. (2004). Potentialities of multivariate approaches in genome-based cancer research: identification of candidate genes for new diagnostics by pls discriminant analysis. *Journal of Chemometrics*, **18**, 125–132. 46

NAES, T. & MARTENS, H. (1985). Comparison of prediction methods for multicollinear data. *Communications in Statistics-Simulation and Computation*, **14**, 545–576. 49

NGUYEN, D.V. & ROCKE, D.M. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, **18**, 39–50. 2, 5, 15, 46

NYGÅRD, S., BORGAN, Ø., LINGJÆRDE, O.C. & STØRVOLD, H.L. (2008). Partial least squares cox regression for genome-wide data. *Lifetime Data Analysis*, **14**, 179–195. 137

OLSHEN, A.B., VENKATRAMAN, E., LUCITO, R. & WIGLER, M. (2004). Circular binary segmentation for the analysis of array-based dna copy number data. *Biostatistics*, **5**, 557–572. 9

PAWITAN, Y. (2001). *In all Likelihood: Statistical Modelling and Inference using Likelihood*. Oxford University Press. 127, 128

PÉREZ-ENCISO, M. & TENENHAUS, M. (2003). Prediction of clinical outcome with microarray data: a partial least squares discriminant analysis (pls-da) approach. *Human Genetics*, **112**, 581–592. 5, 45

PHATAK, A. & DE HOOG, F. (2002). Exploiting the connection between pls, lanczos methods and conjugate gradients: alternative proofs of some properties of pls. *Journal of Chemometrics: A Journal of the Chemometrics Society*, **16**, 361–367. 45, 53

PHATAK, A., REILLY, P. & PENLIDIS, A. (1993). An approach to interval estimation in partial least squares regression. *Analytica Chimica Acta*, **277**, 495–501. 44, 45

RISVIK, H. (2007). Principal component analysis (pca) & nipals algorithm. 19

ROSIPAL, R. & KRÄMER, N. (2006). Overview and recent advances in partial least squares. In *Subspace, latent structure and feature selection*, 34–51, Springer. 6, 48, 75

SAAD, Y. (1992). *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press. 78

SHENDURE, J. & JI, H. (2008). Next-generation dna sequencing. *Nature Biotechnology*, **26**, 1135. 8

SJÖSTRÖM, M., WOLD, S., LINDBERG, W., PERSSON, J.Å. & MARTENS, H. (1983). A multivariate calibration problem in analytical chemistry solved by partial least-squares models in latent variables. *Analytica Chimica Acta*, **150**, 61–70. 28

STONE, M. & BROOKS, R.J. (1990). Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 237–269. 17, 18, 21, 33, 49

SUTTON, M., THIÉBAUT, R. & LIQUET, B. (2018). Sparse partial least squares with group and subgroup structure. *Statistics in Medicine*, **37**, 3338–3356. 6, 10

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288. 1, 4, 48

TIBSHIRANI, R., HASTIE, T., NARASIMHAN, B. & CHU, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, **99**, 6567–6572. 46

TIBSHIRANI, R., SAUNDERS, M., ROSSET, S., ZHU, J. & KNIGHT, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**, 91–108. 130

WHITTAKER, J. (2009). *Graphical models in applied multivariate statistics*. Wiley Publishing. 194

WOLD, H. (1966). Estimation of principal components and related models by iterative least squares. *Multivariate Analysis*, 391–420. 2, 21, 193

WOLD, H. (1975). Soft modelling by latent variables: the non-linear iterative partial least squares (nipals) approach. *Journal of Applied Probability*, **12**, 117–142. 2, 48

WOLD, S., MARTENS, H. & WOLD, H. (1983). The multivariate calibration problem in chemistry solved by the pls method. In *Matrix pencils*, 286–293, Springer. 2, 21, 25

WOLD, S., RUHE, A., WOLD, H. & DUNN, W., III (1984). The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, **5**, 735–743. 21, 23, 28, 33, 218

WOLD, S., JONSSON, J., SJÖRSTRÖM, M., SANDBERG, M. & RÄNNAR, S. (1993). Dna and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures. *Analytica Chimica Acta*, **277**, 239–253. 5

WOLD, S., SJÖSTRÖM, M. & ERIKSSON, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, **58**, 109–130. 19, 32

WORSLEY, K.J. (1997). An overview and some new developments in the statistical analysis of pet and fmri data. *Human Brain Mapping*, **5**, 254–258. 2

ZOU, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**, 1418–1429. 6

ZOU, H. & HASTIE, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (statistical methodology)*, **67**, 301–320. 4

ZOU, H. & ZHANG, H.H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Annals of Statistics*, **37**, 1733. 4

ZOU, H., HASTIE, T. & TIBSHIRANI, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, **15**, 265–286. 6, 130