Sparsity in Partial Least Squares Regression Models

Emeka Calistus Uzochukwu

School of Mathematics

University of Leeds

A thesis submitted in accordance with the requirements for the degree of

Doctor of Philosophy

November 2020

Acknowledgements

Many thanks to my supervisors Prof. Jeanine Houwing-Duistermaat and Prof. John T. Kent, I would not have asked for better supervisors. Thank you for your advice surrounding my research, your support and insight have enabled me to develop as a researcher. I would also like to thank the Leeds Anniversary Research Scholarship for funding my PhD. Finally, I am grateful to my family, you have supported me every step of the way.

Abstract

Data sets with multiple responses and multiple predictor variables are increasingly common. It is known that such data sets often exhibit near multicollinearity and the traditional ordinary least squares (OLS) regression method do not perform well in such a setting because the mean square error of the OLS regression coefficients will be large and prediction performance will be poor. This drawback of OLS is often handled by using well-known dimension reduction methods; the focus in this thesis is Partial Least Squares (PLS).

The following contributions are made in the thesis: (a) Introduce relevant components (RC) models characterized by restrictions on the joint covariance matrix of the response and predictor variables, and show that the univariate (single-response) version of the RC model can be represented as a Krylov model. These representations will shed more light into the understanding of PLS. Also, PLS algorithms are reviewed and presented as estimators of the RC models. (b) Unify various multiple-response regression models under the framework of the RC models, and review some multiple-response PLS methods. In addition, simulation studies are carried out to compare the prediction performance of multivariate PLS (PLS2) methods. (c) Propose novel sparse multivariate PLS (SPLS2) methods for parameter estimation and variable selection, which offers more flexibility compared to known SPLS2 methods, and compare the novel methods against methods in the literature in terms of prediction performance and accuracy in variable selection. (d) Apply the PLS regression methods to a proteomics data set to predict the severity of systemic sclerosis and identify candidate markers. Furthermore, compare the PLS, SPLS and OLS methods with regard to predictive ability using the proteomics data.

Contents

List of Figures

List of Tables

List of Abbreviations

CV	Cross-validation
CV-RMSE	Cross-validated root-mean-square error
lasso	Least Absolute Shrinkage and Selection Operator
MLE	Maximum likelihood estimation
MLR	Multiple Linear Regression
MMLR	Multivariate multiple linear regression
MSE	Mean Squared Error
MVUE	Minimum-Variance Unbiased Estimator
RC	Relevant Components Model
RMSE	Root-mean-square error

Chapter 1

Overview

1.1 Introduction

Large data sets with multiple response variables and multiple predictor variables are common, and such datasets are usually multicollinearity when the number of observations (n) is close to the number of predictor variables (p). Multicollinearity refers to the case when the covariance matrix of the predictor variables has a large condition number (i.e, the ratio of the largest and smallest eigenvalues of the covariance matrix is large). We have near multicollinearity when the covariance matrix of the predictor variables is almost singular. And perfect multicollinearity when the covariance matrix of the predictor variables is singular. For instance, in proteomics studies it is common to measure many proteomics variables (p) on several subjects (n), where $n \approx p$. It is common knowledge that dimension reduction methods often have better estimation and prediction performance over tradition ordinary least square (OLS) method in such a context (Frank and Friedman, 1993). Moreover, when p is large interpretation of OLS estimates become difficult (Sirimongkolkasem and Drikvandi, 2019). To handle the problem of estimation, prediction and interpretation dimension reduction and penalization methods can be used. These alternative methods including ridge, least absolute shrinkage and selection operator (lasso), random forest, principal components regression, and partial least squares regression, can produce biased regression estimates but they may improve prediction performance because the regression coefficients could have smaller variances compared to coefficients from OLS.

The ridge regression (Hoerl and Kennard, 1970) method performs estimation by penalizing the size of the regression coefficients, effectively shrinking them towards zero. Shrinkage means that a limit is placed on the size of the regression estimates, without this shrinkage (penalty) predictors with high correlation would give unstable estimates with large variances (Varmuza and Filzmoser, 2016). However, it does not improve interpretation when p is large (Friedman et al., 2001). An alternative method which can produce stable estimates and improve interpretability is lasso (Tibshirani, 1996). The lasso gives stable regression estimates when $n \approx p$ by also shrinking some regression coefficients to exactly zero. Unlike the ridge regression method which shrinks regression coefficients towards zero, the lasso shrinks coefficients to exactly zero, thereby easing model interpretability. However, the lasso can sometimes have poor selection accuracy when predictor variables are perfectly multicollinear (Zou, 2006; Zou and Hastie, 2005). An improvement on both the lasso and ridge regression methods is the elastic net proposed by Zou and Hastie (2005), which combines the advantages of the lasso and ridge regression methods when estimating the parameters of a model. In other words, the elastic net produces stable estimates by shrinks regression coefficients: it enables model interpretability and can sometimes have better selection accuracy compared to the lasso.

Another technique is the random forest (Breiman, 2001), which is made up of a collection of decision trees (Breiman et al., 1984) and mostly used for classification purposes, but sometimes applied in a regression problem when prediction and interpretation are of interest (Segal, 2004). A random forest contains hundreds of decision trees, and individual decision trees are built from bootstrap samples of the dataset. In random forest regression, the prediction error is calculated as the average prediction error over several predictions, and variables which decrease the prediction error are regarded as important variables (Acharjee et al., 2013). The random forest may ease interpretation but may not have better prediction performance compared to the lasso, ridge and elastic net because variables are remove/added in a discrete manner. In addition, random forest can sometimes be improved by the lasso method (Wang and Wang, 2021).

The alternative to penalization is dimension reduction, which handles multicollinearity through orthogonal transformation of the variables. For example, principle components regression (PCR) and partial least squares (PLS) regression. The PCR (Jolliffe, 1982) methods identifies material information by taking successive linear combinations of the predictor variables. Important information is identified by finding the directions of maximum variation in the predictor variables. The linear combinations associated with directions of maximum variation are referred to as relevant components. Usually the number of relevant components $q << p$ and are used to approximate the original predictor variables; which is an advantage of PCR. The PCR can deal with near or perfect multicollinearity and improve prediction (Cook et al., 2008; Næs and Martens, 1988), however it finds relevant components in the predictor variables without taking into account the response variable, therefore may not capture the true relationship between the variables.

PLS (Wold, 1975) incorporates the response variables when finding relevant components using directions of maximum covariance between the response and predictor variable (Rännar et al., 1995), and handles multicollinearity by a decomposition of the joint covariance matrix of the response and predictor variables Cook et al. (2013).

These methods sometimes have similar prediction and estimation performances, however, in some cases PLS performs better. Compared to the penalization techniques mentioned above, which uses all the predictor variables for prediction, the number of variables (relevant components, latent variables) used by PLS for prediction is usually very small compared to the number of predictor variables; in the presence of multicollinearity. Sirimongkolkasem and Drikvandi (2019) showed that when data is multicollinear and not assumed sparse the lasso, ridge regression, and elastic net performs poorly in terms of prediction and parameter estimation compared to dimension reductions such as PCR, and by extension, PLS (because they both use linear combinations). Fearn (1983) used a real data example to show that when important information is associated with smaller eigenvalues ridge regression may not be suitable. Moreover, since PCR does not incorporate the response variables and uses the components with the largest eigenvalues it can perform poorly when relevant information is associated with the smaller eigenvalues. On the other hand, because PLS considers the response variables when finding relevant information it identifies relevant information even when they are associated with smaller eigenvalues (Cook et al., 2013). Moreover, PLS has better prediction performance compared to the above mentioned methods when important information are related to smaller eigenvalues (Almøy, 1996; Lee et al., 2018; ter Braak, 2009). In real data, important information may be related to smaller eigenvalues and the sparsity assumption may not be of interest, in which case PLS will be the preferred method.

We investigate PLS further and the works of (Helland, 1988, 1990) and Cook et al. (2013) who have explored the theoretical and statistical properties of PLS regression will form the foundation of the study.

1.2 Structure of the thesis

The thesis is organised in chapters as follows.

Chapter 2 introduces the notations and conventions used in the thesis. In addition, it describes linear regression, reviews the limitations of the maximum likelihood estimator for linear regression when predictor variables are nearly or perfectly multicollinear, and introduces alternative estimators. The alternative estimators discussed are lasso regression and orthonormal transformation of predictor variables, which is synonymous to PCR. Furthermore, is shows how these alternative methods handle regression coefficients with large variances.

Chapter 3 introduces a univariate (single-response) version of the relevant components (RC) model which has certain restrictions on the joint covariance matrix of the response and predictor variables. The RC model is an alternative model to the classical regression model discussed in Chapter 2, and provide an insight on how to handle multicollinearity. Furthermore, the RC model is further discussed under the Krylov model, which is associated with the linear independence of the Krylov matrix. The chapter show that the Krylov model is the same as the RC model with a different representation. Also, the chapter reviews various PLS1 algorithms for estimating the parameters of the RC model. The algorithms considered are the nonlinear iterative partial least squares algorithm, Gramm-Schmidt algorithm, and factor method.

Chapter 4 extends the discussion about univariate regression to multivariate (multiple-response) regression, and consolidates various multivariate PLS-type regression models under the framework of RC models, some of which are extensions of the univariate RC models discussed in Chapter 3. The models introduced also have specific restrictions on the joint covariance matrix of the response and predictor variables. Furthermore, the chapter reviews three multivariate-response partial least squares (PLS2) methods for estimating the parameters of different multivariate RC models. The PLS2 methods taken into consideration are the envelope method (EPLS) (Cook et al., 2013), statistically inspired modification of PLS (SIMPLS) (De Jong, 1993), and the expectation-maximization algorithm for PLS (EM-PLS) (el Bouhaddani et al., 2018). Furthermore, these methods are compared in terms of prediction performance using simulated data. The results of the simulation show that the EPLS method has better prediction performance compared to EM-PLS and SIMPLS when n is close to p. And EPLS and SIMPLS have similar performances when n larger than p. Also, the performance of the EM-PLS method depends on the model the data are drawn from.

Chapter 5 deals with the problem of interpretability associated with PLS, especially when the number of predictor variables is large, by introducing the notion of sparsity in PLS. This chapter gives a sparse relevant components (SRC) model which combines the assumptions of active predictor variables and dimension reduction in linear regression. Active predictor variables are predictor variable with nonzero correlation with the response variables. The SRC model also has specific restrictions on the joint covariance matrix of the response and predictor variables. With the assumption of active predictor variables, the interpretability problem related to PLS when p is large can be handled. The chapter also presented methods for estimating the parameters of the model. The methods considered are the enveloped-based sparse PLS (Zhu et al., 2020) and sparse PLS (Chun and Keleş, 2010) methods. In addition, a novel sparse PLS2 method is proposed for estimating the parameters of the model. The novel method is a two-stage technique that reduces the dimension of the predictor variables in the first stage and introduces sparsity in the second stage for selecting active predictor variables. Besides, the novel method is more flexible compared to other methods since it can perform both group and within-group variable selection. Furthermore, simulation studies are carried out to compare the methods in terms of prediction performance and variable selection accuracy. The results show that the methods have similar prediction performance when n is larger than p, and the novel method has better selection accuracy compared to other methods considered when within-group sparsity is of interest.

Chapter 6 applied various PLS methods to a large proteomics data to identify candidate biomarkers for systemic sclerosis. Identification of candidate biomarkers is similar to model interpretability. The data has 3 outcome variables, 451 proteomics variables and 408 patients. The outcome variables are known effects of scleroderma on the skin and lungs of patients, and the proteomics variables are different protein markers measured from the blood of patients. The predictive performance of the PLS methods discussed in previous chapters are compared using the proteomics data, the results show that the EPLS method has a slightly better predictive performance compared to SIMPLS and much better performance compared to other methods considered. Also, the sparse PLS2 methods discussed in previous chapters of the thesis were used to select candidate biomarkers.

Chapter 7 gives a summary of the main results of the thesis and provides future work. Also, the chapter discussed some limitation of the work.

The contributions of the thesis are as follows:

1 Multivariate and univariate PLS-type regression models are unified under the framework of relevant components (RC) models. These models are scattered in the literature and to our knowledge no author has consolidated these models. This unification will make is easier for statisticians to understand the idea behind PLS. Describing the models as restrictions on the joint covariance matrix of the response and predictor variables simplified the connection between different PLS-type models proposed in the literature.

2 We reviewed multivariate PLS methods, and compare the recently proposed multivariate likelihood-based envelope method (Cook et al., 2010), SIMPLS (De Jong, 1993) and EM-PLS (el Bouhaddani et al., 2018) in terms of prediction performance. To our knowledge, the prediction performance of the EM-PLS have not been considered in previous simulation studies (el Bouhad-

dani et al., 2018). The result show that the SIMPLS and envelope methods have better prediction performances compared to EM-PLS when the predictor not highly multicollinear.

3 We propose novel sparse multivariate PLS (SPLS2) methods for identifying active predictor variables. The proposed methods are more flexible compared to previous SPLS2 methods. The first method, modified envelope-based sparse PLS, method can selected few number of active predictor variables compared to other methods. The second method, two-stage sparse PLS, can perform both group and within-group sparsity, to our knowledge this feature has not seen in other sparse PLS methods proposed in the literature.

4 A large proteomics data set is analysed using various PLS methodology to predict the level of severity of systemic sclerosis and identify candidate marker. Moreover, the PLS methods are compared in terms of prediction performance. To our knowledge, these methods were not applied in previous analysis of the dataset.

Chapter 2

Background on linear regression

2.1 Introduction

The focus of this chapter is to introduce the notations and conventions used in the thesis, and to give a review linear regression and maximum likelihood estimator for linear regression when n is larger than p. Furthermore, the large p small n problem of regression coefficients is highlighted and estimators for handling this limitation are explored. The estimators considered are the lasso and a PCR-type technique for linear regression.

2.2 Notations and conventions

This thesis makes use of the following notations and conventions. Scalars will be denoted by italic lower case letters $(x, y, a, b, ...)$. Vectors will be denoted by bold italic lower case letters $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}, \boldsymbol{b}, ...)$ and all vectors are column vectors. Matrices will be denoted by bold italic upper case letters $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{A}, \boldsymbol{B}, ...)$. The dimension of matrices will be denoted by $n \times p$. For instance, $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ means that the matrix $\boldsymbol{X}$ has n rows and p columns, and the symbol $\mathbb{R}$ indicates the set real numbers. The dimension of a vector will be represented by $\mathbb{R}^p$, where p is the number of elements in the vector.

Inverse and transpose of a matrix will be denoted by superscripts -1 and T, respectively. For instance, $\boldsymbol{X}^{-1}$ and $\boldsymbol{X}^T$ are the inverse and transpose of the matrix $\boldsymbol{X}$, respectively. Also, $(\boldsymbol{X}^{-1})^T = \boldsymbol{X}^{-T}$ denotes the transpose of the inverse of $\boldsymbol{X}$.

Parameters will be denoted as follows: let y and $\boldsymbol{x}$ be random variables, then μ_y and $\sigma_{yy} = \sigma_y^2$ are the population mean and variance of y, respectively. Also, $\boldsymbol{\mu_x}$ and $\boldsymbol{\Sigma_{xx}}$ are the population mean vector and covariance matrix of $\boldsymbol{x}$, respectively. The parameter $\boldsymbol{\sigma_{xy}}$ is the population covariance vector between $\boldsymbol{x}$ and y.

Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ be a data matrix; n is the number of sample units and p the number of variables. The sample units (rows) will be indexed by i and the variable (columns) by j. The observation of the jth variable on the ith sample unit will be denoted by x_{ij}. The rows of $\boldsymbol{X}$ will be denoted by $\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, ..., \boldsymbol{x}_n^T$, where $\boldsymbol{x}_i = [x_{i1}, ..., x_{ip}]^T$ $(i = 1, ..., n)$ is a p-dimensional column vector for the ith sample unit, and $\boldsymbol{x}_j = [x_{1j}, ..., x_{nj}]^T$ $(j = 1, ..., p)$ is an n-dimensional vector for the jth variable.

A bar above a scalar or vector will represent the sample mean. For example, $\bar{y}$ and $\bar{\boldsymbol{x}}$ are the sample versions of μ_y and $\boldsymbol{\mu_x}$, respectively, and $\boldsymbol{S_{xx}}$ is the sample version of $\boldsymbol{\Sigma_{xx}}$.

The hat symbol will be used to denote an estimate of an unknown population parameter. For instance, $\hat{\beta}$ is the estimate of the unknown parameter β.

The L_1 norm of a p-dimensional vector, $\boldsymbol{b}$, is defined by

$$\|\boldsymbol{b}\|_1 = \|\boldsymbol{b}\| = \sum_{j=1}^{p} |b_j|,$$

where $|b_j|$ denotes the absolute value of b_j. The L_2 norm of $\boldsymbol{b}$ is defined by

$$\|\boldsymbol{b}\|_2 = \left(\sum_{j=1}^{p} b_j^2 \right)^{\frac{1}{2}}.$$

The Frobenius norm is defined as

$$\|\boldsymbol{B}\|_F = \mathrm{tr}(\boldsymbol{B}^T\boldsymbol{B})^{\frac{1}{2}} = \left(\sum_{j=1}^{p} \lambda_j \right)^{\frac{1}{2}},$$

where $\boldsymbol{B} \in \mathbb{R}^{n \times p}$, λ_j is the jth eigenvalue of $\boldsymbol{B}^T\boldsymbol{B} \in \mathbb{R}^{p \times p}$, and $\mathrm{tr}(\cdot)$ represents the trace.

The Hadamard product is used for the element-wise multiplication of two matrices of equal dimensions and defined as

$$\boldsymbol{A} \circ \boldsymbol{B}$$

A linear subspace spanned by $\{x_1, ..., x_p\}$ will be denoted by $\text{span}(x_1, ..., x_p)$.

The centering matrix $\boldsymbol{H}$ will be used for centering the data matrix and defined as

$$\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T,$$

where $\boldsymbol{I}_n$ is an $n \times n$ identity matrix and $\mathbf{1}$ is a column vector of n ones. The next section gives a review of linear regression.

2.3 Introduction to regression

Regression analysis is a statistical methodology commonly used for exploring the relationship between a response variable and a collection of predictor variables, predicting a new response variable from predictor variables, and evaluating the effect the change in the predictor variables have on the response. The response variable is commonly treated as random, but the predictor variables can be fixed or random. In Section 2.3.1 and 2.3.2, we review the fixed-predictor and random-predictor variables settings for multiple linear regression (MLR). We discuss some limitations associated with parameter estimation in MLR and reviewed alternative estimators.

2.3.1 Population linear regression model

Let $y \in \mathbb{R}$ denote a univariate response variable and $\boldsymbol{x} \in \mathbb{R}^p$ a vector of predictor variables, where p is the number of predictor variables. Classical linear regression model describes the conditional distribution of y given $\boldsymbol{x}$ and uses the population mean and variance to characterize the conditional distribution across the population. The regression model assumes that the mean and variance of the conditional distribution of y given $\boldsymbol{x}$ are given by

$$\begin{aligned} \mathrm{E}[y|\boldsymbol{x}] &= \beta_0 + \boldsymbol{\beta}^T\boldsymbol{x} \quad \text{and} \\ \mathrm{var}[y|\boldsymbol{x}] &= E\left[\epsilon^2\right] = \sigma^2, \end{aligned} \tag{2.1}$$

respectively, where $y|\boldsymbol{x}$ means y given $\boldsymbol{x}$, $\mathrm{E}[y|\boldsymbol{x}]$ is a linear function of $\boldsymbol{x}$ and $\mathrm{var}[y|\boldsymbol{x}]$ is constant in $\boldsymbol{x}$. The random variable $\epsilon = y - \mathrm{E}[y|\boldsymbol{x}]$ is the error which accounts for variation about $\mathrm{E}[y|\boldsymbol{x}]$ and ϵ is assumed to have mean zero and constant variance σ^2. The $\beta_0 \in \mathbb{R}$ is the intercept and $\boldsymbol{\beta} \in \mathbb{R}^p$ is a vector of slopes.

The classical linear regression model treats y as random and $\boldsymbol{x}$ as fixed or random. In this work, we will treat $\boldsymbol{x}$ and y as random variables from a joint multivariate normal distribution:

$$\begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} \sim \mathbf{N}_{p+1}\left[\begin{pmatrix} \boldsymbol{\mu_x} \\ \mu_y \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma_{xx}} & \boldsymbol{\sigma_{xy}} \\ \boldsymbol{\sigma}^T_{\boldsymbol{x}y} & \sigma_{yy} \end{pmatrix}\right], \tag{2.2}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{(p+1)\times(p+1)}$ is the joint covariance matrix of $\boldsymbol{x}$ and y, $\boldsymbol{\Sigma_{xx}} \in \mathbb{R}^{p\times p}$ is the population covariance matrix of $\boldsymbol{x}$, $\boldsymbol{\sigma_{xy}} \in \mathbb{R}^p$ the population covariance between $\boldsymbol{x}$ and y, and $\sigma_{yy} = \sigma_y^2 \in \mathbb{R}$ the population variance of y. Note that $\boldsymbol{\mu_x}$ and μ_y are the marginal means of $\boldsymbol{x}$ and y, respectively.

In this setting, it is possible to study how the variables vary together, however regression models are used for studying conditional distributions, therefore when y and $\boldsymbol{x}$ are random we could study either the conditional distribution of $\boldsymbol{x}$ given y or the conditional distribution of y given $\boldsymbol{x}$ (Cook and Weisberg, 2009). Suppose interest is on the regression of y on $\boldsymbol{x}$; we would require the conditional distribution of $y|\boldsymbol{x}$. The density function of the conditional probability of $y|\boldsymbol{x}$ is denoted by $f(y|\boldsymbol{x})$ and defined as $f(y|\boldsymbol{x}) = \frac{f(y,\boldsymbol{x})}{f(\boldsymbol{x})}$, where $f(y,\boldsymbol{x})$ is the joint density of $\boldsymbol{x}$ and y, and $f(\boldsymbol{x})$ the marginal density of $\boldsymbol{x}$. It can be shown that

$$f(y|\boldsymbol{x}) = \frac{1}{\sigma_{y|\boldsymbol{x}}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-\mu_{y|\boldsymbol{x}}}{\sigma_{y|\boldsymbol{x}}}\right)^2\right], \tag{2.3}$$

where $\mu_{y|\boldsymbol{x}} = \beta_0 + \boldsymbol{\beta x}$ is the conditional mean function, the parameters

$$\begin{aligned} \beta_0 &= \mu_y - \boldsymbol{\sigma}^T_{\boldsymbol{x}y}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{xx}}\boldsymbol{\mu_x}, \\ \boldsymbol{\beta} &= \boldsymbol{\Sigma}^{-1}_{\boldsymbol{xx}}\boldsymbol{\sigma_{xy}}, \quad \text{and} \\ \sigma_{y|\boldsymbol{x}} &= \sigma_y^2 - \boldsymbol{\sigma}^T_{\boldsymbol{x}y}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{xx}}\boldsymbol{\sigma_{xy}}. \end{aligned} \tag{2.4}$$

Hence, the conditional expectation of the probability distribution is

$$\mathrm{E}[y|\boldsymbol{x}] = \beta_0 + \boldsymbol{\beta x} = \mu_y + \boldsymbol{\beta}^T(\boldsymbol{x} - \boldsymbol{\mu_x}). \tag{2.5}$$

In regression analysis, Equation (2.5) is affine invariant. That is, prediction of the response variable is not affected by an affine transformation of $\boldsymbol{x}$. For instance, let $\boldsymbol{A} \in \mathbb{R}^{p\times p}$ be a nonsingular matrix and $\boldsymbol{x}^* = \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{\mu_x})$ an affine transformation of $\boldsymbol{x}$, then the covariance of $\boldsymbol{x}^*$ is $\boldsymbol{\Sigma}^*_{\boldsymbol{xx}} = \boldsymbol{A\Sigma_{xx}A}^T$ and the covariance between

$\boldsymbol{x}^*$ and y is $\boldsymbol{\sigma}^*_{\boldsymbol{x}y} = \boldsymbol{A}\boldsymbol{\sigma}_{\boldsymbol{x}y}$. The linear predictor in Equation (2.5) after an affine transformation is

$$\begin{aligned}
\mathrm{E}[y|\boldsymbol{x}^*] &= \mu_y + \boldsymbol{\sigma}^{*T}_{\boldsymbol{x}y}(\boldsymbol{\Sigma}^*_{\boldsymbol{xx}})^{-1}\boldsymbol{x}^* \\
&= \mu_y + \boldsymbol{\sigma}^T_{\boldsymbol{x}y}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{A}^T)^{-T}\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{\mu}_{\boldsymbol{x}}) \\
&= \mu_y + \boldsymbol{\sigma}^T_{\boldsymbol{x}y}\boldsymbol{A}^T\boldsymbol{A}^{-T}\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{A}^{-1}\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{\mu}_{\boldsymbol{x}}) \\
&= \mu_y + \boldsymbol{\sigma}^T_{\boldsymbol{x}y}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{xx}}(\boldsymbol{x}-\boldsymbol{\mu}_{\boldsymbol{x}}) = \mathrm{E}[y|\boldsymbol{x}],
\end{aligned} \tag{2.6}$$

where $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_p = \boldsymbol{A}^T\boldsymbol{A}^{-T}$, which shows that Equation (2.5) is affine invariant. However, an affine transformation of the predictor variables transforms the population regression coefficients, $\boldsymbol{\beta}$, i.e $\boldsymbol{\beta}$ is affine equivariant:

$$\boldsymbol{\beta}^* = (\boldsymbol{\Sigma}^*_{\boldsymbol{xx}})^{-1}\boldsymbol{\sigma}^*_{\boldsymbol{x}y} = (\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{A}^T)^{-1}\boldsymbol{A}\boldsymbol{\sigma}_{\boldsymbol{x}y} = (\boldsymbol{A}^T)^{-1}\boldsymbol{\Sigma}^{-1}_{\boldsymbol{xx}}\boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{\sigma}_{\boldsymbol{x}y} = \boldsymbol{A}^{-T}\boldsymbol{\beta}. \tag{2.7}$$

This is important as it will help motivate orthogonally invariant estimators later.

2.3.2 Estimation of linear regression model from samples

The data used for estimating μ_y, $\boldsymbol{\beta}$, and $\boldsymbol{\mu}_{\boldsymbol{x}}$ from Equation (2.5) consists of samples for both y and $\boldsymbol{x}$. Let $(y_i, \boldsymbol{x}_i),\ i = 1, ..., n$ be $n(> p)$ independent and identically distributed observations from a normal distribution. The minus twice log-likelihood of $f(y|\boldsymbol{x})$ up to a constant is

$$\mathcal{L} = n\log(\sigma^2_{y|\boldsymbol{x}}) + \mathrm{tr}\left((\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\sigma^{-2}_{y|\boldsymbol{x}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})^T\right), \tag{2.8}$$

where $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$ is the vector of response variables, and $\boldsymbol{X} = \boldsymbol{H}\boldsymbol{X} \in \mathbb{R}^{n\times p}$ is a centered matrix of predictor variables. The values of $\boldsymbol{\beta}$ and $\sigma^2_{y|\boldsymbol{x}}$ that maximizes Equation (2.8) are

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= \boldsymbol{S}^{-1}_{\boldsymbol{xx}}\boldsymbol{s}_{\boldsymbol{x}y} \quad \text{and} \\
\hat{\sigma}^2_{y|\boldsymbol{x}} &= \boldsymbol{s}^2_{y|\boldsymbol{x}} = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)}{n},
\end{aligned} \tag{2.9}$$

where

$$\begin{aligned}
\boldsymbol{S}_{\boldsymbol{xx}} &= \frac{1}{n-1}\sum_{i=1}^n(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^T \\
\boldsymbol{s}_{\boldsymbol{x}y} &= \frac{1}{n-1}\sum_{i=1}^n(\boldsymbol{x}_i-\bar{\boldsymbol{x}})(y_i-\bar{y}), \quad \text{and} \quad \bar{\boldsymbol{x}} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{1}
\end{aligned} \tag{2.10}$$

are the sample covariances of $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ and $\boldsymbol{\sigma}_{\boldsymbol{x}y}$, respectively, $\bar{\boldsymbol{x}} = \hat{\boldsymbol{\mu}}_{\boldsymbol{x}}$ and $\bar{y} = \hat{\mu}$ are the sample means of the variables, and $\mathbf{1}$ is a column vector of n ones. Note that the maximum likelihood estimate $\boldsymbol{s}^2_{y|\boldsymbol{x}}$ is biased and it is often divided by $n-p-1$ instead of n to make it unbiased. In the sample, the linear predictor in Equation (2.5) is replace by

$$\hat{y} = \hat{\mu}_y + \hat{\boldsymbol{\beta}}^T(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_{\boldsymbol{x}}). \tag{2.11}$$

The $\hat{\boldsymbol{\beta}}$ is the minimum-variance unbiased estimator (MVUE) of $\boldsymbol{\beta}$ when the sample size (n) is larger than $p+1$ predictor variables. However, when p is larger than n, $\hat{\boldsymbol{\beta}}$ may not be the best (MVUE) choice because of near or perfect multicollinearity. Particularly, when n is moderate compare to p, the estimated regression coefficients can be unstable because of near multicollinearity. An extreme case is when predictor variables have perfect multicollinearity, as a consequence $\boldsymbol{X}$ is not full-rank and $\hat{\boldsymbol{\beta}}$ will not exist (Almøy, 1996).

Alternative methods can be used when predictor variables have near or perfect multicollinearity; these methods either use $k(< p)$ predictor variables or reduce the dimension of the predictor variables through linear combinations. Two alternative methods are explored in this chapter; these methods include the least absolute shrinkage and selection operator (lasso) and orthonormal transformation of predictor variables amongst others. These methods produce biased estimates of the regression coefficients, but can significantly reduce the mean squared error (MSE) of the estimators, and thus improve prediction.

The MSE which measure the quality of an estimator is defined as

$$\text{MSE}(\hat{\boldsymbol{\beta}}) = \text{E}[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})]. \tag{2.12}$$

It is known that the MSE of an estimator can be expressed as the sum of the variance and squared bias of the estimator:

$$\text{MSE}(\hat{\boldsymbol{\beta}}) = \text{var}(\hat{\boldsymbol{\beta}}) + \text{bias}(\hat{\boldsymbol{\beta}})^2, \tag{2.13}$$

where $\text{var}(\hat{\boldsymbol{\beta}}) = \text{E}\left[(\hat{\boldsymbol{\beta}} - \text{E}[\hat{\boldsymbol{\beta}}])^2\right]$ and $\text{bias}(\hat{\boldsymbol{\beta}}) = \text{E}[\hat{\boldsymbol{\beta}}] - \boldsymbol{\beta}$ are the variance and bias of the estimators, respectively. If the estimator $\hat{\boldsymbol{\beta}}$ is unbiased its MSE is

$$\text{MSE}(\hat{\boldsymbol{\beta}}) = \text{var}(\hat{\boldsymbol{\beta}}). \tag{2.14}$$

By reducing var($\hat{\boldsymbol{\beta}}$), the bias($\hat{\boldsymbol{\beta}}$) can increase which may lead to regression coefficients with smaller MSE, provided the increase in bias is not too large, and hence improve predictive performance. We give a brief description of the methods which can substantially reduce the MSE in subsequent sections.

2.3.3 Orthonormal transformation of predictor variables

The problem of near and perfect multicollinearity can be handled by reducing the dimension of the predictor variables through linear transformations. The affine transformation discussed in Equation (2.6) will not the change predicted values, but it can still lead to high variance of individual regression coefficients and high correlation between the estimated regression coefficients due to near multicollinearity. An orthogonal transformation can instead be used to orthonormalize the predictor variables. For instance, one can find a full rank orthogonal matrix $\boldsymbol{A}^* \in \mathbb{R}^{p\times p}$ such that $\boldsymbol{X}^* = \boldsymbol{X}\boldsymbol{A}^*$ is orthonormal, where ${\boldsymbol{X}^*}^T\boldsymbol{X}^* = \boldsymbol{D} \in \mathbb{R}^{p\times p}$ is a diagonal matrix with zeros in the off-diagonal. Consider the singular value decomposition (SVD) of $\boldsymbol{X}$;

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T, \tag{2.15}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times p}$ is a matrix of left singular vectors ($\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_p$), $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ a matrix of right singular vectors ($\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_p$), and $\boldsymbol{D} \in \mathbb{R}^{p\times p}$ is a diagonal matrix of singular values with $d_{11} \geq d_{22} \geq \ldots d_{pp} \geq 0$. Then the estimated regression coefficients can be written as

$$\hat{\boldsymbol{\beta}} = \boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}^TY. \tag{2.16}$$

In particular, if X_l and X_k are highly correlated, $l \neq k$, then d_{pp} will be small, as a result $\hat{\boldsymbol{\beta}}$ will have high variance. Also, if X_l and X_k are perfectly correlated, $l \neq k$, then d_{pp} will be zero, as a result $\hat{\boldsymbol{\beta}}$ will not exist. By discarding d_{pp} along with its corresponding columns in $\boldsymbol{U}$ and $\boldsymbol{V}$ (the same holds for every d_{jj}, $j = 1, \ldots, p$ which are small or equal to zero) the estimated regression coefficient becomes

$$\tilde{\boldsymbol{\beta}} = \boldsymbol{V}^*(\boldsymbol{D}^*)^{-1}\boldsymbol{U}^{*T}Y, \tag{2.17}$$

which can be viewed as shrinkage estimator (Krämer, 2007), where $\boldsymbol{U}^* \in \mathbb{R}^{n\times k}$, $\boldsymbol{V}^* \in \mathbb{R}^{p\times k}$, and $\boldsymbol{D}^* \in \mathbb{R}^{k\times k}$ with $k < p$ (James et al., 2013). Then $\tilde{\boldsymbol{\beta}}$ is a biased estimator of $\boldsymbol{\beta}$, but its MSE might be small compared to that of $\hat{\boldsymbol{\beta}}$.

2.3.4 Lasso regression

The lasso regression was proposed by Tibshirani (1996) for simultaneous parameter estimation and variable selection. The lasso reduces the number of predictor variables by restricting the size of the regression coefficients. The restriction makes the estimated regression coefficients to be biased, but the MSE of the coefficients may be reduced substantially which can improve the prediction performance of the regression model. Improving predictive performance and forcing some regression coefficients to exactly zero are advantages of the lasso method.

The lasso shrinks the regression coefficients by introducing a penalty into Equation (2.8). The lasso estimator is obtained by optimising the function

$$\mathcal{J}_\lambda(\boldsymbol{\beta}) = \mathcal{L} + \lambda\|\boldsymbol{\beta}\|_1, \tag{2.18}$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$ is the L_1 norm of $\boldsymbol{\beta}$, and $\lambda \geq 0$ is the shrinkage (tuning) parameter which controls the complexity of the model. For $\lambda = 0$ all the regression coefficients will be nonzero. And as the value of λ increases, coefficients will be shrunk towards zero but some coefficient will be exactly zero. Also, if λ is large enough all the coefficients will be exactly zero. Moreover, λ controls the trade-off between the bias and variance of the estimators; as the shrinkage parameter increases the bias increases and the variance decreases and vice versa. This implies that at $\lambda = 0$ we have no bias and as $\lambda \to \infty$ we have zero variance in the limit (Tibshirani, 1996). So a good tuning parameter will strike a balance between bias and variance for lasso estimates. The lasso solution is computed as the parameter λ is varied. The λ can be chosen by cross-validation (CV) .

An insight into the lasso solution can be made by considering orthonormal predictor variables. Let $\boldsymbol{X}^*$ be orthonormal, i.e $\boldsymbol{X}^{*T}\boldsymbol{X}^* = \boldsymbol{I} \in \mathbb{R}^{p\times p}$, where $\boldsymbol{I}$ is the identity matrix, then maximize Equation (2.18), that is

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, \mathcal{J}_\lambda(\boldsymbol{\beta}). \tag{2.19}$$

The derivate of Equation (2.19) with respect to $\boldsymbol{\beta}$ is

$$-\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta} + \frac{\lambda\boldsymbol{\beta}}{2|\boldsymbol{\beta}|} = 0, \tag{2.20}$$

and solving for $\boldsymbol{\beta}$ gives

$$\hat{\beta}_j^{\lambda} = \text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \lambda/2)_+, \tag{2.21}$$

where $\hat{\beta}_j$ is the jth element of the maximum likelihood estimator of $\boldsymbol{\beta}$, sign$(\cdot)$ is the sign function, and $(x)_+ = \max\{x, 0\}$ (Friedman et al., 2001). That is, the lasso estimate translates the regression coefficients by an amount of $\lambda/2$.

2.3.5 Cross-validation

Cross-validation (CV) is used for obtaining several predictions so as to find the optimum complexity of a model (Baumann and Baumann, 2014). For instance, in lasso it is used for obtaining the optimum value of the tuning parameter, λ, that produce the smallest MSE. CV can be described as follows: the observations are randomly split into K folds of approximately equal sizes. The number of folds can be between 2 and n. One fold is left out as the validation set and the remaining K-1 folds are used as a training set to build models which increase in complexity. The validation set is used to obtain predicted values with increasing model complexities. This is repeated until each fold has been used as a validation set. Then, the residuals between the predicted values and y-values are computed for different model complexities, and the MSEs are computed from the residuals. The level of complexity corresponding to the smallest MSE is the optimum model complexity.

CV steps:

(a) Split the observations $\{i = 1, \ldots, n\}$ into K folds of roughly equal sizes, $F_1, \ldots, F_k$.

(b) For $k = 1, \ldots, K$:

  (i) Train on $(\boldsymbol{x}_i, y_i)$, $i \notin F_k$, and validate on $(\boldsymbol{x}_i, y_i)$, $i \in F_k$.

  (ii) For each value of the tuning parameter $\lambda \in \{\lambda_1, \ldots, \lambda_s\}$, compute the estimate $\hat{y}_{\lambda}^{-k}$ on the training set, and note the total error on the validation set;

$$\boldsymbol{e}_k(\lambda) = \sum_{i \in F_k} (y_i - \hat{y}_{\lambda}^{-k}(\boldsymbol{x}_i))^2.$$

(c) For each tuning parameter value, λ, compute the mean error over all the folds, i.e

$$\mathrm{CV}(\lambda) = \frac{1}{n}\sum_{k=1}^{K} \boldsymbol{e}_k(\lambda) = \frac{1}{n}\sum_{k=1}^{K}\sum_{i\in F_k}(y_i - \hat{y}_{\lambda}^{-k}(\boldsymbol{x}_i))^2.$$

(d) Choose the value of the tuning parameter such that

$$\hat{\lambda} = \arg\min_{\lambda}\ \mathrm{CV}(\lambda).$$

Chapter 3

Univariate Partial Least Squares Regression

3.1 Introduction

The previous chapter discussed multiple linear regression in settings when $n > p$, and noted that the maximum likelihood estimator may not exist when $p > n$ due to near or perfect multicollinearity. The lasso and a PCR-type estimator was suggested as ways of handling multicollinearity. These methods are not without limitations as discussed in Chapter 1, and partial least squares (PLS) can deal with some of these limitations.

Univariate (single-response) partial least squares (or PLS1) is a method for fitting multiple linear regression models. The PLS1 reduces the dimension of predictor variables through suitable linear combinations (Frank, 1987; Geladi and Kowalski, 1986). PLS1 is mostly used when the number of predictor variables is large and nearly or exactly multicollinear (Cook et al., 2007), and is often viewed as an algorithm. Further, PLS1 has drawn attention in the statistics community, where investigations to uncover reasons for its good or poor performance in terms of prediction accuracy have been carried out. A major statistical contribution has been the recognition that the PLS estimator can be viewed as an estimator of the parameters for a certain Gaussian model for the joint distribution of the predictor and response variables. The model is given the name *relevant components model* below. Different versions of the model are presented in the literature, but little attempt has been made to unify these models.

The purpose of this chapter is to unify different representations of the relevant components model. In particular, this chapter assumes a joint multivariate normal distribution for the response and predictor variables with certain restrictions on the joint covariance matrix of the variables. The chapter further discusses the relevant components model and shows that it can be represented alternatively as a Krylov model of dimension q, reviews various PLS1 algorithms, and shows how estimators from PLS1 algorithms are estimators of the parameters of the relevant components model and the Krylov model.

3.2 Statistical model

Assume a one-dimensional response variable and p-dimensional predictor variables, y and $\boldsymbol{x} \in \mathbb{R}^p$, respectively, which have the following multivariate normal distribution:

$$\begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} \sim \mathbf{N}_{p+1}\left[\begin{pmatrix} \boldsymbol{\mu_x} \\ \mu_y \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma_{xx}} & \boldsymbol{\sigma_{xy}} \\ \boldsymbol{\sigma}_{\boldsymbol{x}y}^T & \sigma_y^2 \end{pmatrix}\right], \tag{3.1}$$

where $\boldsymbol{\Sigma_{xx}} \in \mathbb{R}^{p\times p}$ is the covariance matrix of $\boldsymbol{x}$, $\boldsymbol{\sigma}_{\boldsymbol{x}y} \in \mathbb{R}^p$ the covariance between $\boldsymbol{x}$ and y, and $\sigma_y^2 \in \mathbb{R}$ the variance of y. The marginal vector $\boldsymbol{\mu_x} \in \mathbb{R}^p$ is the mean vector of $\boldsymbol{x}$ and $\mu_y \in \mathbb{R}$ is the marginal mean of y. Given the model (3.1) with all parameters known, the best linear predictor is

$$\mathrm{E}(y|\boldsymbol{x}) = \mu_y + \boldsymbol{\beta}^T(\boldsymbol{x} - \boldsymbol{\mu_x}), \tag{3.2}$$

where $\boldsymbol{\beta} = \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{\sigma}_{\boldsymbol{x}y} \in \mathbb{R}^{p\times 1}$ is the vector of population regression coefficients.

The model which describes a restriction on $\boldsymbol{\Sigma}$ is the relevant components model (RC). We call the model (Helland, 1988, 1990) the relevant components model of order (q) denoted by $\mathrm{RC}_{\boldsymbol{x}}$, where q $(1 \leq q \leq p)$ represents the number of linear combinations taken from the predictor variable. Note, if $q = p$ there is no restriction on $\boldsymbol{x}$. Alternatively, the relevant component model can be viewed as a Krylov model of dimension q (Inguanez, 2015). In this section, we introduce the relevant components model and then give an alternative formulation of the model.

3.2.1 Relevant components model of order (q)

In the population setting, the relevant components model of order (q), RC$_{\boldsymbol{x}}$, assumes that after a rotation of $\boldsymbol{x}$, two splits of $\boldsymbol{x}$ emerge such that all the information required to predict y is contained in only one part of the split. The part of the split which contains all the information for predicting y will be called the *relevant components* and the other part will be called the *irrelevant components*. Assume the dimensions of the components are known so that the relevant components have dimension q and the irrelevant components have dimension $p - q$ (Helland, 1988, 1990).

Let $\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{p\times p}$ be an orthogonal matrix, such that $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{p\times q}$ and $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-q)}$ are semi-orthogonal matrices of full column rank, with $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 = \boldsymbol{I}_q$, $\boldsymbol{\Gamma}_0^T\boldsymbol{\Gamma}_0 = \boldsymbol{I}_{p-q}$, and $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_0 = \boldsymbol{0}_{p\times(p-q)}$. Starting from $\boldsymbol{x}$, define two linear transformations $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$ to be (Helland, 1992)

$$\boldsymbol{t}_1 = \boldsymbol{\Gamma}_1^T\boldsymbol{x} \quad \text{and} \quad \boldsymbol{t}_0 = \boldsymbol{\Gamma}_0^T\boldsymbol{x}. \tag{3.3}$$

Conversely, $\boldsymbol{x}$ can be recovered from $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$ by

$$\boldsymbol{x} = \boldsymbol{\Gamma}_1\boldsymbol{t}_1 + \boldsymbol{\Gamma}_0\boldsymbol{t}_0 = \boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_1^T\boldsymbol{x} + \boldsymbol{\Gamma}_0\boldsymbol{\Gamma}_0^T\boldsymbol{x}, \tag{3.4}$$

where $\boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_1^T$ is a projection matrix, and $\boldsymbol{\Gamma}_0\boldsymbol{\Gamma}_0^T = \boldsymbol{I} - \boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_1^T$ is the complement of $\boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_1^T$. Assume that the rotation of $\boldsymbol{x}$ partitions the covariance matrix, $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$, such that

$$\boldsymbol{\Gamma}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 \end{pmatrix}. \tag{3.5}$$

where $\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 \in \mathbb{R}^{q\times q}$ (upper block) and $\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 \in \mathbb{R}^{(p-q)\times(p-q)}$ (lower block) are the covariance matrices of $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$, respectively. The zeros in Equation (3.5) are restrictions imposed by the RC$_{\boldsymbol{x}}$ model, and indicates that $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$ are uncorrelated. Assume $\boldsymbol{t}_1$ contain all the information in $\boldsymbol{x}$ for predicting y. Then the joint covariance matrix, $\boldsymbol{\Sigma}$, after a rotation of $\boldsymbol{x}$ is such that

$$\boldsymbol{\Sigma}_{\boldsymbol{\Gamma}} = \begin{pmatrix} \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 & \boldsymbol{0} & \boldsymbol{\Gamma}_1^T\boldsymbol{\sigma}_{\boldsymbol{xy}} \\ \boldsymbol{0} & \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 & \boldsymbol{0} \\ \boldsymbol{\sigma}_{\boldsymbol{xy}}^T\boldsymbol{\Gamma}_1 & \boldsymbol{0} & \sigma_y^2 \end{pmatrix}. \tag{3.6}$$

where $\boldsymbol{\Gamma}_1^T\boldsymbol{\sigma}_{\boldsymbol{x}y} \in \mathbb{R}^q$ the covariance between $\boldsymbol{t}_1$ and y. The covariance between $\boldsymbol{t}_0$ and y is $\boldsymbol{\Gamma}_0^T\boldsymbol{\sigma}_{\boldsymbol{x}y} \in \mathbb{R}^{p-q} = \mathbf{0}$ and indicates that $\boldsymbol{t}_0$ hold no information for predicting y. The RC$_{\boldsymbol{x}}$ model assumes that q is the smallest dimension for which there exist orthogonal matrix $\boldsymbol{\Gamma}$ such that Equation (3.6) holds (Naes and Helland, 1993). As a result we have the following (Cook et al., 2013):

$$\begin{aligned}(a)\ \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}} &= \boldsymbol{\Gamma}_1\operatorname{cov}(\boldsymbol{t}_1)\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0\operatorname{cov}(\boldsymbol{t}_0)\boldsymbol{\Gamma}_0^T\\ &= \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0}\boldsymbol{\Gamma}_0^T \text{ and}\\ (b)\ \boldsymbol{\sigma}_{\boldsymbol{x}y} &= \boldsymbol{\Gamma}_1\operatorname{cov}(\boldsymbol{t}_1, y) + \boldsymbol{\Gamma}_0\operatorname{cov}(\boldsymbol{t}_0, y)\\ &= \boldsymbol{\Gamma}_1\boldsymbol{\eta} + \mathbf{0} = \boldsymbol{\Gamma}_1\boldsymbol{\eta},\end{aligned} \tag{3.7}$$

where the following are function of $\boldsymbol{\Sigma}$; $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1} = \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\Gamma}_1 \in \mathbb{R}^{q\times q}$ and $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0} = \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\Gamma}_0 \in \mathbb{R}^{(p-q)\times(p-q)}$ are covariance matrices of $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$, respectively, $\boldsymbol{\eta} = \operatorname{cov}(\boldsymbol{t}_1, y) \in \mathbb{R}^q$ is the covariance vector between $\boldsymbol{t}_1$ and y, and the $\mathbf{0}$ in Equation (3.7)(b) is because $\boldsymbol{t}_0$ and y are uncorrelated. The results in Equation (3.7) says that all the information in $\boldsymbol{x}$ required to explain the relationship between y and $\boldsymbol{x}$ is contained in $\boldsymbol{t}_1$. Then, the joint covariance matrix $\boldsymbol{\Sigma}$ can be rewritten as (Cook et al., 2013)

$$\boldsymbol{\Sigma} = \begin{pmatrix}\boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0}\boldsymbol{\Gamma}_0^T & \boldsymbol{\Gamma}_1\boldsymbol{\eta}\\ \boldsymbol{\eta}^T\boldsymbol{\Gamma}_1^T & \sigma_{yy}\end{pmatrix}. \tag{3.8}$$

If the RC$_{\boldsymbol{x}}$ model holds, the linear predictor in Equation (3.2) is simplified as

$$y = \mu_y + \boldsymbol{\alpha}^{*^T}\boldsymbol{\Gamma}_1^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}), \tag{3.9}$$

with

$$\boldsymbol{\alpha}^* = \begin{pmatrix}\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\Gamma}_1 & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\Gamma}_0\end{pmatrix}^{-1}\begin{pmatrix}\boldsymbol{\Gamma}_1^T\boldsymbol{\sigma}_{\boldsymbol{x}y}\\ \mathbf{0}\end{pmatrix} = (\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1\boldsymbol{\sigma}_{\boldsymbol{x}y} \in \mathbb{R}^{q\times 1}, \tag{3.10}$$

and

$$\boldsymbol{\beta}_{\boldsymbol{\Gamma}_1} = \boldsymbol{\Gamma}_1(\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{\sigma}_{\boldsymbol{x}y} \tag{3.11}$$

is determined by the relevant components of $\boldsymbol{x}$ alone. The orthogonal matrix, $\boldsymbol{\Gamma}$, can be determined in the population using properties of the Krylov matrix. In Section 3.2.2, we discuss how $\boldsymbol{\Gamma}$ can be determined from the population and present an alternative representation of the RC$_{\boldsymbol{x}}$ model called the Krylov model.

3.2.2 Population $\mathbf{\Gamma}$ and alternative RC$_{\boldsymbol{x}}$ model representation

The orthogonal matrix, $\mathbf{\Gamma}$, can be determined through the linear dependence of columns of the Krylov matrix given by

$$\boldsymbol{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y}) = (\boldsymbol{\sigma}_{\boldsymbol{x}y}, \mathbf{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{x}y}, ..., \mathbf{\Sigma}_{\boldsymbol{xx}}^{j-1}\boldsymbol{\sigma}_{\boldsymbol{x}y}) \in \mathbb{R}^{p\times j}, \tag{3.12}$$

where $\boldsymbol{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$ is called the Krylov matrix of order j, for all $j = 1, \ldots, p$ $(j \leq p)$. The Krylov matrix is connected to the power method, which is used for finding the eigenvalues and eigenvectors of symmetric matrices. The power method generates a sequence of p-dimensional vectors that make up the Krylov matrix (Phatak and De Jong, 1997). Let the Krylov subspace of order j be

$$\begin{aligned}\mathcal{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) &= \text{span}(\boldsymbol{\sigma}_{\boldsymbol{x}y}, \mathbf{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{x}y}, ..., \mathbf{\Sigma}_{\boldsymbol{xx}}^{j-1}\boldsymbol{\sigma}_{\boldsymbol{xy}}) \in \mathbb{R}^{p\times j}\\ &= \text{span}(\mathbf{\Sigma}_{\boldsymbol{xx}}^{k}\boldsymbol{\sigma}_{\boldsymbol{x}y} : k = 0, 1, 2, \ldots, j-1).\end{aligned} \tag{3.13}$$

Note that, any vector in $\mathcal{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$ can be written as a linear combination given by

$$a_1\mathbf{\Sigma}_{\boldsymbol{xx}}^0\boldsymbol{\sigma}_{\boldsymbol{x}y} + a_2\mathbf{\Sigma}_{\boldsymbol{xx}}^1\boldsymbol{\sigma}_{\boldsymbol{xy}} + a_3\mathbf{\Sigma}_{\boldsymbol{xx}}^2\boldsymbol{\sigma}_{\boldsymbol{xy}} + \cdots + a_j\mathbf{\Sigma}_{\boldsymbol{xx}}^{j-1}\boldsymbol{\sigma}_{\boldsymbol{xy}} + a_{j+1}\mathbf{\Sigma}_{\boldsymbol{xx}}^{j}\boldsymbol{\sigma}_{\boldsymbol{xy}},$$

where $a_{j+1} = 0$ and $\mathbf{\Sigma}_{\boldsymbol{xx}}^0 = 1$. It follows that (Inguanez, 2015),

$$\text{span}(\boldsymbol{\sigma}_{\boldsymbol{x}y}, \mathbf{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}^2\boldsymbol{\sigma}_{\boldsymbol{x}y}, \cdots, \mathbf{\Sigma}_{\boldsymbol{xx}}^{j-1}\boldsymbol{\sigma}_{\boldsymbol{x}y}) \subset \text{span}(\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{x}y}, \mathbf{\Sigma}_{\boldsymbol{xx}}^2\boldsymbol{\sigma}_{\boldsymbol{xy}}, \cdots, \mathbf{\Sigma}_{\boldsymbol{xx}}^{j}\boldsymbol{\sigma}_{\boldsymbol{xy}})$$

and,

$$\begin{aligned}\mathbf{\Sigma}_{\boldsymbol{xx}}\,\text{span}(\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}^2\boldsymbol{\sigma}_{\boldsymbol{xy}}, \cdots, \mathbf{\Sigma}_{\boldsymbol{xx}}^{j-1}\boldsymbol{\sigma}_{\boldsymbol{x}y}) &= \text{span}(\mathbf{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}^2\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}^3\boldsymbol{\sigma}_{\boldsymbol{xy}}, \cdots, \mathbf{\Sigma}_{\boldsymbol{xx}}^{j}\boldsymbol{\sigma}_{\boldsymbol{xy}})\\ &\subset \text{span}(\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}^2\boldsymbol{\sigma}_{\boldsymbol{xy}}, \cdots, \mathbf{\Sigma}_{\boldsymbol{xx}}^{j}\boldsymbol{\sigma}_{\boldsymbol{xy}}).\end{aligned}$$

So that $\mathcal{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$ forms a sequence of vector spaces with $\mathcal{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) \subset \mathcal{K}_{j+1}(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$. That is,

$$\begin{aligned}\mathcal{K}_1(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) &= \text{span}(\boldsymbol{\sigma}_{\boldsymbol{xy}}),\\ \mathcal{K}_2(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) &= \text{span}(\boldsymbol{\sigma}_{\boldsymbol{xy}}, \mathbf{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}}) \supset \mathcal{K}_1(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}).\end{aligned} \tag{3.14}$$

Then, $\mathcal{K}_1(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) \subset \mathcal{K}_2(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) \subset \cdots \subset \mathcal{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$. In addition, when two adjacent Krylov subspace are equal, they are also equal to subsequent subspaces. That is, if $\mathcal{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) = \mathcal{K}_{j+1}(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$, then

$$\mathcal{K}_j(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) = \mathcal{K}_{j+1}(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) = \mathcal{K}_{j+2}(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) = \cdots = \mathcal{K}_p(\mathbf{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}). \tag{3.15}$$

In other words, $\mathcal{K}_1(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})$ is contained in $\mathcal{K}_2(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})$, $\mathcal{K}_2(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})$ is contained in $\mathcal{K}_3(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})$, and so on; the subspace increases until the required dimension, j, is reached. At $\mathcal{K}_j(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})$ it remains unchanged as additional dimensions are not required due to linear dependence (Cook et al., 2013; Inguanez and Kent, 2013). Then the matrix, $\boldsymbol{K}_j(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})$, has at most $\min(j, p)$ linearly independent columns, because any set of $\min(j, p) + 1$ vectors of $\boldsymbol{K}_j(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})$ is linearly dependent. Let the required dimension be q such that

$$q = \min\{j : \boldsymbol{\mathcal{K}}_j(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y}) = \boldsymbol{\mathcal{K}}_{j+1}(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})\}.$$

The q is called the Krylov dimension. The Krylov matrix and Krylov subspace of dimension q (Saad, 1992, 2003) are

$$\boldsymbol{K}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y}) = (\boldsymbol{\sigma}_{\boldsymbol{x}y}, \boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{x}y}, ..., \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{q-1}\boldsymbol{\sigma}_{\boldsymbol{x}y}) \in \mathbb{R}^{p\times q} \tag{3.16}$$

and

$$\boldsymbol{\mathcal{K}}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y}) = \mathrm{span}(\boldsymbol{\sigma}_{\boldsymbol{x}y}, \boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{x}y}, ..., \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{q-1}\boldsymbol{\sigma}_{\boldsymbol{x}y}), \tag{3.17}$$

respectively. Let $\boldsymbol{G} = (\boldsymbol{G}_1, \boldsymbol{G}_0) \in \mathbb{R}^{p\times p}$ be an orthogonal basis representing an element of the Grassmann manifold, such that $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{I}_p = \boldsymbol{G}\boldsymbol{G}^T$. The Grassmann manifold denoted by $\mathcal{G}_{q,p}$ is the set of all q-dimensional subspaces of a vector space in $\mathbb{R}^p$. A point in $\mathcal{G}_{q,p}$ is a vector space of the Euclidean space, which may be specified by a $p\times q$ semi-orthogonal matrix whose columns form an arbitrary basis of the vector subspace of interest. We can require $\boldsymbol{G}_1 \in \mathbb{R}^{p\times q}$ to be a semi-orthogonal basis of $\boldsymbol{\mathcal{K}}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{x}y})$ and $\boldsymbol{G}_0 \in \mathbb{R}^{p\times(p-q)}$ its orthogonal completion. The first column of $\boldsymbol{G}_1$, $\boldsymbol{g}_1$, can be chosen to be proportional to $\boldsymbol{\sigma}_{\boldsymbol{x}y}$, i.e,

$$\boldsymbol{g}_1 = \tau\boldsymbol{\sigma}_{\boldsymbol{x}y} \in \mathbb{R}^p, \quad \text{then} \quad \boldsymbol{G}^T\boldsymbol{\sigma}_{\boldsymbol{x}y} = \tau\boldsymbol{e}_1 \in \mathbb{R}^p, \tag{3.18}$$

where τ is the constant of proportionality, and $\boldsymbol{e}_1 = (1, 0, \dots, 0)^T$ is known as the standard basis vector. The product $\boldsymbol{G}^T\boldsymbol{\sigma}_{\boldsymbol{x}y}$ is proportional to $\boldsymbol{e}_1$ because the columns of $\boldsymbol{G}$ are orthogonal to one another, i.e

$$\begin{aligned} &\boldsymbol{G}^T\boldsymbol{\sigma}_{\boldsymbol{x}y} \propto \boldsymbol{G}^T\boldsymbol{g}_1 = \left[\boldsymbol{g}_1^T, \boldsymbol{g}_2^T, \dots, \boldsymbol{g}_p^T\right]\boldsymbol{g}_1 = (1, 0, \dots, 0)^T, \\ &\text{since} \quad \boldsymbol{g}_j^T\boldsymbol{g}_1 = 0 \quad \text{for} \quad j = 2, 3, \dots, p, \end{aligned} \tag{3.19}$$

where $\boldsymbol{g}_j$ are the columns of $\boldsymbol{G}$ (Inguanez, 2015). If $\boldsymbol{x}$ is rotated by $\boldsymbol{G}$ the covariance matrix, $\boldsymbol{G}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}$, will be block diagonal such that

$$\boldsymbol{G}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{G}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_0, \end{pmatrix} \tag{3.20}$$

where $\boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1 \in \mathbb{R}^{q\times q}$ and $\boldsymbol{G}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_0 \in \mathbb{R}^{(p-q)\times(p-q)}$ are upper and lower blocks, respectively. Moreover, if $\boldsymbol{G}$ is chosen with respect to Equation (3.18) it can be shown that $\boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1$ and $\boldsymbol{G}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_0$ are tridiagonal (Sundar and Bhagavan, 1999). Note that all the information needed to explain the relationship between y and $\boldsymbol{x}$ is contained in $\boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1$. The Krylov model of dimension q can then be represented alternatively as

$$\boldsymbol{\Sigma}_{\boldsymbol{G}} = \begin{pmatrix} \boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1 & \mathbf{0} & \tau\boldsymbol{e}_1 \\ \mathbf{0} & \boldsymbol{G}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_0 & \mathbf{0} \\ \tau\boldsymbol{e}_1 & \mathbf{0} & \sigma_{yy} \end{pmatrix} \tag{3.21}$$

Representing the Krylov model as (3.21) also simplifies the linear predictor in Equation (3.2) such that

$$y = \mu_y + \boldsymbol{\alpha}^T\boldsymbol{G}_1^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}), \tag{3.22}$$

with

$$\boldsymbol{\alpha} = \begin{pmatrix} \boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{G}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_0 \end{pmatrix}^{-1} \begin{pmatrix} \tau\boldsymbol{e}_1 \\ \mathbf{0} \end{pmatrix} = \tau(\boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1)^{-1}\boldsymbol{e}_1 \in \mathbb{R}^{q\times 1}. \tag{3.23}$$

Then,

$$\boldsymbol{\beta}_{\boldsymbol{G}_1} = \tau\boldsymbol{G}_1(\boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1)^{-1}\boldsymbol{e}_1 = \tau\boldsymbol{G}_1\boldsymbol{\alpha} \tag{3.24}$$

is the regression estimator.

Note that the RC$_{\boldsymbol{x}}$ model in (3.6) and the Krylov model in (3.21) are identical. Recall that $\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1$ is a diagonal matrix and $\boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1$ is a tridiagonal matrix. The covariance matrix, $\boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1$, can be converted into a diagonal matrix after an additional orthogonal transformation such that

$$\boldsymbol{\Delta}_1\boldsymbol{G}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}_1\boldsymbol{\Delta}_1 = \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1,$$

where $\boldsymbol{\Delta}_1 \in \mathbb{R}^{q\times q}$ is an orthogonal matrix. Then Equations (3.6) and (3.21) are identical such that

$$\boldsymbol{\Delta}\boldsymbol{G}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{G}\boldsymbol{\Delta} = \boldsymbol{\Gamma}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma} \quad \text{and} \quad \tau\boldsymbol{\Delta}\boldsymbol{e}_1 = \boldsymbol{\Gamma}_1\boldsymbol{\eta}, \tag{3.25}$$

where $\boldsymbol{\Delta} = (\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_0) \in \mathbb{R}^{p\times p}$, and $\boldsymbol{\Delta}_0 \in \mathbb{R}^{(p-q)\times(p-q)}$.

3.3 Review of PLS1

PLS1 is regarded as a family of algorithms which produce the same regression estimator if q is known. These algorithms are different only as slight modifications of one another, but they all have the same objective; which is to determine the basis matrix used for rotating $\boldsymbol{x}$. One of the earliest algorithms is the nonlinear iterative partial least squares (NIPALS) (Wold, 1966). However, various PLS1 algorithms have been proposed for use in regression analysis. For instance, the statistically inspired modification of PLS (SIMPLS) algorithm that maximizes the covariance between the response and predictor variables was developed by De Jong (1993). SIMPLS differ slightly from the NIPALS because components are determined from original predictor variables rather than transformed variables (transformed variables are used in NIPALS). Other algorithms for PLS1 include factor method, Gram-Schmidt, conjugate gradient, Lanczos bidiagonalization, kernel algorithm (Boulesteix and Strimmer, 2006; Lindgren et al., 1993; Rännar et al., 1994) etc.

The relationship between these algorithms have been investigated in the literature. For instance, Helland (1988) proved that two versions of the NIPALS algorithm are identical. A description of the relationship between PLS1 and Lanczos bidiagonalization was investigated by Elden (2004) and Bro and Elden (2009) to further promote the use of PLS1 in linear regression analysis. Phatak and de Hoog (2002) and Takane and Loisel (2014) studied PLS1 as a constrained least squares estimator (number of linear combinations, q, is the constraint) which lie in a Krylov subspace and made connections between the NIPALS, conjugate gradient and Lanczos bidiagonalization algorithms. Andersson (2009) proposed three PLS1 algorithms and compared them to six already existing algorithms in terms of numerical stability and speed. It was discovered that some existing algorithms may be suboptimal in terms of estimation and prediction accuracy. Also, some fast algorithms were shown to be unstable when large number of linear combinations of the predictor variables are needed.

However, statisticians are more interested in models than algorithm. Helland (1990) introduced a statistical model (identical to the relevant component model, (3.6)) for dimension reduction and made connection between some parameters of the model and parameters determined using PLS1 algorithms. In addition, Helland developed a maximum likelihood procedure for estimating the parameters of the model (Helland, 1992). Following Helland (1992), Inguanez (2015) proposed a Krylov model, (identical to (3.21)), for dimension reduction and developed an approximate maximum likelihood method for parameter estimation and prediction.

The next section is a continuation of the review on PLS1 with a description of some PLS1 algorithms.

3.4 Description of population PLS1 algorithms

In this section, we describe some PLS1 algorithms when the dimension, q, is known. These algorithms generate different bases but ultimately leads to the same regression estimator (Andersson, 2009). We considered the factor method, Gram-Schmidt and NIPALS algorithms. Furthermore, connections are made between these algorithms and the Krylov subspace.

3.4.1 NIPALS algorithm

The NIPALS algorithm (Abdi et al., 2016) form the foundation for other PLS1 algorithms and is given as follows:

1. Let $\boldsymbol{x}_0 = \boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}$ and $y_0 = y - \mu_y$, and set $\boldsymbol{x}_0 = \boldsymbol{x}$ and $y_0 = y$.

2. For $j = 1, 2, ..., q \leq p$ preform the steps below:

   (i) Determine linear combinations (components) of $\boldsymbol{x}$ while taking into account its relationship with y. First define weights $\boldsymbol{w}_j$ as the covariance between $\boldsymbol{x}$ and y. Second, determine component, t_j, of $\boldsymbol{x}$;

$$\begin{aligned} \boldsymbol{w}_j =& \frac{\operatorname{cov}(\boldsymbol{x}_{j-1}, y_{j-1})}{\| \operatorname{cov}(\boldsymbol{x}_{j-1}, y_{j-1})\|}, \\ t_j =& \frac{\boldsymbol{x}_{j-1}^T \boldsymbol{w}_j}{\|\boldsymbol{x}_{j-1}^T \boldsymbol{w}_j\|}, \quad \text{where} \quad \|\boldsymbol{x}_{j-1}^T \boldsymbol{w}_j\| = (\boldsymbol{w}_j^T \boldsymbol{x}_{j-1} \boldsymbol{x}_{j-1}^T \boldsymbol{w}_j)^{1/2}, \end{aligned}$$

   where the weights, $\boldsymbol{w}_j$, are orthogonal (Höskuldsson, 1988).

(ii) Using least squares determine the loadings, $\boldsymbol{p}_j$ and c_j, of $\boldsymbol{x}$ and y, respectively. These are obtained by regressing $\boldsymbol{x}$ on t_j, and y on t_j:

$$\boldsymbol{p}_j = \mathrm{cov}(\boldsymbol{x}_{j-1}, t_j), \quad \text{and} \quad c_j = \mathrm{cov}(y_{j-1}, t_j).$$

(iii) Obtain new $\boldsymbol{x}$ and y:

$$\begin{aligned} \boldsymbol{x}_j =& \boldsymbol{x}_{j-1} - \boldsymbol{p}_j t_j = \boldsymbol{x}_{j-1}\boldsymbol{Q}_{\boldsymbol{x}_{j-1}\boldsymbol{w}_j} \quad \text{(deflation)}, \\ y_j =& y_{j-1} - c_j t_j. \end{aligned}$$

where $\boldsymbol{Q}_{\boldsymbol{x}_{j-1}\boldsymbol{w}_j} = \boldsymbol{I} - \boldsymbol{x}_{j-1}^T\boldsymbol{w}_j(\boldsymbol{w}_j^T\boldsymbol{x}_{j-1}\boldsymbol{x}_{j-1}^T\boldsymbol{w}_j)^{-1}\boldsymbol{w}_j^T\boldsymbol{x}_{j-1}$.

After q steps, the algorithm establishes a bilinear relationship between the response and predictor variables given by:

$$\begin{aligned} \boldsymbol{x} =& \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{p}_1 t_1 + \boldsymbol{p}_2 t_2 + \cdots + \boldsymbol{p}_q t_q, \\ y =& \mu_y + c_1 t_1 + c_2 t_2 + \cdots + c_q t_q. \end{aligned} \tag{3.26}$$

The linear predictor is

$$\begin{aligned} y_{q,PLS} =& \mu_y + c_1 t_1 + c_2 t_2 + \cdots + c_q t_q \\ =& \mu_y + \boldsymbol{\beta}_{q,PLS}^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}). \end{aligned} \tag{3.27}$$

In this thesis, the $\boldsymbol{w}_j'$s are of interest, and Helland (1988) showed that $\boldsymbol{w}_j \propto \boldsymbol{p}_j$, that is, either can be used as the basis vector.

3.4.2 Factor method

The factor method produces the relationship in Equation (3.27) (Helland, 1990). The factor method begins by projecting $\boldsymbol{x}$ onto two orthogonal subspaces. Let $\boldsymbol{R} = (\boldsymbol{R}_1, \boldsymbol{R}_0) \in \mathbb{R}^{p\times p}$ be an orthogonal matrix, with $\boldsymbol{R}_1 \in \mathbb{R}^{p\times q}$ and $\boldsymbol{R}_0 \in \mathbb{R}^{p\times(p-q)}$ semi-orthogonal matrices of full rank. Then,

$$\boldsymbol{x} = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{R}_1\boldsymbol{R}_1^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) + \boldsymbol{R}_0\boldsymbol{R}_0^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}). \tag{3.28}$$

Equation (3.28) can be rewritten as

$$\begin{aligned} \boldsymbol{x} =& \boldsymbol{\mu}_{\boldsymbol{x}} + \sum_{j=1}^{q} \boldsymbol{r}_{1j}\boldsymbol{r}_{1j}^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) + \sum_{j=1}^{p-q} \boldsymbol{r}_{0j}\boldsymbol{r}_{0j}^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) \\ =& \boldsymbol{\mu}_{\boldsymbol{x}} + \sum_{j=1}^{q} \boldsymbol{r}_{1j} t_{1j} + \sum_{j=1}^{p-q} \boldsymbol{r}_{0j} t_{0j} \\ =& \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{R}_1\boldsymbol{t}_1 + \boldsymbol{R}_0\boldsymbol{t}_0, \end{aligned} \tag{3.29}$$

where $\boldsymbol{r}_{1j}$ and $\boldsymbol{r}_{0j}$ are column vectors of $\boldsymbol{R}_1$ and $\boldsymbol{R}_0$, respectively, $t_{1j} = \boldsymbol{r}_{1j}^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}})$ and $t_{0j} = \boldsymbol{r}_{0j}^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}})$ represent relevant and irrelevant components, respectively. The response, y, is related to $\boldsymbol{x}$ through

$$\begin{aligned} y &= \mu_y + \boldsymbol{\beta}^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) + \epsilon \\ &= \mu_y + \boldsymbol{\beta}^T \left(\sum_{j=1}^{q} \boldsymbol{r}_{1j} t_{1j} + \sum_{j=1}^{p-q} \boldsymbol{r}_{0j} t_{0j} \right) + \epsilon \\ &= \mu_y + \boldsymbol{\beta}^T \sum_{j=1}^{q} \boldsymbol{r}_{1j} t_{1j} + \epsilon, \end{aligned} \tag{3.30}$$

where $\boldsymbol{\beta}^T \sum_{j=1}^{p-q} \boldsymbol{r}_{0j} t_{0j} = 0$ because the regression coefficients are not determined by irrelevant components. Recall that the covariance matrices can be expressed as $\boldsymbol{\Sigma}_{\boldsymbol{xx}} = \boldsymbol{R\Lambda R}^T$ and $\boldsymbol{\sigma}_{\boldsymbol{x}y} = \boldsymbol{R\vartheta}$ after a rotation of $\boldsymbol{x}$, where $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times p}$ is a diagonal matrix. Then, the vector regression coefficient is

$$\boldsymbol{\beta} = \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1} \boldsymbol{\sigma}_{\boldsymbol{x}y} = \boldsymbol{R\Lambda}^{-1}\boldsymbol{\vartheta} = \sum_{j=1}^{p} \lambda_j^{-1} \boldsymbol{r}_j \vartheta_j, \tag{3.31}$$

where ϑ_j is the ith element of $\boldsymbol{\vartheta}$. If $p - q + 1$ values of $\boldsymbol{\Lambda}$ are equal then the corresponding basis vectors in $\boldsymbol{R}$ span the same subspace. This implies that a single vector can be used to represent the set of vectors. Let the chosen basis vector be $\boldsymbol{r}^*$, then the regression coefficient becomes

$$\begin{aligned} \boldsymbol{\beta} &= \sum_{j=1}^{p} \lambda_j^{-1} \boldsymbol{r}_j \vartheta_j \simeq \sum_{j=1}^{q-1} \lambda_j^{-1} \boldsymbol{r}_{1j} \vartheta_j + \lambda_q^{-1} \boldsymbol{r}^* \vartheta_q \\ &= \sum_{j=1}^{q} \lambda_j^{-1} \boldsymbol{r}_{1j} \vartheta_j = \sum_{j=1}^{q} \boldsymbol{r}_{1j} \frac{\vartheta_j}{\lambda_j} = \sum_{j=1}^{q} \boldsymbol{r}_{1j} c_j, \end{aligned} \tag{3.32}$$

where $c_j = \dfrac{\vartheta_j}{\lambda_j}$ is the jth loading for y. Then the linear predictor of Equation (3.30) becomes

$$\begin{aligned} y_{q,f} &= \mu_y + \boldsymbol{\beta}^T \sum_{j=1}^{q} \boldsymbol{r}_{1j} t_{1j} = \mu_y + \sum_{j=1}^{q} c_j \boldsymbol{r}_{1j}^T \sum_{j=1}^{q} \boldsymbol{r}_{1j} t_{1j} \\ &= \mu_y + \sum_{j=1}^{q} c_j \boldsymbol{r}_{1j}^T \boldsymbol{r}_{1j} t_{1j} = \mu_y + \sum_{j=1}^{q} c_j t_{1j}, \end{aligned} \tag{3.33}$$

which has the same representation as Equation (3.27).

3.4.3 Gram-Schmidt algorithm

The Gram-Schmidt (GS) algorithm is used for operating upon a set of vectors to produces an orthonormal set of vectors which are linear combinations of the original set. For instance, let $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p) \in \mathbb{R}^{p\times p}$ be a set of vectors, if $\tilde{\boldsymbol{V}} = (\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}}_p) \in \mathbb{R}^{p\times p}$ is an orthogonal basis obtained by orthonormalizing the columns of $\boldsymbol{V}$, then $\text{span}(\tilde{\boldsymbol{V}}) = \text{span}(\boldsymbol{V})$, where $\tilde{\boldsymbol{V}}^T\tilde{\boldsymbol{V}} = \boldsymbol{I}_p$.

In Section 3.2.1 the semi-orthogonal matrix, $\boldsymbol{G}_1$, was defined as a basis for the Krylov matrix of order q ($\boldsymbol{K}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$). In this section, we show how $\boldsymbol{G}_1$ can be determined from $\boldsymbol{K}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$ using the GS algorithm. Recall that

$$\boldsymbol{K}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}}) = (\boldsymbol{\sigma}_{\boldsymbol{xy}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}}, ..., \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{q-1}\boldsymbol{\sigma}_{\boldsymbol{xy}}) \in \mathbb{R}^{p\times q}.$$

To simplify notations let the column vector of $\boldsymbol{K}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$ be denoted by $\boldsymbol{k}_j = \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{j-1}\boldsymbol{\sigma}_{\boldsymbol{xy}},\ j = 1, \ldots, q$, and let the projection of $\boldsymbol{k}_j$ onto $\boldsymbol{k}_{j'}$, $j \neq j'$, be denoted by

$$\text{proj}_{\boldsymbol{k}_{j'}}(\boldsymbol{k}_j) = \left(\frac{\boldsymbol{k}_j^T\boldsymbol{k}_{j'}}{\boldsymbol{k}_{j'}^T\boldsymbol{k}_{j'}}\right)\boldsymbol{k}_{j'} = c_j^*\boldsymbol{k}_{j'},$$

where $c_j^*,\ j = 1, \ldots, q-1$ is a scalar. The GS algorithm is done sequentially as follows:

$$\begin{aligned}
\text{Set}\quad \boldsymbol{g}_1 &= \boldsymbol{\sigma}_{\boldsymbol{xy}} \subset \text{span}(\boldsymbol{\sigma}_{\boldsymbol{xy}}) \\
\boldsymbol{g}_2 &= \boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}} - c_1^*\boldsymbol{\sigma}_{\boldsymbol{xy}} \subset \text{span}(\boldsymbol{\sigma}_{\boldsymbol{xy}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}}) \\
\boldsymbol{g}_3 &= \boldsymbol{\Sigma}_{\boldsymbol{xx}}^2\boldsymbol{\sigma}_{\boldsymbol{xy}} - c_2^*\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}} - c_1^*\boldsymbol{\sigma}_{\boldsymbol{xy}} \subset \text{span}(\boldsymbol{\sigma}_{\boldsymbol{xy}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\sigma}_{\boldsymbol{xy}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}}^2\boldsymbol{\sigma}_{\boldsymbol{xy}}) \\
&\vdots \\
\boldsymbol{g}_q &= \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{q-1}\boldsymbol{\sigma}_{\boldsymbol{xy}} - \sum_{j=1}^{q-1} c_j^*\boldsymbol{\Sigma}_{\boldsymbol{xx}}^j\boldsymbol{\sigma}_{\boldsymbol{xy}}
\end{aligned} \tag{3.34}$$

This shows that the columns of $\boldsymbol{G}_1$ are linear combinations of $\boldsymbol{K}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$. The column vectors, $\boldsymbol{g}_j$, are orthogonal to one another and can be orthonormalized by normalizing each $\boldsymbol{g}_j$.

3.4.4 Connecting NIPALS and two other algorithms

In this section, connections between the factor method, Gram-Schmidt and NIPALS algorithms are discussed via the weights (transformation matrix).

Connection between Gram-Schmidt and NIPALS

The connection between Gram-Schmidt and NIPALS algorithms can be made by showing that the weights, $\boldsymbol{w}_j$, determined from NIPALS can be represented as basis of the Krylov matrix (Helland, 1988, 1990, 2001).

Recall that from the NIPALS algorithm in Section 3.4.1 weights are defined as $\boldsymbol{w}_j \propto \text{cov}(\boldsymbol{x}_{j-1}, y_{j-1})$ and $\boldsymbol{x}_j = \boldsymbol{x}_{j-1} - \boldsymbol{p}_j t_j = \boldsymbol{x}_{j-1}\boldsymbol{Q}_{\boldsymbol{x}_{j-1}\boldsymbol{w}_j}$. The deflations, $\boldsymbol{x}_j$, are uncorrelated with the components t_j; therefore, t_j's are also uncorrelated with one another. Hence, we can replace $\boldsymbol{x}_j$ by $\boldsymbol{x}$, and y_j by y.

A recurrence relation for $\boldsymbol{w}_j$ can then be defined as

$$\begin{aligned}\boldsymbol{w}_{j+1} &= \text{cov}(\boldsymbol{x} - t_j\boldsymbol{p}_j,\ y) = \text{cov}(\boldsymbol{x}\boldsymbol{Q}_{\boldsymbol{x}\boldsymbol{w}_j}, y)\\ &= \text{cov}(\boldsymbol{x} - \boldsymbol{x}\boldsymbol{x}^T\boldsymbol{w}_j(\boldsymbol{w}_j^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{w}_j)^{-1}\boldsymbol{w}_j^T\boldsymbol{x}, y)\\ &=\boldsymbol{\sigma}_{\boldsymbol{x}y} - \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_j(\boldsymbol{w}_j^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_j)^{-1}\boldsymbol{w}_j^T\boldsymbol{\sigma}_{\boldsymbol{x}y}\end{aligned} \tag{3.35}$$

Let $\boldsymbol{\mathcal{W}}_q \in \mathbb{R}^{p\times q}$ be the span of $\boldsymbol{W}_q = (\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_q)$, where q is the smallest integer such that $\boldsymbol{w}_{q+1} = 0$. Then $\boldsymbol{\mathcal{W}}$ also span the Krylov subspace (Manne, 1987). That is, $\boldsymbol{w}_j$ can be written as a linear combination of $\boldsymbol{\sigma}_{\boldsymbol{x}y}, \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\sigma}_{\boldsymbol{x}y}, \boldsymbol{\Sigma}^2_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\sigma}_{\boldsymbol{x}y}, ..., \boldsymbol{\Sigma}^{q-1}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\sigma}_{\boldsymbol{x}y}$. Starting from Equation (3.35) $\boldsymbol{w}_1 = \boldsymbol{\sigma}_{\boldsymbol{x}y} \in \text{span}(\boldsymbol{\sigma}_{\boldsymbol{x}y})$ with $\boldsymbol{w}_0 = \boldsymbol{0}$. The second weight $\boldsymbol{w}_2$ is

$$\boldsymbol{w}_2 = \boldsymbol{\sigma}_{\boldsymbol{x}y} - \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_1(\boldsymbol{w}_1^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_1)^{-1}\boldsymbol{w}_1^T\boldsymbol{\sigma}_{\boldsymbol{x}y},$$

where $\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_1(\boldsymbol{w}_1^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_1)^{-1}\boldsymbol{w}_1^T\boldsymbol{\sigma}_{\boldsymbol{x}y}$ is $\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}$ multiplied by a linear combination $\boldsymbol{w}_1$, such that

$$\boldsymbol{w}_2 = \boldsymbol{\sigma}_{\boldsymbol{x}y} - \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\tilde{\boldsymbol{w}}_1 \in \text{span}(\boldsymbol{\sigma}_{\boldsymbol{x}y}, \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\sigma}_{\boldsymbol{x}y}). \tag{3.36}$$

Likewise,

$$\boldsymbol{w}_3 = \boldsymbol{\sigma}_{\boldsymbol{x}y} - \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_2(\boldsymbol{w}_2^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_2)^{-1}\boldsymbol{w}_2^T\boldsymbol{\sigma}_{\boldsymbol{x}y},$$

where $\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_2(\boldsymbol{w}_2^T\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{w}_2)^{-1}\boldsymbol{w}_2^T\boldsymbol{\sigma}_{\boldsymbol{x}y}$ is $\boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}$ multiplied by a linear combination $\boldsymbol{w}_2$. That is,

$$\begin{aligned}\boldsymbol{w}_3 &= \boldsymbol{\sigma}_{\boldsymbol{x}y} - \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\tilde{\boldsymbol{w}}_2 = \boldsymbol{\sigma}_{\boldsymbol{x}y} - \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}(\boldsymbol{\sigma}_{\boldsymbol{x}y} - \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\tilde{\boldsymbol{w}}_1)\\ &= \boldsymbol{\sigma}_{\boldsymbol{x}y} - \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\sigma}_{\boldsymbol{x}y} + \boldsymbol{\Sigma}^2_{\boldsymbol{x}\boldsymbol{x}}\tilde{\boldsymbol{w}}_1 \in \text{span}(\boldsymbol{\sigma}_{\boldsymbol{x}y}, \boldsymbol{\Sigma}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\sigma}_{\boldsymbol{x}y}, \boldsymbol{\Sigma}^2_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\sigma}_{\boldsymbol{x}y})\end{aligned} \tag{3.37}$$

This continues until the qth weight. The recurrence relation shows that $\boldsymbol{\mathcal{W}}_1 \subset \boldsymbol{\mathcal{W}}_2 \subset \cdots \subset \boldsymbol{\mathcal{W}}_q = \boldsymbol{\mathcal{K}}_q(\boldsymbol{\sigma}_{\boldsymbol{xy}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}})$ (that is, $\text{span}(\boldsymbol{W}_q) = \text{span}(\boldsymbol{G}_1)$). This shows that the goal of NIPALS is to determine basis of the Krylov subspace which can be used to find parameters of the Krylov model and RC model.

Connection between NIPALS and factor method

The connection between NIPALS and factor method can be made by noting that the weights, $\boldsymbol{w}_j$, from NIPALS and vectors $\boldsymbol{r}_{1j}$ for factor method span the same subspace. In Section 3.4.1, we mentioned that the weights are proportional to the loadings; $\boldsymbol{w}_j \propto \boldsymbol{p}_j$. That is,

$$\begin{aligned}\boldsymbol{p}_j = \text{cov}(\boldsymbol{x}, t_j) &= \text{cov}\left(\boldsymbol{x}, \frac{\boldsymbol{x}^T\boldsymbol{w}_j}{\|\boldsymbol{x}^T\boldsymbol{w}_j\|}\right)\\ &\propto \boldsymbol{x}\boldsymbol{w}_j^T\boldsymbol{x} = \boldsymbol{x}\frac{\text{cov}(\boldsymbol{x}, y)^T}{\|\,\text{cov}(\boldsymbol{x}, y)\|}\boldsymbol{x}\\ &\propto \frac{\text{cov}(\boldsymbol{x}, y)^T}{\|\,\text{cov}(\boldsymbol{x}, y)\|} = \boldsymbol{w}_j^T.\end{aligned}$$

Thus, from Equation (3.26) we have

$$\begin{aligned}\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}} =& \boldsymbol{p}_1 t_1 + \boldsymbol{p}_2 t_2 + \cdots + \boldsymbol{p}_q t_q\\ &\propto \boldsymbol{w}_1 t_1 + \boldsymbol{w}_2 t_2 + \cdots + \boldsymbol{w}_q t_q\\ &\propto \boldsymbol{w}_1\boldsymbol{w}_1^T\boldsymbol{x} + \boldsymbol{w}_2\boldsymbol{w}_2^T\boldsymbol{x} + \cdots + \boldsymbol{w}_q\boldsymbol{w}_q^T\boldsymbol{x}\\ &= \boldsymbol{W}_q\boldsymbol{W}_q^T\boldsymbol{x} = \boldsymbol{W}_q\boldsymbol{t},\end{aligned} \tag{3.38}$$

where $\boldsymbol{W}_q\boldsymbol{W}_q^T\boldsymbol{x}$ is a projection of $\boldsymbol{x}$ onto the column space of $\boldsymbol{W}_q$. It follows that Equation 3.38 and the second term of Equation 3.28 are both projections of $\boldsymbol{x}$ onto a subspace, where the matrices $\boldsymbol{R}_1 \in \mathbb{R}^{p\times q}$ and $\boldsymbol{W}_q \in \mathbb{R}^{p\times q}$ are both orthonormal matrices and hence, span the same subspace (Helland, 1990). That is,

$$\boldsymbol{W}_q\boldsymbol{W}_q^T\boldsymbol{x} \equiv \boldsymbol{R}_1\boldsymbol{R}_1^T\boldsymbol{x}$$

Then, from Section 3.4.4, $\boldsymbol{W}_q$ span the Krylov subspace, thus, the $\text{span}(\boldsymbol{W}_q) = \text{span}(\boldsymbol{R}_1) = \mathcal{K}_q(\boldsymbol{\Sigma}_{\boldsymbol{xx}}, \boldsymbol{\sigma}_{\boldsymbol{xy}})$ and $\sum_{j=1}^q \boldsymbol{r}_{1j} t_{1j} \propto \sum_{j=1}^q \boldsymbol{w}_j t_j$. Therefore, $\boldsymbol{t} \equiv \boldsymbol{t}_1$, thus, $\boldsymbol{t}$ from NIPALS and $\boldsymbol{t}_1$ from factor method span the same subspace. That is,

$\text{span}(\{\boldsymbol{t}_{1j} : j = 1, ..., q\}) = \text{span}(\{\boldsymbol{t}_j : j = 1, ..., q\})$. In other words, the orthogonal matrices from NIPALS and factor method span the same subspace, and their scores span the same subspace.

3.5 Sample PLS1 regression

In previous sections of this chapter, the linear predictor in Equation (3.2), estimators in Equations (3.16) and (3.23), algorithms (NIPALS, factor method and GS algorithms), and connections between algorithms have been made with regard to population parameters which are often unknown and must be estimated. In practice, population parameters must be replaced by sample parameters. That is, population covariances, variances and means are replaced by sample covariances, variances, and means, respectively. For instance, the linear predictor is replace by

$$\hat{y} = \bar{y} + \hat{\boldsymbol{\beta}}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}), \tag{3.39}$$

where $\bar{y}$ is the sample mean of y, $\bar{\boldsymbol{x}}$ the sample mean of $\boldsymbol{x}$, and $\hat{\boldsymbol{\beta}} = \boldsymbol{S}_{\boldsymbol{xx}}^{-1}\boldsymbol{s}_{\boldsymbol{x}y}$ is the vector of estimated regression coefficients, with $\boldsymbol{S}_{\boldsymbol{xx}}$ the estimated sample covariance matrix of $\boldsymbol{x}$ and $\boldsymbol{s}_{\boldsymbol{x}y}$ the estimated vector of sample covariance vector between $\boldsymbol{x}$ and y. The vector $\hat{\boldsymbol{\beta}}$ is the best solution when the sample size, n, is larger than the number of predictor variables, however, when $p > n$ or predictor variables are nearly multicollinear $\hat{\boldsymbol{\beta}}$ might be unstable or may not exist. In such cases the estimators in Equations (3.11) and (3.24) are preferred, and the algorithms discussed in Section 3.4 can be used to calculate the estimates of Equations (3.11) and (3.24). For instance, the $\boldsymbol{G}_1$ can be estimated as

$$\hat{\boldsymbol{G}}_1 = (\boldsymbol{s}_{\boldsymbol{x}y}, \boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{s}_{\boldsymbol{x}y}, ..., \boldsymbol{S}_{\boldsymbol{xx}}^{q-1}\boldsymbol{s}_{\boldsymbol{x}y}) \in \mathbb{R}^{p\times q}.$$

or its linear combination (Naik and Tsai, 2000) and (Helland, 1988), with solution $\hat{\boldsymbol{\beta}}_{\hat{\boldsymbol{G}}_1}$ if q is the true dimension, where

$$\hat{\boldsymbol{\beta}}_{\hat{\boldsymbol{G}}_1} = \hat{\boldsymbol{G}}_1(\hat{\boldsymbol{G}}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\hat{\boldsymbol{G}}_1)^{-1}\hat{\boldsymbol{G}}_1^T\boldsymbol{s}_{\boldsymbol{x}y}. \tag{3.40}$$

In addition, the recurrence relation from the NIPALS is

$$\hat{\boldsymbol{w}}_{j+1} = \boldsymbol{s}_{\boldsymbol{x}y} - \boldsymbol{S}_{\boldsymbol{xx}}\hat{\boldsymbol{w}}_j(\hat{\boldsymbol{w}}_j^T\boldsymbol{S}_{\boldsymbol{xx}}\hat{\boldsymbol{w}}_j)^{-1}\hat{\boldsymbol{w}}_j^T\boldsymbol{s}_{\boldsymbol{x}y}, \tag{3.41}$$

where $\hat{\boldsymbol{W}}_q = (\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2, ..., \hat{\boldsymbol{w}}_q) \in \text{span}(\hat{\boldsymbol{G}}_1)$ as shown in Section 3.4.4.

The number of relevant components, q, was assumed to be known, but must be determined in the sample, e.g by deciding when to stop the algorithm. Cross-validation (CV) is the method commonly used, a description of CV was given in chapter 2. The CV for PLS1 can be performed by replacing the vector of tuning parameters in Section 2.3.5 with $q \in \{1, \ldots, p\}$. Then $\hat{q} = \arg\min_q \text{CV}(q)$.

3.6 Conclusion

The chapter described a RC models for univariate regression and connected the model to the Krylov matrix which provides a different characterization to the RC model. Further, several PLS1 algorithms for estimating the parameters of the RC model were reviewed. And connections between the algorithm were made; which helped to show that PLS1 algorithms estimates different bases of the Krylov subspace. Moreover, the RC and Krylov models sheds more light into PLS1.

Chapter 4

Multivariate Partial Least Squares Regression

4.1 Introduction

Chapter 3 gives an introduction of partial least squares, particulary univariate (single-response) partial least squares (PLS1) regression. PLS1 considers with one response variable and multivariate predictor variables, and a few linear combinations of the predictor variables (relevant components) are used to predict a single response variable. In the current chapter, the focus is multivariate (multiple-response) partial least squares (also known as PLS2) regression, where a few relevant components are used to predict multiple response variables together. The difference between PLS1 and PLS2 is that the former deals with a single response while the latter considers more than one response variables together (Indahl et al., 2009). PLS2 is a family of methods for extracting a few relevant components of the predictor variables that hold all the information for predicting multiple response variables together. If few relevant components are extracted for each response variable, the relevant components for one response variable may be different from that of another response variable (Cook et al., 2013), in which case it is recommended to use PLS2 (Indahl et al., 2009). Moreover, PLS2 can give a collective understanding compared to several PLS1 regression, and several PLS1 regressions will give many parameters which will be difficult to interpret (Wold et al., 2001). However, in PLS2 a small number of linear combinations of both the response and predictor variables can be extracted (Rännar et al., 1995).

PLS is related to other dimension reduction techniques such as principal components regression (PCR) and canonical correlation analysis (CCA). The difference between them is that PLS finds relevant components that maximize covariance between the response ($\boldsymbol{y}$) and predictor ($\boldsymbol{x}$) variables (Höskuldsson, 1988), CCA finds relevant components between response and predictor variables that exhibit maximum correlation (Borga et al., 1997; Kruger and Qin, 2003; Rosipal and Krämer, 2005), and PCR uses the relevant components of maximum variance from the predictor variables to predict the response variable in a linear regression. PLS and CCA both involve finding relevant components between two sets of variable under certain criteria, whereas PCR focuses on one set of variables. In determining relevant components, CCA requires that transformation matrices and relevant components of $\boldsymbol{x}$ and $\boldsymbol{y}$ be orthonormal, while PLS requires that only the transformation matrices be orthonormal (Kruger and Qin, 2003). Besides, PLS uses a deflation procedure to determine subsequent relevant components (see Subsection 3.4.1), while CCA depends on additional constraints.

Furthermore, there is a version of PLS called orthogonal PLS (O-PLS) (Eriksson et al., n.d.; Trygg and Wold, 2002) that require the relevant components to also be orthonormal. The O-PLS and PLS have similar prediction performance but the O-PLS uses a smaller number of components compared to PLS2 (Biagioni et al., 2011). An extension of the O-PLS is the O2-PLS. The O2-PLS applies O-PLS to $\boldsymbol{x}$ and $\boldsymbol{y}$ by treating them as response and predictor variables one at a time in other to extract $\boldsymbol{x}$-orthogonal variation in $\boldsymbol{y}$ and $\boldsymbol{y}$-orthogonal variation in $\boldsymbol{x}$ (Biagioni et al., 2011) used for relating $\boldsymbol{x}$ and $\boldsymbol{y}$. Moreover, O-PLS is different from CCA because O-PLS extracts components in $\boldsymbol{x}$ related to $\boldsymbol{y}$ and components specific to $\boldsymbol{x}$ alone. Since PLS sometimes has better prediction performance compare to PCR (Almøy, 1996) and similar prediction performance with O2-PLS we focus on PLS in this chapter.

As previously mentioned, PLS is a family of algorithms, and the PLS estimator is an estimator of the parameters of a statistical model. This chapter also introduces *relevant components models* that are characterized by certain restrictions on the joint covariance matrix of $\boldsymbol{x}$ and $\boldsymbol{y}$. These restrictions address complexities that accompany large and nearly (or perfectly) multicollinear variables. Furthermore, in this chapter methods for estimating parameters of different relevant components models are presented, and simulation studies are performed to compare the prediction performance of the different methods considered.

4.2 Relevant components models

Multivariate multiple linear regression (MMLR) is used for modelling the relationship between multiple response variables and multiple predictor variables, predicting new responses, and comparing effects of predictor variables on individual response variables. MMLR is different from multiple linear regression (MLR) analysis which has a single response variable and multiple predictor variables. Each response variable can be fitted separately, however MMLR enables the understanding of correlation patterns and inferences based on one response variable alone are sometimes biased.

Let $\boldsymbol{y} \in \mathbb{R}^r$ and $\boldsymbol{x} \in \mathbb{R}^p$ be vectors of response and predictor variables, respectively, that follow a joint multivariate normal distribution given by

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} \sim \mathbf{N}_{p+r} \left[\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu_x} \\ \boldsymbol{\mu_y} \end{pmatrix}, \; \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma_{xx}} & \boldsymbol{\Sigma_{xy}} \\ \boldsymbol{\Sigma}_{\boldsymbol{xy}}^T & \boldsymbol{\Sigma_{yy}} \end{pmatrix} \right], \tag{4.1}$$

where $\boldsymbol{\Sigma_{yy}} \in \mathbb{R}^{r \times r}$ is the covariance matrix of $\boldsymbol{y}$, $\boldsymbol{\Sigma_{xx}} \in \mathbb{R}^{p \times p}$ the covariance matrix of $\boldsymbol{x}$, and $\boldsymbol{\Sigma_{xy}} \in \mathbb{R}^{p \times r}$ the covariance matrix between $\boldsymbol{x}$ and $\boldsymbol{y}$. Given model (4.1) with all parameters known, the best linear predictor under quadratic loss, for a new vector of predictor variables is

$$\boldsymbol{y} = \mathrm{E}(\boldsymbol{y}|\boldsymbol{x}) = \boldsymbol{\mu_y} + \boldsymbol{B}^T(\boldsymbol{x} - \boldsymbol{\mu_x}), \tag{4.2}$$

where $\mathrm{E}(\boldsymbol{y}|\boldsymbol{x})$ is the conditional expectation of $\boldsymbol{y}$ given $\boldsymbol{x}$, and $\boldsymbol{B} = \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{\Sigma_{xy}} \in \mathbb{R}^{p \times r}$ is the matrix of population regression coefficients.

The relevant component model presented in this section assumes a restriction on the joint covariance matrix of $\boldsymbol{x}$ and $\boldsymbol{y}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{(p+r)\times(p+r)}$. We call the model which restricts the covariance matrix of $\boldsymbol{x}$ alone the relevant component (RC) model of order (r, q) denoted by $\mathrm{RC}_{\boldsymbol{x}}$, where r is the dimension of the response variable and q $(1 \leq q \leq p)$ represents the number of linear combinations taken from the predictor variables. The RC$_{\boldsymbol{x}}$ model reduces $\boldsymbol{\Sigma}$ with no restriction on $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$. The model which restricts both $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$ is the relevant component model of order (m, q) denoted by RC$_{\boldsymbol{yx}}$, where m $(1 \leq m \leq r)$ represents the number of linear combinations taken from the response variables. The RC$_{\boldsymbol{yx}}$ model is a sub-model of the RC$_{\boldsymbol{x}}$ model. Note, If $q = p$ there is no restriction on $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ and if $m = r$ there is no restriction on $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$. For instance, the model in (4.1) is the unrestricted model. Table 4.2 shows some of the relevant component models.

Model	Order
RC$_{\boldsymbol{x}}$	$\{r, q \ : \ m = r, \quad 1 \leq q \leq p\}$
RC$_{\boldsymbol{yx}}$	$\{m, q \ : \ 1 \leq m \leq r, \quad 1 \leq q \leq p\}$
Unrestricted	$\{r, p \ : \ m = r, \quad q = p\}$

Table 4.1: *Some relevant components models.*

4.2.1 Relevant components model of order (m, q)

The relevant components model of order (m, q), RC$_{\boldsymbol{yx}}$, assumes that after a rotation, $\boldsymbol{x}$ and $\boldsymbol{y}$ are each split into two parts such that only one part of the split in $\boldsymbol{y}$ and one part of the split in $\boldsymbol{x}$ are required for describing the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$. The splits of $\boldsymbol{x}$ and $\boldsymbol{y}$ which are important for describing a relationship between the variables will be called *relevant components*, and holds all the needed information. The other parts of $\boldsymbol{x}$ and $\boldsymbol{y}$ will be called the *irrelevant components*.

Let $\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{p\times p}$ and $\boldsymbol{\Upsilon} = (\boldsymbol{\Upsilon}_1, \boldsymbol{\Upsilon}_0) \in \mathbb{R}^{r\times r}$ be two orthogonal matrices, such that $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{p\times q}$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-q)}$, $\boldsymbol{\Upsilon}_1 \in \mathbb{R}^{r\times m}$ and $\boldsymbol{\Upsilon}_0 \in \mathbb{R}^{r\times(r-m)}$ are full rank orthonormal matrices ($\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 = \boldsymbol{I}_q$, $\boldsymbol{\Gamma}_0^T\boldsymbol{\Gamma}_0 = \boldsymbol{I}_{p-q}$, $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_0 = \boldsymbol{0}_{p\times(p-q)}$, $\boldsymbol{\Upsilon}_1^T\boldsymbol{\Upsilon}_1 = \boldsymbol{I}_m$, $\boldsymbol{\Upsilon}_0^T\boldsymbol{\Upsilon}_0 = \boldsymbol{I}_{r-m}$, and $\boldsymbol{\Upsilon}_1^T\boldsymbol{\Upsilon}_0 = \boldsymbol{0}_{r\times(r-m)}$). The dimensions of the relevant of $\boldsymbol{x}$ and $\boldsymbol{y}$ are q and m, respectively, and the dimensions of their irrelevant components are $p - q$ and $r - m$, $(p \geq q$ and $r \geq m)$. Define the following linear transformations of $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\boldsymbol{t}_1 = \boldsymbol{\Gamma}_1^T\boldsymbol{x}, \quad \boldsymbol{t}_0 = \boldsymbol{\Gamma}_0^T\boldsymbol{x}, \quad \boldsymbol{v}_1 = \boldsymbol{\Upsilon}_1^T\boldsymbol{y} \quad \text{and} \quad \boldsymbol{v}_0 = \boldsymbol{\Upsilon}_0^T\boldsymbol{y}.$$

Then $\boldsymbol{x}$ and $\boldsymbol{y}$ can each be reconstructed such that

$$\boldsymbol{x} = \boldsymbol{\Gamma}_1\boldsymbol{t}_1 + \boldsymbol{\Gamma}_0\boldsymbol{t}_0, \quad \text{and} \quad \boldsymbol{y} = \boldsymbol{\Upsilon}_1\boldsymbol{v}_1 + \boldsymbol{\Upsilon}_0\boldsymbol{v}_0. \tag{4.3}$$

Assume the rotation of $\boldsymbol{x}$ and $\boldsymbol{y}$ partitions the covariance matrices, $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$, $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{xy}}$ such that the covariance matrix of $\boldsymbol{x}$, $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$, after rotation is

$$\boldsymbol{\Gamma}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 \end{pmatrix}, \tag{4.4}$$

where $\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 \in \mathbb{R}^{q\times q}$ and $\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 \in \mathbb{R}^{(p-q)\times(p-q)}$ are the covariance matrices of $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$, respectively. The zeros in Equation (4.4) are restrictions imposed by the RC$_{\boldsymbol{yx}}$ model, and indicates that $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$ are uncorrelated. The covariance matrix of $\boldsymbol{y}$, $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$, after rotation is

$$\begin{pmatrix} \boldsymbol{\Upsilon}_1^T\boldsymbol{\Sigma}_{\boldsymbol{yy}}\boldsymbol{\Upsilon}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Upsilon}_0^T\boldsymbol{\Sigma}_{\boldsymbol{yy}}\boldsymbol{\Upsilon}_0 \end{pmatrix}, \tag{4.5}$$

where $\boldsymbol{\Upsilon}_1^T\boldsymbol{\Sigma}_{\boldsymbol{yy}}\boldsymbol{\Upsilon}_1 \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Upsilon}_0^T\boldsymbol{\Sigma}_{\boldsymbol{yy}}\boldsymbol{\Upsilon}_0 \in \mathbb{R}^{(r-m)\times(r-m)}$ are the covariance matrices of $\boldsymbol{v}_1$ and $\boldsymbol{v}_0$, respectively. The zeros in Equation (4.5) are restrictions imposed by the RC$_{\boldsymbol{yx}}$ model, and indicates that $\boldsymbol{v}_1$ and $\boldsymbol{v}_0$ are uncorrelated. Also, the covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$, $\boldsymbol{\Sigma}_{\boldsymbol{xy}}$, after rotation is

$$\begin{pmatrix} \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Upsilon}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{4.6}$$

where $\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Upsilon}_1 \in \mathbb{R}^{q\times m}$ is the covariance matrix between $\boldsymbol{t}_1$ and $\boldsymbol{v}_1$. The zeros in Equation (4.6) are also restrictions imposed by the RC$_{\boldsymbol{xy}}$ model, and indicates that $\boldsymbol{v}_1$ and $\boldsymbol{t}_0$ are uncorrelated, $\boldsymbol{t}_1$ and $\boldsymbol{v}_0$ are uncorrelated, and $\boldsymbol{v}_0$ and $\boldsymbol{t}_0$ are uncorrelated. Then, $\boldsymbol{v}_1$ and $\boldsymbol{t}_1$ are the relevant components of $\boldsymbol{y}$ and $\boldsymbol{x}$, respectively. The covariance matrices in Equations (4.4), (4.5) and (4.6) can be combined to form

$$\boldsymbol{\Sigma}_{\boldsymbol{\Gamma},\boldsymbol{\Upsilon}} = \begin{pmatrix} \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 & \mathbf{0} & \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Upsilon}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\Upsilon}_1^T\boldsymbol{\Sigma}_{\boldsymbol{yx}}\boldsymbol{\Gamma}_1 & \mathbf{0} & \boldsymbol{\Upsilon}_1^T\boldsymbol{\Sigma}_{\boldsymbol{yy}}\boldsymbol{\Upsilon}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Upsilon}_0^T\boldsymbol{\Sigma}_{\boldsymbol{yy}}\boldsymbol{\Upsilon}_0 \end{pmatrix}, \tag{4.7}$$

The RC$_{\boldsymbol{yx}}$ model holds if $\boldsymbol{\Sigma}$ has a block decomposition of order (m, q) and does not have a block decomposition for any smaller values of the indices. The $\boldsymbol{\Sigma}$ has a block decomposition of order (m, q) if there exist orthogonal matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Upsilon}$ such that

(4.7) holds. As a result

$$\begin{aligned} (a)\ \boldsymbol{\Sigma}_{\boldsymbol{xx}} &= \boldsymbol{\Gamma}_1 \operatorname{cov}(\boldsymbol{t}_1)\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0 \operatorname{cov}(\boldsymbol{t}_0)\boldsymbol{\Gamma}_0^T \\ &= \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0}\boldsymbol{\Gamma}_0^T \\ (b)\ \boldsymbol{\Sigma}_{\boldsymbol{yy}} &= \boldsymbol{\Upsilon}_1 \operatorname{cov}(\boldsymbol{v}_1)\boldsymbol{\Upsilon}_1^T + \boldsymbol{\Upsilon}_0 \operatorname{cov}(\boldsymbol{v}_0)\boldsymbol{\Upsilon}_0^T \\ &= \boldsymbol{\Upsilon}_1\boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_1}\boldsymbol{\Upsilon}_1^T + \boldsymbol{\Upsilon}_0\boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_0}\boldsymbol{\Upsilon}_0^T \\ \text{and}\quad (c)\ \boldsymbol{\Sigma}_{\boldsymbol{xy}} &= \boldsymbol{\Gamma}_1 \operatorname{cov}(\boldsymbol{t}_1, \boldsymbol{v}_1)\boldsymbol{\Upsilon}_1^T = \boldsymbol{\Gamma}_1\boldsymbol{\Psi}\boldsymbol{\Upsilon}_1^T, \end{aligned} \tag{4.8}$$

where $\boldsymbol{t}_1$ and $\boldsymbol{v}_1$ are called relevant components of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. The $\boldsymbol{t}_0$ and $\boldsymbol{v}_0$ are irrelevant components, $\boldsymbol{\Psi} = \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Upsilon}_1 \in \mathbb{R}^{q\times m}$ is the covariance between $\boldsymbol{t}_1$ and $\boldsymbol{v}_1$, the parameters $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1} = \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 \in \mathbb{R}^{q\times q}$, $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0} = \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 \in \mathbb{R}^{(p-q)\times(p-q)}$, $\boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_1} = \boldsymbol{\Upsilon}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Upsilon}_1 \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_0} = \boldsymbol{\Upsilon}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Upsilon}_0 \in \mathbb{R}^{(r-m)\times(r-m)}$ are symmetric covariance matrices of $\boldsymbol{t}_1$, $\boldsymbol{t}_0$, $\boldsymbol{v}_1$, and $\boldsymbol{v}_0$, respectively. Note that, all the information needed for describing a relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ is contained in $\boldsymbol{t}_1$ and $\boldsymbol{v}_1$. Then, the joint covariance matrix of $\boldsymbol{x}$ and $\boldsymbol{y}$ can be rewritten as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0}\boldsymbol{\Gamma}_0^T & \boldsymbol{\Gamma}_1\boldsymbol{\Psi}\boldsymbol{\Upsilon}_1^T \\ \boldsymbol{\Upsilon}_1\boldsymbol{\Psi}\boldsymbol{\Gamma}_1^T & \boldsymbol{\Upsilon}_1\boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_1}\boldsymbol{\Upsilon}_1^T + \boldsymbol{\Upsilon}_0\boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_0}\boldsymbol{\Upsilon}_0^T \end{pmatrix} \tag{4.9}$$

If the RC$_{\boldsymbol{yx}}$ model holds, the linear predictor in Equation (4.2) is simplified as

$$\boldsymbol{v}_1 = \boldsymbol{B}_{m,q}^T\boldsymbol{t}_1, \tag{4.10}$$

where $\boldsymbol{B}_{m,q} \in \mathbb{R}^{q\times m}$ is the matrix of regression coefficient of $\boldsymbol{v}_1$ on $\boldsymbol{t}_1$.

Isotropic RC$_{\boldsymbol{yx}}$ models

Various isotropic RC$_{\boldsymbol{yx}}$ models can be describes by introducing additional restrictions in the model (4.7). The isotropic models are related to the work of el Bouhaddani et al. (2018). Assume that after rotating $\boldsymbol{x}$ and $\boldsymbol{y}$ the RC$_{\boldsymbol{yx}}$ model holds and in addition, that the covariance matrices of the irrelevant components of $\boldsymbol{x}$ and/or $\boldsymbol{y}$ are isotropic, i.e,

$$\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0} = \sigma\boldsymbol{I}_{p-q} \quad \text{and} \quad \boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_0} = \sigma^*\boldsymbol{I}_{r-m}, \tag{4.11}$$

where σ and σ^* are constants. Note that a *predictor isotropic* RC$_{\boldsymbol{yx}}$ model can be described by requiring that only the covariance matrix of $\boldsymbol{t}_0$ is isotropic. Also, a *response isotropic* RC$_{\boldsymbol{yx}}$ model can be described by requiring that only the covariance matrix of $\boldsymbol{v}_0$ is isotropic.

Assume the covariance matrices of $\boldsymbol{v}_0$ and $\boldsymbol{t}_0$ are isotropic; then $\boldsymbol{\Sigma}$ is given by

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T + \sigma\boldsymbol{\Gamma}_0\boldsymbol{\Gamma}_0^T & \boldsymbol{\Gamma}_1\boldsymbol{\Psi}\boldsymbol{\Upsilon}_1^T \\ \boldsymbol{\Upsilon}_1\boldsymbol{\Psi}\boldsymbol{\Gamma}_1^T & \boldsymbol{\Upsilon}_1\boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_1}\boldsymbol{\Upsilon}_1^T + \sigma^*\boldsymbol{\Upsilon}_0\boldsymbol{\Upsilon}_0^T \end{pmatrix}. \tag{4.12}$$

These additional restrictions simplify the structure of the joint covariance matrix, $\boldsymbol{\Sigma}$. Note that the relationship between $\boldsymbol{v}_1$ and $\boldsymbol{t}_1$ can be described as

$$\boldsymbol{v}_1 = \boldsymbol{B}_{m,q}^T\boldsymbol{t}_1 + \boldsymbol{h}, \tag{4.13}$$

where $\boldsymbol{B}_{m,q} \in \mathbb{R}^{q\times m}$ is the matrix of regression coefficients of $\boldsymbol{v}_1$ on $\boldsymbol{t}_1$, and $\boldsymbol{h} \in \mathbb{R}^m$ is uncorrelated with $\boldsymbol{t}_1$.Then the following covariance matrices can be reconstructed:

$$\begin{aligned} \operatorname{cov}(\boldsymbol{x}, \boldsymbol{y}) &= \operatorname{cov}\left(\boldsymbol{\Gamma}_1\boldsymbol{t}_1 + \boldsymbol{\Gamma}_0\boldsymbol{t}_0, \boldsymbol{\Upsilon}_1\boldsymbol{v}_1 + \boldsymbol{\Upsilon}_0\boldsymbol{v}_0\right) \\ &= \operatorname{cov}\Big(\boldsymbol{\Gamma}_1\boldsymbol{t}_1 + \boldsymbol{\Gamma}_0\boldsymbol{t}_0, \boldsymbol{\Upsilon}_1[\boldsymbol{B}_{m,q}^T\boldsymbol{t}_1 + \boldsymbol{h}] + \boldsymbol{\Upsilon}_0\boldsymbol{v}_0\Big) \\ &= \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{B}_{m,q}\boldsymbol{\Upsilon}_1^T, \end{aligned} \tag{4.14}$$

and

$$\begin{aligned} \operatorname{cov}(\boldsymbol{y}) &= \operatorname{cov}\left(\boldsymbol{\Upsilon}_1\boldsymbol{v}_1 + \boldsymbol{\Upsilon}_0\boldsymbol{v}_0\right) \\ &= \operatorname{cov}\left(\boldsymbol{\Upsilon}_1[\boldsymbol{B}_{m,q}\boldsymbol{t}_1 + \boldsymbol{h}] + \boldsymbol{\Upsilon}_0\boldsymbol{v}_0\right) \\ &= \boldsymbol{\Upsilon}_1(\boldsymbol{B}_{m,q}^T\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{B}_{m,q} + \boldsymbol{\Omega}_h)\boldsymbol{\Upsilon}_1^T + \sigma^*\boldsymbol{I}_r, \end{aligned} \tag{4.15}$$

where $\boldsymbol{\Omega}_h = \operatorname{diag}(\sigma_{11}, \sigma_{22}, \ldots, \sigma_{mm}) \in \mathbb{R}^{m\times m}$ is the covariance matrix of $\boldsymbol{h}$. Then the joint covariance matrix with isotropic errors is

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T + \sigma\boldsymbol{\Gamma}_0\boldsymbol{\Gamma}_0^T & \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{B}_{m,q}\boldsymbol{\Upsilon}_1^T \\ \boldsymbol{\Upsilon}_1\boldsymbol{B}_{m,q}\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T & \boldsymbol{\Upsilon}_1(\boldsymbol{B}_{m,q}^T\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{B}_{m,q} + \boldsymbol{\Omega}_h)\boldsymbol{\Upsilon}_1^T + \sigma^*\boldsymbol{\Upsilon}_0\boldsymbol{\Upsilon}_0^T \end{pmatrix}, \tag{4.16}$$

where $\boldsymbol{B}_{m,q}^T\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{B}_{m,q} + \boldsymbol{\Omega}_h \equiv \boldsymbol{\Omega}_{\boldsymbol{\Upsilon}_1}$ and $\boldsymbol{B}_{m,q}\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1} \equiv \boldsymbol{\Psi}$.

A submodel of the isotropic RC$_{\boldsymbol{yx}}$ model can be constructed by requiring that $q = m$, $\boldsymbol{B}_{q,q} \in \mathbb{R}^{q\times q}$ is diagonal and $\boldsymbol{\Omega}_h$ is isotropic; which is identical to the model proposed by el Bouhaddani et al. (2018).

4.2.2 Relevant components model of order (r, q)

The relevant components model of order (r, q), RC$_{\boldsymbol{x}}$, assumes that after a rotation of $\boldsymbol{x}$, $\boldsymbol{x}$ is split into two parts such that all the information required to predict $\boldsymbol{y}$ is contained in only one part of the split. Assume the dimensions of the components are known so that the relevant components have dimension q and the irrelevant components have dimension $p-q$.

Let $\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{p\times p}$ be an orthogonal matrix, such that $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{p\times q}$ and $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-q)}$ are known semi-orthogonal matrices of full column rank, with $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 = \boldsymbol{I}_q$, $\boldsymbol{\Gamma}_0^T\boldsymbol{\Gamma}_0 = \boldsymbol{I}_{p-q}$, and $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_0 = \mathbf{0}_{p\times(p-q)}$. Starting from $\boldsymbol{x}$, define two linear transformations $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$ to be

$$\boldsymbol{t}_1 = \boldsymbol{\Gamma}_1^T\boldsymbol{x} \quad \text{and} \quad \boldsymbol{t}_0 = \boldsymbol{\Gamma}_0^T\boldsymbol{x}. \tag{4.17}$$

Conversely, $\boldsymbol{x}$ can be recovered from $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$ by

$$\boldsymbol{x} = \boldsymbol{\Gamma}_1\boldsymbol{t}_1 + \boldsymbol{\Gamma}_0\boldsymbol{t}_0 = \boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_1^T\boldsymbol{x} + \boldsymbol{\Gamma}_0\boldsymbol{\Gamma}_0^T\boldsymbol{x}, \tag{4.18}$$

where $\boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_1^T$ is a projection matrix, and $\boldsymbol{\Gamma}_0\boldsymbol{\Gamma}_0^T = \boldsymbol{I} - \boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_1^T$ its complement. Assume that the rotation of $\boldsymbol{x}$ partitions the covariance matrix, $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$, into a block diagonal form such that

$$\boldsymbol{\Gamma}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 \end{pmatrix}, \tag{4.19}$$

where $\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 \in \mathbb{R}^{q\times q}$ and $\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 \in \mathbb{R}^{(p-q)\times(p-q)}$ are the covariance matrices of $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$, respectively. The zeros in Equation (4.19) are restrictions imposed on the RC$_{\boldsymbol{x}}$ model, and indicate that $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$ are uncorrelated. The $\boldsymbol{t}_1$ contains all the information in $\boldsymbol{x}$ for predicting $\boldsymbol{y}$. Then the joint covariance matrix, $\boldsymbol{\Sigma}$, after a rotation of $\boldsymbol{x}$ is

$$\boldsymbol{\Sigma}_{\boldsymbol{\Gamma}} = \begin{pmatrix} \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 & \mathbf{0} & \boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xy}} \\ \mathbf{0} & \boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 & \mathbf{0} \\ \boldsymbol{\Sigma}_{\boldsymbol{xy}}^T\boldsymbol{\Gamma}_1 & \mathbf{0} & \boldsymbol{\Sigma}_{\boldsymbol{yy}} \end{pmatrix}. \tag{4.20}$$

where $\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xy}} \in \mathbb{R}^{q\times r}$ the covariance between $\boldsymbol{t}_1$ and $\boldsymbol{y}$. The covariance between $\boldsymbol{t}_0$ and $\boldsymbol{y}$, i.e $\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xy}} \in \mathbb{R}^{p-q\times r} = \mathbf{0}$ indicating that $\boldsymbol{t}_0$ hold no information for describing $\boldsymbol{y}$. The RC$_{\boldsymbol{x}}$ model assumes that q is the smallest dimension for which there exist orthogonal matrix $\boldsymbol{\Gamma}$ such that (4.20) holds. As a result we have the following:

$$\begin{aligned} (a)\ \boldsymbol{\Sigma}_{\boldsymbol{xx}} &= \boldsymbol{\Gamma}_1 \operatorname{cov}(\boldsymbol{t}_1)\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0 \operatorname{cov}(\boldsymbol{t}_0)\boldsymbol{\Gamma}_0^T \\ &= \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0}\boldsymbol{\Gamma}_0^T \quad \text{and} \\ (b)\ \boldsymbol{\Sigma}_{\boldsymbol{xy}} &= \boldsymbol{\Gamma}_1 \operatorname{cov}(\boldsymbol{t}_1, \boldsymbol{y}) + \boldsymbol{\Gamma}_0 \operatorname{cov}(\boldsymbol{t}_0, \boldsymbol{y}) \\ &= \boldsymbol{\Gamma}_1\boldsymbol{\Phi} + \mathbf{0} = \boldsymbol{\Gamma}_1\boldsymbol{\Phi}, \end{aligned} \tag{4.21}$$

where the following are functions of $\boldsymbol{\Sigma}$; $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1} = \boldsymbol{\Gamma}_1^T \boldsymbol{\Sigma}_{\boldsymbol{xx}} \boldsymbol{\Gamma}_1 \in \mathbb{R}^{q \times q}$ and $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0} = \boldsymbol{\Gamma}_0^T \boldsymbol{\Sigma}_{\boldsymbol{xx}} \boldsymbol{\Gamma}_0 \in \mathbb{R}^{(p-q)\times(p-q)}$ are covariance matrices of $\boldsymbol{t}_1$ and $\boldsymbol{t}_0$, respectively, $\boldsymbol{\Phi} = \mathrm{cov}(\boldsymbol{t}_1, \boldsymbol{y}) \in \mathbb{R}^{q \times r}$ is the covariance matrix between $\boldsymbol{t}_1$ and $\boldsymbol{y}$; the $\mathbf{0}$ in Equation (4.21)(b) is because $\boldsymbol{t}_0$ and $\boldsymbol{y}$ are uncorrelated. The results in Equation (4.21) says that all the information in $\boldsymbol{x}$ required for predicting the multivariate response, $\boldsymbol{y}$, is contained in $\boldsymbol{t}_1$. Then, the joint covariance matrix $\boldsymbol{\Sigma}$ can be rewritten as (Cook et al., 2013)

$$\boldsymbol{\Sigma}_{\boldsymbol{\Gamma}} = \begin{pmatrix} \boldsymbol{\Gamma}_1 \boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1} \boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0 \boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0} \boldsymbol{\Gamma}_0^T & \boldsymbol{\Gamma}_1 \boldsymbol{\Phi} \\ \boldsymbol{\Phi}^T \boldsymbol{\Gamma}_1^T & \boldsymbol{\Sigma}_{\boldsymbol{yy}} \end{pmatrix}. \tag{4.22}$$

If the RC$_{\boldsymbol{x}}$ model holds, the linear predictor in Equation (4.2) can be reduced to essentials as

$$\boldsymbol{y} = \boldsymbol{\mu}_{\boldsymbol{y}} + \boldsymbol{A}_q^T \boldsymbol{\Gamma}_1^T (\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}), \tag{4.23}$$

with

$$\boldsymbol{A}_q = \begin{pmatrix} \boldsymbol{\Gamma}_1^T \boldsymbol{\Sigma}_{\boldsymbol{xx}} \boldsymbol{\Gamma}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Gamma}_0^T \boldsymbol{\Sigma}_{\boldsymbol{xx}} \boldsymbol{\Gamma}_0 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{\Gamma}_1^T \boldsymbol{\Sigma}_{\boldsymbol{xy}} \\ \mathbf{0} \end{pmatrix} = (\boldsymbol{\Gamma}_1^T \boldsymbol{\Sigma}_{\boldsymbol{xx}} \boldsymbol{\Gamma}_1)^{-1} \boldsymbol{\Gamma}_1 \boldsymbol{\Sigma}_{\boldsymbol{xy}} \in \mathbb{R}^{q \times r}, \tag{4.24}$$

and

$$\boldsymbol{B}_q = \boldsymbol{\Gamma}_1 (\boldsymbol{\Gamma}_1^T \boldsymbol{\Sigma}_{\boldsymbol{xx}} \boldsymbol{\Gamma}_1)^{-1} \boldsymbol{\Gamma}_1^T \boldsymbol{\Sigma}_{\boldsymbol{xy}} = \boldsymbol{\Gamma}_1 \boldsymbol{A}_q, \tag{4.25}$$

is the matrix of population regression coefficients. Note that the RC model of order (r, q) is the multivariate version of the RC model of order (q) discussed in Chapter 3.

In the next section, different methods for estimating the parameters of the RC models introduced above are reviewed, and will later be compared with regard to prediction performance.

4.3 Parameter estimation

The purpose of this section is to present methods for estimating the parameters of various relevant components models discussed in Section 4.2. The methods discussed in this section include two maximum likelihood estimation (MLE) methods and a PLS2 algorithm. In particular, the methods considered are used for estimating the parameters of the RC$_{\boldsymbol{x}}$ model and the isotropic RC$_{\boldsymbol{yx}}$ model. These methods are known in the literature as

(a) the envelope method (EPLS) (Cook et al., 2013),

(b) expectation-maximization (EM) algorithm for PLS (EM-PLS) (el Bouhaddani et al., 2018), and

(c) statistically inspired modification of PLS (SIMPLS) (De Jong, 1993).

The envelope method is an MLE method used for estimating the parameters of the RC$_{\boldsymbol{x}}$ model, and can only be used when the sample size, n, is larger than the number of predictor variables, p. The EM-PLS method is also an MLE method used for estimating the parameters of the isotropic RC$_{\boldsymbol{yx}}$ model. The SIMPLS is an algorithm which can be used for estimating the parameters of both the RC$_{\boldsymbol{x}}$ and RC$_{\boldsymbol{yx}}$ models. The SIMPLS algorithm can be used when $r \geq n+1$ and/or $p \geq n+1$. Note that the SIMPLS estimate is different from the MLE.

Let $(\boldsymbol{x}_i^T, \boldsymbol{y}_i^T)$, $i = 1, \dots, n$, be independent and identically distributed random samples of size n, and let the number of relevant components, m and q, be known. Denote by $\boldsymbol{S}_{\boldsymbol{xx}}$, $\boldsymbol{S}_{\boldsymbol{yy}}$, $\boldsymbol{S}_{\boldsymbol{xy}}$, and $\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{x}}$ the sample covariance matrices of $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$, $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$, $\boldsymbol{\Sigma}_{\boldsymbol{xy}}$, and $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}} \in \mathbb{R}^{r\times r}$, respectively, where $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}$ is the conditional covariance matrix of $\boldsymbol{y}$ given $\boldsymbol{x}$. We describe below methods for estimating the parameters for the RC models.

4.3.1 Envelope method for the RC$_{\boldsymbol{x}}$ model

The envelope method maximizes an objective function derive from the multivariate normal distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$. The parameters to be estimated include $\boldsymbol{\Gamma}_1$, $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}$, $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$. Recall that the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$ can be written as the product of the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ and the marginal distribution of $\boldsymbol{x}$, i.e, $f(\boldsymbol{y}, \boldsymbol{x}) = f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x})$. Then, given random sample of size n, minus twice the log-likelihood of the joint distribution, $f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x})$, is

$$-2\mathcal{L} = n\left\{k + \log|\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}| + \log|\boldsymbol{\Sigma}_{\boldsymbol{xx}}| + \operatorname{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1}\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{x}}\right) + \operatorname{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{S}_{\boldsymbol{xx}}\right)\right\}, \tag{4.26}$$

where $k = (p + r)\log(2\pi)$ is a constant,

$$n\boldsymbol{S}_{\boldsymbol{xx}} = \sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\mu}_{\boldsymbol{x}})(\boldsymbol{x}_i - \boldsymbol{\mu}_{\boldsymbol{x}})^T,$$

$$\begin{aligned} n\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{x}} &= \sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{B}^T\boldsymbol{x}_i)(\boldsymbol{y}_i - \boldsymbol{B}^T\boldsymbol{x}_i)^T \\ &= \boldsymbol{S}_{\boldsymbol{yy}} - 2\boldsymbol{B}^T\boldsymbol{S}_{\boldsymbol{xy}} + \boldsymbol{B}^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{B} \\ \boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}} &= \boldsymbol{\Sigma}_{\boldsymbol{yy}} - \boldsymbol{\Sigma}_{\boldsymbol{xy}}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xy}} \end{aligned} \tag{4.27}$$

are the sample covariance matrix of $\boldsymbol{x}$, sample covariance matrix of $\boldsymbol{y}$ given $\boldsymbol{x}$, and population covariance matrix of $\boldsymbol{y}$ given $\boldsymbol{x}$, respectively. Assume the RC$_{\boldsymbol{x}}$ model holds and q is known, then $\boldsymbol{B} = \boldsymbol{\Gamma}_1(\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xy}}$ and $\boldsymbol{t}_{1i} = \boldsymbol{\Gamma}_1^T\boldsymbol{x}_i$ and $\boldsymbol{t}_{0i} = \boldsymbol{\Gamma}_0^T\boldsymbol{x}_i$, have covariance matrices $\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0$, respectively; also $(\boldsymbol{t}_{1i}, \boldsymbol{y}_i)$ is uncorrelated with $(\boldsymbol{t}_{0i})$. Recall that the covariance matrix of $\boldsymbol{t} = (\boldsymbol{t}_1, \boldsymbol{t}_0)$ in Equation (4.5) has inverse and determinant given by

$$\begin{pmatrix} (\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0)^{-1} \end{pmatrix} \quad \text{and} \quad |\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1| \cdot |\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0|, \tag{4.28}$$

respectively. Therefore, $\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{S}_{\boldsymbol{xx}}$, $\mathrm{tr}(\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{S}_{\boldsymbol{xx}})$, and $\log|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|$ in Equation (4.26), are

$$\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{S}_{\boldsymbol{xx}} = \begin{pmatrix} (\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1 & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0)^{-1}\boldsymbol{\Gamma}_0^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0 \end{pmatrix}, \tag{4.29}$$

$$\begin{aligned} \mathrm{tr}(\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{S}_{\boldsymbol{xx}}) &= \mathrm{tr}\left((\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1\right) + \mathrm{tr}\left((\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0)^{-1}\boldsymbol{\Gamma}_0^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0\right) \\ \text{and} \quad \log|\boldsymbol{\Sigma}_{\boldsymbol{xx}}| &= \log|\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1| + \log|\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0|, \end{aligned} \tag{4.30}$$

respectively. Plugging in Equations (4.29) and (4.30) into the log-likelihood, Equation (4.26), we have

$$\begin{aligned} -2\mathcal{L} = n\Bigg\{ & k + \log|\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}| + \mathrm{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}^{-1}\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{x}}\right) + \log|\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1| + \log|\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0| \\ & + \mathrm{tr}\left((\boldsymbol{\Gamma}_1^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1\right) + \mathrm{tr}\left((\boldsymbol{\Gamma}_0^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0)^{-1}\boldsymbol{\Gamma}_0^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0\right)\Bigg\}. \end{aligned} \tag{4.31}$$

If $\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_0)$ is known, the MLEs of the population covariance matrices, $\boldsymbol{\Sigma}_{\boldsymbol{xy}}$, $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$, $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$, in Equation (4.31) are $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{xy}} = \boldsymbol{S}_{\boldsymbol{xy}}$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{xx}} = \boldsymbol{S}_{\boldsymbol{xx}}$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}|\boldsymbol{x}} = \boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{x}}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{yy}} = \boldsymbol{S}_{\boldsymbol{yy}}$, respectively. Then the following reduces to constants

$$\begin{aligned}\operatorname{tr}\Big((\boldsymbol{\Gamma}_1^T\hat{\boldsymbol{\Sigma}}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1\Big) + \operatorname{tr}\Big((\boldsymbol{\Gamma}_0^T\hat{\boldsymbol{\Sigma}}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0)^{-1}\boldsymbol{\Gamma}_0^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0\Big) &= q + (p-q) = p,\\ \operatorname{tr}\Big(\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}|\boldsymbol{x}}^{-1}\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{x}}\Big) &= r\end{aligned} \tag{4.32}$$

Plugging in the estimated covariances gives the following profile log-likelihood

$$\begin{aligned}\mathcal{L}_{pro} &= n\Big\{k' + \log|\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{x}}| + \log|\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1| + \log|\boldsymbol{\Gamma}_0^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0|\Big\}\\ &= n\Big\{k' + \log\Big(|\boldsymbol{S}_{\boldsymbol{yy}} - \boldsymbol{S}_{\boldsymbol{xy}}^T\boldsymbol{\Gamma}_1(\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xy}}||\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1|\Big)\\ &\quad + \log|\boldsymbol{\Gamma}_0^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0|\Big\}\end{aligned} \tag{4.33}$$

where $k' = p + r + k$ is a constant. The log-likelihood, in Equation (4.33), can be optimized in terms of $\boldsymbol{\Gamma}_1$ alone. Then Equation (4.33) can further be expressed as

Let $\mathcal{L}_{pro}^1 = |\boldsymbol{S}_{\boldsymbol{yy}} - \boldsymbol{S}_{\boldsymbol{xy}}^T\boldsymbol{\Gamma}_1(\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xy}}||\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1|$, then

$$\begin{aligned}\mathcal{L}_{pro}^1 &= \boldsymbol{S}_{\boldsymbol{yy}}\boldsymbol{S}_{\boldsymbol{yy}}^{-1}|\boldsymbol{S}_{\boldsymbol{yy}} - \boldsymbol{S}_{\boldsymbol{xy}}^T\boldsymbol{\Gamma}_1(\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xy}}||\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1|\\ &= \boldsymbol{S}_{\boldsymbol{yy}}|\Big(\boldsymbol{I}_r - \boldsymbol{S}_{\boldsymbol{yy}}^{-1}\boldsymbol{S}_{\boldsymbol{xy}}^T\boldsymbol{\Gamma}_1(\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1)^{-1}\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xy}}\Big)||\Big(\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1\Big)|\\ &= \boldsymbol{S}_{\boldsymbol{yy}}\boldsymbol{\Gamma}_1^T|\boldsymbol{S}_{\boldsymbol{xx}} - \boldsymbol{S}_{\boldsymbol{xy}}\boldsymbol{S}_{\boldsymbol{yy}}^{-1}\boldsymbol{S}_{\boldsymbol{xy}}^T|\boldsymbol{\Gamma}_1.\end{aligned} \tag{4.34}$$

In addition, from Equation (4.28) $|\boldsymbol{\Gamma}_0^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_0| = |\boldsymbol{S}_{\boldsymbol{xx}}||\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}^{-1}\boldsymbol{\Gamma}_1|$. Then Equation (4.33) becomes

$$\begin{aligned}\mathcal{L}_{pro} &= n\Big\{k' + \log|\boldsymbol{S}_{\boldsymbol{yy}}||\boldsymbol{\Gamma}_1^T(\boldsymbol{S}_{\boldsymbol{xx}} - \boldsymbol{S}_{\boldsymbol{xy}}\boldsymbol{S}_{\boldsymbol{yy}}^{-1}\boldsymbol{S}_{\boldsymbol{xy}}^T)\boldsymbol{\Gamma}_1| + \log|\boldsymbol{S}_{\boldsymbol{xx}}||\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1|\Big\}\\ &= nk'' + n\log|\boldsymbol{\Gamma}_1^T(\boldsymbol{S}_{\boldsymbol{xx}} - \boldsymbol{S}_{\boldsymbol{xy}}\boldsymbol{S}_{\boldsymbol{yy}}^{-1}\boldsymbol{S}_{\boldsymbol{xy}}^T)\boldsymbol{\Gamma}_1| + n\log|\boldsymbol{\Gamma}_1^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1|\\ &= \mathcal{J}(\boldsymbol{\Gamma}_1),\end{aligned} \tag{4.35}$$

where $k'' = k' + \log|\boldsymbol{S}_{\boldsymbol{yy}}||\boldsymbol{S}_{\boldsymbol{xx}}|$ is a constant. The function $\mathcal{J}(\boldsymbol{\Gamma}_1)$ is invariant under right orthogonal transformations of $\boldsymbol{\Gamma}_1$, i.e $\mathcal{J}(\boldsymbol{\Gamma}_1) = \mathcal{J}(\boldsymbol{\Gamma}_1\boldsymbol{O})$, where $\boldsymbol{O} \in \mathbb{R}^{q\times q}$ is an orthogonal matrix, so the minimization is over the Grassmann manifold and the solution is not unique. Let the objective function be

$$\hat{\mathbf{\Gamma}}_1 = \underset{\mathbf{\Gamma}_1 \in \mathcal{G}}{\arg\min} \left[\mathcal{J}(\mathbf{\Gamma}_1) \right]. \tag{4.36}$$

The goal is to minimize Equation (4.36) over $\mathbf{\Gamma}_1$. After estimating $\hat{\mathbf{\Gamma}}_1$, other parameters that make up the joint covariance matrix in Equation (4.22) are estimated as

$$\begin{aligned} \hat{\mathbf{\Sigma}}_{\boldsymbol{yy}} = \boldsymbol{S}_{\boldsymbol{yy}}, \quad \hat{\mathbf{\Omega}}_{\mathbf{\Gamma}_1} = \hat{\mathbf{\Gamma}}_1^T \boldsymbol{S}_{\boldsymbol{xx}} \hat{\mathbf{\Gamma}}_1, \quad \hat{\mathbf{\Omega}}_{\mathbf{\Gamma}_0} = \hat{\mathbf{\Gamma}}_1^{\perp T} \boldsymbol{S}_{\boldsymbol{xx}} \hat{\mathbf{\Gamma}}_1^{\perp}, \\ \text{and} \quad \hat{\mathbf{\Phi}} = \hat{\mathbf{\Gamma}}_1^T \boldsymbol{S}_{\boldsymbol{xy}}, \end{aligned} \tag{4.37}$$

where $\hat{\mathbf{\Gamma}}_1^{\perp} = \hat{\mathbf{\Gamma}}_0$ is the orthogonal completion of $\hat{\mathbf{\Gamma}}_1$. Moreover, the matrix of regression coefficient is

$$\hat{\boldsymbol{B}}_q = \hat{\mathbf{\Gamma}}_1 (\hat{\mathbf{\Gamma}}_1^T \boldsymbol{S}_{\boldsymbol{xx}} \hat{\mathbf{\Gamma}}_1)^{-1} \hat{\mathbf{\Gamma}}_1^T \boldsymbol{S}_{\boldsymbol{xy}} = \hat{\mathbf{\Gamma}}_1 \hat{\mathbf{\Omega}}_{\mathbf{\Gamma}_1}^{-1} \hat{\mathbf{\Phi}}. \tag{4.38}$$

This estimator of $\hat{\boldsymbol{B}}_q$ depends only on $\text{span}(\hat{\mathbf{\Gamma}}_1)$ so the actual basis is unimportant.

4.3.2 EM algorithm for isotropic RC$_{\boldsymbol{yx}}$ model

The EM algorithm estimates the parameters of different RC models. But the EM algorithm describe here is used for estimating the parameters of the isotropic RC$_{\boldsymbol{yx}}$ models and identical to the EM algorithm described by el Bouhaddani et al. (2018). The parameter to be estimated is $\mathbf{\Theta} = (\mathbf{\Gamma}_1,\ \mathbf{\Upsilon}_1,\ \boldsymbol{B}_{q,q},\ \mathbf{\Omega}_{\mathbf{\Gamma}_1},\ \mathbf{\Omega}_h,\ \sigma,\ \sigma^*)$, where

$$\boldsymbol{B}_{q,q} = \text{diag}(b_{11}, \ldots, b_{qq}) \in \mathbb{R}^{q \times q} \quad \text{and} \quad q = m.$$

The EM algorithm assumes that the complete-data is $(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{t}_{1_i}, \boldsymbol{v}_{1_i})$, where $(\boldsymbol{t}_{1_i}, \boldsymbol{v}_{1_i})$ is consider the missing data and $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ observed data. The EM algorithm alternates between two steps known as the expectation step (E-step) and the maximization step (M-step) until convergence. The E-step estimates first and second moments of the missing data conditional on the observed data and the parameters of the model. The M-step maximizes the E-step over the parameter, $\mathbf{\Theta}$.

Let the density of the complete-data be $f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}_1, \boldsymbol{v}_1)$; then the density can be expressed as the product of conditional and marginal densities, i.e,

$$\begin{aligned} f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t}_1, \boldsymbol{v}_1) &= f(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{t}_1, \boldsymbol{v}_1) f(\boldsymbol{t}_1, \boldsymbol{v}_1) \\ &= f(\boldsymbol{x}|\boldsymbol{t}_1, \boldsymbol{v}_1) f(\boldsymbol{y}|\boldsymbol{t}_1, \boldsymbol{v}_1) f(\boldsymbol{t}_1, \boldsymbol{v}_1) \\ &= f(\boldsymbol{x}|\boldsymbol{t}_1) f(\boldsymbol{y}|\boldsymbol{v}_1) f(\boldsymbol{v}_1|\boldsymbol{t}_1) f(\boldsymbol{t}_1). \end{aligned}$$

The conditional density $f(\boldsymbol{x}|\boldsymbol{t}_1, \boldsymbol{v}_1) = f(\boldsymbol{x}|\boldsymbol{t}_1)$ because $\boldsymbol{x}$ is independent of $\boldsymbol{v}_1$ given $\boldsymbol{t}_1$, and $f(\boldsymbol{y}|\boldsymbol{t}_1, \boldsymbol{v}_1) = f(\boldsymbol{y}|\boldsymbol{v}_1)$ because $\boldsymbol{y}$ is independent of $\boldsymbol{t}_1$ given $\boldsymbol{v}_1$. Moreover, the joint density of the relevant components, $f(\boldsymbol{t}_1, \boldsymbol{v}_1)$, can be written as the product $f(\boldsymbol{v}_1|\boldsymbol{t}_1) f(\boldsymbol{t}_1)$.

Given random samples of size n, the complete-data log-likelihood is the sum of individual log-likelihoods given by

$$\mathcal{L}_{cp}(\boldsymbol{\Theta}) = \log f(\boldsymbol{X}|\boldsymbol{T}_1) + \log f(\boldsymbol{Y}|\boldsymbol{V}_1) + \log f(\boldsymbol{V}_1|\boldsymbol{T}_1) + \log f(\boldsymbol{T}_1), \tag{4.39}$$

where the separate log-likelihoods are

$$\log f(\boldsymbol{X}|\boldsymbol{T}_1) = -\frac{np}{2}\log(\sigma) - \frac{1}{2\sigma}\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\Gamma}_1 \boldsymbol{t}_{1_i})^T(\boldsymbol{x}_i - \boldsymbol{\Gamma}_1 \boldsymbol{t}_{1_i}), \tag{4.40}$$

$$\log f(\boldsymbol{Y}|\boldsymbol{V}_1) = -\frac{nr}{2}\log(\sigma^*) - \frac{1}{2\sigma^*}\sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{\Upsilon}_1 \boldsymbol{v}_{1_i})^T(\boldsymbol{y}_i - \boldsymbol{\Upsilon}_1 \boldsymbol{v}_{1_i}), \tag{4.41}$$

$$\log f(\boldsymbol{V}_1|\boldsymbol{T}_1) = -\frac{nq}{2}|\tilde{\boldsymbol{\Omega}}_{\boldsymbol{h}}| - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{v}_{1_i} - \boldsymbol{B}_{q,q}\boldsymbol{t}_{1_i})^T \tilde{\boldsymbol{\Omega}}_{\boldsymbol{h}}^{-1}(\boldsymbol{v}_{1_i} - \boldsymbol{B}_{q,q}\boldsymbol{t}_{1_i}), \tag{4.42}$$

$$\log f(\boldsymbol{T}_1) = -\frac{n}{2}|\tilde{\boldsymbol{\Omega}}_{\boldsymbol{\Gamma}_1}| - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{t}_{1_i})^T \tilde{\boldsymbol{\Omega}}_{\boldsymbol{\Gamma}_1}^{-1}\boldsymbol{t}_{1_i}. \tag{4.43}$$

The maximum likelihood estimator of $\boldsymbol{\Theta}$ is

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}) = \arg\max_{\boldsymbol{\Theta}} \mathrm{E}\left[\mathcal{L}_{cp}(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Theta}^k)\right], \tag{4.44}$$

where $\mathrm{E}(\cdot)$ is the conditional-expected complete log-likelihood, and $\boldsymbol{\Theta}^k$ is some fixed current estimate of the parameter. Note that the expectations of the conditional and marginal log-likelihoods in Equation (4.39) can be estimated separately.

E-step

Let $\mathrm{E}_{\boldsymbol{Z}}(\cdot) = \mathrm{E}_{\boldsymbol{Z}}(\cdot|\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Theta}^k)$ denote the conditional expectation with respect to some $\boldsymbol{Z}$, the required conditional expectations are itemized below.

(a) $$\begin{aligned}\mathrm{E}_{\boldsymbol{T}_1}[\log f(\boldsymbol{X}|\boldsymbol{T}_1)] &= -\mathrm{E}_{\boldsymbol{T}_1}[(\boldsymbol{X}-\boldsymbol{T}_1\boldsymbol{\Gamma}_1^T)^T(\boldsymbol{X}-\boldsymbol{T}_1\boldsymbol{\Gamma}_1^T)|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\Theta}^k]+k^*\\ &= \mathrm{tr}(-\boldsymbol{X}^T\boldsymbol{X}+2\boldsymbol{X}^T\boldsymbol{\mu}_{\boldsymbol{t}_1}\boldsymbol{\Gamma}_1^T-\boldsymbol{\Gamma}_1\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1}\boldsymbol{\Gamma}_1^T)+k^*,\end{aligned}$$

where k^* is a constant, $\boldsymbol{\mu}_{\boldsymbol{t}_1}=\mathrm{E}_{\boldsymbol{T}_1}[\boldsymbol{T}_1]$ and $\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1}=\mathrm{E}_{\boldsymbol{T}_1}[\boldsymbol{T}_1^T\boldsymbol{T}_1]$, with

$$\begin{aligned}\boldsymbol{\mu}_{\boldsymbol{t}_1} &= (\boldsymbol{x},\boldsymbol{y})\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xyt}_1}\\ \boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1} &= \tilde{\boldsymbol{\Omega}}_{\boldsymbol{\Gamma}_1}-\boldsymbol{\Sigma}_{\boldsymbol{t}_1\boldsymbol{xy}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xyt}_1}+\boldsymbol{\Sigma}_{\boldsymbol{t}_1\boldsymbol{xy}}\boldsymbol{\Sigma}^{-1}\boldsymbol{S}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xyt}_1},\end{aligned} \tag{4.45}$$

where $\boldsymbol{S}$ is the sample covariance matrix of the observed data, and $\boldsymbol{\Sigma}_{\boldsymbol{t}_1\boldsymbol{xy}}$ the covariance between $(\boldsymbol{x},\boldsymbol{y})$ and $\boldsymbol{t}_1$ given by

$$\boldsymbol{\Sigma}_{\boldsymbol{t}_1\boldsymbol{xy}}=\begin{pmatrix}\boldsymbol{\Gamma}_1\tilde{\boldsymbol{\Omega}}_{\boldsymbol{\Gamma}_1}\\ \boldsymbol{\Upsilon}_1\tilde{\boldsymbol{\Omega}}_{\boldsymbol{\Gamma}_1}\boldsymbol{B}_{q,q}\end{pmatrix}. \tag{4.46}$$

To ensure the matrix $\boldsymbol{\Gamma}_1$ is orthonormal a constraint is introduced into (a) which gives

$$\mathrm{E}_{\boldsymbol{T}_1}[\log(\boldsymbol{X}|\boldsymbol{T}_1)]=\mathrm{tr}(-\boldsymbol{X}^T\boldsymbol{X}+2\boldsymbol{X}^T\boldsymbol{\mu}_{\boldsymbol{t}_1}\boldsymbol{\Gamma}_1^T-\boldsymbol{\Gamma}_1\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1}\boldsymbol{\Gamma}_1^T)+\mathrm{tr}((\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1-\boldsymbol{I}_q)\boldsymbol{\Lambda}_{\boldsymbol{t}_1}), \tag{4.47}$$

where $\boldsymbol{\Lambda}_{\boldsymbol{t}_1}\in\mathbb{R}^{q\times q}$ is a matrix of penalties. In a similar manner, the conditional expectation of Equation (4.41) is

(b) $$\begin{aligned}\mathrm{E}_{\boldsymbol{V}_1}[\log f(\boldsymbol{X}|\boldsymbol{V}_1)] = {}&\mathrm{tr}(-\boldsymbol{Y}^T\boldsymbol{Y}+2\boldsymbol{Y}^T\boldsymbol{\mu}_{\boldsymbol{v}_1}\boldsymbol{\Upsilon}_1^T-\boldsymbol{\Upsilon}_1\boldsymbol{C}_{\boldsymbol{v}_1\boldsymbol{v}_1}\boldsymbol{\Upsilon}_1^T)+\\ &\mathrm{tr}((\boldsymbol{\Upsilon}_1^T\boldsymbol{\Upsilon}_1-\boldsymbol{I}_q)\boldsymbol{\Lambda}_{\boldsymbol{v}_1}),\end{aligned}$$

(c) $$\begin{aligned}\mathrm{E}_{\boldsymbol{T}_1}[\log f(\boldsymbol{V}_1|\boldsymbol{T}_1)] &= \mathrm{tr}\,\mathrm{E}[(-\boldsymbol{V}_1^T\boldsymbol{V}_1+2\boldsymbol{V}_1^T\boldsymbol{T}_1\boldsymbol{B}_{q,q}-\boldsymbol{B}_{q,q}\boldsymbol{T}_1^T\boldsymbol{T}_1\boldsymbol{B}_{q,q})|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\Theta}^k]\\ &= \mathrm{tr}\left(-\boldsymbol{V}_1^T\boldsymbol{V}_1+2\mathrm{E}_{\boldsymbol{T}_1}[\boldsymbol{V}_1^T\boldsymbol{T}_1]\boldsymbol{B}_{q,q}-\boldsymbol{B}_{q,q}\mathrm{E}_{\boldsymbol{T}_1}[\boldsymbol{T}_1^T\boldsymbol{T}_1]\boldsymbol{B}_{q,q}\right)\end{aligned}$$

The conditional expectation of Equation (4.43) is

(d) $\mathrm{E}_{\boldsymbol{T}_1}[\log f(\boldsymbol{T}_1)]=-\dfrac{n}{2}|\tilde{\boldsymbol{\Omega}}_{\boldsymbol{\Gamma}_1}|-\dfrac{1}{2}\tilde{\boldsymbol{\Omega}}_{\boldsymbol{\Gamma}_1}^{-1}\mathrm{E}_{\boldsymbol{T}_1}[\boldsymbol{T}_1^T\boldsymbol{T}_1].$

Finally, let $\boldsymbol{E}=\boldsymbol{X}-\boldsymbol{T}_1\boldsymbol{\Gamma}_1^T$, $\boldsymbol{F}=\boldsymbol{Y}-\boldsymbol{V}_1\boldsymbol{\Upsilon}_1^T$, and $\boldsymbol{H}=\boldsymbol{V}_1-\boldsymbol{T}_1\boldsymbol{B}$, then

(e) $\mathrm{E}_{\boldsymbol{E}}[\log f(\boldsymbol{X}|\boldsymbol{T}_1)]=-\dfrac{np}{2}\log(\sigma)-\dfrac{1}{2\sigma}\,\mathrm{tr}(\mathrm{E}_{\boldsymbol{E}}[\boldsymbol{E}^T\boldsymbol{E}]),$

(f) $\mathrm{E}_{\boldsymbol{F}}[\log f(\boldsymbol{Y}|\boldsymbol{V}_1)]=-\dfrac{nr}{2}\log(\sigma^*)-\dfrac{1}{2\sigma^*}\,\mathrm{tr}(\mathrm{E}_{\boldsymbol{F}}[\boldsymbol{F}^T\boldsymbol{F}]),$

(g) and $\mathrm{E}_{\boldsymbol{H}}[\log f(\boldsymbol{V}_1|\boldsymbol{T}_1)]=-\dfrac{n}{2}|\tilde{\boldsymbol{\Omega}}_{\boldsymbol{h}}|-\dfrac{1}{2}\tilde{\boldsymbol{\Omega}}_{\boldsymbol{h}}^{-1}\,\mathrm{tr}(\mathrm{E}_{\boldsymbol{H}}[\boldsymbol{H}^T\boldsymbol{H}]).$

M-step

This step finds $\boldsymbol{\Theta}^{k+1}$ which maximizes the fitted likelihood in the E-step. To do so we differentiate items (a) to (g) with respect to corresponding parameters to get the following.

Differentiating item (a) with respect to $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Lambda}_{\boldsymbol{t}_1}$ and setting them equal to zero gives

- $2\boldsymbol{X}^T\boldsymbol{\mu}_{\boldsymbol{t}_1} - 2\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1}\boldsymbol{\Gamma}_1^T - 2\boldsymbol{\Gamma}_1\boldsymbol{\Lambda}_{\boldsymbol{t}_1} = 2\boldsymbol{X}^T\boldsymbol{\mu}_{\boldsymbol{t}_1} - 2\boldsymbol{\Gamma}_1^T(\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1} + \boldsymbol{\Lambda}_{\boldsymbol{t}_1}) = \boldsymbol{0}$ and $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 - \boldsymbol{I}_q = \boldsymbol{0}$

  Solving item (I) for $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Lambda}_{\boldsymbol{t}_1}$ gives

  $$\boldsymbol{\Gamma}_1 = \boldsymbol{X}^T\boldsymbol{\mu}_{\boldsymbol{t}_1}(\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1} + \boldsymbol{\Lambda}_{\boldsymbol{t}_1})^{-1} \text{ and } \boldsymbol{\Lambda}_{\boldsymbol{t}_1} = \boldsymbol{\Gamma}_1^T\boldsymbol{X}^T\boldsymbol{\mu}_{\boldsymbol{t}_1} - \boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1}, \tag{4.48}$$

  where $\boldsymbol{\Lambda}_{\boldsymbol{t}_1}$ is such that $(\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1} + \boldsymbol{\Lambda}_{\boldsymbol{t}_1})$ is invertible, and $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Lambda}_{\boldsymbol{t}_1}$ satisfy

  $$\boldsymbol{I}_q = \boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 = [(\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1} + \boldsymbol{\Lambda}_{\boldsymbol{t}_1})^{-1}]^T\boldsymbol{\mu}_{\boldsymbol{t}_1}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{\mu}_{\boldsymbol{t}_1}(\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1} + \boldsymbol{\Lambda}_{\boldsymbol{t}_1})^{-1},$$
  $$\text{where } \boldsymbol{\mu}_{\boldsymbol{t}_1}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{\mu}_{\boldsymbol{t}_1} = (\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1} + \boldsymbol{\Lambda}_{\boldsymbol{t}_1})^T(\boldsymbol{C}_{\boldsymbol{t}_1\boldsymbol{t}_1} + \boldsymbol{\Lambda}_{\boldsymbol{t}_1}) = \boldsymbol{L}_{\boldsymbol{t}_1}\boldsymbol{L}_{\boldsymbol{t}_1}^T. \tag{4.49}$$

  The $\boldsymbol{L}_{\boldsymbol{t}_1}\boldsymbol{L}_{\boldsymbol{t}_1}^T$ is a Cholesky decomposition, and $\boldsymbol{L}_{\boldsymbol{t}_1}$ is lower triangular matrix. In a similar manner, differentiating item (b) and solving for $\boldsymbol{\Upsilon}_1$ and $\boldsymbol{\Lambda}_{\boldsymbol{v}_1}$ gives

- $\boldsymbol{\Upsilon}_1 = \boldsymbol{Y}^T\boldsymbol{\mu}_{\boldsymbol{v}_1}(\boldsymbol{C}_{\boldsymbol{v}_1\boldsymbol{v}_1} + \boldsymbol{\Lambda}_{\boldsymbol{v}_1})^{-1}$ and $\boldsymbol{\Lambda}_{\boldsymbol{v}_1} = \boldsymbol{\Upsilon}_1^T\boldsymbol{Y}^T\boldsymbol{\mu}_{\boldsymbol{v}_1} - \boldsymbol{C}_{\boldsymbol{v}_1\boldsymbol{v}_1}$.

  Taking the derivative of item (c) with respect to $\boldsymbol{B}_{q,q}$ and setting it equal to zero gives

- $\mathrm{E}_{\boldsymbol{T}_1}\left[\boldsymbol{V}_1^T\boldsymbol{T}_1\right] - \boldsymbol{B}_{q,q}\mathrm{E}_{\boldsymbol{T}_1}\left[\boldsymbol{T}_1^T\boldsymbol{T}_1\right] = \boldsymbol{0}$, where
  $\boldsymbol{B}_{q,q} = \mathrm{E}_{\boldsymbol{T}_1}\left[\boldsymbol{T}_1^T\boldsymbol{T}_1\right]^{-1}\mathrm{E}_{\boldsymbol{T}_1}\left[\boldsymbol{V}_1^T\boldsymbol{T}_1\right]$

  The derivative of item (d) gives

- $n\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1} - \mathrm{E}_{\boldsymbol{T}_1}[\boldsymbol{T}_1^T\boldsymbol{T}_1] = \boldsymbol{0}$
  then, $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1} = \dfrac{1}{n}\mathrm{E}_{\boldsymbol{T}_1}[\boldsymbol{T}_1^T\boldsymbol{T}_1]$.

  For the parameters, $(\sigma, \sigma^*, \sigma_h)$, differentiating (e), (f), and (g) and setting them equal to zero and solving for σ, σ^*, and σ_h we have

- $\sigma = \frac{1}{np}\mathrm{E}_{\boldsymbol{E}}[\boldsymbol{E}^T\boldsymbol{E}]$,
  $\sigma^* = \frac{1}{nr}\mathrm{E}_{\boldsymbol{F}}[\boldsymbol{F}^T\boldsymbol{F}]$, and
  $\boldsymbol{\Omega}_{\boldsymbol{h}} = \frac{1}{n}\mathrm{E}_{\boldsymbol{H}}[\boldsymbol{H}^T\boldsymbol{H}]$, respectively

Then for some initial value, $\boldsymbol{\Theta}^0$, alternate between the E-step and M-step until convergence, where

$$\begin{aligned}
\boldsymbol{\Gamma}_1^{k+1} &= \boldsymbol{X}^T\mathrm{E}[\boldsymbol{T}_1](\boldsymbol{L}_{\boldsymbol{\Gamma}_1}^T)^{-1}\\
\boldsymbol{\Upsilon}_1^{k+1} &= \boldsymbol{Y}^T\mathrm{E}[\boldsymbol{V}_1](\boldsymbol{L}_{\boldsymbol{\Upsilon}_1}^T)^{-1}\\
\boldsymbol{B}_{q,q}^{k+1} &= \mathrm{E}[\boldsymbol{V}_1^T\boldsymbol{T}_1]\left(\mathrm{E}[\boldsymbol{T}_1^T\boldsymbol{T}_1]\right)^{-1}\circ\boldsymbol{I}_q\\
\tilde{\boldsymbol{\Omega}}_{\boldsymbol{\Gamma}_1}^{k+1} &= \frac{1}{n}\mathrm{E}[\boldsymbol{T}_1^T\boldsymbol{T}_1]\circ\boldsymbol{I}_q\\
(\sigma_e)^{k+1} &= \frac{1}{np}\operatorname{tr}\mathrm{E}[\boldsymbol{E}^T\boldsymbol{E}]\\
(\sigma_f)^{k+1} &= \frac{1}{nr}\operatorname{tr}\mathrm{E}[\boldsymbol{F}^T\boldsymbol{F}]\\
\tilde{\boldsymbol{\Omega}}_{\boldsymbol{h}}^{k+1} &= \frac{1}{n}\mathrm{E}[\boldsymbol{H}^T\boldsymbol{H}]\circ\boldsymbol{I}_q
\end{aligned}$$

with $k = 1, 2, \ldots,$. The product $\circ$ is the Hadamard product and ensures that the off diagonal elements are zero.

4.3.3 SIMPLS algorithm

The SIMPLS algorithm is one of the earliest PLS algorithms for estimating the parameters of different relevant components models. Two versions of the SIMPLS algorithm are found in the literature; one estimates the parameters of the RC$_{\boldsymbol{x}}$ model, and the other estimates the parameters of the RC$_{\boldsymbol{yx}}$ model (Boulesteix and Strimmer, 2006). We focus on finding the parameters of the RC$_{\boldsymbol{x}}$ model. The SIMPLS algorithm computes $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{p\times q}$ so that the squared sample covariance between $\boldsymbol{y}$ and linear transformations of $\boldsymbol{x}$ is maximal under certain conditions. The conditions are that the relevant components are mutually uncorrelated and $\boldsymbol{\Gamma}_1$ is semiorthogonal, i.e,

$$\begin{aligned}
\boldsymbol{\gamma}_{j+1} &= \arg\max_{\boldsymbol{\gamma}} \boldsymbol{\gamma}^T\boldsymbol{S}_{\boldsymbol{xy}}\boldsymbol{S}_{\boldsymbol{xy}}^T\boldsymbol{\gamma},\\
&\text{subject to } \ \boldsymbol{\gamma}^T\boldsymbol{S}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_{1j} = 0 \ \ \boldsymbol{\gamma}^T\boldsymbol{\gamma} = 1,
\end{aligned} \tag{4.50}$$

where $\boldsymbol{\gamma}_j$, $j = 0, \ldots, q$ are columns of $\boldsymbol{\Gamma}_1$, and $\boldsymbol{\gamma}_0 = \mathbf{0}$. The SIMPLS algorithm computes the columns of $\boldsymbol{\Gamma}_1$ sequentially using the objective function Equation (4.50). The Spectral decomposition (SD) can be implemented to compute $\boldsymbol{\Gamma}_1$; the principal eigenvector of $\boldsymbol{S_{xy}S_{xy}^T}$ is used to compute the first relevant component. The relevant component is then used to find the loading vectors; which are used to project out the contribution of the first relevant component from $\boldsymbol{S_{xy}S_{xy}^T}$. The process is repeated until, q say, relevant components are determined. In Table 4.2, the SIMPLS algorithm of De Jong (1993) is presented;

Determine squared cross-product	$\boldsymbol{J} = \boldsymbol{S_{xy}S_{xy}^T}$
For $k = 1, \ldots, q$	
$k = 1$, determine SD of	$\boldsymbol{J}$
$k > 1$, determine SD of	$\boldsymbol{J} - \boldsymbol{P}(\boldsymbol{P}^T\boldsymbol{P})^{-1}\boldsymbol{P}^T\boldsymbol{J}$
The weights (basis)	$\boldsymbol{\gamma}$ = first eigenvector
The scores	$\boldsymbol{t}_1 = \boldsymbol{X\gamma}$
The loadings	$\boldsymbol{p} = \dfrac{\boldsymbol{X}^T\boldsymbol{t}_1}{(\boldsymbol{t}_1^T\boldsymbol{t}_1)}$
Store $\boldsymbol{\gamma}$, $\boldsymbol{t}_1$, and $\boldsymbol{p}$ in	$\boldsymbol{\Gamma}_1$, $\boldsymbol{T}$, and $\boldsymbol{P}$, respectively

Table 4.2: *SIMPLS algorithm*

4.4 Simulation Study

In section 4.3, three PLS2 methods for estimating the parameters of RC$_{\boldsymbol{x}}$ and isotropic RC$_{\boldsymbol{yx}}$ models were reviewed. In this section, the predictive performance of the estimation methods are compared using different settings of simulated datasets to find out when the methods perform better. The settings covered include

(i) large and small sample size,

(ii) degree of multicollinearity in $\boldsymbol{x}$,

(iii) number of relevant components, and

(iv) the amount of variation explained by relevant and irrelevant components.

The datasets were generated from the RC$_{\boldsymbol{x}}$ model using n observation from a multivariate regression with $r = 3$ response variables, and p predictor variables; dimension for the relevant components is q, and the variables are multivariate normal with zero mean. The covariance matrix, $\mathbf{\Sigma}_{\boldsymbol{xx}}$ was generated as $\mathbf{\Sigma}_{\boldsymbol{xx}} = \mathbf{\Gamma}_1\mathbf{\Omega}_{\mathbf{\Gamma}_1}\mathbf{\Gamma}_1^T + \mathbf{\Gamma}_0\mathbf{\Omega}_{\mathbf{\Gamma}_0}\mathbf{\Gamma}_0^T$, where $\mathbf{\Omega} = (\mathbf{\Omega}_{\mathbf{\Gamma}_1}, \mathbf{\Omega}_{\mathbf{\Gamma}_0})$ is a diagonal matrix containing distinct eigenvalues of $\rho^{|i-j|} \in \mathbb{R}^{p\times p}$, $\mathbf{\Gamma} = (\mathbf{\Gamma}_1, \mathbf{\Gamma}_0)$ was constructed by orthonormalizing a p by p matrix of uniform (0,1) random variables, and $\boldsymbol{B}$ was generated as $\mathbf{\Gamma}_1\mathbf{\Omega}_{\mathbf{\Gamma}_1}^{-1}\mathbf{\Phi}$, where $\mathbf{\Phi} \in \mathbb{R}^{q\times r}$ was generated as a matrix of uniform (0,2) random variables. Finally, $\mathbf{\Sigma}_{\boldsymbol{y}|\boldsymbol{x}} = 0.8^{|i-j|} \in \mathbb{R}^{r\times r}$. Table 4.3 show the settings considered in the simulation study,

n	p	q	ρ
30	5	2	0.01
100	20	3	0.8
		10	

Table 4.3: *Overview of simulation settings. Small and large values of ρ correspond to low and high multicollinearity, respectively.*

where large values ρ corresponds to near multicollinearity in $\boldsymbol{x}$. The root-mean-square error (RMSE) used for comparison was calculated using the Frobenius norm of a matrix given by

$$\|\boldsymbol{D}_{\boldsymbol{y}}\| = \sqrt{\sum_i \sum_j |a_{ij}|^2}, \tag{4.51}$$

where $\boldsymbol{D}_{\boldsymbol{y}}$ is the difference between actual and predicted matrix of response variables.

The envelope, SIMPLS, and EM-PLS estimators were obtained with the R (R Core Team, 2020) packages `Renvlp` (Lee and Su, 2018), `plsr`, and `PO2PLS`, respectively. The results shown in the figures below are the averages of cross-validated RMSEs over 100 replications in each scenario. The RMSEs are compared over several components for the different regression method; envelope (EPLS), SIMPLS, EM algorithm (EM-PLS) and OLS. Note that the R package `PO2PLS` is used for estimating parameters of the isotropic RC$_{\boldsymbol{yx}}$ model alone, and the largest number of relevant components $q \leq \min(r, p)$. However, we are interested in investigating its predictive performance when data are drawn from the RC$_{\boldsymbol{x}}$ model.

4.4.1 Comparing different methods in terms of predictive performance

Scenario 1: The first simulation was designed to compare the predictive performance of the methods when the variation in the relevant part is large compared to variation in the irrelevant part. To simulated the data we used $r = 3$ response variables, $p = 20$ predictor variables, $q = 3$ relevant components, $n = 30$ random samples, and the predictor variables are nearly multicollinear (i.e, $\rho = 0.8$). The results are in Figure 4.1 and shows that in this setting EPLS, EM-PLS, and SIMPLS methods outperformed the OLS method with regard to CV-RMSE. The required number of components is 3 for SIMPLS and EM-PLS, and 4 for EPLS. Note that the value of the cross-validated root-mean-square error (CV-RMSE) for the EPLS method for 3 and 4 components are similar. When $q = 3$, the CV-RMSE for the methods are EPLS $=$ 1.20, SIMPLS $=$ 1.22, EM-PLS $=$ 1.31, and OLS $=$ 2.16. Note that OLS uses all the predictor variables, and therefore will have the same value across the components.

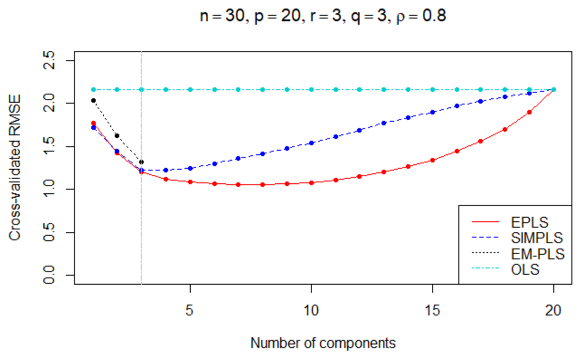


Figure 4.1: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, n is small and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

Using the same simulation setting the sample size was increased to n =100, and the results are in Figure 4.2. The results show that with a large sample size all the methods (EPLS, SIMPLS, EM-PLS and OLS) are similar when $q \geq 3$.

Scenario 2: In the second simulation, the relevant and irrelevant part share the total variation in the data. Also, we used $r = 3$ response variables, $p = 20$ predictor variables, $q = 3$ relevant components, $n = 30$ random samples, and the predictor

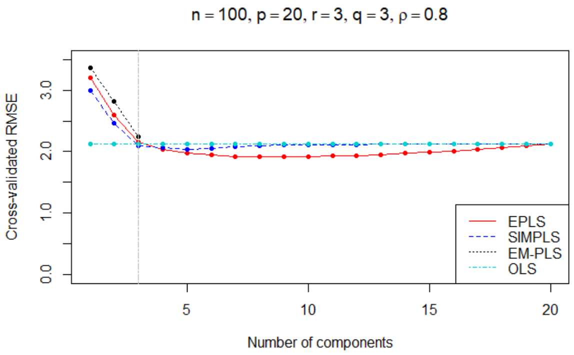


Figure 4.2: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, n is large and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

variables are not multicollinear (i.e, $\rho = 0.01$). The results are in Figure 4.3. We see that EPLS has better prediction performance compared to the other methods, and EM-PLS and SIMPLS are similar in terms of RMSE. When $q = 3$ the CV-RMSEs are 1.80 (EPLS), 2.24 (SIMPLS), 2.41 (EM-PLS), and 3.44 (OLS) and shows that EPLS outperformed other methods. Also, the PLS2 methods outperformed the OLS method when n is small.

Furthermore, using the same setting, the sample size was increased to n =100, the results are presented in Figure 4.4 and show that with an increase in the sample size SIMPLS used a fewer number of components compared to EPLS to find the smallest CV-RMSE. The EPLS method uses more components than SIMPLS and performs better than other methods. The OLS and SIMPLS methods are identical after three components. Also, the EM-PLS has the poorest performance with regard to RMSE compared to other methods considered. When $q = 3$ the CV-RMSEs are 3.25 (EPLS), 3.47 (SIMPLS), 4.02 (EM-PLS), and 3.38 (OLS) and shows that EPLS performs better than other methods.

Scenario 3: For the third simulation study, the number of predictor variables was reduced to check how the methods will perform with regard to prediction. We took $n = 30$ samples from a multivariate regression with $r = 3$ response variables, $p = 5$ predictor variables, $q = 2$ relevant components, and the predictor variables are near multicollinear (i.e, $\rho = 0.8$). Also, in this setting the variability in the relevant part is large compare to the irrelevant part. The results are in Figure 4.5 and shows

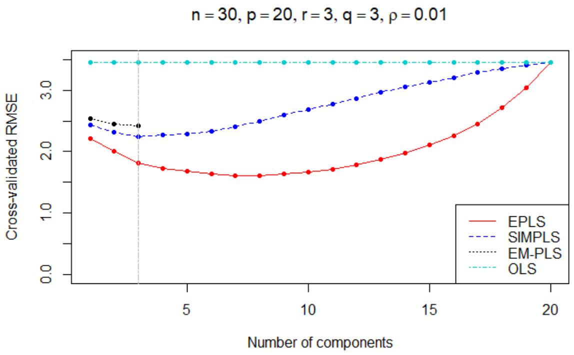


Figure 4.3: *CV-RMSEs over 100 replications when total variation in the data is shared by both the relevant part and the irrelevant part, n is small and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

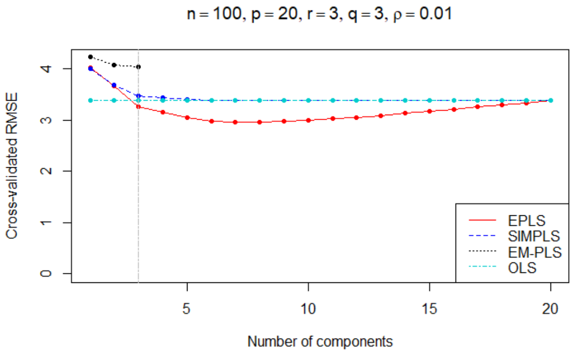


Figure 4.4: *CV-RMSEs over 100 replications when total variation in the data is shared by both the relevant part and the irrelevant part, n is large and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

that EPLS, SIMPLS and OLS performs slightly better than EM-PLS with regard to RMSE. Moreover, the sample size was increased to 100; results are presented in Figure 4.6 and shows that with an increase in sample size other methods performed slightly better than EM-PLS. For $q = 2$ the CV-RMSEs are 3.28 (EPLS), 3.27 (SIMPLS), 3.52 (EM-PLS), and 3.26 (OLS). This indicates that EM-PLS may have a poor prediction performance when p is small and the variability in the relevant part is large compared to the irrelevant part.

Further, using the same simulation setting the prediction performance of the methods were compared for cases when the predictor variables are not multicollinear (i.e, $\rho = 0.01$). The results in Figures 4.7 and 4.8 show that EM-PLS has the poorest performance with regard to RMSE compared to EPLS, SIMPLS, and OLS. And

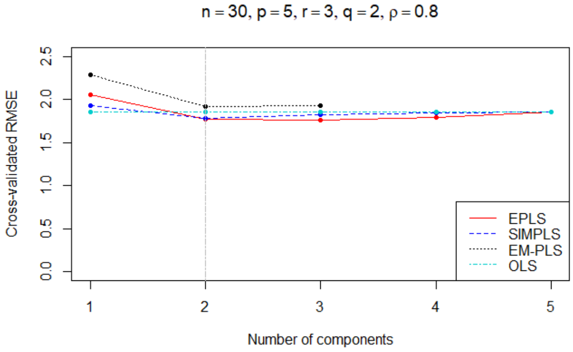


Figure 4.5: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, the number of predictor variables is small, n is small and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

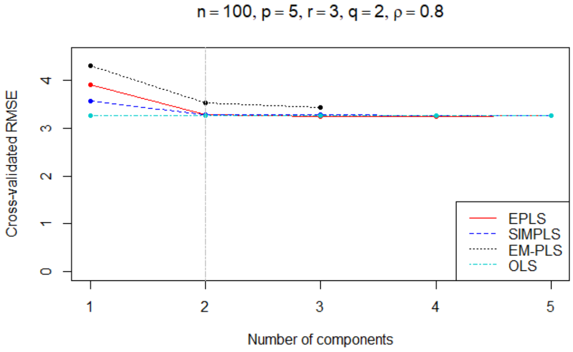


Figure 4.6: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, the number of predictor variables is small, n is large and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

with an increase in sample size from $n = 30$ to $n = 100$ the EM-PLS methods did not improve. For $q = 2$ the CV-RMSEs are 4.09 (EPLS), 4.17 (SIMPLS), 5.59 (EM-PLS), and 4.15 (OLS) and shows that EPLS performs slightly better than other methods. This suggests that when the number of predictor is small and not multicollinear EPLS, SIMPLS, and OLS outperforms EM-PLS.

Scenario 4: In this scenario, the number of relevant components is larger compared to other scenarios. Also, in this setting the variability in the relevant part is large compare to the irrelevant part. We took $n = 30$ and $n = 100$ samples from a multivariate regression with $r = 3$ response variables, $p = 20$ predictor variables, $q = 10$ relevant components, and the predictor variables are near multicollinear (i.e, $\rho = 0.8$). The results for $n = 30$ and $n = 100$ are given in Figures 4.9 and 4.10,

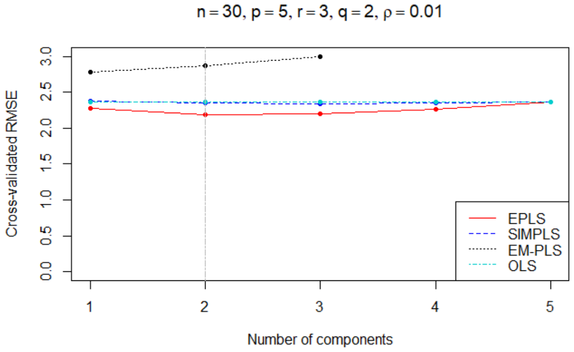


Figure 4.7: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, n is small and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

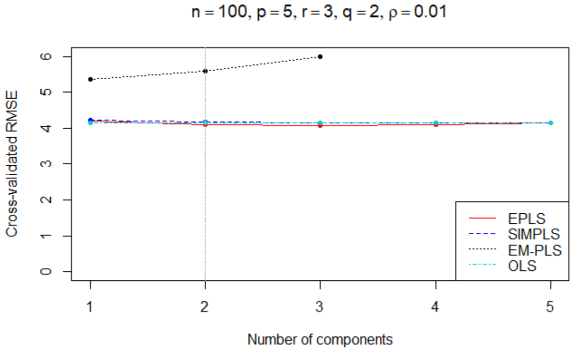


Figure 4.8: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, n is large and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

respectively. We see that EM-PLS is outperformed by the other methods in terms prediction irrespective of the sample size. In addition, the prediction perform of OLS, EPLS, and SIMPLS are similar for $q \geq 10$ when $n = 100$. However, with $n = 30$ the EPLS method has the smallest RMSE at $q = 10$ and preforms better than other methods. This suggests that the EPLS method is better in terms of prediction when n is small and q is moderate.

Scenario 5: Here, the settings are similar to that of scenario 4, but the predictor variables are not multicollinear. The results are given in Figures 4.11 and 4.12, and shows that EPLS performs better than other methods when $n = 30$. For $n = 100$ OLS, EPLS, and SIMPLS are similar after $q = 7$ components. And EM-PLS is similar to EPLS and SIMPLS when $q = 3$. In this scenario, the EPLS outperforms

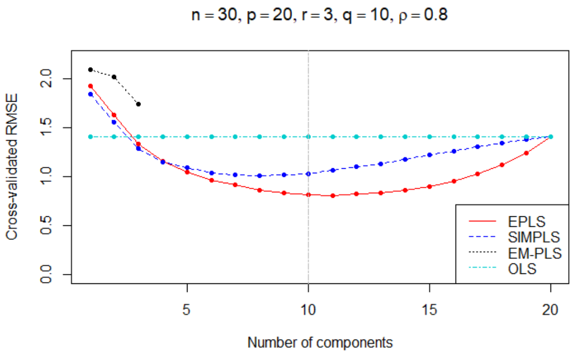


Figure 4.9: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, n is small and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

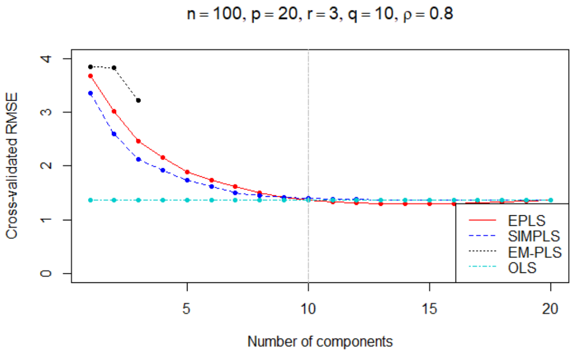


Figure 4.10: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, n is large and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

other methods when sample size is small, where the CV-RMSEs are 0.99 (EPLS), 1.38 (SIMPLS), and 1.62 (OLS) when $q = 10$.

In the scenarios that follow, the variability in the irrelevant part is large compared to the relevant components. The settings used in the previous scenarios will be used and the prediction performance of the methods will be compared.

Scenario 6: We generated the data using $r = 3$ response variables, $p = 20$ predictor variables, $q = 3$ relevant components, $n = 30$ random samples, and the predictor variables are nearly multicollinear (i.e, $\rho = 0.8$). The results are presented in Figure 4.13 and shows that the PLS2 methods outperforms OLS with regard to prediction performance. Also, the PLS2 methods are similar when $q = 3$ where the CV-

RMSEs are 2.79 (EPLS), 3.23 (SIMPLS), and 3.02 (EM-PLS). In addition, for the three PLS2 methods the smallest RMSE for the first component is the smallest. Furthermore, as sample size increased to $n = 100$ all the methods are similar; the results are shown in Figure 4.14.

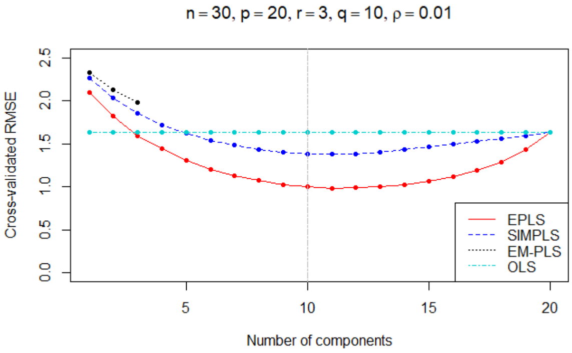


Figure 4.11: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, n is small and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

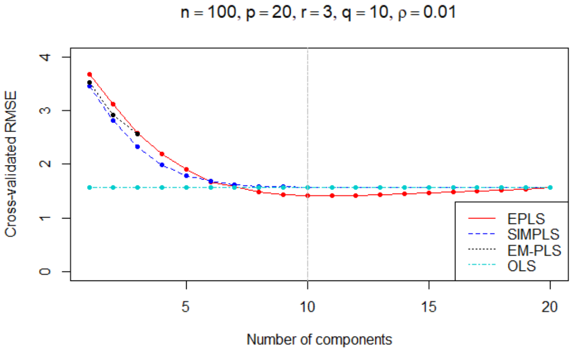


Figure 4.12: *CV-RMSEs over 100 replications when variation in the relevant part is large compared to the irrelevant part, n is large and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

Scenario 7: Data was generated using $r = 3$ response variables, $p = 20$ predictor variables, $q = 3$ relevant components, $n = 30$ random samples, and the predictor variables are not multicollinear (i.e, $\rho = 0.01$). The results appear in Figure 4.15 and shows that the PLS2 methods outperforms OLS with regard to prediction performance. And the EPLS method performs better than the EM-PLS and SIMPLS methods with CV-RMSEs equal to 1.86 (EPLS), 2.32 (SIMPLS), and 2.68

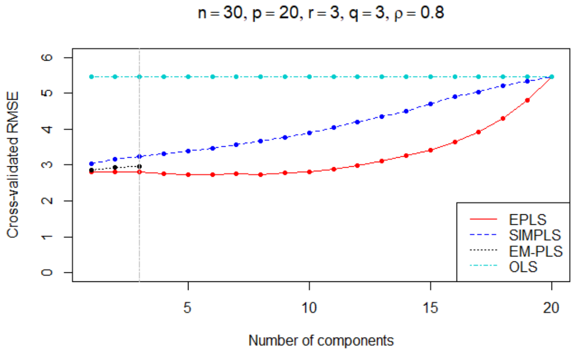


Figure 4.13: *CV-RMSEs over 100 replications when variation in the irrelevant part is large compared to the relevant part, n is small and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

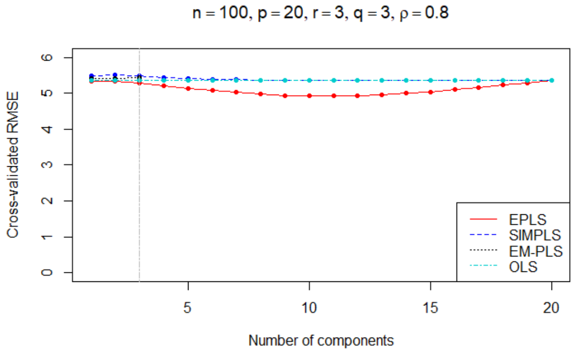


Figure 4.14: *CV-RMSEs over 100 replications when variation in the irrelevant part is large compared to the relevant part, n is large and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

(EM-PLS). Furthermore, with $n = 100$ EPLS, SIMPLS, and OLS outperforms the EM-PLS method; the results are given in Figure 4.16. Moreover, the smallest CV-RMSE for EPLS is for $q = 6$ components, which is more than the number used by SIMPLS.

Scenario 8: In this scenario, the generated data has $r = 3$ response variables, $p = 20$ predictor variables, $q = 10$ relevant components, $n = 30$ random samples, and the predictor variables are nearly multicollinear (i.e, $\rho = 0.8$). The results are shown in Figure 4.17, we see that EPLS has better prediction performance compared to OLS, SIMPLS, and EM-PLS. And EM-PLS and SIMPLS has similar performance and both performs better than OLS. Also, EPLS, SIMPLS, and EM-PLS requires 10, 8, and 1 components, respectively. Suggesting that in this setting EPLS requires

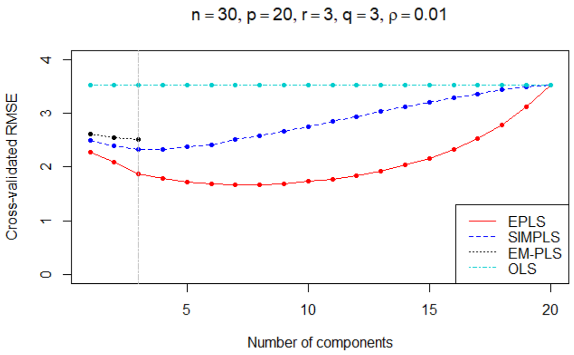


Figure 4.15: *CV-RMSEs over 100 replications when variation in the irrelevant part is large compared to the relevant part, n is small and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

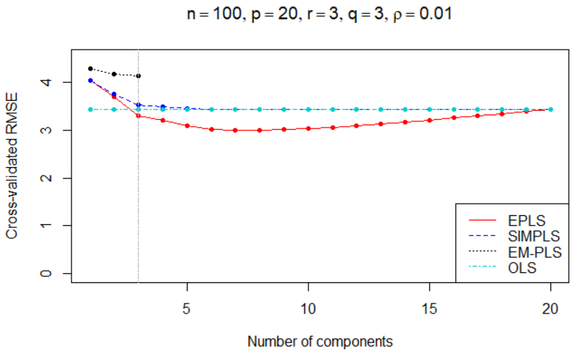


Figure 4.16: *CV-RMSEs over 100 replications when variation in the irrelevant part is large compared to the relevant part, n is large and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

more components compared to EM-PLS and SIMPLS to achieve its best prediction. Further, the sample size was increased to $n = 100$ and the results are shown in Figure 4.18. Here, the EPLS, SIMPLS, and OLS have similar performances when $q \geq 10$ components. Besides, the EPLS, SIMPLS, and OLS methods outperformed the EM-PLS methods.

Scenario 9: In this scenario, the generated data has $r = 3$ response variables, $p = 20$ predictor variables, $q = 10$ relevant components, $n = 30$ random samples, and the predictor variables are not multicollinear (i.e, $\rho = 0.01$). The results are in Figure 4.19 and we see that EPLS outperformed OLS, SIMPLS, and EM-PLS with regard to RMSE . And EM-PLS has the poorest performance compared to other methods. Further, the sample size was increased to $n = 100$ and the results are

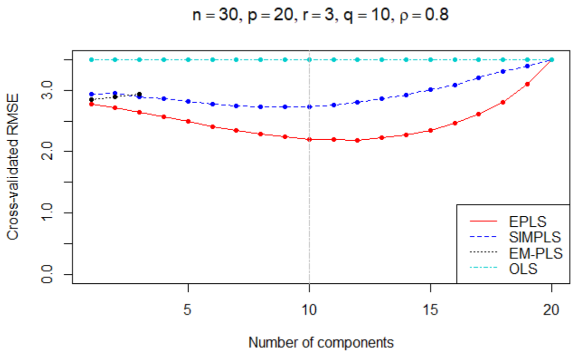


Figure 4.17: *CV-RMSEs over 100 replications when variation in the irrelevant part is large compared to the relevant part, n is small and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

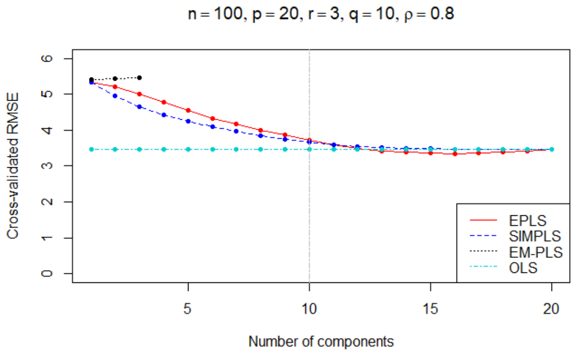


Figure 4.18: *CV-RMSEs over 100 replications when variation in the irrelevant part is large compared to the relevant part, n is large and the predictor variables are nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

shown in Figure 4.20. Here, the PLS2 methods have similar prediction performance for the first three components. Also, EPLS, SIMPLS, and OLS have similar RMSE performances after $q = 6$ components. This suggests that in this setting the methods are indistinguishable when $n = 100$ and $q \geq 6$ (except EM-PLS).

4.4.2 Conclusion

Several models that decompose the joint covariance matrix of the response and predictor variables were introduce, some of which are in the literature. We unified models under the umbrella of the RC models and showed that the models differ in terms of restrictions applied on the joint covariance matrix. The models presented were

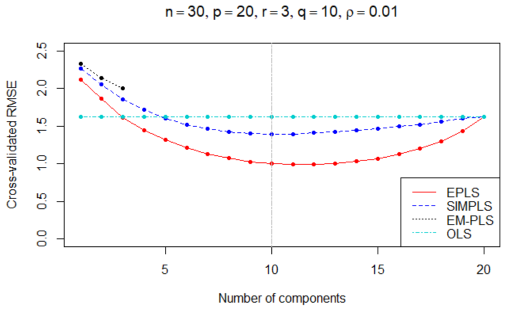


Figure 4.19: *CV-RMSEs over 100 replications when variation in the irrelevant part is large compared to the relevant part, n is small and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

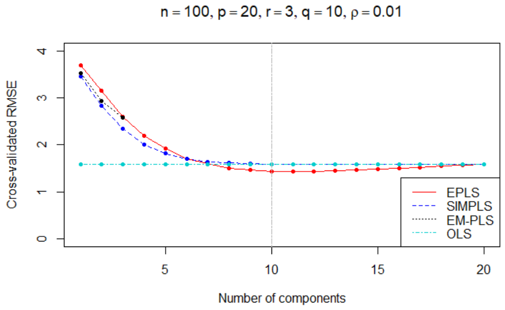


Figure 4.20: *CV-RMSEs over 100 replications when variation in the irrelevant part is large compared to the relevant part, n is large and the predictor variables are not nearly multicollinear. The vertical grey line corresponds the true number of relevant components.*

discussed under the decomposition of predictor variables alone and decomposition of response and predictor variables simultaneously. Also, isotropic and non-isotropic models were considered, where isotropic models have more restrictions compared to non-isotropic models. Futher, we reviewed three methods for estimating the parameters of the models and compared the prediction ability of the methods using several settings of data simulated from the $\mathrm{RC}_{\boldsymbol{x}}$ model. The methods considered include the EPLS (Cook et al., 2013), SIMPLS (De Jong, 1993), and EM-PLS (el Bouhaddani et al., 2018). The simulation results showed that the PLS2 methods have better prediction performance compared to OLS when n is small and the predictor variables are nearly multicollinear. However, when n is large the OLS and PLS2 methods have similar prediction performances. Besides, the EPLS method performs better than SIMPLS and EM-PLS with regard to prediction performance when the sample size is small and predictor variables are not nearly multicollinear, this because the

EPLS method estimates and ignores the irrelevant components (Cook et al., 2013). The predictive performance of EM-PLS is similar to SIMPLS when the p is large, but performs poorly when n is large compared to when n is small. The poor performance of EM-PLS is because the data was simulated from a non-isotropic RC$_{\boldsymbol{x}}$ model and the EM-PLS method is meant to for isotropic RC$_{\boldsymbol{yx}}$ models. Further simulations (not included) show that when data are generated from the isotropic model the predictive performance of the EM-PLS method is comparable to the EPLS and SIMPLS methods when n is large.

Chapter 5

Sparse Multivariate Partial Least Squares

5.1 Introduction

Partial least squares (PLS) regression was introduced as an estimator of the parameters of two statistical models (i.e, the RC$_{\boldsymbol{x}}$ and RC$_{\boldsymbol{yx}}$ models) in Chapters 3 and 4. The models were described as restrictions on the joint covariance matrix of the response and predictor variables. The restriction is that after a rotation of $\boldsymbol{x}$ (and $\boldsymbol{y}$) the covariance matrix of $\boldsymbol{x}$ (and $\boldsymbol{y}$) will be block diagonal and only one block contains the needed information.

In regression analysis, sometimes the interest is both to improve prediction performance and enable model interpretability. PLS is known to have better prediction performance compared to OLS when predictor variables are nearly multicollinear (Almøy, 1996; Helland and Almøy, 1994). However, it will not ease model interpretability especially when the number of predictor variables is large (Andries and Martin, 2013; Mehmood et al., 2012) because relevant components are linear combinations of all the predictor variables. Model interpretability is associated with the principle of sparsity, which assumes that only a few predictor variables have nonzero correlations with the response variables. Several methods have been proposed for variable selection in multivariate linear regression analysis, and their theoretical and empirical justification have been explored.

The lasso (Tibshirani, 1996) is one of the popular methods for simultaneous parameter estimation and variable selection in regression models, and has been extended to the adaptive lasso (Zou, 2006) approach to allow coefficients to have different penalties and to improve selection accuracy when the number of predictor variables far exceeds the sample size. Fan and Li (2001) presented a penalized least squares approach which can account for stochastic errors, as they are usually ignored when selecting variables. A penalization technique which accounts for the dependency structure of a multivariate response variable in a regression model was proposed by Wang (2015) by constructing a penalized conditional log-likelihood of each response on other responses and predictor variables. Some authors such as Yuan and Lin (2006) and Simon and Tibshirani (2012) presented a penalized approach for selecting grouped variables for prediction while Simon et al. (2013) extended the group lasso method to accommodate within-group sparsity. Dondelinger et al. (2020) proposed a penalization method to aid similarity between coefficients of different groups when they have a high-dimensional structure. Furthermore, Huang et al. (2010) developed a theoretical justification for the used of group sparsity when the group structure is known, and Knight and Fu (2000) discussed the asymptotic behaviours of several lasso-type penalty functions in a regression model. Unified frameworks for penalization are presented in Wang and Leng (2007) and Negahban et al. (2012). The literature on variable selection in linear regression models is extensive, but the focus of this chapter is sparsity in partial least squares.

PLS can be improved by combining the assumptions that only a few relevant components are required and only a few predictor variables have nonzero correlation with the response variables (Chun and Keleş, 2010; Kawano et al., 2015; Zhu et al., 2020).

In this chapter, a sparse relevant components (SRC) model that combines the assumptions of sparsity and relevant component is introduced. This model also places restrictions on the joint covariance matrix of the response and predictor variables. In addition, different methods for estimating the parameters of the model will be discussed. Moreover, we propose two methods for parameter estimation; a modified envelope-based PLS (ME-SPLS) and a two-stage SPLS (2S-SPLS). The ME-SPLS method finds relevant components and enables sparsity of predictor variables simultaneously. The 2S-SPLS method is a two-stage technique which finds relevant components in the first stage and imposes sparsity in the second stage.

5.2 A model for sparse regression

The proposed SRC model, just as in chapters 3 and 4, can be described in terms of a decomposition of the joint covariance matrix between the predictor and response variables. The model assumes that some predictor variables have zero covariances with the response variables. The set of predictor variables that have nonzero covariance with the response variables will be called the active set, while the set that do not will be called the inactive set. The subscripts $\mathcal{I}$ and $\mathcal{A}$ are used if a term is related with the inactive or active predictor variables; respectively. Without loss of generality, assume the predictor variables are ordered such that the first $p_{\mathcal{A}}$ predictor variables are active, $\mathcal{A} = \{1, 2, \dots, p_{\mathcal{A}}\}$, and the remaining $p_{\mathcal{I}} = p - p_{\mathcal{A}}$ are inactive, where $p_{\mathcal{A}}$ is the number of active predictor variables and $p_{\mathcal{I}}$ is the number of inactive predictor variables, $p_{\mathcal{A}} + p_{\mathcal{I}} = p$ (Chun and Keleş, 2010; Zhu et al., 2020). Let the vector of predictor variables $\boldsymbol{x}^T = (\boldsymbol{x}_{\mathcal{A}}^T, \boldsymbol{x}_{\mathcal{I}}^T) \in \mathbb{R}^p$, where $\boldsymbol{x}_{\mathcal{A}} \in \mathbb{R}^{p_{\mathcal{A}}}$ and $\boldsymbol{x}_{\mathcal{I}} \in \mathbb{R}^{p_{\mathcal{I}}}$ are the active and inactive sets, respectively. The active predictor variables are nearly multicollinear and, hence, can be decomposed into relevant and irrelevant components. In addition, the active and inactive sets are correlated. Let $\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2) \in \mathbb{R}^{p \times p}$ be a rotation matrix with restrictions such that

$$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Gamma}_{1_{\mathcal{A}_1}} & \boldsymbol{\Gamma}_{2_{\mathcal{A}_2}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{p_{\mathcal{I}}} \end{pmatrix}, \quad \text{with } \boldsymbol{\Gamma}_1 = \begin{pmatrix} \boldsymbol{\Gamma}_{1_{\mathcal{A}_1}} \\ \boldsymbol{0} \end{pmatrix} \text{ and } \boldsymbol{\Gamma}_2 = \begin{pmatrix} \boldsymbol{\Gamma}_{2_{\mathcal{A}_2}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{p_{\mathcal{I}}} \end{pmatrix}, \tag{5.1}$$

where $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{p \times q}$, $\boldsymbol{\Gamma}_2 \in \mathbb{R}^{p \times (p-q)}$, $\boldsymbol{\Gamma}_{1_{\mathcal{A}_1}} \in \mathbb{R}^{p_{\mathcal{A}} \times q}$, $\boldsymbol{\Gamma}_{2_{\mathcal{A}_2}} \in \mathbb{R}^{p_{\mathcal{A}} \times (p_{\mathcal{A}} - q)}$, and $\boldsymbol{I}_{p_{\mathcal{I}}} \in \mathbb{R}^{p_{\mathcal{I}} \times p_{\mathcal{I}}}$. Let components $\boldsymbol{t}^*$ be defined such that

$$\boldsymbol{t}^* = \boldsymbol{\Gamma}^T \boldsymbol{x} = (\boldsymbol{\Gamma}_{1_{\mathcal{A}_1}}^T \boldsymbol{x}_{\mathcal{A}},\ \boldsymbol{\Gamma}_{2_{\mathcal{A}_2}}^T \boldsymbol{x}_{\mathcal{A}},\ \boldsymbol{x}_{\mathcal{I}}). \tag{5.2}$$

The SRC model which combines sparsity and relevant components assumes that the $\operatorname{cov}(\boldsymbol{x}, \boldsymbol{y})$ is captured by only $\operatorname{cov}(\boldsymbol{\Gamma}_{1_{\mathcal{A}_1}}^T \boldsymbol{x}_{\mathcal{A}}, \boldsymbol{y})$ alone. The covariance matrix of the model is given by

$$\boldsymbol{\Sigma}^* = \begin{pmatrix} \boldsymbol{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_1}} & \boldsymbol{0} & \boldsymbol{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_1} \boldsymbol{x}_{\mathcal{I}}} & \boldsymbol{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_1} \boldsymbol{y}} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_2}} & \boldsymbol{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_2} \boldsymbol{x}_{\mathcal{I}}} & \boldsymbol{0} \\ \boldsymbol{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_1} \boldsymbol{x}_{\mathcal{I}}} & \boldsymbol{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_2} \boldsymbol{x}_{\mathcal{I}}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_{\mathcal{I}}} & \boldsymbol{0} \\ \boldsymbol{\Sigma}^T_{\boldsymbol{t}^*_{\mathcal{A}_1} \boldsymbol{y}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}} \end{pmatrix}, \tag{5.3}$$

where $\mathbf{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_1}} \in \mathbb{R}^{q\times q}$ and $\mathbf{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_2}} \in \mathbb{R}^{(p_{\mathcal{A}}-q)\times(p_{\mathcal{A}}-q)}$ are covariance matrices of the relevant and irrelevant parts of the active set, and $\mathbf{\Sigma}_{\boldsymbol{x}_{\mathcal{I}}} \in \mathbb{R}^{p_{\mathcal{I}}\times p_{\mathcal{I}}}$ is the covariance matrix of the inactive predictor variables. Also, $\mathbf{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_1}\boldsymbol{y}} \in \mathbb{R}^{q\times r}$ is the covariance matrix between the relevant part of the active set and the response variable, $\mathbf{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_1}\boldsymbol{x}_{\mathcal{I}}} \in \mathbb{R}^{q\times p_{\mathcal{I}}}$ is the covariance between inactive predictor variables and relevant components of the active predictor variables, $\mathbf{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_2}\boldsymbol{x}_{\mathcal{I}}} \in \mathbb{R}^{(p_{\mathcal{A}}-q)\times p_{\mathcal{I}}}$ is the covariance between inactive predictor variables and irrelevant components of the active predictor variables and $\mathbf{0}$ is a matrix of zeros indicating uncorrelated partitions of $\mathbf{\Sigma}^*$.

Note that the covariance matrices, $\mathbf{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_1}}$ and $\mathbf{\Sigma}_{\boldsymbol{t}^*_{\mathcal{A}_2}}$, have the same structure as the covariance matrices of the Krylov model or RC model (detail in Chapter 3).

5.3 Review of variable selection in PLS regression

In previous sections of this chapter, the principle of sparsity and relevant components were introduced. For the remainder of this chapter, we will refer to them as variable selection and component extraction, respectively.

Regularization methods have been used for variable selection in linear regression (Tibshirani, 1996), relevant basis identification in principal component analysis (Guan and Dy, 2009; Zou et al., 2006), covariance estimation in analysis of covariance(Rothman et al., 2008) and so on. Variable selection can be a preprocessing step before the development of a predictive model (Friedman et al., 2001; James et al., 2013) or part of the model fitting procedure; the two procedures can be implemented in PLS. Eriksson et al. (2006) proposed a technique for variable selection in PLS known as variable importance in projection (VIP). VIP involves using the contributions of predictor variables when determining the basis vectors as indicators of their importance to the regression model. Other algorithms for variable selection in PLS involves adding or removing variables from the PLS procedure depending on whether or not they improve the predictive performance of the model (Mehmood et al., 2012; Osborne et al., 1997). In Section 5.3.1, we review several regularization methods for estimating the parameters of the SRC model in (5.3). These methods selects variables by imposing a penalty function on some parameters of the model. Here, we consider independent random samples of size n, with r responses variables and p predictor variables, from a multivariate normal distribution with zero mean.

5.3.1 Regularized PLS regression methods

For variable selection and component extraction, PLS forces some parameters of the model to exactly zero (Boulesteix and Strimmer, 2006; Zhu et al., 2020). The penalized parameters are either the regression coefficients or the relevant basis matrix, $\boldsymbol{\Gamma}_1$. By forcing parameters to exactly zero the predictor variables that show strong relationship with the response variables can be identified. For instance, Huang et al. (2004) introduce a penalized PLS1 approach which penalizes the last PLS1 estimator. Using the response and relevant components the following predictor is constructed

$$\begin{aligned} y_{q,PLS} =& \mu_y + c_1 t_1 + c_2 t_2 + \cdots + c_q t_q \\ =& \mu_y + \boldsymbol{\beta}_{q,PLS}^T(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}), \end{aligned}$$

where $t_i^{\prime} s$ are column vectors of relevant components, $\boldsymbol{t}_1 \in \mathbb{R}^{n\times q}$, and c_j's are estimated loadings for the response variable. Then a soft-threshold penalty is imposed on the $\tilde{\beta}_j \in \boldsymbol{\beta}_{q,PLS}$'s to force small regression coefficients to exactly zero;

$$\hat{\beta}_j = \text{sign}(\tilde{\beta}_j)(|\tilde{\beta}_j| - \lambda)_+ \quad j = 1, \ldots, p,$$

where λ is the tuning parameter, and sign$(\cdot)$ is the sign function. Hence, the jth predictor variable, x_j, will be declared inactive if $\hat{\beta}_j = 0$. Chun and Keleş (2010) proposed a sparse PLS (SPLS) regression method where the penalty was imposed on a substitute basis vector rather than the original basis vector (loadings). According to Jolliffe et al. (2003) penalizing the original basis vector alone may not give basis vectors that are sparse enough. The proposed objective function is

$$\begin{aligned} (\boldsymbol{\gamma}, \boldsymbol{c}^*) = \underset{\boldsymbol{\gamma},\boldsymbol{c}^*}{\arg\min} \Big[& -\kappa\boldsymbol{\gamma}^T\boldsymbol{M}\boldsymbol{M}^T\boldsymbol{\gamma} + (1-\kappa)(\boldsymbol{c}^* - \boldsymbol{\gamma})^T\boldsymbol{M}\boldsymbol{M}^T(\boldsymbol{c}^* - \boldsymbol{\gamma}) + \\ & \lambda_1\|\boldsymbol{c}^*\|_1 + \lambda_2\|\boldsymbol{c}^*\|_2^2 \Big] \\ & \text{subject to} \quad \boldsymbol{\gamma}^T\boldsymbol{\gamma} = 1, \end{aligned} \tag{5.4}$$

where $\boldsymbol{M} = \boldsymbol{S}_{\boldsymbol{x}\boldsymbol{y}} \in \mathbb{R}^{p\times r}$ is the covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$, $\boldsymbol{\gamma} \in \mathbb{R}^p$ is a basis vector, and $\boldsymbol{c}^* \in \mathbb{R}^p$ is a substitute weight. Also $\|\cdot\|_2^2$ and $\|\cdot\|_1$ are norms introduced to solve the problem of dependence and interpretation, respectively, $\kappa \in (0, 1)$ controls the trade-off between the first and second terms. However, the basis vectors are estimated separately which can result in different predictor variables being selected in different iterations, and may not reduce the number of predictor variables substantially.

For variable selection with NIPALS, Lee et al. (2011) proposed a modification of the algorithm. A linear relationship between two covariance matrices was constructed; the first is the covariance between the response and predictor variables ($\boldsymbol{M}$), and the second is the covariance between the response variables and relevant components ($C = Y^T\boldsymbol{t} \in \mathbb{R}^r$). The relationship is given by

$$\boldsymbol{M}^T = C\boldsymbol{\gamma}^T + \boldsymbol{e}$$

where the basis, $\boldsymbol{\gamma} \in \mathbb{R}^p$, is the vector coefficients of the model, and $\boldsymbol{e} \in \mathbb{R}^{r\times p}$ is the irrelevant information. To enforce sparsity, an elastic-net penalty was introduced to down-weight coefficients; the elastic net penalty is used as it can select more than n variables when $n > p$ (a feature lacking for the lasso penalty) and it is not outperformed by the ridge method in terms of prediction when predictor variables are highly correlated (Zou and Hastie, 2005). The function to be minimized is

$$\sum_{j=1}^{p}\left\{\frac{1}{2}\left(M_j^T - C\gamma_j\right)^T\left(M_j^T - C\gamma_j\right) + \lambda_1|\gamma_j| + \lambda_2\gamma_j^2\right\},$$

where $M_j^T \in \mathbb{R}^r$ is the jth column of $\boldsymbol{M}^T$, γ_j is the jth element of $\boldsymbol{\gamma}$, and $|\cdot|$ is the absolute value function. In the methods presented above, small values in the column of $\boldsymbol{\Gamma}_1$ are forced to zero one column at a time, which may not lead to a substantial reduction in the number of active predictor variables, because different columns may force different rows of $\boldsymbol{\Gamma}_1$ to exactly zero. Liu et al. (2014) proposed a method which performs variable selection jointly rather than sequentially by penalizing the "best" q relevant basis together. And proposed the jointly sparse PLS regression objective function given by

$$\begin{gathered}\boldsymbol{\Gamma}_1 = \underset{\boldsymbol{\Gamma}_1}{\arg\min}\left[-\sum_{j=1}^{q}\boldsymbol{\gamma}_j^T\boldsymbol{M}\boldsymbol{M}^T\boldsymbol{\gamma}_j + \lambda\sum_{i=1}^{p}\|\boldsymbol{\gamma}_i\|_2\right]\\ \text{subject to} \quad \boldsymbol{\gamma}_j^T\boldsymbol{\gamma}_j = 1, \quad \boldsymbol{\gamma}_k^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\gamma}_j = 0, \quad j \neq k,\end{gathered}$$

where $\boldsymbol{\gamma}_j$ is the jth column of $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{p\times q}$, and $\boldsymbol{\gamma}_i$ is the ith row of $\boldsymbol{\Gamma}_1$. As mention previously, the rows are the groups and the L_2 penalty on $\boldsymbol{\Gamma}_1$ behaves like the L_1

penalty at the group level, that is, for a large enough value of λ an entire row will be set to zero. This is synonymous to the L_1 penalty, which sets individual entries to zero. That is, for the L_1 penalty there is one element in each group. If an entry in a row is nonzero the entire row is regarded as important. However, if each group has exactly one entire L_1 penalty will be similar to the L_2 penalty (Friedman et al., 2010*a*).

The methods reviewed thus far computes the columns of $\boldsymbol{\Gamma}_1$ sequentially and uses no distributional assumptions. Likelihood-based methods which estimate parameters together are discussed in Section 5.3.2, which could provide additional gains in terms of estimation.

5.3.2 Likelihood-based methods

A likelihood-based method for simultaneous variable selection and component extraction is reviewed in this section. In particular, focus is on the recently proposed envelope-based sparse PLS of Zhu et al. (2020).

Envelope-based sparse PLS method

The envelope method (detail in chapter 4) (Cook et al., 2013) provides more gains in efficiency than SIMPLS (De Jong, 1993) because it extracts and then ignores variation in the data which is irrelevant, and bases estimation and prediction on relevant information. In simulation studies, it was found that the envelope method performs better than traditional PLS2 methods in terms of predictions and variable selection (Cook et al., 2013; Zhu et al., 2020). Recall that from chapter 4 the objective function of the envelope method for estimating the relevant basis matrix, $\boldsymbol{\Gamma}_1$, is given by

$$\begin{aligned}\hat{\boldsymbol{\Gamma}}_1 &= \underset{\boldsymbol{\Gamma}_1\in\mathcal{G}}{\arg\min}\left\{n\log|\boldsymbol{\Gamma}_1^T(\boldsymbol{S_{xx}}-\boldsymbol{S_{xy}}\boldsymbol{S_{yy}}^{-1}\boldsymbol{S_{xy}}^T)\boldsymbol{\Gamma}_1| + n\log|\boldsymbol{\Gamma}_1^T\boldsymbol{S_{xx}}^{-1}\boldsymbol{\Gamma}_1|\right\} \\ &= \underset{\boldsymbol{\Gamma}_1\in\mathcal{G}}{\arg\min}\left\{\mathcal{J}(\boldsymbol{\Gamma}_1)\right\},\end{aligned} \tag{5.5}$$

where $\boldsymbol{S_{xy}}$ is the sample covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$, $\boldsymbol{S_{xx}}$ and $\boldsymbol{S_{yy}}$ are the sample covariance matrices of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and estimation is performed over the

Grassmann manifold, $\mathcal{G}$. Using $\hat{\boldsymbol{\Gamma}}_1$ from Equation (5.5), the regression coefficients and other parameters can be estimated (detail in Chapter 4). Estimating the regression coefficients makes it possible to predict new response variables; however, selection of active predictor variables may not be possible since all the regression coefficients are nonzero. To select active predictor variables using the envelope method, Zhu et al. (2020) imposed a penalty function on $\boldsymbol{\Gamma}_1$ to force some parameters to exactly zero; the method is known as the envelope-based sparse PLS (E-SPLS) method. In Zhu et al. (2020), the goal was to select the predictor variables which jointly explain all the response variables together. The group lasso penalty was introduced into Equation (5.5) for group selection to give the following penalized objective function

$$\hat{\boldsymbol{\Gamma}}_1 = \underset{\boldsymbol{\Gamma}_1 \in \mathcal{G}}{\arg\min} \left\{ \mathcal{J}(\boldsymbol{\Gamma}_1) + \lambda \sum_{j=1}^{p} w_j \|\boldsymbol{\gamma}_j\|_2 \right\}, \tag{5.6}$$

where $\boldsymbol{\gamma}_j$ is the jth row (group) of $\boldsymbol{\Gamma}_1$, $w_j = 1/\|\hat{\boldsymbol{\gamma}}_j\|_2^{\vartheta}$ the ith group weight, $\vartheta = (0.5, 1, 2, 4, 8)$ regulates the size of the weights, and $\hat{\boldsymbol{\gamma}}_j$ is data dependent and can be estimates from the envelope method (Zou et al., 2006). The $\lambda \geq 0$ is the tuning parameter, and $\|\cdot\|_2$ is the L_2 norm. Penalizing the rows of $\boldsymbol{\Gamma}_1$ in order to force some regression coefficients to zero is appropriate because the rows of $\boldsymbol{\Gamma}_1$ act as surrogate regression coefficients and can be used to determine the contributions of the predictor variables (Mehmood et al., 2012). Using the group lasso penalty, if the jth row is shrunk to exactly zero the corresponding row of the regression coefficient is shrunk to exactly zero. That is, in E-SPLS predictor variables are inactive if their corresponding rows in the basis matrix, $\boldsymbol{\Gamma}_1$, are zero and active if they are nonzero. Without loss of generality, the solution $\hat{\boldsymbol{\Gamma}}_1$ from Equation (5.6) will be identical to the matrix in Equation (5.1) and given by

$$\hat{\boldsymbol{\Gamma}}_1 = \begin{pmatrix} \hat{\boldsymbol{\Gamma}}_{1_{\mathcal{A}_1}} \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^{p \times q}. \tag{5.7}$$

The basis matrix is vital for determining the regression coefficients given by

$$\hat{\boldsymbol{B}} = \hat{\boldsymbol{\Gamma}}_1 (\hat{\boldsymbol{\Gamma}}_1^T \boldsymbol{X}^T \boldsymbol{X} \hat{\boldsymbol{\Gamma}}_1)^{-1} \hat{\boldsymbol{\Gamma}}_1^T \boldsymbol{X}^T \boldsymbol{Y} = \begin{pmatrix} \hat{\boldsymbol{B}}_{\mathcal{A}} \\ \mathbf{0} \end{pmatrix}. \tag{5.8}$$

Then, the matrix of regression coefficients for the active predictor variables is denoted by $\hat{\boldsymbol{B}}_{\mathcal{A}} \in \mathbb{R}^{p_{\mathcal{A}} \times r}$. This method can be more efficient in terms of variable selection and component extraction than those previously proposed in the literature (Zhu et al., 2020). However, E-SPLS is limited because of the assumption made in Equation (5.7); that a group relationship exists and all response variables are explained by the same predictor variables.

In the next section, we propose a modified E-SPLS (ME-SPLS) and a novel two-stage SPLS (2S-SPLS) methods for variable selection and extraction, which are more flexible than E-SPLS (Zhu et al., 2020) and SPLS (Chun and Keleş, 2010).

5.3.3 Modified E-SPLS method

The E-SPLS method may not reduce the number of components substantially because it considers groups of variables using a group lasso penalty. For the group lasso penalty, if a group has one nonzero parameter it can cause all other parameters in that group to be nonzero. For instance, if $\boldsymbol{\mu}$ is a p-dimensional vector of means of a p random variables, $\boldsymbol{x}$, the penalty function $||\boldsymbol{\mu}||_2$ will be zero only when each μ_i, $j = 1, ..., p$ is zero.

We propose imposing a lasso penalty on $\boldsymbol{\Gamma}_1$ to force more coefficients to exactly zero. The lasso penalty function can force more parameters to exactly zero compared to the group lasso penalty, because with large values of λ, $\boldsymbol{\Gamma}_1$ will contain single elements in each column. For instance, let $p = 4$ and $q = 2$, the basis $\boldsymbol{\Gamma}_1$ and the regression coefficient matrix after a lasso penalty is introduced are

$$\boldsymbol{\Gamma}_1 = \begin{pmatrix} \gamma_{11} & 0 \\ 0 & \gamma_{22} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{B} = \begin{pmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \tag{5.9}$$

even when the true structure of $\boldsymbol{\Gamma}_1$ is

$$\boldsymbol{\Gamma}_1 = \begin{pmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \\ \gamma_{31} & \gamma_{32} \\ 0 & 0 \end{pmatrix} \tag{5.10}$$

Because $\boldsymbol{\Gamma}_1$ is a semi-orthogonal basis, if Equation (5.9) holds, the elements γ_{11} and γ_{22} will be ± 1 (up to permutation) if λ is sufficiently large. This implies that for appropriate values of λ the penalty will induce a non-group structure on the relevant basis matrix, and force parameters to zero resulting in the selection of a smaller number of active predictor variables. This provides more flexibility compared to the E-SPLS of Zhu et al. (2020). We propose an objective function for ME-SPLS given by

$$\hat{\boldsymbol{\Gamma}}_1 = \underset{\boldsymbol{\Gamma}_1 \in \mathcal{G}}{\arg\min} \left\{ \mathcal{J}(\boldsymbol{\Gamma}_1) + \lambda \sum_{j=1}^{p} \sum_{l=1}^{q} |\gamma_{jl}| \right\}, \tag{5.11}$$

The penalty is applied to elements of $\boldsymbol{\Gamma}_1$ rather than its rows, where γ_{jl} denote entries in the jth row and lth column of $\boldsymbol{\Gamma}_1$.

5.3.4 A two-stage sparse PLS method (2S-SPLS)

The methods discussed above for simultaneous variable selection and component extraction achieves sparsity in the regression coefficient by penalizing the matrix of relevant basis. We propose an alternative method which penalizes the regression coefficients directly. The method proposed here is a two-stage method which extracts components in one stage and penalizes regression coefficients in the next stage. In the component extraction stage, the matrix of relevant basis with the appropriate number of dimension, say q, is estimated from either the envelope method, Equation (5.5), or other regularization methods (Section 5.3.1) and then used to linearly transform the predictor variables such that

$$\boldsymbol{X} \in \mathbb{R}^{n\times p} \longrightarrow \boldsymbol{T}^* = \boldsymbol{X}\hat{\boldsymbol{\Gamma}}_1 \in \mathbb{R}^{n\times q}, \tag{5.12}$$

where $q < p$, $\hat{\boldsymbol{\Gamma}}_1 \in \mathbb{R}^{p\times q}$ is an estimated matrix of relevant basis which has no sparse structure, and $\boldsymbol{T}^*$ is the matrix of relevant components. Then the multivariate multiple linear regression model of $\boldsymbol{Y}$ on $\boldsymbol{T}^*$ is

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{T}^*\boldsymbol{B}^* + \boldsymbol{E} \\ &= \boldsymbol{X}\hat{\boldsymbol{\Gamma}}_1\boldsymbol{B}^* + \boldsymbol{E}, \end{aligned} \tag{5.13}$$

where $\boldsymbol{B}^* \in \mathbb{R}^{q\times r}$ is the matrix of regression coefficients for the regression of $\boldsymbol{Y}$ on $\boldsymbol{T}^*$ and $\boldsymbol{E} \in \mathbb{R}^{n\times r}$ is normally distributed with mean $\mathbf{0}$ and variance $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{t}^*} \in \mathbb{R}^{r\times r}$. The coefficients, $\boldsymbol{B}^*$, of the regression model, Equation (5.13), is estimated assuming $\hat{\boldsymbol{\Gamma}}_1$ is known. Let $\boldsymbol{T}^*$ be random, the joint likelihood of $\boldsymbol{Y}$ and $\boldsymbol{T}^*$ can be represented as the product of the conditional distribution of $\boldsymbol{Y}|\boldsymbol{T}^*$ and the marginal distribution of

$\boldsymbol{T}^*$. Then the joint log-likelihood of the response variables and relevant components up to a constant is

$$\mathcal{L} = \frac{1}{n}\operatorname{tr}\left\{(\boldsymbol{Y} - \boldsymbol{T}^*\boldsymbol{B}^*)^T(\boldsymbol{Y} - \boldsymbol{T}^*\boldsymbol{B}^*)\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1}\right\} + \log|\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{t}^*}| + \log|\boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{t}^*}|. \tag{5.14}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{t}^*} \in \mathbb{R}^{q\times q}$ is the covariance matrix of the relevant components and $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{t}^*} \in \mathbb{R}^{r\times r}$ is the conditional covariance matrix of $\boldsymbol{y}$ given $\boldsymbol{t}^*$. Maximization is over the parameters, $\boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{t}^*}$, $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{t}^*}$ and $\boldsymbol{B}^*$, where

(a) $\boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{t}^*} = \boldsymbol{\Gamma}_1\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\Gamma}_1$,

(b) $\boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{y}} = \boldsymbol{\Gamma}_1\boldsymbol{\Sigma}_{\boldsymbol{xy}}$

(c) $\boldsymbol{\Sigma}_{\boldsymbol{y}|\boldsymbol{t}^*} = \boldsymbol{\Sigma}_{\boldsymbol{yy}} - \boldsymbol{\Sigma}_{\boldsymbol{yt}^*}\boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{t}^*}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{y}}$,

(d) $\boldsymbol{B}^* = \boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{t}^*}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{t}^*\boldsymbol{y}}$,

The solution can be found by first estimating parameters (a), (b), and (c) using maximum likelihood with $\boldsymbol{\Gamma}_1$ known. Recall that $\boldsymbol{\Gamma}_1$ can be estimated as $\hat{\boldsymbol{\Gamma}}_1$ using for example envelope method in Equation (5.5), and by maximum likelihood the covariance matrices can be estimated as $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{yy}} = \boldsymbol{S}_{\boldsymbol{yy}}$, $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{xx}} = \boldsymbol{S}_{\boldsymbol{xx}}$, and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{xy}} = \boldsymbol{S}_{\boldsymbol{xy}}$. Hence, we have

$$\begin{aligned}
&(a)\quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{t}^*\boldsymbol{t}^*} = \hat{\boldsymbol{\Gamma}}_1\hat{\boldsymbol{\Sigma}}_{\boldsymbol{xx}}\hat{\boldsymbol{\Gamma}}_1,\\
&(b)\quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{t}^*\boldsymbol{y}} = \hat{\boldsymbol{\Gamma}}_1\hat{\boldsymbol{\Sigma}}_{\boldsymbol{xy}}\\
&(c)\quad \hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}|\boldsymbol{t}^*} = \hat{\boldsymbol{\Sigma}}_{\boldsymbol{yy}} - \hat{\boldsymbol{\Sigma}}_{\boldsymbol{yt}^*}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{t}^*\boldsymbol{t}^*}^{-1}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{t}^*\boldsymbol{y}},
\end{aligned}$$

Then we have a profile log-likelihood in terms of $\boldsymbol{B}^*$ given by

$$\mathcal{L}_{\text{pro}} = \frac{1}{n}\operatorname{tr}\left\{(\boldsymbol{Y} - \boldsymbol{T}^*\boldsymbol{B}^*)^T(\boldsymbol{Y} - \boldsymbol{T}^*\boldsymbol{B}^*)\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1}\right\} + \log|\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}| + \log|\boldsymbol{S}_{\boldsymbol{t}^*\boldsymbol{t}^*}|, \tag{5.15}$$

where $\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*} = \hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}|\boldsymbol{t}^*}$ and $\boldsymbol{S}_{\boldsymbol{t}^*\boldsymbol{t}^*} = \hat{\boldsymbol{\Sigma}}_{\boldsymbol{t}^*\boldsymbol{t}^*}$ are sample covariances. Then the objective function for estimating the regression coefficient $\boldsymbol{B}^*$ is

$$\hat{\boldsymbol{B}}^* = \underset{\boldsymbol{B}^*\in\mathbb{R}}{\arg\min}\left[\mathcal{L}_{\text{pro}}(\boldsymbol{B}^*)\right]. \tag{5.16}$$

It is more beneficial to reformulate the objective function in terms of the regression coefficients $\boldsymbol{B}$. Recall from Equations (5.12) and (5.13) that,

$$\boldsymbol{T}^* = \boldsymbol{X}\hat{\boldsymbol{\Gamma}}_1, \quad \boldsymbol{B} = \boldsymbol{\Gamma}_1\boldsymbol{B}^*, \quad \text{then} \quad \boldsymbol{B}^* = \boldsymbol{\Gamma}_1^T\boldsymbol{B}. \tag{5.17}$$

An objective function in terms of $\boldsymbol{B}$ is given by

$$\begin{aligned}\hat{\boldsymbol{B}} &= \underset{\boldsymbol{B}\in\mathbb{R}}{\arg\min}\left[\frac{1}{n}\operatorname{tr}\left\{(\boldsymbol{Y}-\boldsymbol{T}^{*}\boldsymbol{\Gamma}_1^T\boldsymbol{B})^T(\boldsymbol{Y}-\boldsymbol{T}^{*}\boldsymbol{\Gamma}_1^T\boldsymbol{B})\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1}\right\}+\log|\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}|+\log|\boldsymbol{S}_{\boldsymbol{t}^*\boldsymbol{t}^*}|\right]\\ &= \underset{\boldsymbol{B}\in\mathbb{R}}{\arg\min}\left[\mathcal{L}_{\text{pro}}(\boldsymbol{B})\right].\end{aligned} \tag{5.18}$$

Using Equation (5.18) the regression coefficients $\boldsymbol{B}$ can be penalized and estimated directly.

To force regression coefficients to exactly zero, a penalty on $\boldsymbol{B}$ can be introduced into Equation (5.18). The penalty to be introduces depends on the goal of the study. If interest is to select only the predictor variables which explain the response variables together a group-lasso penalty can be introduced (Yuan and Lin, 2006), but if group structure is not of interest the lasso penalty can be used (Tibshirani, 1996). In some studies, is may be important to introduce sparsity within the group, i.e allowing for the possibility of some zero coefficients within the groups, in which case a sparse group-lasso penalty will be more appropriate (Friedman et al., 2010*a*; Simon et al., 2013). To enforce a group structure the L_2 norm is imposed on the rows (groups) of $\boldsymbol{B}$ and introduced into Equation (5.18). On the other hand, a within group structure can be implemented by introducing L_2 and L_1 norms. The objective function is given by

$$\hat{\boldsymbol{B}} = \underset{\boldsymbol{B}\in\mathbb{R}}{\arg\min}\left[\mathcal{L}_{\text{pro}}(\boldsymbol{B})+\lambda_1\sum_{j=1}^{p}\|\boldsymbol{b}_j\|_2+\lambda_2\sum_{j=1}^{p}\sum_{l=1}^{r}\tau_{jl}|b_{jl}|\right], \tag{5.19}$$

where $\boldsymbol{b}_j\in\mathbb{R}^r$ is the jth row of $\boldsymbol{B}$ (with each row representing a group), b_{jl} is the lth element of the jth row of $\boldsymbol{B}$, $\|\cdot\|_2$ is the L_2 norm, and $|\cdot|$ is the absolute value function. The tuning parameters $\lambda_1>0$ and $\lambda_2>0$ controls the amount of shrink on the group and within group penalties, respectively. The weights $\tau_{jl}=1/|\hat{b}_{jl}^{est}|$ controls the sizes of each coefficient in $\boldsymbol{B}$; where $\hat{b}_{jl}^{est}$ is an estimate obtained from a different method e.g envelope method. The first and second penalties enforces group and within group sparsity, respectively.

The function, Equation (5.19), takes advantage of the non-differentiability of $\|\boldsymbol{b}_j\|_2$ and $|b_{jl}|_2$ at $\|\boldsymbol{b}_j\|_2=\mathbf{0}$ and $|b_{jl}|=0$, respectively, and sets groups of coefficients to exactly zero. Without a penalty on $\boldsymbol{B}$ (that is, $\lambda_1=0$ and $\lambda_2=0$), the optimal solution for $\boldsymbol{B}$ is the matrix of regression coefficients, e.g for the envelope method. When $\lambda_1 = 0$ regression coefficients are forced to zero without any structure. Furthermore, when $\lambda_2 = 0$ a group structure is imposed on the regression coefficients. Solving Equation (5.19) for $\boldsymbol{B}$ is a convex optimization problem, hence, there is a global optimum $\hat{\boldsymbol{B}}$. That is, there is a zero subgradient of Equation (5.19) at $\hat{\boldsymbol{B}}$. The subgradient can be determined by taking the derivative of Equation (5.19) with respect to $\boldsymbol{B}$, which gives

$$\frac{2}{n}(\boldsymbol{\Gamma}_1 \boldsymbol{T}^{*T}\boldsymbol{Y} - \boldsymbol{\Gamma}_1 \boldsymbol{T}^{*T}\boldsymbol{T}^{*}\boldsymbol{\Gamma}_1^T \boldsymbol{B})\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} = \lambda_1 \boldsymbol{B}' + \lambda_2 \boldsymbol{B}'' \tag{5.20}$$

where $\boldsymbol{B}' \in \mathbb{R}^{p\times r}$ is a matrix with rows

$$\boldsymbol{b}_j' = \begin{cases} \frac{\boldsymbol{b}_j}{||\boldsymbol{b}_j||_2}, & \text{if } \ \boldsymbol{b}_j \neq \boldsymbol{0} \\ ||\boldsymbol{b}_j||_2 < 1 & \text{if } \ \boldsymbol{b}_j = \boldsymbol{0}, \end{cases}$$

and $\boldsymbol{B}'' \in \mathbb{R}^{p\times r}$ is a matrix with elements

$$b_{jl}'' = \begin{cases} \frac{b_{jl}}{|b_{jl}|}, & \text{if } \ b_{jl} \neq 0 \\ b_{jl} \in [-1, 1] & \text{if } \ b_{jl} = 0. \end{cases}$$

5.4 Oracle property

To investigate whether the estimators considered are good estimators, we have to verify if the estimators are consistent in terms of estimation and variable selection. This is known as the oracle property (Fan and Li, 2001; Fan et al., 2004). Consistency here is in a limited setting when n gets large with all other aspects of the problem held fixed. A modelling procedure is called an oracle procedure if it possesses the following oracle properties:

a) Choosing the correct set of active predictor variables with probability tending to 1 as n increases.

b) The identified set of parameters are root-n consistent (Zou, 2006). That is, $\sqrt{n}(\hat{\boldsymbol{B}}_{\mathcal{A}}^0 - \boldsymbol{B}_{\mathcal{A}}) \sim \boldsymbol{N}_{p_{\mathcal{A}}}(\boldsymbol{0}_{p_{\mathcal{A}}}, \boldsymbol{\Sigma}_{\mathcal{A}})$, where $\boldsymbol{\Sigma}_{\mathcal{A}}$ is the covariance matrix of the active predictor variables, $\boldsymbol{B}_{\mathcal{A}}$ is the estimator of the active set and $\hat{\boldsymbol{B}}_{\mathcal{A}}^0$ is the oracle estimator.

That is, the penalized estimator is asymptotically equivalent to the true estimator after discarding all predictor variables whose true regression coefficients are zero. An oracle procedure must satisfy selection and estimation consistency, i.e (a) and (b) above.

5.5 Implementation of the methods

In this section, the implementation of the 2S-SPLS, E-SPLS and SPLS methods are presented. The algorithms associated with these methods are based on repeated updates of the starting values. The algorithms are described as follows:

5.5.1 SPLS algorithm

Chun and Keleş (2010) proposed the following algorithm for the SPLS objective function in Equation (5.4). Recall that $\mathcal{A}$ is the set of active predictor variables and q the number of relevant components. Let $\boldsymbol{X}_{\mathcal{A}}$ denote the a submatrix of $\boldsymbol{X}$ containing the active predictor variables.

```
Step 1: set $\boldsymbol{B} = \mathbf{0}$, $\mathcal{A} = \{\cdot\}$, $q = 1$, and $\boldsymbol{X}_1 = \boldsymbol{X}$.
Step 2: while $q^* \leq p$,
    (a) find $\hat{\boldsymbol{\gamma}}$ by solving Equation (5.4) with $\boldsymbol{M} = \boldsymbol{X}_1^T \boldsymbol{Y}$.
    (b) update $\mathcal{A} = \{i : \hat{\gamma}_i \neq \mathbf{0}\} \cup \{i : \hat{\boldsymbol{b}}_i \neq \mathbf{0}\}$.
    (c) using $q^*$ number of components fit PLS with $\boldsymbol{X}_{\mathcal{A}}$ predictor variables and
    (d) update $\hat{\boldsymbol{B}}$ using the PLS estimates in $(c)$,
        update $q^*$ with $q^* \longleftarrow q^* + 1$
        update $\boldsymbol{X}_1$ with $\boldsymbol{X}_{1\mathcal{A}} \longleftarrow \boldsymbol{X}_{1\mathcal{A}}\Big(\boldsymbol{I} - \boldsymbol{P}_{\mathcal{A}}(\boldsymbol{P}_{\mathcal{A}}^T \boldsymbol{P}_{\mathcal{A}})^{-1} \boldsymbol{P}_{\mathcal{A}}^T\Big)$, where
        $\boldsymbol{P}_{\mathcal{A}} = \boldsymbol{X}_{\mathcal{A}}^T \boldsymbol{X}_{\mathcal{A}} \boldsymbol{\Gamma}_{1_{\mathcal{A}_1}} \Big(\boldsymbol{\Gamma}_{1_{\mathcal{A}_1}}^T \boldsymbol{X}_{\mathcal{A}}^T \boldsymbol{X}_{\mathcal{A}} \boldsymbol{\Gamma}_{1_{\mathcal{A}_1}}\Big)^{-1}$
```

Simulations studies (Chun and Keleş, 2010) suggest that a $\kappa < 0.5$ often avoids local solution issues, and setting λ_2 equal to a very large value gives an estimator that depends only on λ_1, which makes computation easier. Furthermore, λ_1 is chosen through cross-validation.

5.5.2 E-SPLS algorithm

Recall that Equations (5.6) and (5.11) are invariant under right orthogonal transformation, that is for any orthogonal matrix $\boldsymbol{O} \in \mathbb{R}^{q\times q}$, $f(\boldsymbol{\Gamma}_1\boldsymbol{O}) = f(\boldsymbol{\Gamma}_1)$ (Edelman et al., 1998). Let $\mathcal{S}_{\boldsymbol{\Gamma}_1}$ denote the column space of $\boldsymbol{\Gamma}_1$. Then $\mathcal{S}_{\boldsymbol{\Gamma}_1}$ is in the Grassmann manifold;

$$\mathcal{S}_{\boldsymbol{\Gamma}_1} = \{\boldsymbol{\Gamma}_1\boldsymbol{O} | \boldsymbol{O} \in \mathbb{R}^{q\times q}, \boldsymbol{O}^T\boldsymbol{O} = \boldsymbol{I}_q\} \in \mathcal{G}_{(q,p)}, \tag{5.21}$$

where $\mathcal{G}_{(q,p)}$ is the Grassmann manifold of all q-dimensional subspaces in $\mathbb{R}^p$. The target is to find the subspace of $\hat{\boldsymbol{\Gamma}}_1$ in Equations (5.6) and (5.11). Let $\boldsymbol{D} = (\nabla f(\boldsymbol{\Gamma}_1))^T\boldsymbol{\Gamma}_2 \in \mathbb{R}^{q\times(p-q)}$ be the matrix of directional derivative of f: the rate of change in the direction of $\boldsymbol{\Gamma}_2$. Also, let $\boldsymbol{Z} \in \mathbb{R}^{p\times p}$ be a skew-symmetric matrix given by (Gallivan et al., 2003)

$$\boldsymbol{Z} = \begin{pmatrix} \boldsymbol{0}_q & \boldsymbol{D} \\ -\boldsymbol{D}^T & \boldsymbol{0}_{p-q} \end{pmatrix} \tag{5.22}$$

The algorithm updates the starting basis of $\boldsymbol{\Gamma}$ via a right orthogonal matrix multiplication (rotation). For each step the update is

$$\boldsymbol{\Gamma}_{(t+1)} = \boldsymbol{\Gamma}_{(t)} \exp\{\delta\boldsymbol{Z}\}, \tag{5.23}$$

where $\delta \in (0,1)$ is the step size. The $\boldsymbol{D}$ and $\boldsymbol{Z}$ are updated for each iteration until a stopping rule is reached. A stopping rule that often guarantees a maximizer is that the norm of $\boldsymbol{D}$ should be sufficiently small. The algorithm is as follows (Adragni et al., 2012):

Step 1: At $t = 0$, let $\boldsymbol{\Gamma}_{(0)} \in \mathbb{R}^{p\times p}$ be an initial matrix

Step 2: Repeat until $\|\boldsymbol{D}\| < \epsilon$,

(a) Compute the directional derivative $\boldsymbol{D}$ and construct the skew-symmetric matrix $\boldsymbol{Z}$.

(b) Update $\boldsymbol{\Gamma}_{(t+1)} = \boldsymbol{\Gamma}_{(t)} \exp\{\delta\boldsymbol{Z}\}$ such that $f(\boldsymbol{\Gamma}_{1(t+1)}) > f(\boldsymbol{\Gamma}_{1(t)})$.

Step 3: The first q-dimensional columns of $\boldsymbol{\Gamma}$ at the last iteration is a basis of $\hat{\mathcal{S}}_{\boldsymbol{\Gamma}_1}$.

For each iteration, values of $\delta \in (0,1)$ are used to obtain several $\boldsymbol{\Gamma}_{t+1}$, and the $\boldsymbol{\Gamma}_{t+1}$ such that $f(\boldsymbol{\Gamma}_{1(t+1)}) > f(\boldsymbol{\Gamma}_{1(t)})$ is used in the next iteration.

5.5.3 2S-SPLS algorithm

Let $\boldsymbol{X}^* = \boldsymbol{T}^*\boldsymbol{\Gamma}_1^T \in \mathbb{R}^{n\times p}$, then Equation (5.20) can be written as

$$\begin{aligned}
\boldsymbol{X}_k^{*T}\Big(\boldsymbol{Y} - \sum_{j=1}^{p}\boldsymbol{X}_j^*\boldsymbol{b}_j\Big)\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} &= \lambda_1\boldsymbol{b}'_k + \lambda_2\boldsymbol{b}''_{lk}\\
\boldsymbol{X}_k^{*T}\Big(\boldsymbol{Y} - \sum_{j\neq k}\boldsymbol{X}_j^*\boldsymbol{b}_j - \boldsymbol{X}_k^*\boldsymbol{b}_k\Big)\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} &= \lambda_1\boldsymbol{b}'_k + \lambda_2\boldsymbol{b}''_{lk}\\
\Big(\boldsymbol{X}_k^{*T}(\boldsymbol{Y} - \sum_{j\neq k}\boldsymbol{X}_j^*\boldsymbol{b}_j) - \boldsymbol{X}_k^{*T}\boldsymbol{X}_k^*\boldsymbol{b}_k\Big)\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} &= \lambda_1\boldsymbol{b}'_k + \lambda_2\boldsymbol{b}''_{lk}\\
\Big(\boldsymbol{X}_k^{*T}\boldsymbol{r}_{(-k)} - \boldsymbol{X}_k^{*T}\boldsymbol{X}_k^*\boldsymbol{b}_k\Big)\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} &= \lambda_1\boldsymbol{b}'_k + \lambda_2\boldsymbol{b}''_{lk}\\
-\boldsymbol{X}_k^{*T}\boldsymbol{X}_k^*\boldsymbol{b}_k\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} - \lambda_1\frac{\boldsymbol{b}_k}{\|\boldsymbol{b}_k\|_2} &= -\boldsymbol{X}_k^{*T}\boldsymbol{r}_{(-k)}\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} + \lambda_2\boldsymbol{b}''_{lk}\\
\Big(\boldsymbol{X}_k^{*T}\boldsymbol{X}_k^*\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} - \frac{\lambda_1}{\|\boldsymbol{b}_k\|_2}\Big)\hat{\boldsymbol{b}}_k &= S\Big(\boldsymbol{X}_k^{*T}\boldsymbol{r}_{(-k)}\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1}, \lambda_2\Big)
\end{aligned}$$

$$\hat{\boldsymbol{b}}_k = \frac{S\Big(\boldsymbol{X}_k^{*T}\boldsymbol{r}_{(-k)}\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1}, \lambda_2\Big)}{\boldsymbol{X}_k^{*T}\boldsymbol{X}_k^*\boldsymbol{S}_{\boldsymbol{y}|\boldsymbol{t}^*}^{-1} - \frac{\lambda_1}{\|\boldsymbol{b}_k\|_2}} \tag{5.24}$$

where $\boldsymbol{r}_{(-k)} = \boldsymbol{Y} - \sum_{j\neq k}\boldsymbol{X}_j^*\boldsymbol{b}_j$ is the partial residual and $S(z,\lambda) = \text{sign}(z)(|z| - \lambda)_+$ soft-thresholding operator (Friedman et al., 2010*b*). Since the penalty is separable between groups a blockwise descent algorithm can be used. The algorithm updates $\hat{\boldsymbol{b}}_k$ using Equation (5.24) for each group until convergence.

Step 1: Start with initial matrix $\boldsymbol{B} = \boldsymbol{B}^0$.

Step 2: Update each group (row) separately using Equation (5.24).

Step 3: Start with new $\boldsymbol{B}^1$ and repeat Step 2.

Step 4: Repeat step 3 for new $\boldsymbol{B}^t$, $t = 2, 3, ...$ until convergence (for instance, a negligible change in $\boldsymbol{B}$)

5.6 Numerical study

In this section, simulation studies were carried out to compare the performance of our proposed methods (i.e, ME-SPLS and 2S-SPLS) with existing methods (i.e, E-SPLS and SPLS) in terms of predictive ability, accuracy in variable selection, bias and variance of the estimators. Also, we compared the sparse methods against non-sparse methods to investigate whether there is any change in predictive performance. The sparse methods include 2S-SPLS, ME-SPLS, E-SPLS, SPLS in Equations (5.19), (5.11), (5.6) and (5.4), respectively, and the non-sparse methods are the envelope method and the SIMPLS algorithm discussed in Chapter 4. The study was performed for group sparsity and within-group sparsity.

5.6.1 Simulation

The data was generated using the following settings; we simulated 50 datasets made up of n observations from the model

$$\boldsymbol{y} = \boldsymbol{B}^T\boldsymbol{x} + \sigma\boldsymbol{e}, \quad \text{where} \quad \boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_{\mathcal{A}} \\ \mathbf{0} \end{pmatrix},$$

with $r = 3$ response variables, and $p_{\mathcal{A}} = 4$ active predictor variables. The covariance matrix of $\boldsymbol{x}$, $\boldsymbol{\Sigma}_{\boldsymbol{xx}} = \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1}\boldsymbol{\Gamma}_1^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0}\boldsymbol{\Gamma}_0^T$, where $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_1} = \rho_1\boldsymbol{I}_q$ is the covariance of the relevant part of $\boldsymbol{x}$. The $\boldsymbol{\Omega}_{\boldsymbol{\Gamma}_0} = (\rho_2\boldsymbol{I}_{p_{\mathcal{A}}-q}, \rho_3\boldsymbol{I}_{p_{\mathcal{I}}})$ with $\rho_2\boldsymbol{I}_{p_{\mathcal{A}}-q}$ the covariance of the irrelevant part, and $\rho_3\boldsymbol{I}_{p_{\mathcal{I}}}$ covariance of the inactive set, $\boldsymbol{x}_{\mathcal{I}}$. The components of $\boldsymbol{e} \in \mathbb{R}^r$ are normally distributed with $\delta^{|k-l|}$ the correlation between e_k and e_l. The columns of $\boldsymbol{B}$ are coefficients for each response variable, where $\mathbf{0}$ is a matrix of zero indicating that the corresponding predictor variables are inactive. The $\rho = (\rho_1, \rho_2, \rho_3)$, δ, n, p, q, σ, and $\boldsymbol{B}_{\mathcal{A}}$ were varied to generate different scenarios of the data as follows;

$$\boldsymbol{B}_{\mathcal{A}^1} = \begin{pmatrix} -2 & 4 & 3 \\ 1 & 2.5 & 4 \\ 2 & 1.5 & 5 \\ 4 & -2 & -1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{B}_{\mathcal{A}^2} = \begin{pmatrix} -3 & 5 & 1 \\ 3 & 0 & 0 \\ 2 & 1 & -2 \\ 4 & 0 & 3 \end{pmatrix}.$$

where $\boldsymbol{B}_{\mathcal{A}^1}$ and $\boldsymbol{B}_{\mathcal{A}^2}$ are for group and within-group sparsity, respectively. The zeros in rows of $\boldsymbol{B}_{\mathcal{A}^2}$ indicate that response variables are associated with different set of predictor variables. Table 5.1 shows the different scenarios considered.

n	p	q	ρ	δ	σ
30, 50, 100	12	2	(100,0.8)	0.1	1
500, 1000	100	3		0.9	6
2000, 5000	150				
10000	200				

Table 5.1: *Overview of simulated data. Small and large values of ρ (and δ) correspond to low and high multicollinearity, respectively.*

Parameters for E-SPLS were computed using R codes obtained via personal correspondence with the authors Zhu et al. (2020). The parameters of SPLS, SIMPLS and envelope methods were computed using the R package `spls`, `plsr`, and `Renvlp`, respectively. The parameters of ME-SPLS (Equation (5.11)) and 2S-SPLS (Equation (5.19)) were estimated using R codes which are available in the Appendix A.3.

The number of components and tuning parameters for E-SPLS, SPLS, and ME-SPLS were chosen using 5-fold crossvalidation (also used by the authors, **?** and Chun and Keleş (2010)). For 2S-SPLS, the relevant basis was computed using the envelope method and the tuning parameters were selected using Bayesian Information Criterion (BIC), because BIC performs better than CV if the true model has a finite dimension and is included the potential models (Shao, 1997). The predictive performance was evaluated via mean-squared error, MSE $= \mathrm{E}[(\boldsymbol{y} - \hat{\boldsymbol{y}})^2]$.

Selection performance was evaluated using true positive rate (TPR), true negative rate (TNR), and accuracy (ACC), which are given by

$$\mathrm{TPR} = \frac{1}{50}\sum_{i=1}^{50}\frac{\text{number of estimated true nonzeros}}{\text{total number of true nonzeros}} \tag{5.25}$$

$$\mathrm{TNR} = \frac{1}{50}\sum_{i=1}^{50}\frac{\text{number of estimated true zeros}}{\text{total number of true zeros}} \tag{5.26}$$

$$\mathrm{ACC} = \frac{1}{50}\sum_{i=1}^{50}\frac{\text{number of estimated true nonzeros} + \text{number of estimated true zeros}}{\text{total number of true nonzeros} + \text{total number of true zeros}} \tag{5.27}$$

A meaningful zero is approximated by 10^{-5}. The bias of $\hat{\boldsymbol{B}}_{\mathcal{A}}$ for the 50 replicates was computed through $\mathrm{bias}(\hat{\boldsymbol{B}}_{\mathcal{A}}) = \|\mathrm{Ave}(\hat{\boldsymbol{B}}_{\mathcal{A}}) - \boldsymbol{B}_{\mathcal{A}}\|_F$, where $\mathrm{Ave}(\hat{\boldsymbol{B}}_{\mathcal{A}})$ is the average of the 50 replicates and $\|\cdot\|_F$ is the Frobenius norm. The standard deviation (std) was calculated using $\mathrm{std}(\hat{\boldsymbol{B}}_{\mathcal{A}}) = \sqrt{\mathrm{tr}(\mathrm{cov}(\hat{\boldsymbol{B}}_{\mathcal{A}}))}$.

5.6.2 Comparing different methods in terms of predictive performance

In this section, we present results showing the predictive performance of the different methods discussed in previous sections.

In the first study, we used $n = 50$ observations from a multivariate regression with $r = 3$ response variables, $p = 12$ predictor variables, and $q = 3$ relevant components. The covariance matrix of $\boldsymbol{x}$ has $\rho = (4, 0.8, 1)$, $\boldsymbol{\Gamma}_{\mathcal{A}}$ was constructed by orthogonalizing a matrix of uniform (0,1) random variables, and $\boldsymbol{B}_{\mathcal{A}} = \boldsymbol{B}_{\mathcal{A}^1}$ (for group sparsity). The $\delta = 0.95$ is the largest correlation in $\boldsymbol{y}$, and $\sigma = 1$ is the size of the noise. In this scenario, the variation in the relevant part is larger than the variation in the irrelevant part, and correlation in the response is high. Five-fold cross-validation was used to calculate the MSEs of the datasets. The results are shown in Figure 5.1. We can see that the likelihood-based methods (E-SPLS, EPLS and 2S-SPLS) have similar performances across the number of components. Moreover, the likelihood-based methods outperforms SPLS and SIMPLS with cross-validated (CV) MSEs. The CV-MSEs are 0.5813, 0.6363, and 0.5787 for E-SPLS, EPLS and 2S-SPLS, respectively, and 1.2607, 1.7832 for SPLS and SIMPLS, respectively, when $q = 3$. In addition, all the prediction methods are close when $q \geq 7$.

When the sample size was increase to $n = 100$ the relative predictive performance of the methods did not change; the results are shown in Figure 5.2.

The setting used in the third example is similar to the first but $n = 100$ and $\sigma = 6$. In this setting, the size of the noise is large. The results are given in Figure 5.3 and shows that the predictive performance of 2S-SPLS, EPLS and SPLS are similar, and require only $q = 2$ components. The SIMPLS and E-SPLS did not perform well compared to 2S-SPLS, EPLS and SPLS when $q = 2$. For $q = 2$ the CV-MSEs of the methods are 4.1564 (2S-SPLS), 4.0525 (EPLS), 4.2433 (SPLS), 4.5093 (SIMPLS), and 4.84 (E-SPLS). The methods have similar performances after $q = 3$ number of components. This suggests that the 2S-SPLS and SPLS methods have better prediction performance compare to E-SPLS when the noise is large. Also, the

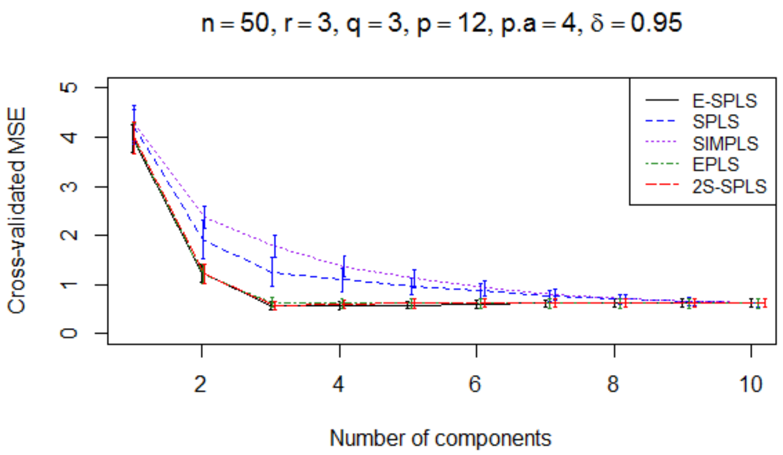


Figure 5.1: *CV-MSEs over 50 replications when variation in the relevant part is large compared to the irrelevant part. The width of the vertical lines on the curves are ± the standard deviations and shows the amount of variability of the CV-MSEs over the 50 replicates. The noise is, sigma = 1, sample size is moderate, the response variables are highly correlated, and predictor variables are nearly multicollinear.*

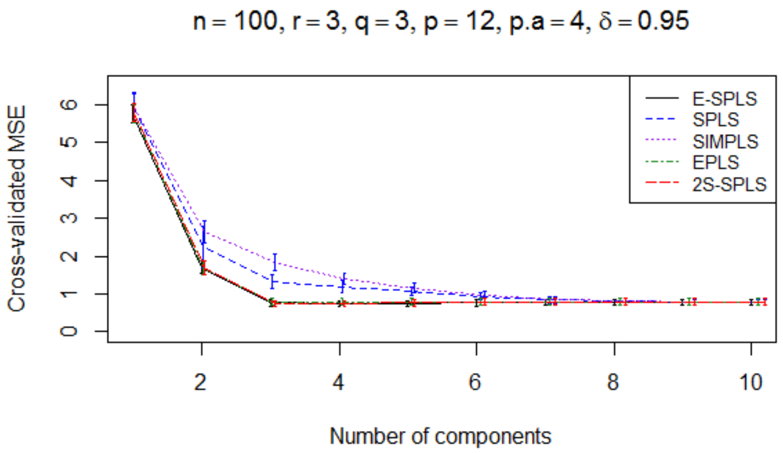


Figure 5.2: *CV-MSEs over 50 replications when variation in the relevant part is large compared to the irrelevant part. The width of the vertical lines on the curves are ± the standard deviations and shows the amount of variability of the CV-MSEs over the 50 replicates. The noise is, sigma = 1, sample size is large, the response variables are highly correlated, and predictor variables are nearly multicollinear.*

2S-SPLS, EPLS and SPLS methods require only $q = 2$ components while E-SPLS require $q = 4$ components.

The results shown in Figure 5.4 has the following settings; $n = 50$, $\rho = (4, 0.8, 1)$, $\delta = 0.1$, and $\sigma = 6$. Here, there is little correlation among the response variables, and the size of the noise is large. The results are not very different from those of

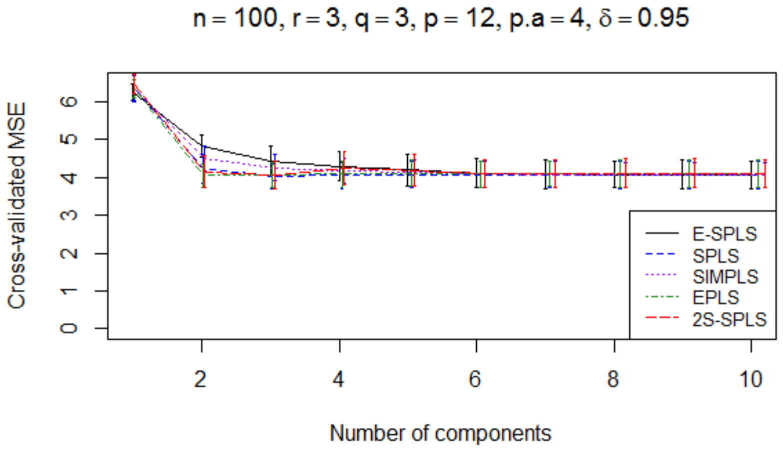


Figure 5.3: *CV-MSEs over 50 replications when variation in the relevant part is large compared to the irrelevant part. The width of the vertical lines on the curves are ± the standard deviations and shows the amount of variability of the CV-MSEs over the 50 replicates. The noise is, sigma = 6, sample size is large, the response variables are highly correlated, and predictor variables are nearly multicollinear.*

Figure 5.3 re-enforcing that when the noise is large E-SPLS may not be the best method to use for predicting new responses.

In the next simulation scheme, we generated data using $n = 50$, $\rho = (40, 0.8, 1)$, $\delta = 0.95$, and $\sigma = 6$. In this scenario, variation in the relevant part is much larger than the variation in the irrelevant part, correlation in the response variable is high, and the noise is large. The results are in Figure 5.5 and shows that all the methods considered have similar and better prediction performance compared to SIMPLS. Comparing the results in Figures 5.4 and 5.5, we see that the performance of E-SPLS improved when the variation in the relevant part increased irrespective of the size of the noise.

The previous simulation settings have more variation in the relevant part compared to the irrelevant part. In the simulation studies that follow, we consider cases when variation in irrelevant part is larger than the variation in the relevant part. In the first scheme, $n = 50$, $p = 12$, $r = 3$, $q = 3$, $\rho = (0.8, 30, 100)$, $\delta = 0.1$, and $\sigma = 6$. Here, there is not correlation in the response variables and the noise is large. The results are shown in Figure 5.6 and suggests that the likelihood-based methods 2S-SPLS (5.19), E-SPLS (5.6) and EPLS (5.5) performs better than SPLS (5.4) and SIMPLS when variation in the relevant part is much smaller than the variation in

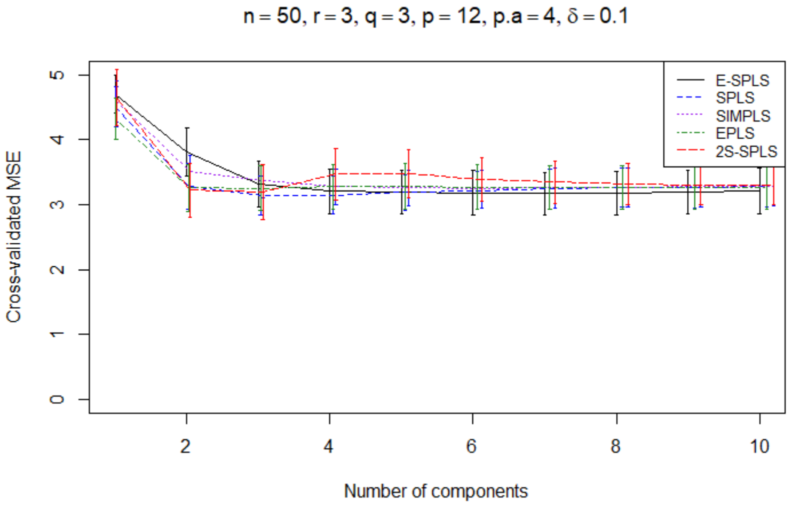


Figure 5.4: *CV-MSEs over 50 replications when variation in the relevant part is large compared to the irrelevant part. The width of the vertical lines on the curves are* ± *the standard deviations and shows the amount of variability of the CV-MSEs over the 50 replicates. The noise is, sigma* = 6*, sample size is moderate, the response variables are not correlated, and predictor variables are nearly multicollinear.*

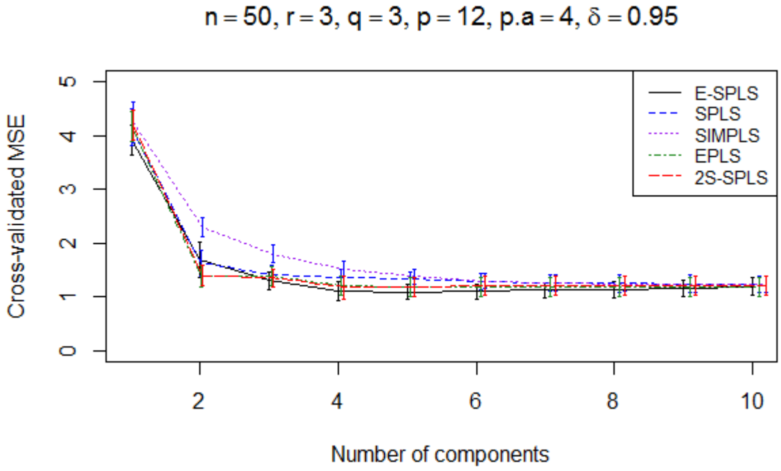


Figure 5.5: *CV-MSEs over 50 replications when variation in the relevant part is large compared to the irrelevant part. The width of the vertical lines on the curves are* ± *the standard deviations and shows the amount of variability of the CV-MSEs over the 50 replicates. The noise is, sigma* = 6*, sample size is moderate, the response variables are highly correlated, and predictor variables are nearly multicollinear.*

the irrelevant part. For $q = 3$ the CV-MSEs are 2.5993 (2S-SPLS), 2.6847 (EPLS), 2.5719 (E-SPLS), 3.025 (SPLS), and 3.0797 (SIMPLS). An increase in the sample size did not change the comparative predictive performance of the methods; the results are shown in Figure 5.7.

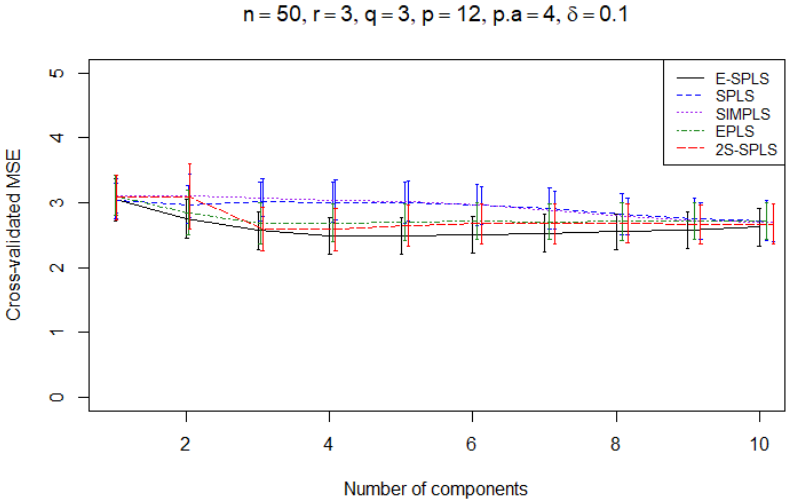


Figure 5.6: *CV-MSEs over 50 replications when variation in the relevant part is large compared to the irrelevant part. The width of the vertical lines on the curves are* ± *the standard deviations and shows the amount of variability of the CV-MSEs over the 50 replicates. The noise is, sigma* = 6*, sample size is moderate, the response variables are not correlated, and predictor variables are nearly multicollinear.*

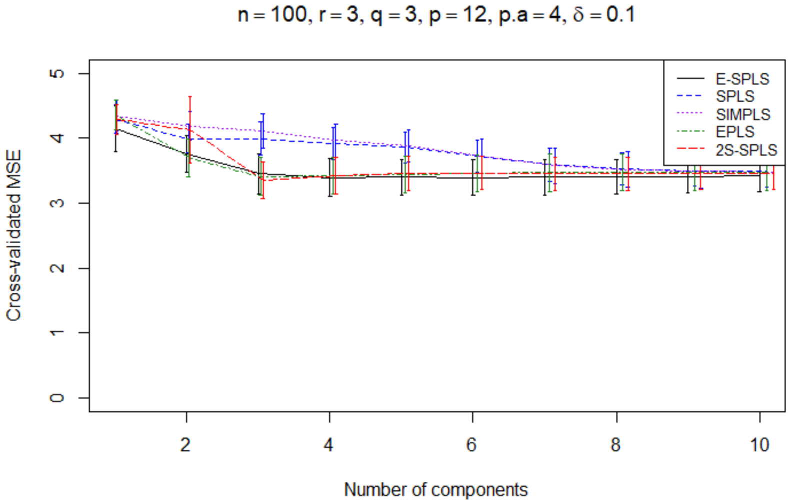


Figure 5.7: *CV-MSEs over 50 replications when variation in the irrelevant part is large compared to the relevant part. The width of the vertical lines on the curves are* ± *the standard deviations and shows the amount of variability of the CV-MSEs over the 50 replicates. The noise is, sigma* = 6*, sample size is large, the response variables are not correlated, and predictor variables are nearly multicollinear.*

The final study is identical to the previous setting but the largest correlation in $\boldsymbol{y}$ is $\delta = 0.95$. The results are shown in Figure 5.8 and suggests that 2S-SPLS and EPLS are slightly better than E-SPLS, when $q = 2$ is the best number of

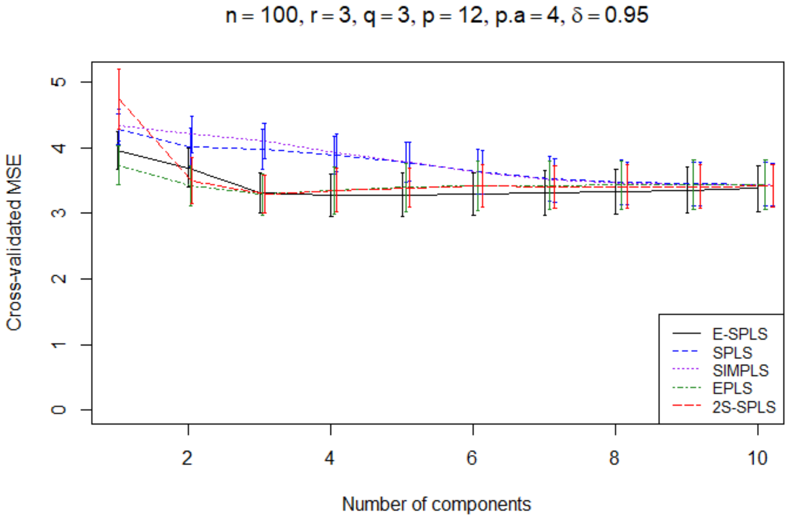


Figure 5.8: *CV-MSEs over 50 replications when variation in the irrelevant part is large compared to the relevant part. The width of the vertical lines on the curves are* ± *the standard deviations and shows the amount of variability of the CV-MSEs over the 50 replicates. The noise is, sigma* = 6*, sample size is moderate, the response variables are highly correlated, and predictor variables are nearly multicollinear.*

relevant components. Again, the likelihood based methods outperform the SPLS and SIMPLS methods. The predictive performance of the methods are similar after $q = 8$ components. The results suggests that high correlation in the response variable improves the performance of 2S-SPLS.

5.6.3 Comparing the sparse methods in terms of selection accuracy

In this section, various simulation studies were conducted to compare the selection accuracy of the regularized methods (2S-SPLS, E-SPLS and SPLS) considered in this chapter.

We begin with simulations studying the selection performance for group sparsity. In the first study, we used $p = 12$, $r = 3$, $q = 2$, $\rho = (4, 0.8, 1)$, $\delta = 0.95$, and $\sigma = 1$. In this scenario, the number of predictor variables is fixed while sample size, n, increase from 30 to 500, the variation in the relevant part is larger than the variation in the irrelevant part, correlation in $\boldsymbol{y}$ is high, and the noise is small. Table 5.2 gives the averages of TPR, TNR and ACC from the 50 datasets, it shows that 2S-SPLS and E-SPLS have similar selection performance and are slightly better than SPLS when n is moderate.

	E-SPLS			SPLS			2S-SPLS		
n	ACC	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR
30	1	1	1	0.9694	1	0.95	0.99	1	0.985
50	1	1	1	0.973	1	0.96	0.9994	1	0.9992
100	1	1	1	1	1	1	0.9994	1	0.9992
500	1	1	1	1	1	1	1	1	1

Table 5.2: *Selection performance for group lasso with $n > p$, $p = 12$, $r = 3$, $p_{\mathcal{A}} = 4$, $q = 2$, $\delta = 0.95$, and $\sigma = 1$. Variation in the relevant part is large compared to the irrelevant part.*

Using the same settings, the standard deviations and bias of each element of $\boldsymbol{B}_{\mathcal{A}}$ was calculated from the 50 estimates. The results are shown in Table 5.3. The decrease in the standard deviations as sample size increases follows the pattern of a $\sqrt{n}$-consistent estimator. Additionally, there is no considerable difference in the standard deviations of the methods. But, the estimated bias for 2S-SPLS and E-SPLS are small compared to the bias of SPLS, which suggests that the estimates of the likelihood based methods (2S-SPLS and E-SPLS) are better.

Standard Deviations								
n	30	50	100	500	1000	2000	5000	10000
E-SPLS	0.2041	0.2006	0.1374	0.0616	0.0471	0.0316	0.0198	0.0147
SPLS	0.1886	0.1859	0.1129	0.0527	0.0365	0.0267	0.0175	0.0122
2S-SPLS	0.2432	0.2019	0.1367	0.0616	0.0470	0.0314	0.0195	0.0143
Bias								
E-SPLS	73.9619	73.8471	73.5728	74.0493	73.8583	73.8549	73.8796	73.856
SPLS	84.4117	84.5654	83.8969	83.6924	83.8127	83.7426	83.7316	83.7404
2S-SPLS	74.1755	73.8621	73.5855	74.0597	73.8691	73.8669	73.8885	73.8627

Table 5.3: *Standard deviation and bias of estimates for group lasso with $n > p$, $p = 12$, $r = 3$, $p_{\mathcal{A}} = 4$, $q = 2$, $\delta = 0.95$, and $\sigma = 1$. Variation in the relevant part is large compared to the irrelevant part.*

In the next simulation study, we have $q = 3$, $\sigma = 6$, and $\delta = 0.1$. In this setting, the noise is larger and correlation in $\boldsymbol{y}$ is lower compared to the previous setting. The results are presented in Table 5.4, and we can infer that for small sample size E-SPLS performs better than SPLS and 2S-SPLS. Also, E-SPLS and 2S-SPLS perform better compared to SPLS for a relatively large sample, $n = 100$. The results suggest that when the noise is large the likelihood-based methods perform better than SPLS.

The results of the standard deviations and bias for this setting were computed and shown in Table 5.5. For small sample size E-SPLS and SPLS have smaller

	E-SPLS			SPLS			2S-SPLS		
n	ACC	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR
30	0.9683	0.985	0.96	0.9317	0.92	0.9375	0.9333	0.9133	0.9433
50	0.985	0.98	0.9875	0.9483	0.93	0.9575	0.9306	0.935	0.9483
100	1	1	1	0.9567	0.95	0.96	0.975	1	0.9625
500	1	1	1	0.9617	0.96	0.9625	0.9956	1	0.9933

Table 5.4: *Selection performance for group lasso with $n > p$, $p = 12$, $r = 3$, $p_{\mathcal{A}} = 4$, $q = 3$, $\delta = 0.1$, and $\sigma = 6$. Variation in the relevant part is large compared to the irrelevant part.*

standard deviations compared to 2S-SPLS. However, for larger samples the methods are competitive. The biases of SPLS are slightly larger than those of E-SPLS and 2S-SPLS. These results suggests that when the noise is large and sample size is small 2S-SPLS is not the preferred method because the standard deviation is large for this setting.

The next simulation study considers the setting when the variation in the irrelevant part is large compared to the variation in the relevant part. Also, correlation in $\boldsymbol{y}$ in high and the noise is large. The following setting was used; $p = 12$, $r = 3$, $q = 2$, $\rho = (0.8, 30, 100)$, $\delta = 0.95$, and $\sigma = 6$. The results of the selection accuracy, standard deviation and bias are shown in Tables 5.6 and 5.7, respectively. The results did not show appreciable difference among the methods, except for $n = 30$ where E-SPLS outperforms SPLS and 2S-SPLS with regard to the TPR and standard deviation.

Standard Deviations								
n	30	50	100	500	1000	2000	5000	10000
E-SPLS	0.4965	0.3659	0.2419	0.0946	0.0684	0.0474	0.0304	0.0220
SPLS	0.4589	0.4066	0.3119	0.1142	0.0821	0.0466	0.0330	0.0234
2S-SPLS	0.9923	0.6728	0.2719	0.0949	0.0674	0.0457	0.0305	0.0221
Bias								
E-SPLS	77.835	77.3667	77.2008	77.3426	77.2979	77.2248	78.8678	77.2756
SPLS	81.3762	80.8539	79.9853	79.0464	78.8299	78.7251	78.8678	78.8828
2S-SPLS	76.3430	76.3439	76.7103	77.2741	77.2644	77.2135	77.2917	77.2785

Table 5.5: *Standard deviation and bias of estimates for group lasso with $n > p$, $p = 12$, $r = 3$, $p_{\mathcal{A}} = 4$, $q = 3$, $\delta = 0.1$, and $\sigma = 6$. Variation in the relevant part is large compared to the irrelevant part.*

In the next simulation study performance of the methods were compared when there is within group sparse in the regression coefficients. The variation in the relevant part is large compared to variation in the irrelevant part. Also, correlation in $\boldsymbol{y}$ in high and the noise is small. The following setting was used; $p = 12$, $r = 3$, $q = 3$, $\rho = (4, 0.8, 1)$, $\delta = 0.95$, and $\sigma = 1$. The results of the selection accuracy, standard deviation and bias are in Tables 5.8 and 5.8, respectively. The results show that in this case the 2S-SPLS outperformed other methods in terms of selection accuracy, highlighting the flexibility of the 2S-SPLS method.

	E-SPLS			SPLS			2S-SPLS		
n	ACC	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR
30	0.9767	1	0.965	0.9583	0.925	0.975	0.9339	0.9267	0.9375
50	0.9983	1	0.9975	0.9817	0.975	0.985	0.975	0.964	0.98
100	1	1	1	0.995	0.985	1	0.98	0.99	0.975
500	1	1	1	1	1	1	1	1	1

Table 5.6: *Selection performance for group lasso with $n > p$, $p = 12$, $r = 3$, $p_{\mathcal{A}} = 4$, $q = 2$, $\delta = 0.95$, and $\sigma = 6$. Variation in the irrelevant part is large compared to the relevant part.*

Standard Deviations								
n	30	50	100	500	1000	2000	5000	10000
E-SPLS	0.5560	0.4619	0.2846	0.1158	0.0892	0.0580	0.0395	0.0256
SPLS	0.7288	0.6223	0.1778	0.1687	0.1293	0.0970	0.0619	0.0409
2S-SPLS	0.8387	0.5491	0.3208	0.1409	0.0846	0.0579	0.0403	0.0256
Bias								
E-SPLS	76.0094	75.6702	75.8408	75.7923	75.7450	75.7456	75.7510	75.7409
SPLS	79.2349	77.6916	76.5189	76.1758	75.8004	75.7024	75.8784	75.8281
2S-SPLS	74.3464	74.4964	75.2090	75.4287	75.3788	75.3981	75.4317	75.4246

Table 5.7: *Standard deviation and bias of estimates for group lasso with $n > p$, $p = 12$, $r = 3$, $p_{\mathcal{A}} = 4$, $q = 2$, $\delta = 0.95$, and $\sigma = 6$. Variation in the irrelevant part is large compared to the relevant part.*

	E-SPLS			SPLS			2S-SPLS		
n	ACC	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR
30	0.945	0.889	0.9637	0.8472	0.973	0.805	0.929	0.957	0.919
50	0.945	0.889	0.9637	0.9089	0.973	0.881	0.961	0.956	0.962
100	0.945	0.889	0.9637	0.9089	0.973	0.882	0.968	0.993	0.961
500	0.945	0.889	0.9637	0.9167	1	0.889	0.982	1	0.976

Table 5.8: *Selection performance for within-group sparsity with $n > p$, $p = 12$, $r = 3$, $p_{\mathcal{A}} = 4$, $q = 3$, $\delta = 0.95$, and $\sigma = 1$. Variation in the relevant part is large compared to the irrelevant part.*

n	30	50	100	500	1000	2000	5000	10000
E-SPLS	0.2705	0.2084	0.1481	0.0671	0.0506	0.0316	0.0208	0.0142
SPLS	0.4439	0.243	0.2348	0.0776	0.0548	0.0361	0.0237	0.0167
2S-SPLS	0.4637	0.422	0.2576	0.1334	0.1232	0.0934	0.0911	0.0813
Bias								
E-SPLS	53.1479	53.3416	53.2189	53.3088	53.259	53.302	53.299	53.290
SPLS	56.2254	55.5882	55.2754	55.210	55.155	55.150	55.133	55.126
2S-SPLS	51.3129	51.7865	51.2136	51.3887	51.3983	51.371	51.325	51.301

Table 5.9: *Standard deviation and bias of estimates for within-group sparsity with $n > p$, $p = 12$, $r = 3$, $p_{\mathcal{A}} = 4$, $q = 3$, $\delta = 0.95$, and $\sigma = 1$. Variation in the relevant part is large compared to the irrelevant part.*

5.7 Conclusion

This chapter introduced a sparse RC model which combines the assumption of variable selection and component extraction when finding information in predictor variable that is relevant for predicting the response variables. The model explains that only a linear combination of the active variables are required in a regression model for the response. Different methods for estimating the parameters of the model were reviewed including E-SPLS (Zhu et al., 2020), SPLS (Chun and Keleş, 2010) e.t.c. Furthermore, we proposed a novel method (2S-SPLS) for estimating the parameters of the sparse RC model which is more flexible compared to other methods in the literature. That is, the 2S-SPLS method can perform group and within-group sparsity while other methods in the literature are for group sparsity alone.

The methods for estimating the parameters of the model were compared with regard to prediction performance and variable selection accuracy using simulated data. The results show that the methods have similar prediction performance when the variation in the relevant part is larger than the variation in the irrelevant part - when sample size is moderate or large compared to the number of predictor variables. On the other hand, the MLE methods performed better than the algorithms when variation in the irrelevant is large compared to the variation in the relevant part. In addition, the degree of correlation in the response variables and the size of the noise affects the prediction performance of the methods: increase in correlation and noise reduces the predictive performance of the methods, which is similar to the results of Rimal et al. (2019).

Also, the E-SPLS method performed better the SPLS and 2S-SPLS methods in terms of variable selection accuracy when the sample size is small compared to the number of predictor variables, but when sample is large the methods are comparable. However, when within-group sparsity is of interest the 2S-SPLS method performed better than E-SPLS and SPLS methods. The decreasing trend of the standard deviation of the regression coefficients shows that the methods are root-n consistent. These results show that when sample size is large 2S-SPLS is robust to in terms of variable selection compared to other methods.

Chapter 6

Modelling Scleroderma Disease

6.1 Introduction

Systemic sclerosis (SSc; also called scleroderma) is an autoimmune disease, which affects the skin and different organs in the body such as the lungs and blood vessels. The disease manifests in the form of hardening of the skin and deterioration in lung function. The disease often starts with the thickening of the fingers or face, before progressing to elbows and knees subsequently leading to the thickening and scarring of lung tissues. SSc is defined as either limited cutaneous or diffused cutaneous depending on its severity. The disease is defined as limited cutaneous SSc (lcSSc) if skin thickening affects fingers or face, and as diffuse cutaneous SSc (dcSSc) if skin thickening extends to elbows and knees (Nihtyanova et al., 2014; Pokeerbux et al., 2019). SSc is a rare disease, affecting from 7 people per million to 489 people per million most of which are women (Bernatsky et al., 2009). It is estimated that the number of women with SSc is 4 times the number of men. However, women are more likely to survive from the disease compared to men (Barnes and Mayes, 2012). Figure 6.1 shows an example of the hands of a patient with early-stage SSc.

The extent of skin thickening and deterioration of lung function are outcomes used to measure the severity of the disease. These outcomes are total modified Rodnan skin score (MRSS) for measuring skin thickness, and diffuse capacity for carbon monoxide (DLCO) and forced vital capacity (FVC) for measuring lung function deterioration. The different degrees of MRSS are

- 0 = normal skin

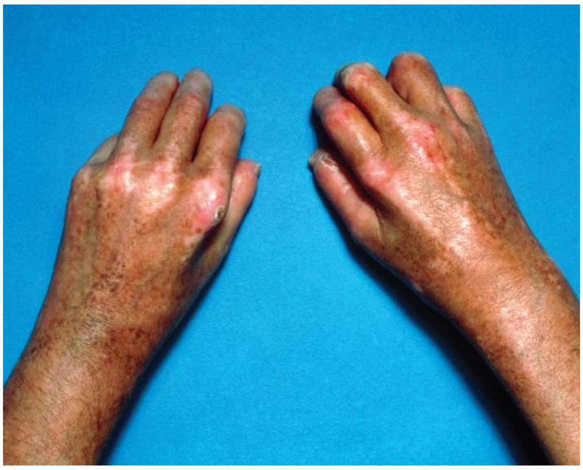


Figure 6.1: *Hardening skin of the hands of a patient with early-stage systemic sclerosis (Michael, 2018).*

- 1 = possible skin thickening
- 2 = definite skin thickening but mobile
- 3 = skin more thickened and fixed to deeper tissue.

Different parts of the body (face to foot) are scored separately and summed up to get the MRSS score for each patient. FVC and DLCO are measured by checking the percentage drop in lung function. Patients are checked regularly in the hospital to monitor progression; a patient's disease progression was classified as significant if there is a 15% drop in FVC or DLCO.

The FVC and DLCO are expensive to measure and it may be preferable to use biomarkers that are cheaper to measure to monitor patients progression. A biomarker is a known measure for assessing the severity or presence of a disease (Califf, 2018). The proteomics markers (or potential biomarkers) considered in this chapter are measurements of different proteins from the blood of patients. Detailed understanding of the markers are beyond the scope of the project.

Because SSc is a rare disease, the number of patients is limited, but the number of proteomics markers is large (451) so the dataset is nearly multicollinear. The goal of this chapter is to predict the level of severity of outcomes using important information (few relevant components) and find markers that are related to the outcomes (MRSS, DLCO, and FVC) together, which will reduce hospital visits and reduce the cost of measuring the outcome. Moreover, we assume that the data are drawn from the RC$_{\boldsymbol{x}}$ model discussed in chapter 4 and use the dimension reduction methods discussed in Chapters 3, 4, and 5 for parameter estimation, perform prediction, and to identify candidate markers. The methods considered include univariate (single-outcome) partial least squares (PLS1), multivariate (multiple-outcome) partial least squares (PLS2), and sparse multivariate partial least squares (SPLS2). The estimated regression coefficients obtained from these methods have smaller mean square error compared to the estimates of the ordinary least squares (OLS) method when the variables are nearly multicollinear (Cook et al., 2007; De Jong, 1995; Helland, 1988).

The PLS1 methods in Chapter 3 will be used to compute components of the proteomics markers for predicting SSc outcomes separately. The PLS1 methods considered are the statistically inspired modification of PLS (SIMPLS) and the envelope-based PLS (EPLS). Subsequently the PLS2 methods in Chapter 4 will be used to extract components from the markers for predicting the outcomes together. The PLS2 methods applied are SIMPLS, EPLS and an expectation-maximization algorithm for PLS2 (EM-PLS).

Finally, the SPLS2 methods in Chapter 5 will be used for selecting markers that are related to the outcomes together and predict outcomes jointly. The SPLS2 methods used are the sparse SIMPLS (SPLS) (Chun and Keleş, 2010), envelope-based sparse PLS (E-SPLS) (Zhu et al., 2020), and the novel two-stage SPLS (2S-SPLS) methods.

The SIMPLS and SPLS are algorithms and computes the PLS components sequentially. Whereas, the EPLS, E-SPLS and 2S-SPLS are likelihood-based methods and computes the maximum likelihood estimates (MLE) of the components together. Further, 2S-SPLS computes the components in first stage and regression coefficients in the second stage. The methods and how they are used in the analysis are presented in Table 6.1.

Method	Variants	Application
PLS1	SIMPLS, EPLS	prediction
PLS2	SIMPLS, EPLS, EM-PLS	prediction
SPLS	SPLS, E-SPLS, 2S-SPLS	prediction, variable selection
OLS		prediction

Table 6.1: *The list of methods, their variants, how they will be used.*

6.2 Data description

The data consists of patients clinically diagnosed with SSc between 1995 and 2015 in Leeds. The ages of the patients are between 26 and 82, among which are 101 women and 19 men. The number of patients with dcSSc is 39, and the number with lcSSc is 81. Furthermore, the lung outcomes (DLCO and FVC) are continuous random variables, while the skin thickening outcome (MRSS) is a discrete random variable, but are all modelled as continuous random variables. The data required cleaning before applying the methods. The next section presents data cleaning and descriptive statistics.

6.2.1 Data cleaning

Before cleaning, the data consists of $p = 451$ markers, $r = 3$ outcome variables, and $n = 408$ patients (observations). Some of the markers have detection limits (DL) and NAs. A detection limit is the lowest measurable amount of a marker that can be distinguished from the absence of that marker. We ignored a marker when the number of DLs and/or NAs is greater than or equal to 10 (i.e, number of DLs + number of NAs ≥ 10). As a result, the number of markers reduced from $p = 451$ to $p = 196$. For markers with the number of DLs and/or NAs less than 10 we set the values with DL at the detection limits. After cleaning the data set by considering the markers, observations with NAs were ignored resulting in a decrease from $n = 408$ to $n = 264$.

For outlier detection, the histograms of the $p = 196$ markers were plotted and values isolated at the tails of the histograms were identified as outliers. Twelve values were identified as outliers and were ignored.

The data were collected at different time points, but three fixed time points contain the desired proteomics data. Table 6.2 gives a summary of the number of observations for the three fixed time points after data cleaning and shows that the initial time point has the largest number of patients. In the remainder of the chapter, only time point zero was considered because the number of patients in other time points are small; otherwise we would have considered multiple time points.

Time point	Month zero	Month six	Month twelve
n	96	27	32

Table 6.2: *The number of observations at three fixed time points after data cleaning.*

The dimensions of the variables at month (timepoint) zero after data cleaning are shown in Table 6.3.

Patients	Outcomes	Markers
$n = 96$	$r = 3$	$p = 196$

Table 6.3: *Dimensions of the outcomes and markers at time point zero after data cleaning.*

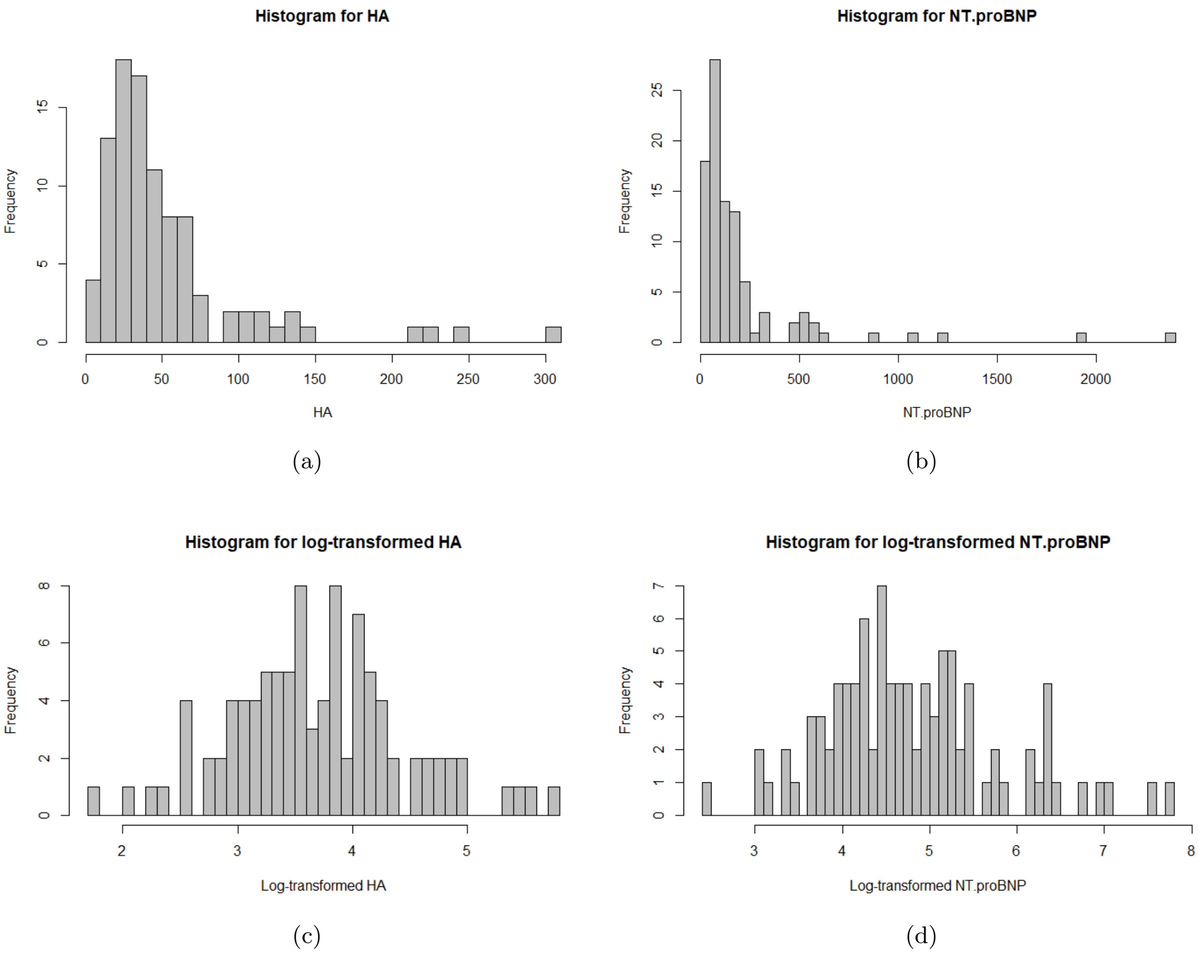
Upon inspecting the histogram of the variables, the outcomes and some markers are not normally distributed and the natural logarithmic transformation ($log_e(x + c)$, $c = 1$) was used to make the data approximately normally distributed; in order to satisfy the RC model assumption. The constant c is included in order to avoid having negative infinity values as some outcomes have zero values. Figure 6.2 show the histograms for two markers before and after the log-transformation was applied. The markers were skewed to the right before the transformation and approximately normally distributed after transformation.

The mean, mode, and median values for FVC (Table 6.4), are slightly different from each other suggesting that its distribution is approximately normal. Also, the standard deviation and the range of FVC is small suggesting that the spread of the outcome is small. The same inference can be made for DCLO. The median and mode of MRSS are equal but smaller than the mean. Furthermore, the range and standard deviation suggest that the spread of the MRSS is large compared to the spread of FVC and DLCO. Moreover, the histograms of the outcome variables are given in Figure 6.3 and shows that FVC and DLCO are approximately normally distributed, and MRSS follows a Poisson distribution with mean equal to 1.238.

	Mean	Median	Mode	Standard deviation	Minimum	Maximum
FVC.	4.592	4.663	4.291	0.252	3.761	4.997
DLCO	4.092	4.135	4.190	0.299	3.045	4.605
MRSS	1.238	1.099	1.099	0.883	0	3.219

Table 6.4: *Descriptive statistics of the three outcome variables*

The Pearson correlation coefficients (PCC) among the outcomes are given in Table

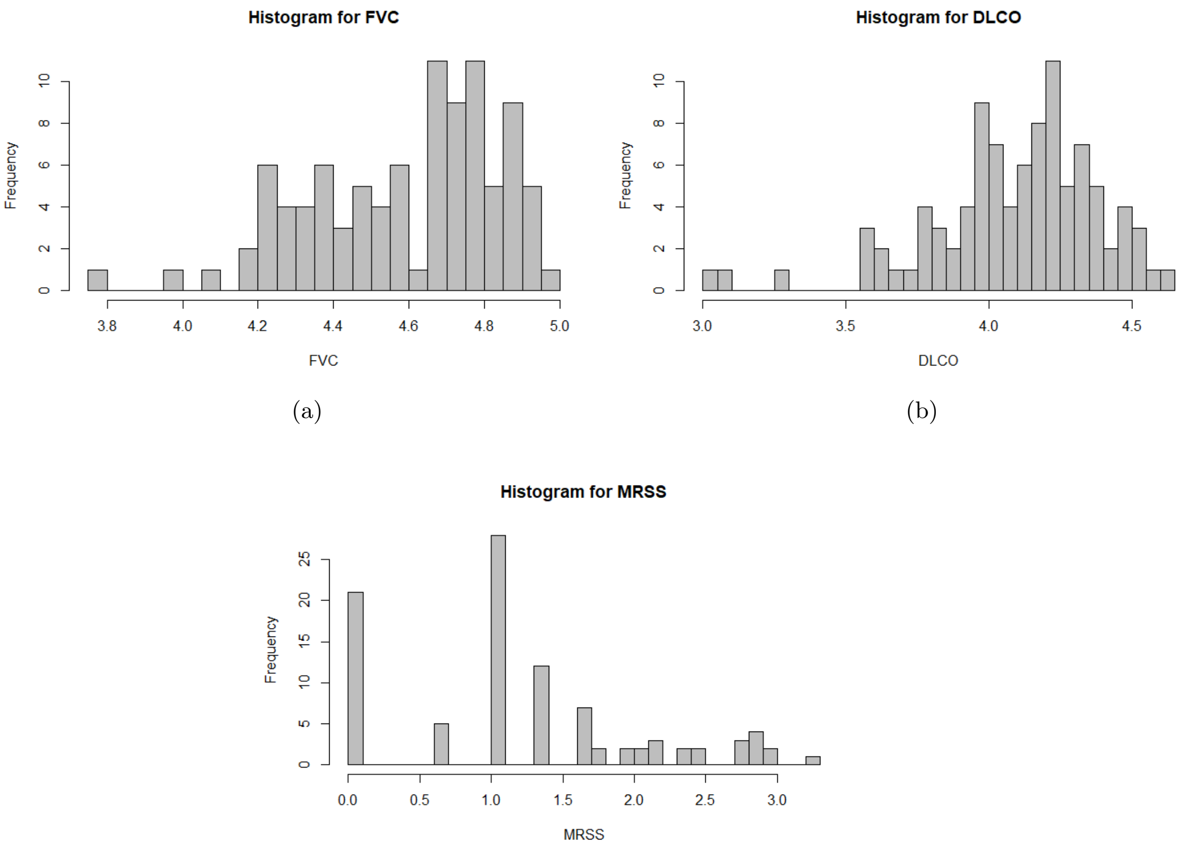


(a) (b)

(c) (d)

Figure 6.2: *The histograms of two markers before and after log-transformation.*

6.5. The PCCs shows that the outcome variables are correlated, and it might be appropriate to model them together. The correlation between the outcomes for lung function is larger than the correlation between the lung functions and MRSS. Also, PCC among the markers are given in Figure 6.4. The blue-clustered and orange-clustered regions of the plot suggest that some markers are correlated. Moreover, the condition number of the covariance matrix of the proteomics markers is 4.97×10^{19} which shows that the markers are nearly multicollinear (Belsley, 1993). The largest correlation between markers is approximately 0.9. Furthermore, several of the correlation between markers are greater than 0.5. This suggests that a dimension reduction method such as PLS might be appropriate for modelling the data.



(a)

(b)

(c)

Figure 6.3: *Histograms for individual outcome variables.*

	FVC	DLCO	MRSS
FVC.	1	0.6642	-0.4261
DLCO	0.6642	1	-0.2645
MRSS	-0.4261	-0.2645	1

Table 6.5: *The PCCs among the three outcomes variables; FVC, DCLO, and MRSS.*

The outcomes and markers were standardized using

$$X_{\text{standardized}} = \frac{X - \bar{x}}{s_{xx}}, \tag{6.1}$$

where $X \in \mathbb{R}^{n\times 1}$ is the unstandardized variable, $X_{\text{standardized}}$ the standardized variable, $\bar{x}$ is the sample mean of X, and s_{xx} is the sample standard deviation of X.

A common practice is to perform preliminary analysis using simple linear regressions (SLR) of each outcome on each proteomics marker and choose markers based on p-values. For instance, we performed SLR for each SSc outcome using each of $p = 196$

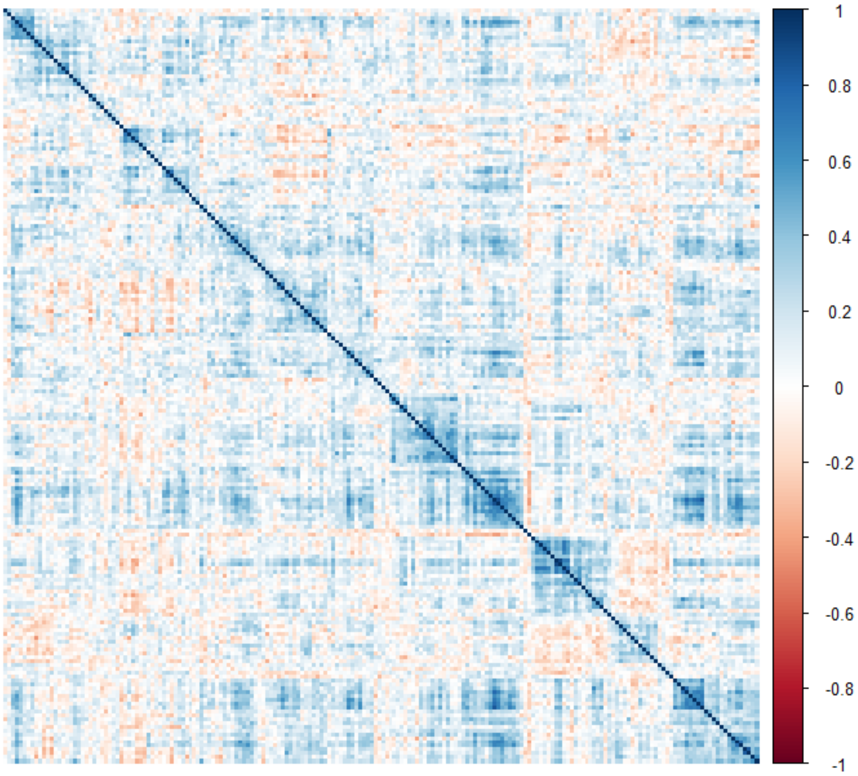


Figure 6.4: *Correlogram for the markers: condition number of is 4.97×10*^{19}.

markers. The number of statistically significant markers (p-value=0.001) for each outcome variable are in Table 6.6 and the MSE for the outcomes each significant marker is approximately 0.8. However, SLR does not use enough information available about the outcomes and does not take into account the correlation in the data and does not avoid multiple testing (Bunea et al., 2006). In particular, excluding some predictor variables may result in loss of information (Helland, 1988).

Outcomes	FVC	MRSS	DLCO
Number of markers	18	6	14
Condition number	75.819	7.294	29.017

Table 6.6: *Number of markers selected for outcome using simple linear regression.*

In the next section, multiple linear regression via the lasso will be used in another preliminary analysis to select initial markers.

6.2.2 Lasso regression

The envelope-based PLS (EPLS) method which will be used for prediction can not be applied when $n < p$. To avoid this problem, the lasso was first used as a preprocessing step for variable selection. This preprocessing step was performed for each SSc outcome, and markers with nonzero coefficients where selected.

Ten-fold cross-validation via the mean square error criterion was used to determine the optimal tuning parameter. The optimal tuning parameter for each fold was extracted and used to identify the markers with nonzero regression coefficients. So there are 10 tuning parameters (one for each fold) and 10 subsets of selected markers. The selected markers were grouped according to the number of times they appear in the 10 subsets. Table 6.7 shows the markers selected for each outcome variable.

The column titled *Mean of* $\hat{\boldsymbol{\beta}}$ corresponds to the mean of each regression coefficient across the subsets. For example, the marker *Leptin.R.* appeared in 10 subsets, and the mean of the corresponding regression coefficient from the models is 0.5202. The markers that appear in 10 subsets are classified as most important markers, followed by the markers that appear in 9 subsets, and so on. For instance, *Leptin.R.* is among the most important markers for MRSS. Also, *Endoglin* and *SP.D* are the most important markers for DLCO.

In the next section, the SSc data set will be analysed using the PLS methods discussed in previous chapters of the thesis. And we used only markers which were selected via the lasso.

MRSS	
Selected by 10 models	Mean of $\hat{\boldsymbol{\beta}}$
Leptin.R.	0.5202
Periostin	0.4274
SCFR.	0.3887
IL.2.receptor.alpha.	0.3412
Apo.E.	0.3385
HGF.receptor.	0.1900
Prostasin	0.0937
Gelsolin	0.0664
AGP.1.	0.0547
Selected by 9 models	
ENA.78.	0.0920
KLK.7.	0.0615
Selected by 8 models	
Thrombospondin.1	0.0469
Selected by 6 models	
AB.40.	1.4199
Tweak.	1.1050
CLU.	0.5967
AAT.	0.5391
VKDPS.	0.3900
LRG1.	0.2829
Resistin	0.2450
LAP	0.2388
CA.15.3.	0.1903
LCN1.	0.1776
Apo.D.	0.1602
ANGPTL4.	0.1555
HB.EGF.	0.1410
GRO.alpha.	0.1340
PON.1.	0.1302
A.disintegrin	0.1133
Fib.1C.	0.1053
GDF.11.	0.1002
SHBG.	0.0921
PIINP	0.0880
CD40.L.	0.0869
Omentin	0.0854
TN.X.	0.0770
EN.RAGE	0.0717
..NT.proBNP.	0.0662
Apo.A.II.	0.0569
CCL15.	0.0344
Selected by 5 models	
TIMP.1	0.4088
MCP.4.	0.1108
ST2	0.1016
MMP.9..total.	0.0658
LH.	0.0484

DLCO	
Selected by 10 models	Mean of $\hat{\boldsymbol{\beta}}$
Endoglin	0.1713
SP.D.	0.0608
Selected by 8 models	
vWF.	0.0752
CEACAM6.	0.0147
Selected by 7 models	
A2Macro.	0.2173
ICAM.1.	0.1268
Tetranectin	0.1116
RAGE.	0.0759
DPPIV.	0.0736
AGP.1.	0.0349
Selected by 6 models	
THP.	1.0651
AAT.	0.2304
1RII.	0.1262
LGL.	0.1026
HGF.receptor.	0.0703
MCP.1.	0.0460
Leptin.R.	0.0207
MMP.1.	0.0177
CRP.	0.0126
Selected by 5 models	
KLK.7.	0.1433
P.Selectin	0.0755
HSP.70.	0.0643
CD40.L.	0.0632
IgM.	0.0512
Apo.H.	0.0462
TSH.	0.0354
Uteroglobin	0.0329
VEGF.	0.0275
MMP.9..total	0.0235

FVC	
Selected by 10 models	Mean of $\hat{\boldsymbol{\beta}}$
Apo.A.I.	0.1696
tPA.	0.1205
SP.D.	0.0664
RAGE.	0.0643
Periostin	0.0359
CRP.	0.0334
Selected by 7 models	
AGP.1.	0.0570
NT.proBNP	0.0332
Gelsolin	0.032
PPP.	0.0151
Selected by 6 models	
OPG.	0.1074
Aggrecan.	0.0580
Fetuin.A	0.0452
VDBP.	0.0379
E.Cad.	0.0366
HE4	0.0221
Omentin	0.0207
MCP.4.	0.0153
Selected by 5 models	
CEACAM6.	0.0063

Table 6.7: *Selected markers with nonzero regression coefficients for MRSS, DLCO, and FVC. The markers are grouped according to the number of times they were selected.*

6.3 Data analysis

In this section, the SSc data set will be analysed using univariate partial least squares (PLS1), multivariate partial least squares (PLS2), and sparse multivariate partial least squares (SPLS2) methods. And the predictive performance of the PLS methods will compared to determine which method is best suited for the data.

6.3.1 Univariate partial least squares (PLS1) regression

In this section, the outcomes were modelled individually using PLS1 and OLS methods. The PLS1 methods used are SIMPLS and EPLS. The number of markers used for each outcome are shown in Table 6.8, and we used markers that appear in least 6 models. Eighteen (18) markers appeared in at least 6 subsets for FVC, 19 for DLCO, and 39 for MRSS, and the condition number for each subset indicates that the markers are multicollinear.

The data were randomly split into a training set and a test set. The training set will be used for selecting the required number components, and the test set will be used for checking the predictive performance of the methods on a new set. The training set has 80 observations, and the test set has 16 observations. We chose 80 to have sufficient data to fit the models.

Outcomes	FVC	DLCO	MRSS
Number of markers	18	19	39
condition number	18.18	13.26	49.54

Table 6.8: *The number of markers selected for each outcome variable.*

Ten-fold cross-validation via mean-square-error was used on the training set to select the number of components. Figure 6.5 show the cross-validated root-mean-square error (CV-RMSE) for FVC across $p = 18$ components. And shows that the EPLS method performs slightly better than SIMPLS and OLS with regard to CV-RMSE. Moreover, the EPLS and SIMPLS methods require 3 and 1 components, respectively. Note that OLS is a straight line because all the markers are used for estimation and prediction. The test set was used to compare the predictive performance of the methods. The fourth column of Table 6.9 gives the root-mean-square error prediction (RMSEP) and suggests that EPLS, SIMPLS, and OLS have similar performances in terms of RMSEP.

Figure 6.6 show the CV-RMSE for DLCO, and suggests that EPLS and SIMPLS

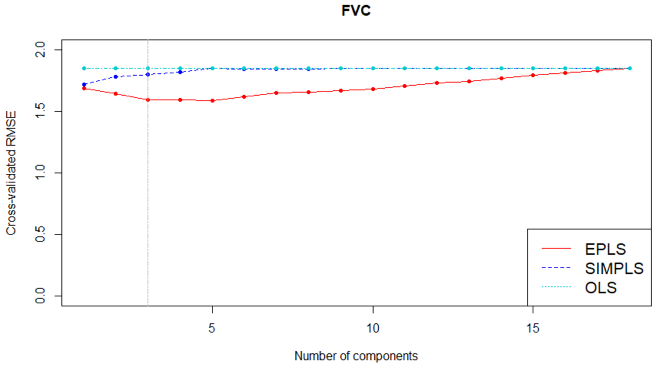


Figure 6.5: *CV-RMSEs for SSc data using the training set. The CV-RMSEs for EPLS, SIMPLS and OLS correspond to the red, blue and green lines, respectively. The points on the lines are the CV-RMSEs for each component. The vertical grey line is the smallest CV-RMSE.*

FVC			
CN = 18.1811			
Method	q^*	CV-RMSE	RMSEP
EPLS	3	1.5913	3.2319
SIMPLS	1	1.7188	3.5784
OLS	18	1.8478	3.2385

Table 6.9: *The optimal number of components for each method, and the corresponding CV-RMSE and RMSEP values.*

require 2 and 1 components, respectively. The figure also shows that EPLS performs slightly better than SIMPLS and OLS with regards to RMSE, and EPLS and SIMPLS perform similarly with regard to RMSE. In addition, the test set was used to compare the predictive performance of the methods. The fourth column of Table 6.10 gives the RMSEP and suggests that the methods have similar performance in terms of predicting new outcomes.

DLCO			
CN = 14.8262			
Method	q^*	CV-RMSE	RMSEP
EPLS	2	1.5718	3.1015
SIMPLS	1	1.7644	3.0035
OLS	19	1.9690	3.2485

Table 6.10: *The optimal number of components for each method, and the corresponding CV-RMSE and RMSEP values.*

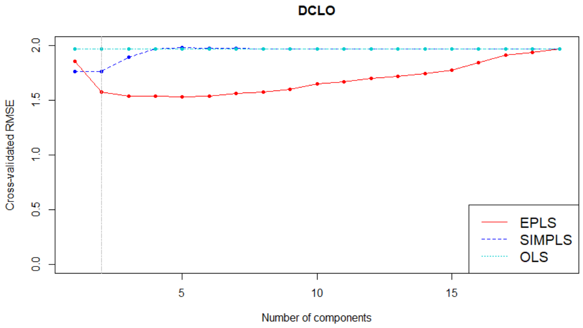


Figure 6.6: *CV-RMSEs for SSc data using the training set. The CV-RMSEs for EPLS, SIMPLS and OLS corresponds to the red, blue and green lines, respectively. The points on the lines are the CV-RMSEs for each component. The vertical grey line is the smallest CV-RMSE.*

Figure 6.7 shows the CV-RMSE for MRSS, and indicates that EPLS requires 4 components compared to SIMPLS which needs 1. Also, EPLS performs better than SIMPLS and OLS in terms of RMSE. In addition, the test set was used to compare the predictive performance of the methods. The fourth column of Table 6.11 gives the RMSEP and shows that EPLS and SIMPLS have similar RMSE performances, and both perform better than the OLS method.

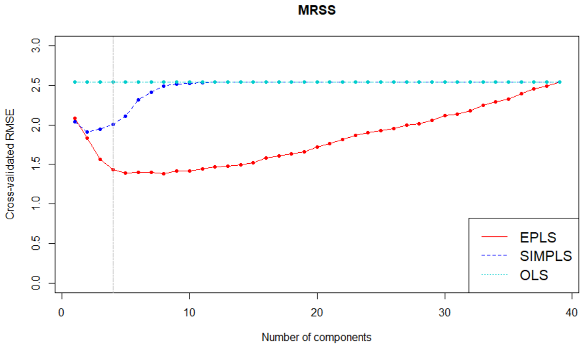


Figure 6.7: *CV-RMSEs for SSc data using the training set. The CV-RMSEs for EPLS, SIMPLS and OLS corresponds to the red, blue and green lines, respectively. The points on the lines are the CV-RMSEs for each component. The vertical grey line is the smallest CV-RMSE.*

6.3.2 Multivariate partial least squares (PLS2) regression

In the previous section, the outcome variables were modelled separately using the selected markers for each outcome variable. In this section, the goal is to model the outcome variables jointly to see how the PLS2 methods perform with regard

MRSS			
CN = 49.5442			
Method	q^*	CV-RMSE	RMSEP
EPLS	4	1.4614	3.3451
SIMPLS	1	2.0701	3.3202
OLS	39	2.5447	4.3308

Table 6.11: *The optimal number of components for each method, and the corresponding CV-RMSE and RMSEP values.*

to prediction and the method that has a better perform compared to others. The markers which appear in six or more models in Table 6.7 were combined to form a new set of $p = 65$ markers.

The methods used for estimation and prediction are SIMPLS, EM-PLS, OLS, and EPLS. A ten-fold cross-validation was used to select the number of components for the PLS methods. Because, the correlation between lung outcomes, DLCO and FVC, is approximately 0.66, and will be modelled jointly. Note that the maximum number of components for EM-PLS is $\min\{r = 2, p = 33\}$; $p = 33$ is the number of combined markers for DLCO and FVC. Figure 6.8 shows the plot of CV-RMSEs for EPLS, SIMPLS, EM-PLS, and OLS, and indicates that EPLS performs better than other methods with regard to RMSE. However, the PLS2 methods are similar in the first component and outperformed the OLS method. The smallest CV-RMSE for EPLS is at the $7th$ component, but its value is similar to that of the $5th$ component. Table 6.12 gives the number of components chosen for each method and the corresponding CV-RMSE values.

Using the chosen number of components for each method, the matrix of regression coefficients were obtained and the RMSEP computed on a test set. The results in the fourth column of Table 6.12 shows that OLS and EPLS have better RMSEP performance compared to SIMPLS and EPLS.

Furthermore, the three outcomes were modelled together, and the prediction performance of the methods were assessed. Note that the maximum number of components for EM-PLS is $\min\{r = 3, p = 65\}$. The results in Figure 6.9 shows that the PLS2 methods outperformed the OLS method in terms of RMSE. And EPLS performs better than SIMPLS and EM-PLS, but require more components. For instance,

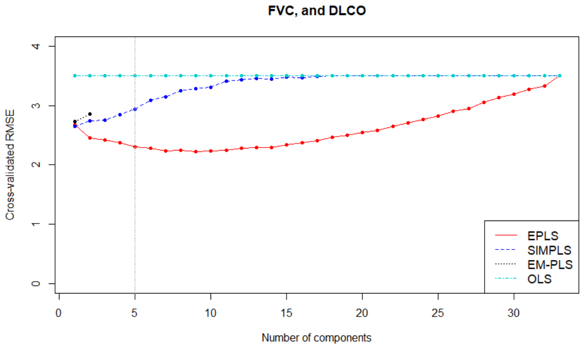


Figure 6.8: *CV-RMSEs for SSc data using the training set. The CV-RMSEs for EPLS, SIMPLS and OLS corresponds to the red, blue and green lines, respectively. The points on the lines are the CV-RMSEs for each component. The vertical grey line is the smallest CV-RMSE.*

FVC and DLCO			
CN = 43.6556			
Method	q^*	CV-RMSE	RMSEP
EPLS	5	2.2994	4.8041
SIMPLS	1	2.6535	5.2292
EM-PLS	1	2.7345	5.2175
OLS	33	3.5048	4.9396

Table 6.12: *The optimal number of components for each method, the corresponding CV-RMSE and RMSEP values.*

EPLS require 6 components compared to SIMPLS and EM-PLS which require only 1 component, each. However, the RMSEP used for assessing the prediction performance on the test set are similar for EPLS and SIMPLS as shown in the fourth column of Table 6.13, and shows that the PLS2 methods outperform OLS.

FVC, MRSS, and DLCO			
CN = 1211.59			
Method	q^*	CV-RMSE	RMSEP
EPLS	6	2.9588	6.5343
SIMPLS	1	3.7266	6.6755
EM-PLS	1	4.0325	7.3329
OLS	65	10.2836	12.4039

Table 6.13: *The optimal number of components for each method, the corresponding CV-RMSE and RMSEP values.*

In addition, we investigate the loadings of PLS2 methods because the loadings from

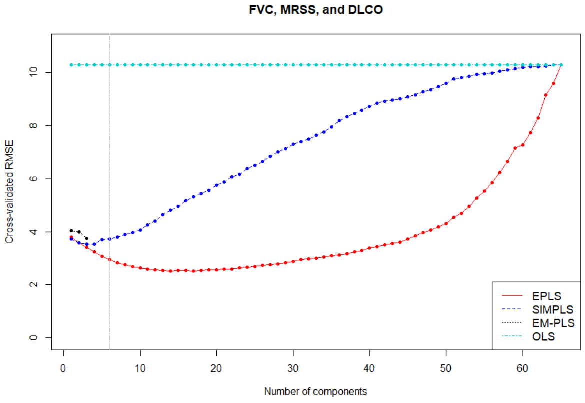
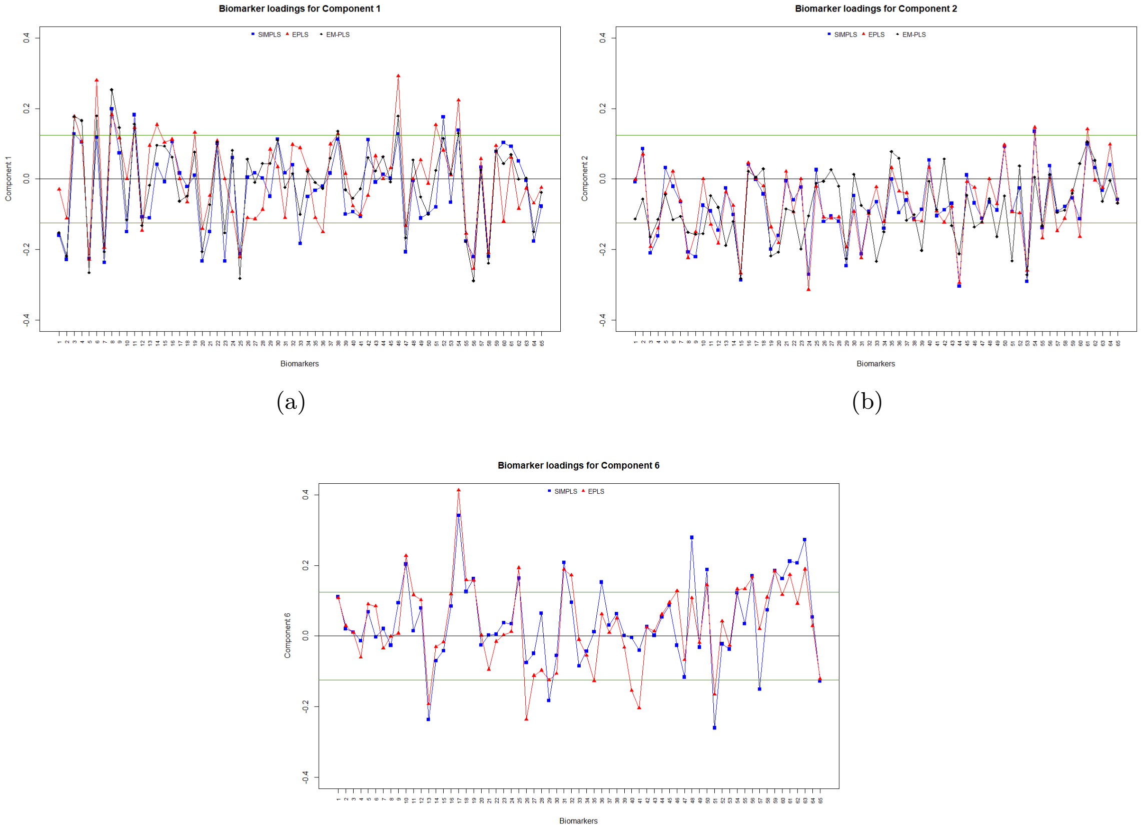


Figure 6.9: *CV-RMSEs for SSc data using the training set. The CV-RMSEs for EPLS, SIMPLS, EM-PLS and OLS corresponds to the red, blue, black and green lines, respectively. The points on the lines are the CV-RMSEs for each component. The vertical grey line is the smallest CV-RMSE.*



(a)

(b)

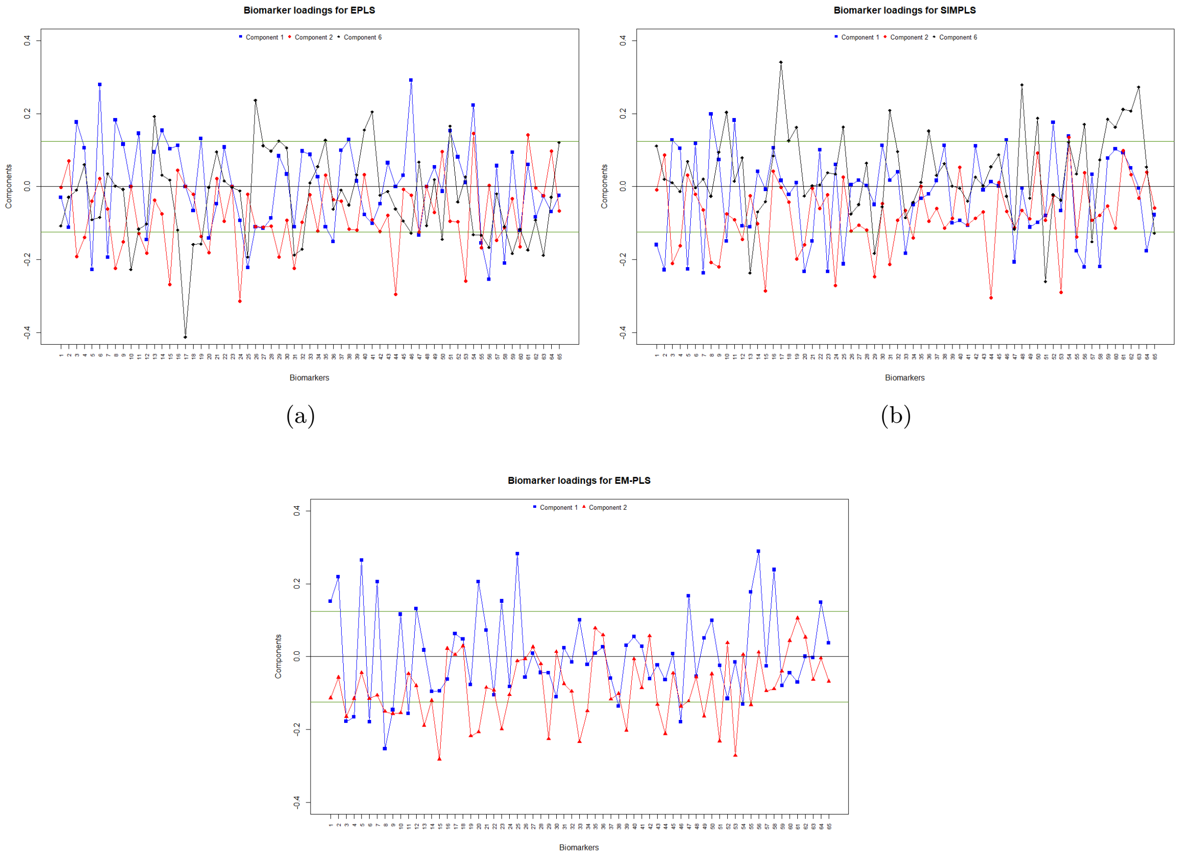
(c)

Figure 6.10: *Loadings per component for the joint modelling of the three outcomes. The red dots connected by blue, red, black lines are the loadings for SIMPLS, EPLS, and EM-PLS, respectively.*

PLS are commonly used for assessing the importance of markers (Mehmood et al., 2012, 2011). For instance, markers with large loading values are considered more important than markers with small loading values. Also, positive loadings show that a marker and component are positively correlated and vice versa, and if the markers contribute equally to a components their loadings will be equal. The loadings from the SIMPLS method where orthogonalized before comparing to loadings from other methods, which are orthogonal. The loadings for components 1, 2, and 6 from the EPLS, SIMPLS, and EM-PLS methods are shown in Figures 6.10 and 6.11. Figure 6.10 compares the loadings of different components for the PLS2 methods used, and Figure 6.11 shows the loadings of different components for each PLS2 method considered. The numbers 1 to 65 are labels for the markers. The results on Figure 6.10 show that the loadings for SIMPLS and EPLS are similar across all the components considered. However, the loading vector of EM-PLS is only similar to the first loading vector of SIMPLS and EPLS but slightly different for loadings 2. Moreover, the green horizontal line is the value of the loadings if all the markers contributed equally; that is a cut-off. We can see that the number loadings outside the cut-off is smaller than the number within the cut-off. For the first component, the markers labelled 46 and 56 have positive and negative correlations, respectively, with the components, and have strong effect on the first component, which suggests that they may have strong effect on the responses. For instance, marker 46 will increase skin thickness and increase lung functions. Most of the markers have negative correlations with the second component with markers 24, 44, and 53 having the most effects. The plots on 6.11 suggests that for EM-PLS method, the first loadings vector is related to markers that have positive relationship with the first component, and the second loadings are related to markers that have negative relationship with the second component. For EPLS and SIMPLS, the first and second loadings are related to markers with negative relationships with the corresponding components. These relationships sometimes reflect a marker's relationship with the outcomes, which suggests that component 2 of EM-PLS, many markers have a negative effect on the outcomes.

Also, markers that are have similar loading values may be correlated indicating that these markers have similar contributions in the components and outcomes. Further, relationship between markers that share positive and negative loading values is weak suggesting that the effect of positive loading values does not depend on those of negative loading values, and may impact the response variables differently.

Some of the loadings are close to zero indicating that some markers have little influence on the outcomes and suggest that potentially active markers can be identified by shrinking some loadings to exactly zero. In Section 6.3.3, the SPLS2 methods will be used to identify candidate biomarkers.



(a)

(b)

(c)

Figure 6.11: *Loadings per component for the joint modelling of the three outcomes. The red, blue, and black lines are the loadings for components 1, 2, and 6, respectively.*

6.3.3 Sparse multivariate PLS (SPLS2) regression

The PLS methods used in previous sections focused on prediction alone. In this section, the goal is to identify potentially active biomarkers for SSc. We used the

SPLS2 methods discussed in Chapter 5 of the thesis which are E-SPLS, SPLS, and 2S-SPLS. Moreover, we compared the prediction performance of the sparse and non-sparse PLS2 methods. Using the training data the predictive performance of the methods were computed and plotted across all components as shown in Figure 6.11.

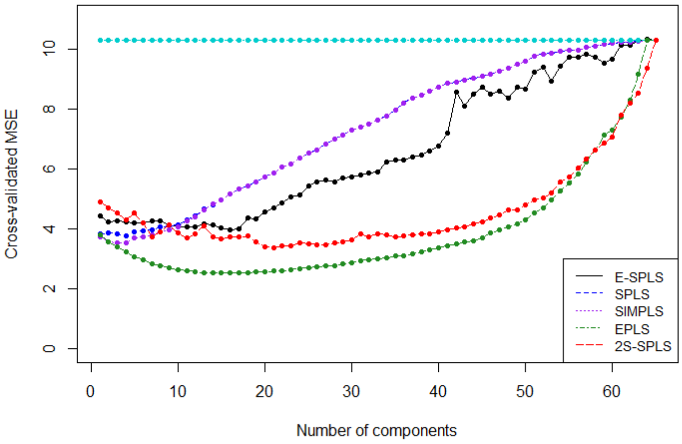


Figure 6.12: *CV-RMSEs for SSc data using the training set. The CV-RMSEs for SPLS, E-SPLS, 2S-SPLS, EPLS, SIMPLS, and OLS corresponds to the blue, black, red, purple, green, and light green lines, respectively. The points on the lines are the CV-RMSEs for each component.*

The results show that the EPLS method has the smallest RMSE compared to the other methods, and the non-sparse PLS methods performed slightly better than the sparse methods. In addition, SPLS and SIMPLS are similar and require only one component, and 2S-SPLS and E-SPLS method are also similar in terms of RMSE from component 1 to component 18. The required number of components and the corresponding CV-RMSE for the methods are given in Table 6.14. The table shows that 2S-SPLS and EPLS require more components compared to other methods. For this dataset, the results also show that the non-sparse PLS methods have smaller RMSEP compared to the sparse PLS methods, which is because the non-sparse methods uses more predictors compared to the sparse methods (Friedman et al., 2001; van Wieringen, 2015) and due to the small sample size of the test set. However, the values of the CV-RMSE for the sparse and non-sparse PLS methods are comparable.

The SPLS2 methods shrunk some regression coefficients to exactly zero. The markers corresponding to nonzero regression coefficients are the active markers. The plots of the regression coefficients for each method are given in Figure 6.13. The figures show that E-SPLS selected fewer markers compared to SPLS and 2S-SPLS. Furthermore, the markers selected by each method are in given in Table 6.15 and shows that the methods selected different markers. For the markers selected by the 2S-SPLS method, the results suggests that *Periostin* and *PSP.D* both have adverse effects on the outcomes: *Periostin* and *PSP.D* increase skin thickness (*MRSS*) and reduces lung functions (*DLCO* and *FVC*). Also, *Apo.E* reduces lung functions and skin thickness, in other words it may be helpful for returning the skin to normal but may be the detriment of the lungs (although the coefficients of the lungs here are close to zero and may have no effect on the lungs). The *Endoglin* seem to increase the lung function and skin thickness, that is, it may improve lung functions but worsen the skin condition, although the coefficient for *MRSS* is small and the effect of *Endoglin* on skin may be negligible. The results of E-SPLS suggests that the *AGP.1* biomarker improves lung functions and reduces skin thickening, where as *CRP* worsens lung functions and increases the thickness of skin. The biomarkers for SPLS can be interpreted accordingly. Overall, *CRP*, *Periostin*, and *PSP.D* have adverse effects of the outcomes, also *Endoglin* and*Apo.E* have adverse effects on the skin and lungs, respectively. In addition, *AGP.1* has a positive effect on both the lungs and skin.

The markers with nonzero coefficients are weakly correlated with some markers with zero coefficients suggesting that they have negligible effect on the contribution of nonzero coefficient on response variables (Kraha et al., 2012). Also, the largest correlation between markers with nonzero coefficients and markers with zero coefficients is approximately 0.5, this indicates that the contributions of the zero-coefficient markers to the nonzero-coefficient marker is small.

6.3.4 Conclusion

In this chapter, we analysed a proteomics data set of patients with SSc to predict patients outcomes and identify candidate biomarkers. We proposed that only a few components and a few biomarkers are required for modelling the outcome variables (MRSS, FVC, and DLCO), and applied various PLS methods discussed in previous

FVC, MRSS, and DLCO			
CN = 316.7813			
Method	q^*	CV-RMSE	RMSEP
E-SPLS	1	4.4465	8.2219
2S-SPLS	7	3.7308	8.0223
SPLS	1	3.8359	8.0165
EPLS	6	2.9588	6.5343
SIMPLS	1	3.7266	6.6755
OLS	65	10.2836	12.4039

Table 6.14: *The optimal number of components for each method, the corresponding CV-RMSE and RMSEP values.*

Methods		
2S-SPLS	E-SPLS	SPLS
Periostin	AGP.1	Periostin
PSP.D	CRP	PSP.D
Apo.E		CRP.
CD40.L		Leptin.R
CCL15		HE4
Endoglin		IL.2.receptor.alpha
		Prostasin
		PIINP
		vWF
		CEACAM6
		ICAM.1

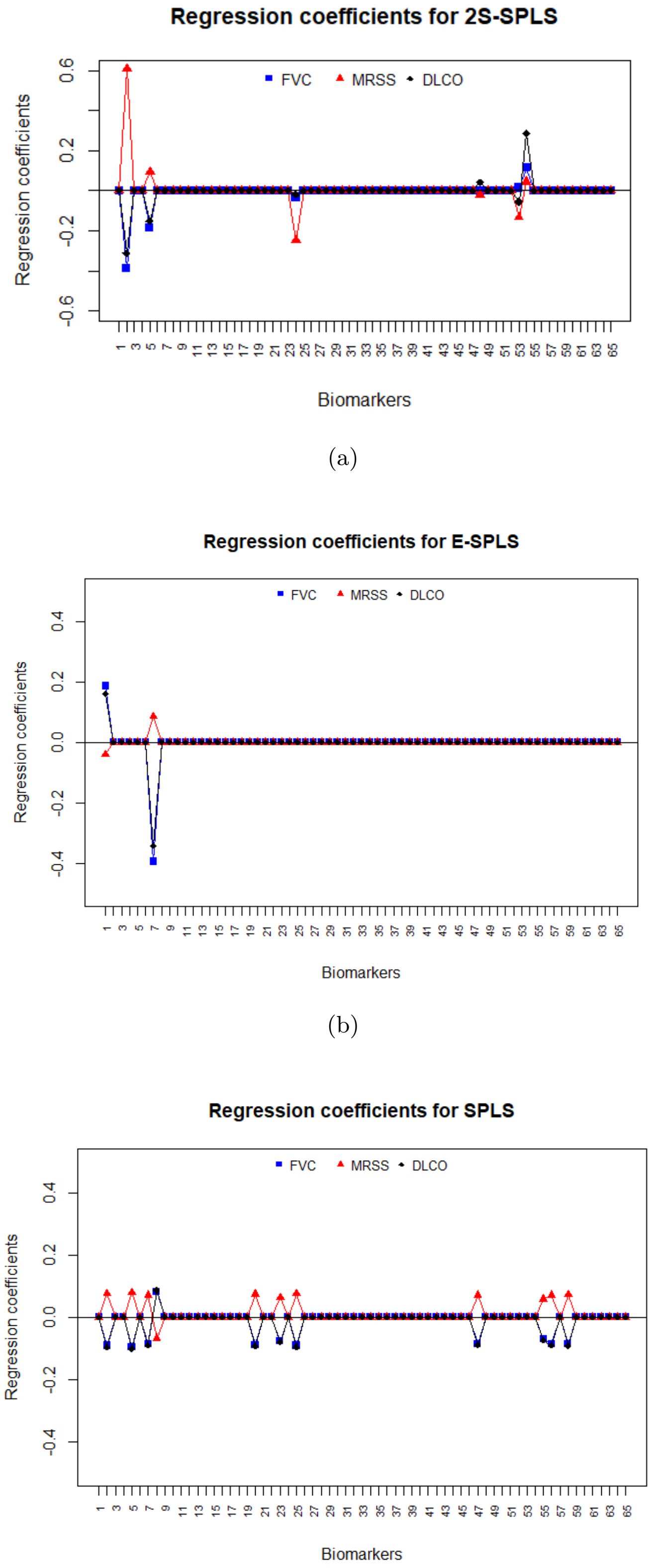
Table 6.15: *List of candidate biomarkers selected by each method.*

chapters to estimate the parameters of the model.

We observed that because of the near multicollinearity present in the data set (as indicated by the condition number) only a few components of the proteomics data set were required to predict the SSc outcomes. The results suggests that for this data set better predictions can be made by summarizing the information in the markers about SSc in a fewer number of variables (components) rather than use all the markers. The predictive performance of the methods when the outcomes are considered separately suggests that the EPLS method has better prediction performance compared to other methods when the biomarkers are nearly multicollinear (see condition numbers in Tables 6.9, 6.10, and 6.11). This result is similar to the results of the simulation setting when the sample size is small compared to the number of predictor variables in Chapter 4. The results are also similar for when the outcomes are modelled together: EPLS has lower root-mean-square error compared to other methods. However, the EPLS method uses more components compared to SIMPLS and EM-PLS methods but increasing the number of components used for SIMPLS and EM-PLS methods will not improve prediction performance. This suggests that for this data set, where biomarkers are nearly multicollinear EPLS should be used.

To determine candidate biomarkers the loadings of the PLS methods were examined and some loadings were close to zero suggesting that the corresponding biomarkers may not be important for the outcomes, which we investigated further using sparse PLS methods. The prediction performance of the sparse PLS methods where compared to that of the non-sparse PLS methods, and the results suggests that the non-sparse PLS methods perform better that the sparse PLS methods. The difference in predictive performance may be because the sparse PLS methods use a smaller number of biomarkers compared to the non-sparse PLS methods, moreover the number of observations is small ($n = 80$) compared to the number of biomarkers ($p = 65$), which can influence the performance of the methods. In particular, the EPLS method outperformed other methods with regard to predictive performance.

In addition, the sparse PLS methods were used to select candidate biomarkers and there seem to be an overlap in the selected biomarkers with many biomarkers being different. However, further investigate show that the biomarkers selected by the different methods are correlated, which could explain why different methods select different markers. Also, the small sample size might be another reason the methods selecting different markers.



(a)

(b)

(c)

Figure 6.13: *Regression coefficients for the joint modelling of the three outcomes. The red dots connected by blue, red, black lines are the coefficients for FVC, MRSS, andDLCO, respectively.*

Chapter 7

Conclusion and Future work

7.1 Introduction

This chapter summarizes the main results of the thesis and includes suggestions, which could improve some aspects of the research. Further, some future works are proposed as offshoots of the research.

7.2 Summary of the thesis

Large data sets are associated with near or perfect multicollinearity, and the OLS method cannot handle such data sets. The thesis is concerned with models and methods that can deal with near or perfect multicollinearity in a large data set. The main results of the thesis are summarised below.

In Chapter 3, we introduced a decomposable model called the relevant components ($\mathrm{RC}_{\boldsymbol{x}}$) model, which has specific restrictions on the joint covariance matrix of the response and predictor variables. This description of the model sheds more light into the understanding of PLS regression. The $\mathrm{RC}_{\boldsymbol{x}}$ model is appropriate for dealing with near or perfect multicollinearity because the required information in $\boldsymbol{x}$ is summarised in a few important components. Moreover, we showed that the model can be represented as a Krylov model of dimension q (Inguanez, 2015). Besides, we reviewed several PLS1 algorithms for estimating the parameters of the $\mathrm{RC}_{\boldsymbol{x}}$ model and showed that the algorithms aim to compute a semi-orthogonal matrix which spans the Krylov subspace. The semi-orthogonal matrix is then used for estimating the parameters of the model and perform prediction.

In Chapter 4, we integrated some models under the framework of the RC model for multiple-response regression analysis when the predictor variables are nearly or perfectly multicollinear. The RC models, in the chapter, were introduced separately in the literature, by providing a unified model framework, it becomes clear that they are alternative models for dealing with multicollinear in the variables but have different restrictions on the joint covariance matrix of the response and predictor variables. These models were discussed under (a) reduction of the predictor variable alone, and (b) reduction of the response and predictor variables. Besides, some isotropic RC models were discussed, which were more restricted compared to other RC models. Furthermore, three PLS2 methods for estimating the parameters of the models were reviewed and compared in terms of predictive ability through simulation studies. The results show that these methods perform better than the OLS method in terms of prediction when the data are nearly multicollinear.

In Chapter 5, we addressed the problem of variable selection in multiple-response linear regression when the predictor variables are nearly multicollinear. We introduced a sparse relevant components (SRC) model which combines the assumptions that (a) only a few relevant components from the predictor variables are related to the responses and (b) some predictor variables have zero covariances with the response variables. The SRC model also has restrictions on the joint covariance matrix of the response and predictor variables. The predictor variables having nonzero covariances with the responses are regarded as active predictor variables and vice versa. Also, we reviewed several sparse PLS methods for estimating the parameters of the SRC model. The methods force some rows of the estimated semi-orthogonal matrix or regression coefficients to exactly zero; rows of the semi-orthogonal matrix correspond to rows of the regression coefficients. Further, we proposed a novel method for estimating the parameters of the SRC model, which shrinks the regression coefficients directly to exactly zero. The novel method is a two-stage technique which reduces the dimension of the predictor variables in the first stage and introduces sparsity for selecting active predictor variables in the second stage. Also, the novel method has an added advantage compared to methods applied in the literature. The novel method can (a) select predictor variables which explain the multiple response variables together (group sparsity), and (b) select predictor variables that are related to subsets of the responses (within-group sparsity). Moreover, simulation studies were done to compare the methods in terms of prediction performance and variable selection accuracy. The results show that the proposed method performs better than other methods in terms of variable selection when with-group sparsity is needed.

In Chapter 6, we analysed a real data set of patients with systemic sclerosis (SSc). The data has three outcome variables, 196 proteomics variables, 98 patients, and some proteomics variables are nearly multicollinear. The outcome variables are known effects of SSc on the skin and lungs of patients, and the proteomics variables are different protein markers taken from the blood of patients. We proposed that the data are drawn from the RC$_{\boldsymbol{x}}$ model and applied various PLS methods to the data to predict the severity of SSc outcomes and select candidate biomarkers; selecting candidate biomarkers can reduce the number of hospital visits made by patients. The results show that better SSc predictions can be made when a few linear combinations of the proteomics variables used rather than using all the proteomic variables.

7.3 Summary of results

From the thesis we can draw the following conclusions:

1 PLS is a family of algorithms for estimating the parameters of various RC models introduced in Chapter 3 4, and 5. Moreover, the RC models provide a fresh perspective into the literature surrounding PLS methods. Also, the overarching of PLS algorithms is to determine a semi-orthogonal matrix with spans the Krylov subspace, and the semi-orthogonal matrix is used for estimating all the parameters of the model. In addition, the single-response RC model can be represented alternatively as Krylov model of order q.

2 From the simulation studies of Chapters 4 and 5, and the real data example of Chapter 6 we can conclude that PLS and SPLS methods outperform the OLS method in terms of predictive ability when predictor variables are nearly multicollinear. Furthermore, the predictive ability of the EPLS method

is moderately better than that of other PLS2 methods when the number of relevant components is large compared to the number of observations. This result is true whether the predictor variables are nearly multicollinear or not. The EM-PLS method had poor prediction performance when the number of predictor variables is small compared to the number of observation. However, the EM-PLS method is used for estimating the parameters of an isotropic RC model and data used in the simulation studies were drawn from the $\text{RC}_{\boldsymbol{x}}$ model.

3 From the results of Chapter 6, we conclude that the proposed two-stage SPLS method offers an added advantage in terms of variable selection when selection is based on within-group sparsity. Simulation studies confirmed that the proposed method outperformed other SPLS2 methods with regard to selection accuracy when the response variables are not related with the same set of predictor variables.

7.4 Publishable material

The publishable material from the thesis are as follows:

1 ***Title: Model and Prediction Comparison for Multivariate Partial least Squares Regression.*** This paper will contain a review of different relevant components regression models, and compare the predictive performance of various PLS2 methods not yet considered in the literature such as EM-PLS (el Bouhaddani et al., 2018).

2 ***Title: Sparse Multivariate Partial Least Squares Regression.*** The paper will propose new sparse PLS methods which are more flexible with regard to variables selection when compared to methods in the literature. The proposed method can account for both group and within-group sparsity in variable selection.

3 ***Title: Partial Least Squares Regression for modelling Systemic Sclerosis using proteomics markers.*** This paper will analysis a proteomics data set of patients with systemic sclerosis using sparse and non-sparse

PLS methods. The goal is to predict multiple outcome variables jointly and select candidate biomarkers related to the multiple outcome variables together.

7.5 Improvement of the study and future work

In the simulation studies of Chapter 4, the predictive performance of the methods were compared by simulating data from the RC$_{\boldsymbol{x}}$ model. The study can further be expanded to contain examples comparing the predictive performance of the methods when data are simulated from the RC$_{\boldsymbol{yx}}$ and isotropic RC$_{\boldsymbol{yx}}$ models. Also, while simulating from the RC$_{\boldsymbol{x}}$ model the number of response variables was fixed at $r = 3$; further studies can be carried out for settings when $r \geq 3$.

In Chapter 5, a function was written in R software for estimating the parameters of the novel two-stage SPLS method. The function takes a long time to run, so it may be beneficial to improve the function in terms of speed. Moreover, it might be better to use an algorithm which alternates between the component extraction stage and variable selection stage until convergence. The simulation studies in the thesis only considered settings when $n > p$, more scenarios can be considered with $n < p$. Moreover, in Chapter 6 observations with NAs where ignored, and it could be useful to estimate the NAs using multiple imputation before applying the methods.

The thesis assumed that the response variables are normally distributed. For future work, (a) it would be interesting to consider RC models that can accommodate situations when some response variables are discrete and others are continuous. For instance, the MRSS outcome of SSc is Poisson distributed, while DLCO and FVC are normally distributed. A joint model can be used to model the outcomes together while taking into account the multicollinearity in the markers. (b) Moreover, PLS2 methods can be developed for longitudinally measured predictor variables to accommodate the situation when the data are measure at different time points. For example, the required proteomics data were measured at three fixed time points, but we analysed only the first time point. Also, as some data sets have predictor variables with detection limits, PLS2 methods can be proposed for handling detection limits in predictor variables.

Appendix A

R Codes

A.1 R script for comparing prediction performance of PLS2 methods

A.2 R script for comparing prediction performance of PLS2 methods

```
#devtools::install_github('selbouhaddani/PPLS/Package/PPLS')
library(mvtnorm); library(PPLS); library(pracma); library(Renvlp); library(pls);
library(PPLS);library(matrixStats);library(fBasics); library(Hmisc); library(ggplot2);
library(reshape2); library(simrel); library(future.apply)library(MASS);
#devtools::install_github("selbouhaddani/PO2PLS@RCpp")
library(OmicsPLS); library(PO2PLS)
#_______________________________________________________________________________________________


#___________ data simulation __________________________________
#_______________________________________________________________________________________
rm(list = ls())
set.seed(100)
runs <- 100 #zhu used 200 replications
n <- 30 # zhu used 50 to 1000
p <- 5
r <- 3
q <- 2
delt <- 0.8
rho <- 0.01

#IRn30p5r3q2delta0_8rho0_8

gramschmidt <- function(x) {
  x <- as.matrix(x)
  # Get the number of rows and columns of the matrix
  n <- ncol(x)
  m <- nrow(x)

  # Initialize the Q and R matrices
  q <- matrix(0, m, n)
  r <- matrix(0, n, n)

  for (j in 1:n) {
    v = x[,j] # Step 1 of the Gram-Schmidt process v1 = a1
    # Skip the first column
    if (j > 1) {
      for (i in 1:(j-1)) {
        r[i,j] <- t(q[,i]) %*% x[,j] # Find the inner product (noted to be q^T a
        #earlier) Subtract the projection from v which causes v to become perpendicular
        # to all columns of Q
        v <- v - r[i,j] * q[,i]
      }
    }
    # Find the L2 norm of the jth diagonal of R
    r[j,j] <- sqrt(sum(v^2))
    # The orthogonalized result is found and stored in the ith column of Q.
    q[,j] <- v / r[j,j]
  }

  # Collect the Q and R matrices into a list and return
  qrcomp <- list('Q'=q, 'R'=r)
  return(qrcomp)
}

gamma <- gramschmidt(matrix(rnorm(p*p, 0, 1), p, p))$Q

gamma1 <- gamma[,1:q]
gamma0 <- gamma[,(q+1):p]

sig <- matrix(0, p,p)
for(i in 1:p){
  for(j in 1:p){
    sig[i,j] <- rho^(abs(i - j))
  }
}

# omega <- eigen(sig)$values
# sigx <- gamma1%*%diag(omega[1:q], q) %*%t(gamma1) +
#gamma0%*%diag(omega[(q+1):p], (p-q)) %*%t(gamma0)

omega <- sort(eigen(sig)$values, decreasing = F)
sigx <- gamma1%*%diag(omega[1:q], q) %*%t(gamma1) +
  gamma0%*%diag(omega[(q+1):p], (p-q)) %*%t(gamma0)

sigy.x <- matrix(0, r,r)
for(i in 1:r){
  for(j in 1:r){
    sigy.x[i,j] <- delt^(abs(i - j))
  }
}

#sigy.x <- diag(c(2,1.5,0.5), r)

Beta <- gamma1%*%matrix(runif(q*r, 0, 2), q, r)

X <- list(); E <- list(); Y <- list()
for(k in 1:runs){
  X[k] <- list(mvrnorm(n, rep(0, p), sigx))
  E[k] <- list(1*mvrnorm(n, rep(0, r), sigy.x))
  Y[k] <- list(scale(X[[k]]%*%Beta + E[[k]]))
  X[k] <- list(scale(X[[k]]))
}
#cor(X[[1]])


plan(multiprocess)
#______________________Using the Envelope
fold = k = 10
groups <- sample(rep(seq_len(fold), length.out = n))

myenvCv <- function(X,Y,k){
  #Y <- as.matrix(Y[[1]]);  X <- as.matrix(X[[1]])
  Y <- as.matrix(Y);  X <- as.matrix(X)
  a <- dim(Y);  n <- a[1];  r <- a[2];  p <- ncol(X); p <- p-1

  fitt <- plsr(Y ~ X, ncomp = p, method = "simpls")$loadings

  M <- list()
  efit <- matrix(M,k,p)
  for(i in 1:k){
    for(j in 1:p){
      efit[i,j] <- list(xenv(X[groups!=i,], Y[groups!=i,], u = j, asy = F,
                              as.matrix(fitt[,1:j])))
    }
  }

  prederror <- matrix(M,k,p)
  for(i in 1:k){
    for(j in 1:p){
      prederror[i,j] <- list(as.matrix(Y[groups==i,] - as.matrix(X[groups==i,])
                                       %*%efit[[i,j]]$beta))
    }
  }

  ecv <- matrix(0,k,p)
  for(i in 1:k){
    for(j in 1:p){
      ecv[i,j] <- norm(prederror[i,j][[1]], type = "F")
    }
  }

  return(envpls = colMeans(ecv))

}
envrepp1 <- matrix(unlist(future_lapply(1:runs, function(i) myenvCv(X=X[[i]],
                                                                     Y=Y[[i]], k=fold))),
                   ncol=p-1, byrow=T)
ecv_mn1 <- colMeans(envrepp1); ecv_sd1 <- colSds(envrepp1)

sink("EIRn30p5r3q2delta0_8rho0_01.txt")
print(envrepp1)
print(colMeans(envrepp1))
print(colSds(envrepp1))
sink()


#_______________________Using the SIMPLS

mysimplsCv <- function(X,Y,k){
  Y <- as.matrix(Y)
  X <- as.matrix(X)
  a <- dim(Y)
  n <- a[1]
  r <- a[2]
  p <- ncol(X)


  sfit <- NULL
  for(i in 1:k){
    sfit[i] <- list(plsr(Y[groups!=i,] ~ X[groups!=i,], ncomp = p, method = "simpls"))
  }



  cv <- matrix(0,k,p)
  for(i in 1:k){
    for(j in 1:p){
      cv[i,j] <- norm( as.matrix( Y[groups==i,] - X[groups==i,]%*%
                                   as.matrix(sfit[[i]]$coefficients[,,j]) ),
                       type="F")
    }
  }

  return(splsCv <- colMeans(cv))
}
simplsrepp1 <- matrix(unlist(future_lapply(1:runs, function(i) mysimplsCv(X=X[[i]], Y=Y[[i]],
                                                                         k=fold))),
                      ncol=p, byrow=T)
scv_mn1 <- colMeans(simplsrepp1); scv_sd1 <- colSds(simplsrepp1)

sink("SIRn30p5r3q2delta0_8rho0_01.txt")
print(simplsrepp1)
print(colMeans(simplsrepp1))
print(colSds(simplsrepp1))
sink()



#_______________________Using the EM-PLS

# myPPLSCv <- function(X, Y, k, emnum){
#   Y <- as.matrix(Y);   X <- as.matrix(X)
#   a <- dim(Y)
#   n <- a[1];   r <- a[2]
#   p <- ncol(X)
#   M=list()
#   pfit <- matrix(M,k,r)
#   for(i in 1:k){
#     for(j in 1:r){
#       pfit[i,j] <- list(PPLS_simult(X=as.matrix(X[groups!=i,]),
#                 Y=as.matrix(Y[groups!=i,]), a = j, EMsteps = emnum, atol = 1e-09,
#type = "SVD"))
#     }
#   }
#   pp.pred <- matrix(0, k,r)
#   for(i in 1:k){
#     for(j in 1:r){
#       pp.pred[i,j] <- norm(as.matrix(Y[groups==i,] -
#             X[groups==i,]%*%ginv(pfit[i,j][[1]]$estimates$W%*%
#t(pfit[i,j][[1]]$Expectations$mu_T)%*%
#             pfit[i,j][[1]]$Expectations$mu_T%*%t(pfit[i,j][[1]]$estimates$W))%*%
#pfit[i,j][[1]]$estimates$W%*%t(pfit[i,j][[1]]$Expectations$mu_T)%*%
#pfit[i,j][[1]]$Expectations$mu_U%*%
#             t(pfit[i,j][[1]]$estimates$C)),type="F")
#     }
#   }
#   return <- colMeans(pp.pred)
# }
# PPLSrepp1 <- matrix(unlist(future_lapply(1:runs, function(i) myPPLSCv(X=X[[i]],
#                                                    Y=Y[[i]], k=fold, emnum = 50))),
#ncol=r, byrow=T)
# pcv_mn1 <- colMeans(PPLSrepp1); pcv_sd1 <- colSds(PPLSrepp1)




myPPLSCv <- function(X, Y, k, emnum){
  Y <- as.matrix(Y);  X <- as.matrix(X)
  a <- dim(Y);  n <- a[1];   r <- a[2];  p <- ncol(X)

  # Y <- as.matrix(Y);   X <- as.matrix(X)

  M=list();  pfit <- matrix(M,k,r)
  for(i in 1:k){
    for(j in 1:r){
      pfit[i,j] <- list(PO2PLS(X[groups!=i,], Y[groups!=i,], r = j, 0, 0, steps=emnum,
                               tol=0.000001))
    }
  }


  pp.pred <- matrix(0, k,r)
  for(i in 1:k){
    for(j in 1:r){
 pp.pred[i,j] <- norm(as.matrix(Y[groups==i,] - X[groups==i,] %*% ginv(pfit[i,j][[1]]$params$W
                              %*% pfit[i,j][[1]]$params$SigT %*% t(pfit[i,j][[1]]$params$W))
                              %*% pfit[i,j][[1]]$params$W %*% pfit[i,j][[1]]$params$SigT %*%
                            pfit[i,j][[1]]$params$B %*% t(pfit[i,j][[1]]$params$C)),
type="F")
    }
  }

  return <- colMeans(pp.pred)
}
PPLSrepp1 <- matrix(unlist(future_lapply(1:runs, function(i) myPPLSCv(X=X[[i]],
                                          Y=Y[[i]], k=fold, emnum = 150))), ncol=r, byrow=T)
pcv_mn1 <- colMeans(PPLSrepp1); pcv_sd1 <- colSds(PPLSrepp1)


sink("PIRn30p5r3q2delta0_8rho0_01.txt")
print(PPLSrepp1)
print(colMeans(PPLSrepp1))
print(colSds(PPLSrepp1))
sink()



#_____________________Using the OLS
mlmCV <- function(X, Y, k){
  Y <- as.matrix(Y); X <- as.matrix(X);  a <- dim(Y);
  n <- a[1];  r <- a[2];  p <- ncol(X)
  #k=10
  mlm1 <- NULL
  for(i in 1:k){
    mlm1[i] <- list(lm(cbind(Y[groups!=i,]) ~ X[groups!=i,]))
  }

  Yhat1 <- NULL
  for(i in 1:k){
    Yhat1[i] <- list( X[groups==i,]%*%coef(mlm1[[i]])[-1,] )
  }

  mlmnorm <- rep(0, k)
  for(i in 1:k){
    mlmnorm[i]=norm(as.matrix(Y[groups==i,]-Yhat1[[i]]), type = "F")
  }
  return(rep(mean(mlmnorm),p))
}
mlmrepp <- matrix(unlist(future_lapply(1:runs, function(i) mlmCV(X=X[[i]], Y=Y[[i]], k=fold))),
                 ncol=p, byrow=T)
lcv_mn <- colMeans(mlmrepp); lcv_sd <- colSds(mlmrepp)

sink("LIRn30p5r3q2delta0_8rho0_01.txt")
print(mlmrepp)
print(colMeans(mlmrepp))
print(colSds(mlmrepp))
sink()



#___________________ Plots ______________________________________________________
x <- c(1:p)
```

```
plot(x,c(ecv_mn1, lcv_mn[1]), type="o", col="red", ylab="Cross-validated␣RMSE",
     xlab=expression("Number␣of␣components"), lty=1, pch=20,
     ylim = c(0, 3), main=expression(list(n==30, p==5, r==3, q==2, rho==0.01)))
#errbar(x, ecv_mn1, ecv_mn1 + ecv_sd1, ecv_mn1 - ecv_sd1, add = T, col = "red",
# cap=0.01, pch = 20, lwd = 0.5, errbar.col = "red")
lines(x,scv_mn1, type="o",pch=20, col="blue", lty=2)
#errbar(x*1.02, scv_mn1, scv_mn1 + scv_sd1, scv_mn1 - scv_sd1, add = T,
# cap=0.01, col = "blue", pch = 20, lwd = 0.5, errbar.col = "blue")
lines(c(1:r), pcv_mn1, type="o", col="black",pch=20, lty=3)
#errbar(c(1:r), pcv_mn1, pcv_mn1 + pcv_sd1, pcv_mn1 - pcv_sd1, add = T,
# cap=0.01, col = "black", pch = 20, lwd = 0.5, errbar.col = "black")
lines(x,lcv_mn, type = "o",pch=20, col = "cyan3", lty=4)
#errbar(x*1.04, lcv_mn, lcv_mn + lcv_sd, lcv_mn - lcv_sd, add = T,
# col = "cyan3", cap=0.01, pch = 20, lwd = 0.5, errbar.col = "cyan3")
# lines(x, rep(mean(unlist(evar1)), p), col = "purple", lty = 5)
abline(v=q, col = "gray", lty = 6)
legend("bottomright", legend=c("EPLS","SIMPLS","EM-PLS", "OLS"),
       col=c("red", "blue", "black", "cyan3"), lty=1:4,cex=1)
```

A.3 R script for variable selection using 2S-SPLS method

```
#_______________ Data generation
rm(list = ls())
runs <- 50
r <- 3
q <- 3
p <- 12
pa <- 4
pi <- p - pa
pa.q <- pa - q
n <- 10000
delt <- 0.95

gramschmidt <- function(x) {
  x <- as.matrix(x)
  # Get the number of rows and columns of the matrix
  n <- ncol(x)
  m <- nrow(x)

  # Initialize the Q and R matrices
  q <- matrix(0, m, n)
  r <- matrix(0, n, n)

  for (j in 1:n) {
    v = x[,j] # Step 1 of the Gram-Schmidt process v1 = a1
    # Skip the first column
    if (j > 1) {
      for (i in 1:(j-1)) {
        r[i,j] <- t(q[,i]) %*% x[,j] # Find the inner product (noted to be q^T a earlier)
        # Subtract the projection from v which causes v to become perpendicular
        #to all columns of Q
        v <- v - r[i,j] * q[,i]
      }
    }
    # Find the L2 norm of the jth diagonal of R
    r[j,j] <- sqrt(sum(v^2))
    # The orthogonalized result is found and stored in the ith column of Q.
    q[,j] <- v / r[j,j]
  }

  # Collect the Q and R matrices into a list and return
  qrcomp <- list('Q'=q, 'R'=r)
  return(qrcomp)
}

gamma.pa <- gramschmidt(matrix(rnorm(pa*pa, 0, 1), pa, pa))$Q
gamma.a1 <- rbind(gamma.pa[,1:q], matrix(0, pi, q))
gamma.a2 <- rbind(matrix(gamma.pa[,(q+1):pa], pa, pa.q), matrix(0, pi, pa.q))
gamma.i <- rbind(matrix(0, pa, pi), diag(1, pi))
gamma0 <- cbind(gamma.a2, gamma.i)

gamma.both <- cbind(gamma.a1, gamma.a2, gamma.i)

omega1 <- diag(4, q)
omega.01 <- diag(0.8, (pa.q))
omega.02 <- diag(1, pi)
omega0 <- as.matrix(bdiag(omega.01, omega.02))

sigx <- gamma.a1%*%omega1%*%t(gamma.a1) + gamma0%*%omega0%*%t(gamma0)

sigy.x <- matrix(0, r,r)
for(i in 1:r){
  for(j in 1:r){
    sigy.x[i,j] <- delt^(abs(i - j))
  }
}

Beta <- rbind(matrix(c(-3,  5,  1,
                       3,  0,  0,
                       2,  1, -2,
                       4,  0,  3), pa, r, byrow = T), matrix(0, pi, r)); Beta


Beta1 <- rbind(matrix(c(-3,  5,  1,
                        3,  5e-4,  5e-4,
                        2,  1, -2,
                        4,  5e-4,  3), pa, r, byrow = T), matrix(5e-4, pi, r)); Beta1


X <- list(); E <- list(); Y <- list()
sigxx <- list(); sigxxyy <- list(); sigyy <- list(); Sxy <- list(); Sigs <- list()
for(k in 1:runs){
  X[k] <- list(mvrnorm(n, rep(0, p), sigx))
  E[k] <- list(mvrnorm(n, rep(0, r), sigy.x))
  Y[k] <- list(scale(X[[k]]%*%Beta + E[[k]]))
  X[k] <- list(scale(X[[k]]))
  sigxx[k] <- list(cov(X[[k]])*(n - 1)/n)
  sigxxyy[k] <- list(cov(X[[k]],Y[[k]])*(n - 1)/n)
  sigyy[k] <- list(cov(Y[[k]])*(n - 1)/n)
  Sxy[k] <-list(sigxx[[k]] - sigxxyy[[k]] %*%solve(sigyy[[k]] )%*%t(sigxxyy[[k]]))
  Sigs[k] <- list(list(Sxy = Sxy[[k]], sigxx = sigxx[[k]], sigxxyy = sigxxyy[[k]]))
}


#____________________ Initial value for 2S-SPLS

myenvCv <- function(X,Y){
  Y <- as.matrix(Y);
  X <- as.matrix(X);
  a <- dim(Y);
  n <- a[1];
  r <- a[2];
  p <- ncol(X);
  M <- list()

  efit <- matrix(M,p)
  for(j in 1:p){
    efit[j] <- list(xenv(X, Y, u = j, asy = F))
  }

  return(efit)
}
```

```
em <- lapply(1:runs, function(i) myenvCv(X[[i]], Y[[i]]))

#_______________________ 2S-SPLS
Gamma.fixed1 <- function(par, X, Y, G, w, low, high, intr){
  n <- nrow(X)
  r <- ncol(Y)
  p <- ncol(X)
  obj <- function(par, X, Y, G, pen, w){
    Y <- as.matrix(Y)
    X <- as.matrix(X)

    X=scale(X, center=T, scale=F)
    Y=scale(Y, center=T, scale=F)

    G <- as.matrix(G)
    n <- nrow(X)
    r <- ncol(Y)
    p <- ncol(X)
    q <- ncol(G)

    B <- matrix(par, p, r)

    pen <- pen
    Z <- X%*%G
    sigZY <- cov(Z, Y)*(n - 1)/n
    sigZ <- cov(Z)*(n - 1)/n
    sigY <- cov(Y)*(n - 1)/n
    sigY.Z <- sigY - t(sigZY)%*%solve(sigZ)%*%sigZY

    g.norma <- function(x1, x2){
      r.norm <- matrix(0,p,r)
      for(i in 1:p){
        for(j in 1:r){
          r.norm[i,j] <- abs(x1[i,j])*(1/abs(x2[i,j])^0.5)
        }
      }
      return(sum(r.norm))
    }

    g.norm1 <- function(x){
      r.norm <- matrix(0,p,r)
      for(i in 1:p){
        for(j in 1:r){
          r.norm[i,j] <- abs(x[i,j])
        }
      }
      return(sum(r.norm))
    }

    g.norm2 <- function(x){
      r.norm1 <- rep(0,p)
      for(i in 1:p){
        r.norm1[i] <- sqrt(sum(x[i,]^2))
      }
      return(sum(r.norm1))
    }

    g.normp <- function(x1, x2){
      r.norm1 <- rep(0,p)
      for(i in 1:p){
        r.norm1[i] <- sqrt(sum(x1[i,]^2))*(1/ sqrt(sum(x2[i,]^2)))^2
      }
      return(r.norm1)
    }

    #J<-tr((1/n)*t(Y-Z%*%t(G)%*%B)%*%(Y-Z%*%t(G)%*%B)%*%solve(sigY.Z))+0.5*g.norma(B,w)+
    #pen*g.normp(B,w)
    J<-tr((1/n)*t(Y-Z%*%t(G)%*%B)%*%(Y-Z%*%t(G)%*%B)%*%solve(sigY.Z)) + pen*g.norma(B, w) +
      0.0005*g.norm2(B)
    #J <- tr((1/n)*t(Y - Z%*%t(G)%*%B)%*%(Y - Z%*%t(G)%*%B)%*%solve(sigY.Z)) +
    #pen*g.norma(B, w)
    #J <- tr((1/n)*t(Y - Z%*%t(G)%*%B)%*%(Y - Z%*%t(G)%*%B)%*%solve(sigY.Z)) +
    #pen*g.norm1(B)
    #J <- tr((1/n)*t(Y - Z%*%t(G)%*%B)%*%(Y - Z%*%t(G)%*%B)%*%solve(sigY.Z)) +
    #pen*g.norm2(B) + pen*g.norm1(B)
    #J <- tr((1/n)*t(Y - Z%*%t(G)%*%B)%*%(Y - Z%*%t(G)%*%B)) + pen*g.norm2(B)
    #____________________________________________________________________________________________
    return(J)
  }

  gridd <- seq(low, high, intr)
  fit4 <- lapply(gridd, function(k) nlm(f=obj, p=par, X=X, Y=Y, G=G, pen=k, w=w,
                                      iterlim = 2000, gradtol = 1e-16))

  mini.fit <- rep(0, length(gridd))
  for(k in 1:length(gridd)){
    mini.fit[k] <- fit4[[k]]$minimum
  }

  para1 <- NULL
  for(k in 1:length(gridd)){
    para1[k] <- list(round(matrix(fit4[[k]]$estimate,p,r), 5))
  }

  bic.fit <- rep(0, length(gridd))
  for(k in 1:length(gridd)){
    bic.fit[k] <- mini.fit[k] + length(as.vector(para1[[k]])[ which(!as.vector(para1[[k]]) == 0)])
      log(n)
  }

  bic.fit.min <- which.min(bic.fit)
  opt.Beta <- para1[[bic.fit.min]]
  opt.lambda <- gridd[bic.fit.min]
  #______________________________________________________________________________________________

  return(list(para1 = para1, mini.fit = mini.fit, bic.fit = bic.fit,
              bic.fit.min = bic.fit.min, opt.Beta = opt.Beta, opt.lambda = opt.lambda))
}



gf <- lapply(1:runs, function(i) Gamma.fixed1(em[[i]][[q]]$beta, X[[i]], Y[[i]],
                                             em[[i]][[q]]$Gamma, w=em[[i]][[q]]$beta,
                                             low=0, high=0.007, intr=0.001))
gf.beta1 <- NULL
for(i in 1:runs){
  gf.beta1[[i]] <- round(gf[[i]]$opt.Beta, 5)
}
gf.beta1

gf.beta <- NULL
for(i in 1:runs){
  gf.beta[[i]] <- as.vector(round(gf[[i]]$opt.Beta, 5))
}

my.table <- function(actual, predicted){
  actual <- as.vector(actual);  predicted <- as.vector(predicted)
  table.new <- table(ifelse(actual==0 & predicted==0, "TN",
                          ifelse(actual !=0 & predicted !=0, "TP",
                                 ifelse(actual==0 & predicted !=0, "FP", "FN"))))
  return(table.new)
}

#___ table ______________________
```

```
tabs3 <- NULL
for(i in 1:runs){
  tabs3[[i]] <- list(my.table(as.vector(Beta), gf.beta[[i]]))[[1]]
}

#___ Accuracy rate ______________
acur3 <- rep(0, runs)
tpr3 <- rep(0, runs)
tnr3 <- rep(0, runs)
for(i in 1:runs){
  acur3[i] <- (ifelse(!is.na(tabs3[[i]]["TP"]), tabs3[[i]]["TP"], 0) +
                 ifelse(!is.na(tabs3[[i]]["TN"]), tabs3[[i]]["TN"], 0)) /
    (ifelse(!is.na(tabs3[[i]]["TP"]), tabs3[[i]]["TP"], 0) +
       ifelse(!is.na(tabs3[[i]]["TN"]), tabs3[[i]]["TN"], 0) +
       ifelse(!is.na(tabs3[[i]]["FP"]), tabs3[[i]]["FP"],0)+ifelse(!is.na(tabs3[[i]]["FN"]),
                                                             tabs3[[i]]["FN"],0))

  tnr3[i]  <- ifelse(!is.na(tabs3[[i]]["TN"]), tabs3[[i]]["TN"], 0)/
    (ifelse(!is.na(tabs3[[i]]["TN"]), tabs3[[i]]["TN"], 0) +
       ifelse(!is.na(tabs3[[i]]["FP"]), tabs3[[i]]["FP"], 0))
  tpr3[i]  <- ifelse(!is.na(tabs3[[i]]["TP"]), tabs3[[i]]["TP"], 0)/
    (ifelse(!is.na(tabs3[[i]]["TP"]), tabs3[[i]]["TP"], 0) +
       ifelse(!is.na(tabs3[[i]]["FN"]), tabs3[[i]]["FN"], 0))
}
acur3
mean(acur3)
tpr3
mean(tpr3)
tnr3
mean(tnr3)


#________________ bias of estimates ______________________
gf.oracle.beta <- matrix(NA, runs, r*pa)
for(i in 1:runs){
  gf.oracle.beta[i,] <- as.vector(round(gf[[i]]$opt.Beta[1:pa,], 7))
}

gf.bias <- colMeans(gf.oracle.beta) - as.vector(Beta[1:pa,])
norm_vec <- function(x) sqrt(sum(x^2))
bias.gf <- norm_vec(as.vector(gf.bias))^2
bias.gf


#_____________ standard deviation of estimates ______________________
gf.oracle.beta <- matrix(NA, runs, r*pa)
for(i in 1:runs){
  gf.oracle.beta[i,] <- as.vector(round(gf[[i]]$opt.Beta[1:pa,], 7))
}
#gf.oracle.beta

gf.total.variance <- sum(eigen(cov(gf.oracle.beta))$values)
gf.total.variance
gf.std <- sqrt(gf.total.variance)
gf.std




#_______________________________________________
#_____________ SPLS method ______________________
spls.beta <- NULL
for(i in 1:runs){
  spls.beta[i] <- list( spls(X[[i]], Y[[i]], q, eta = 0.8, kappa=0.5, select="simpls",
                        fit="simpls", scale.x=F, scale.y=F, eps=1e-4, maxstep=5000,
                        trace=F)$betamat )
}
```

```
new.spls.beta1 <- NULL
for(i in 1:runs){
  new.spls.beta1[[i]] <- round(spls.beta[[i]][[q]], 7)
}
# sink("new.spls.beta1_100.txt")
# print(new.spls.beta1)
# sink()


#___________________vectorize beta_________________________________________
new.spls.beta <- NULL
for(i in 1:runs){
  new.spls.beta[[i]] <- as.vector(round(spls.beta[[i]][[q]], 7))
}

my.table <- function(actual, predicted){
  actual <- as.vector(actual);  predicted <- as.vector(predicted)
  table.new <- table(ifelse(actual==0 & predicted==0, "TN",
                            ifelse(actual !=0 & predicted !=0, "TP",
                                   ifelse(actual==0 & predicted !=0, "FP", "FN"))))
  return(table.new)
}

#___ table ______________________
tabs1 <- NULL
for(i in 1:runs){
  tabs1[[i]] <- list(my.table(as.vector(Beta), new.spls.beta[[i]]))[[1]]
}

#___ Accuracy rate ______________
acur1 <- rep(0, runs)
tpr1 <- rep(0, runs)
tnr1 <- rep(0, runs)
for(i in 1:runs){
  acur1[i] <- (ifelse(!is.na(tabs1[[i]]["TP"]), tabs1[[i]]["TP"], 0) +
                 ifelse(!is.na(tabs1[[i]]["TN"]), tabs1[[i]]["TN"], 0)) /
    (ifelse(!is.na(tabs1[[i]]["TP"]), tabs1[[i]]["TP"], 0) +
       ifelse(!is.na(tabs1[[i]]["TN"]), tabs1[[i]]["TN"], 0) +
       ifelse(!is.na(tabs1[[i]]["FP"]), tabs1[[i]]["FP"], 0) +
       ifelse(!is.na(tabs1[[i]]["FN"]), tabs1[[i]]["FN"], 0))

  tnr1[i]  <- ifelse(!is.na(tabs1[[i]]["TN"]), tabs1[[i]]["TN"], 0)/
    (ifelse(!is.na(tabs1[[i]]["TN"]), tabs1[[i]]["TN"], 0) +
       ifelse(!is.na(tabs1[[i]]["FP"]), tabs1[[i]]["FP"], 0))
  tpr1[i]  <- ifelse(!is.na(tabs1[[i]]["TP"]), tabs1[[i]]["TP"], 0)/
    (ifelse(!is.na(tabs1[[i]]["TP"]), tabs1[[i]]["TP"], 0) +
       ifelse(!is.na(tabs1[[i]]["FN"]), tabs1[[i]]["FN"], 0))
}
acur1
mean(acur1)
tpr1
mean(tpr1)
tnr1
mean(tnr1)


#________________ bias of estimates ______________________
spls.oracle.beta <- matrix(NA, runs, r*pa)
for(i in 1:runs){
  spls.oracle.beta[i,] <- as.vector(round(spls.beta[[i]][[q]][1:p.a,], 7))
}

spls.bias <- colMeans(spls.oracle.beta) - as.vector(Beta[1:pa,])
norm_vec <- function(x) sqrt(sum(x^2))
bias.spls <- norm_vec(as.vector(spls.bias))^2
bias.spls
```

```
#________________ standard deviation of estimates ______________________
spls.oracle.beta <- matrix(NA, runs, r*pa)
for(i in 1:runs){
  spls.oracle.beta[i,] <- as.vector(round(spls.beta[[i]][[q]][1:p.a,], 7))
}
#spls.oracle.beta

spls.total.variance <- sum(eigen(cov(spls.oracle.beta))$values)
spls.total.variance
spls.std <- sqrt(spls.total.variance)
spls.std
```

References

Abdi, H., Vinzi, V. E., Russolillo, G., Saporta, G. and Trinchera, L. (2016), *The Multiple Facets of Partial Least Squares Methods: PLS, Paris, France, 2014*, Vol. 173, Springer. 26

Acharjee, A., Finkers, R., Visser, R. G. and Maliepaard, C. (2013), 'Comparison of regularized regression methods for˜ omics data', *Metabolomics* **3**(3), 1. 2

Adragni, K. P., Cook, R. D., Wu, S. et al. (2012), 'Grassmannoptim: An r package for grassmann manifold optimization', *Journal of Statistical Software* **50**(5), 1–18. 79

Almøy, T. (1996), 'A simulation study on comparison of prediction methods when only a few components are relevant', *Computational statistics & data analysis* **21**(1), 87–107. 4, 13, 35, 65

Andersson, M. (2009), 'A comparison of nine pls1 algorithms', *Journal of Chemometrics* **23**(10), 518–529. 25, 26

Andries, E. and Martin, S. (2013), 'Sparse methods in spectroscopy: an introduction, overview, and perspective', *Applied spectroscopy* **67**(6), 579–593. 65

Barnes, J. and Mayes, M. D. (2012), 'Epidemiology of systemic sclerosis: incidence, prevalence, survival, risk factors, malignancy, and environmental triggers', *Current opinion in rheumatology* **24**(2), 165–170. 94

Baumann, D. and Baumann, K. (2014), 'Reliable estimation of prediction errors for qsar models under model uncertainty using double cross-validation', *Journal of cheminformatics* **6**(1), 47. 16

Belsley, D. A. (1993), 'Conditioning diagnostics collinearity and weak data in regression', *Journal-Operational Research Society* **44**, 88–88. 99

Bernatsky, S., Joseph, L., Pineau, C., Belisle, P., Hudson, M. and Clarke, A. (2009), 'Scleroderma prevalence: Demographic variations in a population-based sample', *Arthritis Care & Research* **61**(3), 400–404. 94

Biagioni, D. J., Astling, D. P., Graf, P. and Davis, M. F. (2011), 'Orthogonal projection to latent structures solution properties for chemometrics and systems biology data', *Journal of Chemometrics* **25**(9), 514–525. 35

Borga, M., Landelius, T. and Knutsson, H. (1997), *A unified approach to pca, pls, mlr and cca*, Linköping University, Department of Electrical Engineering. 35

Boulesteix, A.-L. and Strimmer, K. (2006), 'Partial least squares: a versatile tool for the analysis of high-dimensional genomic data', *Briefings in bioinformatics* **8**(1), 32–44. 25, 50, 69

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32. 2

Breiman, L., Friedman, J., Stone, C. J. and Olshen, R. A. (1984), *Classification and regression trees*, CRC press. 2

Bro, R. and Elden, L. (2009), 'Pls works', *Journal of Chemometrics* **23**(2), 69–71. 25

Bunea, F., Wegkamp, M. H. and Auguste, A. (2006), 'Consistent variable selection in high dimensional regression via multiple testing', *Journal of statistical planning and inference* **136**(12), 4349–4364. 101

Califf, R. M. (2018), 'Biomarker definitions and their applications', *Experimental Biology and Medicine* **243**(3), 213–221. 95

Chun, H. and 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**(1), 3–25. 5, 66, 67, 69, 73, 78, 82, 92, 96

Cook, R. D., Forzani, L. et al. (2008), 'Principal fitted components for dimension reduction in regression', *Statistical Science* **23**(4), 485–501. 3

Cook, R. D., Li, B. and Chiaromonte, F. (2010), 'Envelope models for parsimonious and efficient multivariate linear regression', *Statistica Sinica* pp. 927–960. 6

Cook, R. D. and Weisberg, S. (2009), *Applied regression including computing and graphics*, Vol. 488, John Wiley & Sons. 11

Cook, R. D. et al. (2007), 'Fisher lecture: Dimension reduction in regression', *Statistical Science* **22**(1), 1–26. 18, 96

Cook, R., Helland, I. and Su, Z. (2013), 'Envelopes and partial least squares regression', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **75**(5), 851–877. 3, 4, 5, 21, 23, 34, 42, 43, 63, 64, 71

De Jong, S. (1993), 'Simpls: an alternative approach to partial least squares regression', *Chemometrics and intelligent laboratory systems* **18**(3), 251–263. 5, 6, 25, 43, 51, 63, 71

De Jong, S. (1995), 'Pls shrinks', *Journal of chemometrics* **9**(4), 323–326. 96

Dondelinger, F., Mukherjee, S. and Initiative, A. D. N. (2020), 'The joint lasso: high-dimensional regression for group structured data', *Biostatistics* **21**(2), 219–235. 66

Edelman, A., Arias, T. A. and Smith, S. T. (1998), 'The geometry of algorithms with orthogonality constraints', *SIAM journal on Matrix Analysis and Applications* **20**(2), 303–353. 79

el Bouhaddani, S., Uh, H.-W., Hayward, C., Jongbloed, G. and Houwing-Duistermaat, J. (2018), 'Probabilistic partial least squares model: Identifiability, estimation and application', *Journal of Multivariate Analysis* . 5, 6, 39, 40, 43, 46, 63, 120

Elden, L. (2004), 'Partial least-squares vs. lanczos bidiagonalization—i: analysis of a projection method for multiple regression', *Computational Statistics & Data Analysis* **46**(1), 11–31. 25

Eriksson, L., Johansson, E., Kettaneh-Wold, N., Trygg, J., Wikström, C. and Wold, S. (2006), *Multi-and megavariate data analysis*, Vol. 1, Umetrics Sweden. 68

Eriksson, L., Wold, S. and Trygg, J. (n.d.), 'O2pls® for improved analysis and visualization of complex data'. 35

Fan, J. and Li, R. (2001), 'Variable selection via nonconcave penalized likelihood and its oracle properties', *Journal of the American statistical Association* **96**(456), 1348–1360. 66, 77

Fan, J., Peng, H. et al. (2004), 'Nonconcave penalized likelihood with a diverging number of parameters', *The Annals of Statistics* **32**(3), 928–961. 77

Fearn, T. (1983), 'A misuse of ridge regression in the calibration of a near infrared reflectance instrument', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **32**(1), 73–79. 3

Frank, I. E. (1987), 'Intermediate least squares regression method', *Chemometrics and Intelligent Laboratory Systems* **1**(3), 233–242. 18

Frank, L. E. and Friedman, J. H. (1993), 'A statistical view of some chemometrics regression tools', *Technometrics* **35**(2), 109–135. 1

Friedman, J., Hastie, T. and Tibshirani, R. (2001), *The elements of statistical learning*, Vol. 1, Springer series in statistics New York. 2, 16, 68, 112

Friedman, J., Hastie, T. and Tibshirani, R. (2010*a*), 'A note on the group lasso and a sparse group lasso', *arXiv preprint arXiv:1001.0736* . 71, 76

Friedman, J., Hastie, T. and Tibshirani, R. (2010*b*), 'Regularization paths for generalized linear models via coordinate descent', *Journal of statistical software* **33**(1), 1. 80

Gallivan, K. A., Srivastava, A., Liu, X. and Van Dooren, P. (2003), Efficient algorithms for inferences on grassmann manifolds, *in* 'IEEE Workshop on Statistical Signal Processing, 2003', IEEE, pp. 315–318. 79

Geladi, P. and Kowalski, B. R. (1986), 'Partial least-squares regression: a tutorial', *Analytica chimica acta* **185**, 1–17. 18

Guan, Y. and Dy, J. (2009), Sparse probabilistic principal component analysis, *in* 'Artificial Intelligence and Statistics', pp. 185–192. 68

Helland, I. S. (1988), 'On the structure of partial least squares regression', *Communications in statistics-Simulation and Computation* **17**(2), 581–607. 4, 19, 20, 25, 27, 30, 32, 96, 101

Helland, I. S. (1990), 'Partial least squares regression and statistical models', *Scandinavian Journal of Statistics* pp. 97–114. 4, 19, 20, 25, 27, 30, 31

Helland, I. S. (1992), 'Maximum likelihood regression on relevant components', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 637–647. 20, 26

Helland, I. S. (2001), 'Some theoretical aspects of partial least squares regression', *Chemometrics and Intelligent Laboratory Systems* **58**(2), 97–107. 30

Helland, I. S. and Almøy, T. (1994), 'Comparison of prediction methods when only a few components are relevant', *Journal of the American Statistical Association* **89**(426), 583–591. 65

Hoerl, A. E. and Kennard, R. W. (1970), 'Ridge regression: Biased estimation for nonorthogonal problems', *Technometrics* **12**(1), 55–67. 2

Höskuldsson, A. (1988), 'Pls regression methods', *Journal of chemometrics* **2**(3), 211–228. 26, 35

Huang, J., Zhang, T. et al. (2010), 'The benefit of group sparsity', *The Annals of Statistics* **38**(4), 1978–2004. 66

Huang, X., Pan, W., Park, S., Han, X., Miller, L. W. and Hall, J. (2004), 'Modeling the relationship between lvad support time and gene expression changes in the human heart by penalized partial least squares', *Bioinformatics* **20**(6), 888–894. 69

Indahl, U. G., Liland, K. H. and Næs, T. (2009), 'Canonical partial least squares—a unified pls approach to classification and regression problems', *Journal of Chemometrics: A Journal of the Chemometrics Society* **23**(9), 495–504. 34

Inguanez, M. B. (2015), Regularization in Regression: Partial Least Squares and Related Models, PhD thesis, University of Leeds. 19, 22, 24, 26, 117

Inguanez, M. B. and Kent, J. T. (2013), 'An approximate maximum likelihood interpretation of partial least squares (pls) sco. 2013 conference'. 23

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013), *An introduction to statistical learning*, Vol. 112, Springer. 14, 68

Jolliffe, I. T. (1982), 'A note on the use of principal components in regression', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **31**(3), 300–303. 3

Jolliffe, I. T., Trendafilov, N. T. and Uddin, M. (2003), 'A modified principal component technique based on the lasso', *Journal of computational and Graphical Statistics* **12**(3), 531–547. 69

Kawano, S., Fujisawa, H., Takada, T. and Shiroishi, T. (2015), 'Sparse principal component regression with adaptive loading', *Computational Statistics & Data Analysis* **89**, 192–203. 66

Knight, K. and Fu, W. (2000), 'Asymptotics for lasso-type estimators', *Annals of statistics* pp. 1356–1378. 66

Kraha, A., Turner, H., Nimon, K., Zientek, L. and Henson, R. (2012), 'Tools to support interpreting multiple regression in the face of multicollinearity', *Frontiers in psychology* **3**, 44. 113

Krämer, N. (2007), 'An overview on the shrinkage properties of partial least squares regression', *Computational Statistics* **22**(2), 249–273. 14

Kruger, U. and Qin, S. J. (2003), 'Canonical correlation partial least squares', *IFAC Proceedings Volumes* **36**(16), 1603–1608. 35

Lee, D., Lee, W., Lee, Y. and Pawitan, Y. (2011), 'Sparse partial least-squares regression and its applications to high-throughput data analysis', *Chemometrics and Intelligent Laboratory Systems* **109**(1), 1–8. 70

Lee, M. and Su, Z. (2018), ‘Renvlp: An r package for efficient estimation in multivariate analysis using envelope models’. 52

Lee, S. Y., Mediani, A., Maulidiani, M., Khatib, A., Ismail, I. S., Zawawi, N. and Abas, F. (2018), ‘Comparison of partial least squares and random forests for evaluating relationship between phenolics and bioactivities of neptunia oleracea’, *Journal of the Science of Food and Agriculture* **98**(1), 240–252. 4

Lindgren, F., Geladi, P. and Wold, S. (1993), ‘The kernel algorithm for pls’, *Journal of Chemometrics* **7**(1), 45–59. 25

Liu, T.-Y., Trinchera, L., Tenenhaus, A., Wei, D. and Hero, A. O. (2014), ‘Jointly sparse global simpls regression’, *arXiv preprint arXiv:1408.0318* . 70

Manne, R. (1987), ‘Analysis of two partial-least-squares algorithms for multivariate calibration’, *Chemometrics and Intelligent Laboratory Systems* **2**(1-3), 187–197. 30

Mehmood, T., Liland, K. H., Snipen, L. and Sæbø, S. (2012), ‘A review of variable selection methods in partial least squares regression’, *Chemometrics and Intelligent Laboratory Systems* **118**, 62–69. 65, 68, 72, 110

Mehmood, T., Martens, H., Sæbø, S., Warringer, J. and Snipen, L. (2011), ‘A partial least squares based algorithm for parsimonious variable selection’, *Algorithms for Molecular Biology* **6**(1), 27. 110

Michael, H. (2018), ‘Early diagnosis and management of sclerosis’.
URL: *prescriber.co.uk/article/early-diagnosis-and-management-of-systemic-sclerosis/* 95

Naes, T. and Helland, I. S. (1993), ‘Relevant components in regression’, *Scandinavian journal of statistics* pp. 239–250. 21

Næs, T. and Martens, H. (1988), ‘Principal component regression in nir analysis: viewpoints, background details and selection of components’, *Journal of chemometrics* **2**(2), 155–167. 3

Naik, P. and Tsai, C.-L. (2000), 'Partial least squares estimator for single-index models', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **62**(4), 763–771. 32

Negahban, S. N., Ravikumar, P., Wainwright, M. J., Yu, B. et al. (2012), 'A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers', *Statistical science* **27**(4), 538–557. 66

Nihtyanova, S. I., Schreiber, B. E., Ong, V. H., Rosenberg, D., Moinzadeh, P., Coghlan, J. G., Wells, A. U. and Denton, C. P. (2014), 'Prediction of pulmonary complications and long-term survival in systemic sclerosis', *Arthritis & rheumatology* **66**(6), 1625–1635. 94

Osborne, S. D., Künnemeyer, R. and Jordan, R. B. (1997), 'Method of wavelength selection for partial least squares', *Analyst* **122**(12), 1531–1537. 68

Phatak, A. and de Hoog, F. (2002), 'Exploiting the connection between pls, lanczos methods and conjugate gradients: alternative proofs of some properties of pls', *Journal of Chemometrics* **16**(7), 361–367. 25

Phatak, A. and De Jong, S. (1997), 'The geometry of partial least squares', *Journal of Chemometrics: A Journal of the Chemometrics Society* **11**(4), 311–338. 22

Pokeerbux, M., Giovannelli, J., Dauchet, L., Mouthon, L., Agard, C., Lega, J.-C., Allanore, Y., Jego, P., Bienvenu, B., Berthier, S. et al. (2019), 'Survival and prognosis factors in systemic sclerosis: data of a french multicenter cohort, systematic review, and meta-analysis of the literature', *Arthritis research & therapy* **21**(1), 86. 94

R Core Team, . (2020), 'R: a language and environment for statistical computing. r foundation for statistical computing website'. 52

Rännar, S., Geladi, P., Lindgren, F. and Wold, S. (1995), 'A pls kernel algorithm for data sets with many variables and few objects. part ii: Cross-validation, missing data and examples', *Journal of Chemometrics* **9**(6), 459–470. 3, 34

Rännar, S., Lindgren, F., Geladi, P. and Wold, S. (1994), 'A pls kernel algorithm for data sets with many variables and fewer objects. part 1: Theory and algorithm', *Journal of Chemometrics* **8**(2), 111–125. 25

Rimal, R., Almøy, T. and Sæbø, S. (2019), 'Comparison of multi-response prediction methods', *Chemometrics and Intelligent Laboratory Systems* **190**, 10–21. 93

Rosipal, R. and Krämer, N. (2005), Overview and recent advances in partial least squares, *in* 'International Statistical and Optimization Perspectives Workshop" Subspace, Latent Structure and Feature Selection"', Springer, pp. 34–51. 35

Rothman, A. J., Bickel, P. J., Levina, E., Zhu, J. et al. (2008), 'Sparse permutation invariant covariance estimation', *Electronic Journal of Statistics* **2**, 494–515. 68

Saad, Y. (1992), *Numerical methods for large eigenvalue problems*, Manchester University Press. 23

Saad, Y. (2003), *Iterative methods for sparse linear systems*, SIAM. 23

Segal, M. R. (2004), 'Machine learning benchmarks and random forest regression'. 2

Shao, J. (1997), 'An asymptotic theory for linear model selection', *Statistica sinica* pp. 221–242. 82

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2013), 'A sparse-group lasso', *Journal of computational and graphical statistics* **22**(2), 231–245. 66, 76

Simon, N. and Tibshirani, R. (2012), 'Standardization and the group lasso penalty', *Statistica Sinica* **22**(3), 983. 66

Sirimongkolkasem, T. and Drikvandi, R. (2019), 'On regularisation methods for analysis of high dimensional data', *Annals of Data Science* **6**(4), 737–763. 1, 3

Sundar, S. and Bhagavan, B. (1999), 'Comparison of krylov subspace methods with preconditioning techniques for solving boundary value problems', *Computers & Mathematics with Applications* **38**(11-12), 197–206. 24

Takane, Y. and Loisel, S. (2014), On the pls algorithm for multiple regression (pls1), *in* 'International Conference on Partial Least Squares and Related Methods', Springer, pp. 17–28. 25

ter Braak, C. J. (2009), 'Regression by l1 regularization of smart contrasts and sums (roscas) beats pls and elastic net in latent variable model', *Journal of Chemometrics: A Journal of the Chemometrics Society* **23**(5), 217–228. 4

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 267–288. 2, 15, 65, 68, 76

Trygg, J. and Wold, S. (2002), 'Orthogonal projections to latent structures (o-pls)', *Journal of Chemometrics: A Journal of the Chemometrics Society* **16**(3), 119–128. 35

van Wieringen, W. N. (2015), 'Lecture notes on ridge regression', *arXiv preprint arXiv:1509.09169* . 112

Varmuza, K. and Filzmoser, P. (2016), *Introduction to multivariate statistical analysis in chemometrics*, CRC press. 2

Wang, H. and Leng, C. (2007), 'Unified lasso estimation by least squares approximation', *Journal of the American Statistical Association* **102**(479), 1039–1048. 66

Wang, H. and Wang, G. (2021), 'Improving random forest algorithm by lasso method', *Journal of Statistical Computation and Simulation* **91**(2), 353–367. 2

Wang, J. (2015), 'Joint estimation of sparse multivariate regression and conditional graphical models', *Statistica Sinica* pp. 831–851. 66

Wold, H. (1966), 'Estimation of principal components and related models by iterative least squares. in multivariate analysis'. 25

Wold, H. (1975), Path models with latent variables: The nipals approach, *in* 'Quantitative sociology', Elsevier, pp. 307–357. 3

Wold, S., Sjöström, M. and Eriksson, L. (2001), ‘Pls-regression: a basic tool of chemometrics’, *Chemometrics and intelligent laboratory systems* **58**(2), 109–130. 34

Yuan, M. and Lin, Y. (2006), ‘Model selection and estimation in regression with grouped variables’, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 49–67. 66, 76

Zhu, G., Su, Z. et al. (2020), ‘Envelope-based sparse partial least squares’, *The Annals of Statistics* **48**(1), 161–182. 5, 66, 67, 69, 71, 72, 73, 74, 82, 92, 96

Zou, H. (2006), ‘The adaptive lasso and its oracle properties’, *Journal of the American statistical association* **101**(476), 1418–1429. 2, 66, 77

Zou, H. and Hastie, T. (2005), ‘Regularization and variable selection via the elastic net’, *Journal of the royal statistical society: series B (statistical methodology)* **67**(2), 301–320. 2, 70

Zou, H., Hastie, T. and Tibshirani, R. (2006), ‘Sparse principal component analysis’, *Journal of computational and graphical statistics* **15**(2), 265–286. 68, 72