Tilburg University

Extending principal covariates regression for high-dimensional multi-block data

Park, S.

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Link to publication in Tilburg University Research Portal

Citation for published version (APA):
Park, S. (2023). *Extending principal covariates regression for high-dimensional multi-block data*. [s.n.].

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 09. Dec. 2023

Extending Principal Covariates Regression for High-dimensional Multi-block Data

Proefschrift ter verkrijging van de
graad van doctor aan Tilburg University
op gezag van de rector magnificus,
prof. dr. W.B.H.J. van de Donk,

in het openbaar te verdedigen
ten overstaan van een door het college
voor promoties aangewezen commissie
in de Aula van de Universiteit op
vrijdag 17 november 2023 om 10:00 uur

door

Soogeun Park

geboren op 4 januari 1994
te Seoul, Republiek Korea

Promotores:	prof. dr. J.K. Vermunt (Tilburg University) prof. dr. E. Ceulemans (KU Leuven)
Copromotor:	dr. K. Van Deun (Tilburg University)
Leden promotiecommissie:	prof. dr. C.S. Strobl (University of Zurich) prof. dr. H.A.L. Kiers (University of Groningen) prof. dr. ir. A.T. Tenenhaus (L2S, CentraleSupélec) prof. dr. M.C. Kaptein PDEng (Tilburg University)

©2023 Soogeun Park, The Netherlands. All rights reserved. No parts of this thesis may be reproduced, stored in a retrieval system or transmitted in any form or by any means without permission of the author. Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd, in enige vorm of op enige wijze, zonder voorafgaande schriftelijke toestemming van de auteur.

Table of Contents

Chapter 1

Introduction

1.1 Background

Availability of rich volumes of data has become more commonplace than ever in many research fields. Researchers are increasingly working with datasets where large or even huge blocks of variables concerning the same observation units are gathered from multiple data sources. Such a data setup is known as ‘multiblock data’ (A. Tenenhaus & Tenenhaus, 2011), and there are many examples of it. In health science, predictor variables from self-reported questionnaires, brain imaging and eye tracking all obtained for the same set of individuals are used in conjunction to study the mechanisms behind nicotine addiction (Kang et al., 2012). Likewise, in clinical psychology, in order to investigate the pathways that lead to eating disorders, gene expression data and questionnaire data are analyzed together (Steiger, Labonté, Groleau, Turecki, & Israel, 2013). For data consisting of many variables - both in the single and multiblock setting - summary variables are often introduced by combining the predictor variables to identify predictive mechanisms behind an outcome. Capturing the information shared among predictor variables, the summary variables can be understood as representations of the predictive mechanisms. When multiple blocks of data are available, this approach of summary variables gives rise to a unique opportunity to reveal predictive mechianisms of a multi-source nature; mechanisms linked with predictor variables originating from multiple data sources can be found. For the eating disorder example, only by the multiblock data setup, a mixture of genetic and environmental variables, such as an interaction between certain genenetic susceptibilities and exposure to childhood abuse, can be discovered as a predictive mechanism. Integrated analysis of multiple sources is therefore considered as a promising approach in many fields for obtaining a comprehensive picture behind an outcome, since it can uncover these mechanisms characterized by a blend of variables from multiple sources.

However, capturing the predictive mechanisms that encompass multiple sources is a challenging task, as they tend to manifest themselves in a subtle way. It is common that mechanisms stemming from individual data blocks are more pronounced in a multiblock dataset (Van Deun, Smilde, Van Der Werf, Kiers, & Van Mechelen, 2009). This is particularly the case when these individual blocks are much larger than other blocks, or when the blocks are heterogeneous (i.e. the scales on which the variables are measured are vastly different across the blocks). A starting point in tackling this issue is by distinguishing two types of predictive mechanisms; while the block-specific mechanism is known as a 'distinctive process', the predictive mechanism in relation with multiple blocks is called a 'common process'. To be concrete, as determinants of eating disorders, the abovementioned example of *interaction between genetic susceptibilities and childhood abuse* would be a common process pertaining to the gene expression and questionnaire blocks, while *home environment* would be a distinctive process concerning only the questionnaire block. Only by a refined approach that completely sets the two processes apart, it is possible to reveal the subtle common processes and fully reach the potential harboured within multiblock data.

Moreover, predictive modelling with multiblock data is complicated by several issues regarding predictor variables. First, multiblock datasets often contain predictor variables that are collinear (highly correlated) with each other (this is known as multicollinearity). This is an inherent issue for a high-dimensional setup (i.e. datasets with a larger number of variables than observations). When not taken care of, multicollinearity makes estimated model coefficients unstable and leads to overfitting; the obtained results are heavily dependent on the specific sample of data in hand rather than correctly reflecting the true nature of the population (Babyak, 2004). Second, as blocks of predictor variables collected without a specific aim are joined together, it is common for multiblock datasets to consist of many predictor variables that are redundant for the research question of interest. In the nicotine addiction example, brain imaging data concerning most areas of the brain may not be relevant for addiction. Presence of many redundant predictors blows up the number of coefficients to be estimated, which not only leaves researchers with a burden to inspect an infeasibly elaborate model, but also inflates the risk of falsely identifying a predictive effect by chance. Therefore, a method that finds a subset of important predictor variables amidst the irrelevant ones and also treats multicollinearity is required, in order to construct a concise and accurate model for prediction. Figure 1.1 illustrates a multiblock data setup and the common and distinctive processes along with the important and unimportant predictor variables.

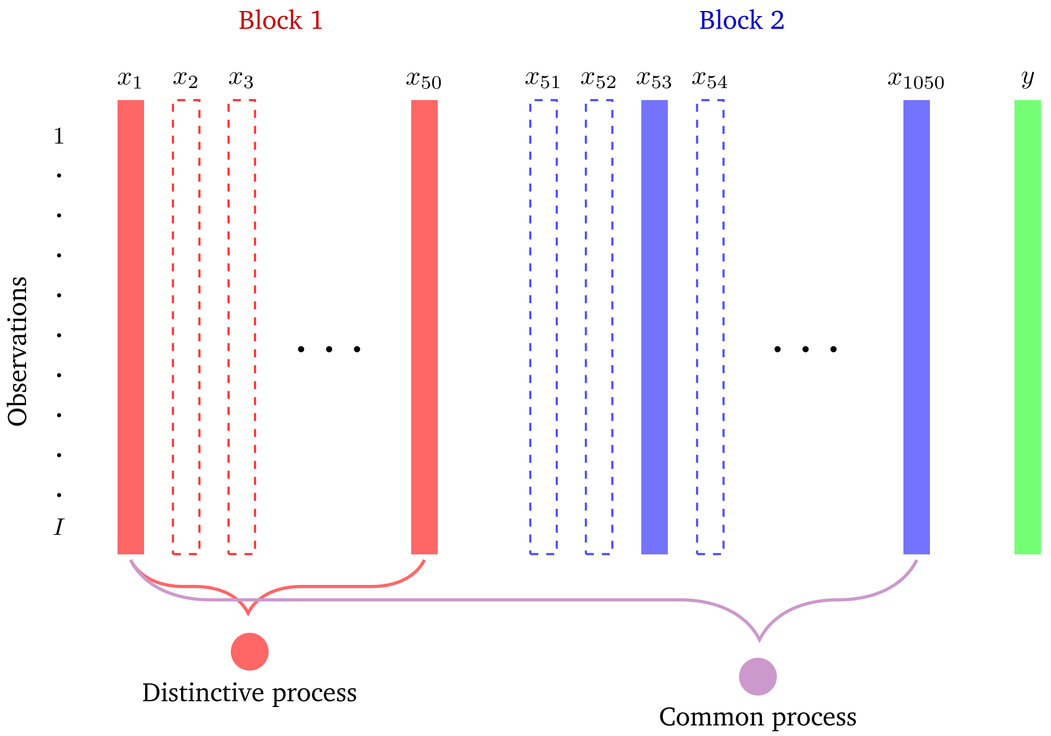


Figure 1.1. Example multiblock data setup with common and distinctive processes. The row indices 1 to I refer to the observations. The columns x_1 to x_{50} indicate the predictor variables in the first block while x_{51} to x_{1050} are predictors in the second block. The outcome variable is indicated by y. The rectangles with dashed line borders indicate predictors that are redundant to the common and distinctive predictive processes behind the outcome, while the filled rectangles refer to relevant predictors.

Challenges present within multiblock data are not only confined to predictor variables, but can also concern the outcome variable. The outcome variable may be continuous or categorical. While the nicotine addiction can be measured by a number of cigarettes smoked per day which would be a continuous variable, the eating disorder may be determined by a medical diagnosis which would be a categorical variable. A method with versatility to address the two types of variables and address both regression and classification problems is needed. Moreover, there may be settings with many outcome variables to target for. The abovementioned complication of redundant predictor variables applies here; some outcome variables may not be relevant to the research question. It is also likely that certain outcome variables would not have strong links with the predictor variables in hand, rendering themselves difficult to predict given the available predictors. A method that can filter out these redundant outcome variables is necessary, as they may obscure the relevant predictive relationships concerning other outcome variables.

1.2 Method

A classical yet powerful tool that provides a good basis for overcoming these challenges is principal component analysis (PCA). It is a dimension reduction technique that compresses the large set of variables into a smaller set of summary variables known as principal components, in such a way that the amount of information compressed in the lower dimensions of principal components is maximized (Jolliffe, 1986). For a dataset $\mathbf{X}$ concerning I observation units and J variables, PCA with R (R is a pre-specified number smaller than I and J) components models the data by the following equations:

$$\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{P}^\top + \mathbf{E} \tag{1.1}$$

with $\mathbf{W}$ a $J \times R$ matrix of weights, $\mathbf{P}$ a $J \times R$ matrix of loadings and $\mathbf{E}$ a $I \times J$ matrix of residual variables. The weights dictate the combination of the variables into the principal components ($\mathbf{T} = \mathbf{X}\mathbf{W}$) with $\mathbf{T}$ denoting the $I \times R$ matrix of principal component scores. On the other hand, the loadings indicate the relationship between the components and the variables. The principal components are considered as representations of the processes measured by the variables, and the weights and the loadings are studied to interpret the principal components.

Extensions of PCA that address the challenges of multiblock data already exist. Nevertheless, they are purely designed for exploration of model structures underneath the data, without the focus on prediction of outcomes. To treat multicollinearity and the redundant variables problem, sparse PCA has been proposed by imposing regularization penalties on the coefficients (H. Shen & Huang, 2008; Zou, Hastie, & Tibshirani, 2006). The regularization penalties, first introduced for the method of linear regression, force certain coefficients that correspond to redundant variables to zero, which drops these variables out of the model (Tibshirani, 1996). This approach of regularization has been the workhorse for modelling with high-dimensional data, as it not only provides a concise model pertaining to a subset of essential variables, but also prevents the problem of multicollinearity and overfitting (McNeish, 2015). Sparse PCA borrows these advantages by regularizing the weights or the loadings; it allows correct identification of a subset of variables that have substantive links with the components.

In the same vein, extensions of PCA that explicitly aim to detach the common and distinctive processes from each other have also been devised. Simultaneous component analysis (SCA; Kiers & ten Berge, 1989) is a framework of such methods. SCA finds principal components from a supermatrix that concatenates the multiple data blocks concerning the same observations. By default, these components are derived in relation to all data blocks in the dataset. However, by introducing constraints, two types of components - *distinctive and common components* - can be identified, serving as representations for distinctive and common processes (see for example Schouteden, Van Deun, Pattyn, & Van Mechelen, 2013). Distinctive components are found such that they are only associated with a single data block, whereas common components are connected with multiple blocks. Moreover, these SCA approaches have been further adapted to also cater for the presence of collinear and redundant variables by inducing sparsity on the coefficients in the same manner as in sparse PCA (de Schipper & Van Deun, 2018; Gu & Van Deun, 2016). These extensions are able to attain interpretable common and distinctive components that are associated with a small subset of variables.

In setting up a model to predict an outcome, the component scores from PCA and its aforementioned extensions can be employed to fit a regression model, instead of using the original set with a large number of variables. This is a well-known approach known as principal component regression (PCR; Jolliffe, 1982). However, this two-step approach has the shortcoming that the prediction of the outcomes is not considered at the first step of finding the component scores; the components are derived with the only aim to summarize the variables in hand. Components that represent important predictive influences may therefore be omitted in this setting. For instance, in the nicotine addiction example, a component reflecting a brain activity with a crucial link to addiction may be missed due to the presence of other components that better summarize the brain imaging variables in hand.

Instead of this two-step approach of PCR, multivariate methods that provide summary variables in lower dimensions that also incorporate the prediction problem have been proposed and used widely in many disciplines including chemometrics (S. Wold, Sjöström, & Eriksson, 2001), multi-omics (Lê Cao, Rossouw, Robert-Granié, & Besse, 2008) and marketing (Hair, Ringle, & Sarstedt, 2011). Principal covariates regression (PCovR; De Jong & Kiers, 1992) belongs to this family of multivariate methods and it finds ‘principal covariates’ which account for a maximal amount of information in both the predictor and outcome variables. Let $\mathbf{X}$ and $\mathbf{Y}$ refer to J predictor variables and L outcome variables concerning I observations. PCovR uses the following equations to model the predictor and outcome variables simultaneously:

$$\begin{aligned}\mathbf{Y} =&\mathbf{XWP}^{(Y)^\top} + \mathbf{E}^{(Y)}\\ \mathbf{X} =&\mathbf{XWP}^{(X)^\top} + \mathbf{E}^{(X)}\end{aligned} \tag{1.2}$$

where the weights $\mathbf{W}$ in this setting dictate how the predictor variables are combined into the principal covariates. The first line concerns the model for the outcome variables, with $\mathbf{P}^{(Y)}$ a $L \times R$ matrix of regression coefficients and $\mathbf{E}^{(Y)}$ the residual matrix for the outcomes. On the other hand, the second line provides the model for the predictor variables, with $\mathbf{P}^{(X)}$ a $J \times R$ matrix of loadings and $\mathbf{E}^{(Y)}$ the residual matrix for the predictors. According to these model equations, principal covariates are considered to reflect the underlying processes composed of predictor variables that play a predictive role in the outcome variables. It is worth noting that the PCovR model is identical to the models employed by PCR or another popular multivariate tool called partial least squares (PLS; H. Wold, 1982; S. Wold, Martens, & Wold, 1983), although the estimated models differ across the three methods. More detailed comparisons between PCovR and these other multivariate methods can be found in the following chapters. Additionally, from the perspective of machine learning that distinguishes statistical methods into supervised and unsupervised methods, PCovR can be considered as being in between the two. It carries out the unsupervised task of finding a lower dimensional representation of the predictor variables, while also fulfilling the supervised task of predicting the outcome variables.

In light of the challenges of multiblock data, whereas a sparse extension of PCovR that tackles the problem of collinear and redundant variables has been devised (sparse PCovR; Van Deun, Crompvoets, & Ceulemans, 2018), developments that also disentangle the common and distinctive processes from each other have not yet been put forward. Namely, extensions analogous to those aforementioned in the PCA setting that accommodate multiblock data have not been proposed in the context of PCovR. To this end, this dissertation sets its focus at extending the PCovR method to overcome the challenges present within multiblock data. The methods will uncover common and distinctive predictive processes behind the outcome of interest, and concisely represent them by basing the components on a small set of important variables. Both continuous and categorical outcome variables will be considered by the methods, as well as the setting with multiple outcome variables. Owing to the intimate connection between PCA and PCovR, we have also conducted research on topics within sparse PCA, with an aim to seek out possible future directions for sparse PCovR research. These sparse PCA studies are also included in the dissertation given their direct relevance to the PCovR extensions. In the following, we provide the outline of the dissertation.

1.3 Outline of the dissertation

In Part I, we propose three extensions of PCovR for high-dimensional data from multiple sources. Chapter 2 provides a multiblock extension to sparse PCovR. By imposing a constraint that can force the entire set of coefficients corresponding to certain data blocks to zero (called the zero block constraint), the method is able to explicitly discern between common and distinctive covariates, which reflect the common and distinctive predictive processes. Further sparsity is induced by regularization penalties, as done in sparse PCovR (Van Deun et al., 2018). A simulation study was conducted to comparatively assess the method against similar methods with respect to the quality of prediction and the retrieval of the true processes underlying data. The method was found to be better at prediction than the two-stage PCR approach that first derives the common and distinctive components; the two-stage approach may omit an important process. At the same time, the zero block constraint and the penalties together were able to construct models that reflect the true underlying processes better than the PLS method designed for multiblock data. By employing an empirical dataset concerning different types of measurements made on potato samples, we demonstrated that the novel method has competitive performance for predicting the sensory experience while uncovering an interpretable model. This chapter was published in *Journal of Chemometrics*.

In Chapter 3, we expanded the multiblock sparse PCovR method to address a classification problem. By combining the method put forward in Chapter 2 with the generalized linear model framework, a logistic regression variant of multiblock sparse PCovR was devised. Moreover, instead of using a zero block constraint which entailed an excessive number of model estimations, a regularization penalty that forces the entire set of coefficients from a data block to zero was incorporated, reducing the computational load significantly. Two different types of penalties were therefore employed to identify common and distinctive covariates and also to impose sparsity. In a simulation study, we found that the method outperforms a classifier based on PLS in both classification and recovery of underlying processes. Practical value of our method was illustrated with an empirical dataset concerning questionnaires on three different members from the same family (mother, father and child). The method classified the families well, while constructing a concise model with common and distinctive covariates. This chapter was published in *Behavior Research Methods*.

Whereas Chapter 2 and 3 placed the stress in the variable selection problem of the predictors, we looked towards incorporating variable selection of outcome variables in Chapter 4. A sparse PCovR method that excludes unimportant variables at both ends of predictor and outcome variables was proposed therein. Being one of the first studies on the simultaneous selection of predictors and outcomes, it is a relevant direction as the problem characterized by the excess of unnecessary variables also pertains to outcome variables. In comparison to sparse PCovR and sparse PLS methods that do not explicitly remove redundant outcome variables, the novel method was reported to be better at outcome prediction in the simulation study. We adopted a dataset regarding a cold study that includes 16 symptoms associated with cold and flu as the outcomes and many other variables under various themes such as blood chemistry and health practices as predictors. By using the novel PCovR method, we were able to identify 10 symptoms that are relevant to being affected by the cold virus and a group of predictors that are important for predicting these symptoms. It was also found to have better prediction quality than the two competing methods under comparison. This chapter is currently under review.

In Part II, we looked into possible opportunities to further refine the PCovR methodology for large and multiblock data by taking a step back and visiting topcis within sparse PCA. Since sparse PCovR is rooted in sparse PCA, chapters in this part are of high relevance not only to the previous chapters but also to future developments of sparse PCovR. Chapter 5 sheds light on important issues that arise by imposing sparsity to the PCA problem. It is well-known that the solutions of non-sparse PCA are found by singular value decomposition (SVD). In this setting, weights, loadings and right singular vectors are equal to each other. However, they are no longer equal to each other in sparse PCA where the weights and loadings are made sparse. This loss of equality is often overlooked in the literature, leading to complications with respect to simulation studies and algorithm initialization strategies. We pointed out that commonly used setups for simulation studies are not comprehensive, and at times resulting in optimistic conclusions on sparse PCA methods. Also, the risk of only relying on the right singular vectors to initialize the algorithms was demonstrated by our simulation study. The consequences of choosing between weights or loadings to be sparse have been highlighted by the simulation study and by employing two empirical datasets. The relevance of these findings is not limited to sparse PCA, but also to extensions of PCovR presented in this dissertation and to sparse PLS, as these methods share the same structures for modelling the data. This chapter was published in *Behavior Research Methods*.

Lastly, Chapter 6 proposes a new algorithm for solving a sparse PCA problem. We considered a sparse PCA problem formulated with a maximization problem to optimize, which is different from the PCovR methods devised in Part I. An iterative thresholding algorithm was derived on the basis of the minorization-maximization numerical procedure. The algorithm has guarantees for optimality, meaning that the solution found by the algorithm is optimal with respect to the objective function. To our knowledge, it is the first work within this formulation of sparse PCA that proves local optimality. On top of the local optimality, as maximization problems of sparse PCA have been known to be more feasible for very large sets of data than minimization problems, this chapter implies a next promising direction for sparse PCovR extensions.

Part I

Extending Principal Covariates Regression

Chapter 2

Sparse Common and Distinctive Covariates Regression

Having large sets of predictors from multiple sources concerning the same observation units and the same criterion is becoming increasingly common in chemometrics. When analyzing such data, chemometricians often have multiple objectives: prediction of the criterion, variable selection, and identification of underlying processes associated to individual predictor sources or to several sources jointly. Existing methods offer solutions regarding the first two aims of uncovering the predictive mechanisms and relevant variables therein for a single block of predictor variables; but, the challenge of uncovering joint and distinctive predictive mechanisms and the relevant variables therein in the multisource setting still needs to be addressed. To this end, we present a multiblock extension of principal covariates regression which aims to find the complex mechanisms in which several or single sources may be involved; taken together, these mechanisms predict an outcome of interest. We call this method Sparse Common and Distinctive Covariates Regression (SCD-CovR). Through a simulation study, we demonstrate that SCD-CovR provides competitive solutions when compared with related methods. The method is also illustrated via an application to a publicly available dataset.

Keywords: Multiblock data, Principal covariates regression, Common and distinctive processes, Data integration, Variable selection

Park, S., Ceulemans, E. & Van Deun, K. (2020). Sparse common and distinctive covariates regression. *Journal of Chemometrics*, page e3270.

2.1 Introduction

When predicting an outcome by a number of predictor variables, there often is the additional aim to obtain insight in the mechanisms at play. For example, when modeling vaccine efficacy as a function of mRNA transcription rates soon after vaccination (Nakaya et al., 2011) setting up a prediction tool was not the only aim. The authors also wanted to understand the involved biological processes by finding - in the transcriptomics data - those biological pathways that are associated to the efficacy of the vaccine. To obtain an even deeper understanding of the system under study often large and heterogeneous collections of data are used that result in several blocks of predictors pertaining to the same observation units. A prominent example is multi-omics studies. These are used to obtain a better understanding of disease mechanisms by jointly studying several features of the biological system (e.g., genomic, transcriptomic, and proteomic data collected from the same sample of patients and controls; Hasin, Seldin, & Lusis, 2017. Obtaining insights from such large multiblock data implies revealing 1) the relevant features in the system, and 2) the orchestration of the system (which features act jointly and which ones act individually in shaping the outcome). For example, the emergence of asthma is known to depend on a complex interplay between genetic susceptibility and environmental exposure (Gallagher et al., 2011). A complicating factor in the analysis of the data, is that they often consist of large collections of untargeted variables which implies that it is the data analyst's task to sort out the relevant predictors from the variables that are irrelevant for the process under study. Moreover, such selection of variables is necessary to ease the interpretation of the resulting model and to address model inconsistency in the high-dimensional setting of (many) more variables than cases (Van Deun et al., 2019).

Within chemometrics, Partial Least Squares (PLS) and Principal Covariates Regression (PCovR) are popular methods which target the twofold goals of deriving the components that represent the underlying processes and predicting the criterion variables. Variants of the methods suited for multiblock data have been devised and shown to be useful at extracting insight about the mechanisms while predicting the criterion variable. Examples include incorporating information on physical properties of intermediate granules when modeling the relationship between process variables and crushing strength of finished tablets (Westerhuis & Coenegracht, 1997), predicting sensory attributes of carrot genotypes via finding joint mechanisms concerning dry matter content, non-volatile and volatile compounds (Kreutzmann, Svensson, Thybo, Bro, & Petersen, 2008), and mapping an interrelated model between consumer preference and sensory information such as odour and flavour pertaining to different flavoured water samples (Måge, Menichelli, & Næs, 2012). As these multiblock methods are subject to interpretational difficulties due to a large number of predictors, sparse PCovR (SPCovR) and sparse PLS (SPLS) were devised to provide solutions that perform variable selection (Lê Cao et al., 2008; Van Deun et al., 2018). Furthermore, viewing each block of predictors as representative of a part of the system under study, multiblock data may present two different types of underlying predictive mechanisms; those that pertain only to variables from a single predictor block and the mechanisms that require joint involvement of variables from multple predictor blocks. We denote the two types of mechanisms by distinctive and (partially) common mechanisms, respectively (with partially indicating mechanisms that pertain to variables from multiple though not all blocks). Identification of these mechanisms has not been fully addressed in the context of criterion prediction by the existing methods.

On the other hand, for purely explorative purposes (this is, only revealing underlying mechanisms without trying to predict a criterion), methods that specifically aim to capture common and distinctive processes have been put forward. Simultaneous component analysis (SCA) with distinctive and common components (DISCO-SCA), Joint and individual variation explained (JIVE) and similar other approaches aim to unravel the structure of the underlying processes by separating common and distinctive mechanisms (Lock, Hoadley, Marron, & Nobel, 2013; Schouteden et al., 2013). Måge, Smilde, and van der Kloet (2019) provided a comprehensive comparison of the performance of several of these approaches under varying data structures while Smilde et al. (2017) proposed a general framework for the methods devised to decompose multiblock data into common and distinctive processes. Moreover, to attain more interpretable solutions especially with high dimensional data, sparse methods have been developed that capture the common and distinctive processes by incorporating particular penalty terms or pre-specified structures (de Schipper & Van Deun, 2018; Gu & Van Deun, 2016; Van Deun, Wilderjans, Van den Berg, Antoniadis, & Van Mechelen, 2011).

Along these lines of research, a method is needed that serves the twofold goals of obtaining insightful predictive models in the setting of high dimensional multi-block data. As discussed, such a method should incorporate predictor selection and uncover the common and distinctive predictive mechanisms. The development of such a method could be envisaged both along the PLS and PCovR lines. Yet, in comparison to SPLS, SPCovR has been shown to be more effective in recovering the underlying processes (Van Deun et al., 2018) and it also offers more flexibility concerning the importance assigned to the dual aim of prediction of the criterion variable and the reconstruction of the predictor variables. Therefore, the current paper focuses on PCovR and integrates the sparse PCovR and SCA methods in the new sparse common and distinctive covariates regression method

(SCD-CovR). We evaluate the performance of SCD-CovR by comparing it with other methods that are characterized by similar goals such as sparse generalized canonical correlation analysis (SGCCA) which is based on PLS (A. Tenenhaus et al., 2014).

The paper is arranged as follows. First, we describe SCD-CovR in detail, followed by a brief overview of existing related methods. Then, simulation studies that comparatively demonstrate the performance of SCD-CovR and other methods are presented and their results are discussed. Finally, we conclude the paper by formulating some limitations and directions for future research. The implementation of SCD-CovR was done in R and it can be found on Github: `https://github.com/soogs/SCD-CovR`, along with the code used to generate the results reported in this paper.

2.2 SCD-CovR

We will use the following notation throughout the paper: scalars, vectors and matrices are denoted by italic lowercase, bold lowercase and bold uppercase letters respectively. Transposing is indicated by the superscript $^\top$. Lowercase subscripts running from 1 to corresponding uppercase letters denote indexing: $i \in \{1, 2, \ldots, I\}$. Subscript $_C$ indicates concatenation of multiple data blocks, while superscripts $^{(X)}$ and $^{(y)}$ highlight affiliation with predictor and criterion variables, respectively.

2.2.1 Model and objective function

SCD-CovR models a criterion in function of multiple blocks of predictors all obtained from the same set of observation units. Let $\mathbf{X}_k$ be a column-centered matrix containing the scores of the I observation units on the J_k predictors in the kth predictor block; with $k \in \{1, 2, \ldots, K\}$. Also, let $\mathbf{y}$ be a centered vector containing the I scores on the criterion.

The SCD-CovR model is based on the well-known PCA model which takes the following formulation for $\mathbf{X}_k$:

$$\mathbf{X}_k = \mathbf{X}_k \mathbf{W}_k (\mathbf{P}_k^{(X)})^\top + \mathbf{E}^{(X)} \tag{2.1}$$

where $\mathbf{W}_k$ and $\mathbf{P}_k^{(X)}$ are $J_k \times R$ matrices of component weights and loadings, respectively. To identify the solution, usually the constraint $(\mathbf{P}_k^{(X)})^\top \mathbf{P}_k^{(X)} = \mathbf{I}_R$ is added under a principal axes orientation. The weights define how the predictors are combined into the R principal components (namely, $\mathbf{T}_k = \mathbf{X}_k \mathbf{W}_k$, implying $t_{ir} = \sum_{j_k} x_{ij_k} w_{j_k r}$) while the loadings express the relationship between them. $\mathbf{E}^{(X)}$

is used to denote the matrix of residuals. This formulation is known as the weight-based model (Van Deun et al., 2011).

PCovR explicitly models the criterion as a function of the components in the PCA model (2.1):

$$\mathbf{y} = \mathbf{X}_k \mathbf{W}_k \mathbf{p}^{(y)} + \mathbf{e}^{(y)} \tag{2.2}$$

with $\mathbf{p}_k^{(y)}$ the vector of R regression coefficients and $\mathbf{e}^{(y)}$ the residuals pertaining to the criterion. The twofold aim of PCovR in reconstructing $\mathbf{X}_k$ and predicting $\mathbf{y}$ is expressed by the objective function to be minimized (De Jong & Kiers, 1992):

$$L(\mathbf{W}_k, \mathbf{P}_k^{(X)}, \mathbf{p}^{(y)}) = \alpha \frac{\left\|\mathbf{y} - \mathbf{X}_k \mathbf{W}_k \mathbf{p}^{(y)}\right\|_2^2}{\|\mathbf{y}\|_2^2} + (1-\alpha) \frac{\left\|\mathbf{X}_k - \mathbf{X}_k \mathbf{W}_k (\mathbf{P}_k^{(X)})^\top\right\|_2^2}{\|\mathbf{X}_k\|_2^2} \tag{2.3}$$

with $0 \leq \alpha \leq 1$ a known constant. The α parameter specifies the balance between modeling the criterion and modeling the block of predictors. With α set at 0, the method is identical to PCA followed by regression, while at 1 it becomes equivalent to linear regression (namely $\hat{y}_i = \sum_r \hat{p}_r^{(y)} \hat{t}_{ir} = \sum_r (\sum_{j_k} \hat{p}_r^{(y)} x_{ij_k} \hat{w}_{j_k r}) = \sum_{j_k} (\sum_r \hat{p}_r^{(y)} \hat{w}_{j_k r}) x_{ij_k}$, with $\sum_r \hat{p}_r^{(y)} \hat{w}_{j_k r}$ as a regression coefficient for the j_kth predictor). How to optimally balance α has been explicitly explored by Vervloet, Van Deun, Van den Noortgate, and Ceulemans (2013). Note that to identify the PCovR solution, De Jong and Kiers (1992) introduced the constraint $\mathbf{T}^\top \mathbf{T} = \mathbf{I}_R$. As pointed out by Vervloet et al. (2013), the solution is still subject to rotational freedom.

As PCA and PCovR construct the components by linearly combining all the predictors, the interpretation of the components can be difficult, especially when the number of predictors grows large. The solutions can also be inconsistent in the high-dimensional setup (Johnstone & Lu, 2009). To overcome these issues, Zou et al. (2006) devised a sparse PCA method that imposes regularization penalties on the objective function. Note that sparse implies that many of the component weights are penalized to become zero. A sparse variant of PCovR, SPCovR, was also developed in a similar manner (Van Deun et al., 2018). SPCovR finds the solutions by minimizing the following objective function:

$$\begin{aligned} L(\mathbf{W}_k, \mathbf{P}_k^{(X)}, \mathbf{p}^{(y)}) = \alpha \frac{\left\|\mathbf{y} - \mathbf{X}_k \mathbf{W}_k \mathbf{p}^{(y)}\right\|_2^2}{\|\mathbf{y}\|_2^2} + (1-\alpha) \frac{\left\|\mathbf{X}_k - \mathbf{X}_k \mathbf{W}_k (\mathbf{P}_k^{(X)})^\top\right\|_2^2}{\|\mathbf{X}_k\|_2^2} \\ + \lambda_L \left|\mathbf{W}_k\right|_1 + \lambda_R \left\|\mathbf{W}_k\right\|_2^2 \end{aligned} \tag{2.4}$$

such that $(\mathbf{P}_k^{(X)})^\top \mathbf{P}_k^{(X)} = \mathbf{I}_R$ and with $\lambda_L \geq 0$, $\lambda_R \geq 0$ and $\alpha \geq 0$. The regularization parameters are the lasso, with $|\mathbf{W}_k|_1 = \sum_{j_k,r} |w_{j_k r}|$, and the ridge $\|\mathbf{W}_k\|_2^2 = \sum_{j_k,r} w_{j_k r}^2$, together forming the elastic net (Zou & Hastie, 2005). The former shrinks and forces certain weights to be exactly zero, while the latter only shrinks the estimates. Therefore, the lasso penalty is employed to obtain sparse weights while the ridge penalty is required to ensure stable estimates under high-dimensionality. It can also be seen that when both of the tuning parameters λ_L and λ_R are 0, the PCovR formulation (2.3) is retrieved. Note that because of the penalties, the SPCovR model is identified and not subject to rotational freedom. However, the components pertain to permutational freedom and sign invariance.

SPCovR and the above methods only target data with a single predictor block and hence do not address the questions associated with multiple predictor blocks. These questions can be answered by performing a joint decomposition of the K predictor blocks into components by imposing a SCA model (Kiers & ten Berge, 1989):

$$\mathbf{X}_C = \mathbf{X}_C \mathbf{W}_C (\mathbf{P}_C^{(X)})^\top + \mathbf{E}^{(X)} \tag{2.5}$$

where $\mathbf{X}_C = [\mathbf{X}_1, \ldots, \mathbf{X}_K]$ (of size $I \times \sum_{k=1}^K J_k^{(X)}$) denotes the supermatrix that concatenates the predictor blocks. Consequently, $\mathbf{W}_C$ and $\mathbf{P}_C^{(X)}$ are weight and loading matrices of size $\sum_{k=1}^K J_k^{(X)} \times R$. Hence, the criterion variable can be modeled using the SCA weights:

$$\mathbf{y} = \mathbf{X}_C \mathbf{W}_C \mathbf{p}^{(y)} + \mathbf{e}^{(y)} \tag{2.6}$$

with $\mathbf{p}_C^{(y)}$ a vector of R regression coefficients.

As the interpretation of SCA solutions is even more challenging, sparse SCA methods were devised (Van Deun et al., 2011). Furthermore, a sparse SCA method that explicitly models common and distinctive processes was proposed. This method, sparse common and distinctive SCA (SCaDS), minimizes the following objective function (de Schipper & Van Deun, 2018):

$$L(\mathbf{W}_C, \mathbf{P}_C^{(X)}) = \left\| \mathbf{X}_C - \mathbf{X}_C \mathbf{W}_C (\mathbf{P}_C^{(X)})^\top \right\|_2^2 + \lambda_L |\mathbf{W}_C|_1 + \lambda_R \|\mathbf{W}_C\|_2^2 \tag{2.7}$$

such that $(\mathbf{P}_C^{(X)})^\top \mathbf{P}_C^{(X)} = \mathbf{I}_R$ and subject to *zero block constraints* on $\mathbf{W}_C$ that fix block-specific sets of weights - pertaining to one or several predictor blocks - to zero. This implies that the component is determined only by predictors of those blocks for which the weights have not been fixed to zero. Common components are obtained by not placing such zero block constraints on the component. The lasso penalty is used in addition to the zero block constraints to achieve sparseness within the common and distinctive components. As an alternative to using such a fixed structure, sparse multi-block PCA methods which rely on a group lasso penalty (which has the property to shrink entire groups of coefficients to zero) have also been proposed (Gu & Van Deun, 2016).

Building upon SCaDS and SPCovR, we propose the SCD-CovR that predicts the criterion, while providing sparse solutions that capture the common and distinctive processes in the predictor blocks. SCD-CovR implies minimizing the following objective function:

$$\begin{aligned} L(\mathbf{W}_C, \mathbf{P}_C^{(X)}, \mathbf{p}^{(y)}) = \alpha \frac{\left\|\mathbf{y} - \mathbf{X}_C \mathbf{W}_C \mathbf{p}^{(y)}\right\|_2^2}{\left\|\mathbf{y}\right\|_2^2} + (1-\alpha) \frac{\left\|\mathbf{X}_C - \mathbf{X}_C \mathbf{W}_C (\mathbf{P}_C^{(X)})^\top\right\|_2^2}{\left\|\mathbf{X}_C\right\|_2^2} \\ + \lambda_L \left|\mathbf{W}_C\right|_1 + \lambda_R \left\|\mathbf{W}_C\right\|_2^2 \end{aligned} \tag{2.8}$$

such that $(\mathbf{P}_C^{(X)})^\top \mathbf{P}_C^{(X)} = \mathbf{I}_R$, and subject to zero block constraints on $\mathbf{W}_C$.

As in SCaDS, common and distinctive components can be obtained with SCD-CovR through the zero block constraints on $\mathbf{W}_C$. Similarly as for SPCovR, the components both account for variation in the criterion *and* predictor variables with α allowing to flexibly tune prediction and reconstruction. The $\mathbf{W}_C$ weights can be examined to understand which predictors define the derived common and distinct components. It is also easy to see that this method is an adaptation of PCovR. When λ_L and $\lambda_R = 0$ are equal to zero and with the absence of the zero block constraints, the formulation is identical to PCovR.

2.2.2 Algorithm

To solve the optimization problem defined in (2.8) we use an alternating procedure where the loadings $\mathbf{P}_C^{(X)}$ and the regression coefficients $\mathbf{p}^{(y)}$ are solved for conditional upon fixed values for the weights $\mathbf{W}_C$ and vice versa. A schematic outline of the algorithm is given here below. The optimization procedure that we propose here closely follows those proposed for SCaDS and SPCovR (de Schipper & Van Deun, 2018; Van Deun et al., 2018). This procedure boils down to solving for all components together (unlike deflation methods that solve for each component in turn) and using a coordinate descent procedure to solve the conditional elastic net problem to estimate the sparse weights. More details on the procedure can be found in the Appendix. The alternating routine ensures that the loss is non-increasing and the algorithm converges to a stationary point, usually a local minimum. To avoid local minima problems, we recommend to use multiple random and a rational starting value based on PCovR.

Algorithm 2.1 SCD-CovR

```
Inputs:
    $\mathbf{X}_C$ and $\mathbf{y}$, number of components $R$, weighting parameter $\alpha$, regularization
    parameters $\lambda_L$ and $\lambda_R$, maximum number of iterations $T$, convergence threshold
    $\epsilon \geq 0$

Initialize:
    $\mathbf{W}_C \leftarrow \mathbf{W}_C{}^{(0)}$, $\mathbf{P}_C^{(X)} \leftarrow \mathbf{P}_C^{(X)}{}^{(0)}$, $\mathbf{p}^{(y)} \leftarrow \mathbf{p}^{(y)}{}^{(0)}$, $L_0 \leftarrow$ Initial loss,
    Loss difference $d \leftarrow 1$, Iteration counter $t \leftarrow 1$

while $t < T$ or $\epsilon < d$ do
    Conditional estimation of $\mathbf{P}_C^{(X)}{}^{(t)}$ and $\mathbf{p}^{(y)}{}^{(t)}$ given $\mathbf{W}_C{}^{(t-1)}$
    Conditional estimation of $\mathbf{W}_C{}^{(t)}$ given $\mathbf{P}_C^{(X)}{}^{(t)}$ and $\mathbf{p}^{(y)}{}^{(t)}$
    $L_u \leftarrow$ updated loss given $\mathbf{W}_C{}^{(t)}$, $\mathbf{P}_C^{(X)}{}^{(t)}$ and $\mathbf{p}^{(y)}{}^{(t)}$
    $d \leftarrow L_0 - L_u$
    $t \leftarrow t + 1$
    $L_0 \leftarrow L_u$
end while
```

2.2.3 Model selection

To use our proposed SCD-CovR method, values have to be provided for the number of components R, the weighting parameter α, the number of (partially) common and distinct components, and the ridge and lasso regularization parameters λ_L and λ_R. In order to select a suitable model, these parameters need to be tuned according to some optimality criterion. Several model selection strategies exist targeting different optimality criteria. These include cross-validation which is often recommended within the literature for methods involving regularization parameters. To optimize the optimality criterion, a grid search can be used that exhaustively compares all possible combinations of the tuning values for the different parameters. A sequential approach where each parameter is tuned in turn can also be considered as it was demonstrated to work well for cross-validation for PCovR (Vervloet, Van Deun, Van den Noortgate, & Ceulemans, 2016). As cross-validation is computationally costly if we consider all combinations of the tuning parameters, we therefore opt to use the sequential approach in the simulation study and the empirical application. The procedures are implemented slightly differently in these two sections because no oracle information is available for the empirical example. However, in general, the procedures first optimize R, λ_R and α simultaneously, followed by tuning the zero block constraints and λ_L. An interesting feature of the sparse PCA or PCovR methods with sparse weights instead of loadings is that the level of sparsity does not closely relate to the amount of variance explained; models comprised of components with very sparse weights can account for a comparable amount of variance as models that are much less or barely sparse (de Schipper & Van Deun, 2018). The weights are used to construct the component scores and these can be approximated very well with few non-zero weights. This even means that distinctive components can still account for a considerable amount of variance in the data block(s) for which the component has all zero weights.

2.2.4 Related methods

SCD-CovR is a method with three main objectives. It (a) predicts a criterion, (b) recovers the underlying common and distinctive predictor mechanisms via dimension reduction, and (c) derives sparse and therefore interpretable components. The method offers a solution that achieves all of these objectives in a balanced and a flexible manner. This section lists other component based methods devised to fulfill and balance these multiple objectives. When prediction is the only objective, methods with more emphasis on prediction may outperform SCD-CovR. In a similar vein, Smilde, Westerhuis, and Boque (2000) commented that PLS usually yields better prediction if the multiple blocks are analyzed as one single 'superblock'. Accounting for the multiblock structure helps in revealing meaningful insights but may come with lower prediction quality. On the other hand, applying a componentwise approach or explicitly taking into account the multi-block structure regularizes the problem. As such procedures safeguard against overfitting, they may improve the prediction quality especially in unstable settings (e.g., high dimensional data).

A method often used to aim both at prediction and modeling the variation in the block of predictors is principal component regression (PCR). This method first performs PCA and then, in a second and separate step, regresses the criterion on the components. The PCA step can be performed with SCaDS (leading to PCR-SCaDS) to also meet the objectives of finding common and distinctive mechamisms and having sparse component weights. It is closely related to SCD-CovR, as the components found by PCR-SCaDS are equal to the SCD-CovR components that we would obtain if we set the weighting parameter α to zero. Moreover, both methods encourage the recovery of the common and distinctive structure by imposing zero block constraints on the weights matrix. In comparison to SCD-CovR, PCR-SCaDS does not take the regression problem into consideration when deriving the components, implying that the processes that underlie the predictors would be retrieved with higher quality. However, simultaneously, PCR-SCaDS suffers from the weakness that predictor components that explain a lot of variance in the criterion may not be recovered (Vervloet et al., 2016).

SGCCA is another component-based method that addresses the multiple goals of simultaneous prediction and modeling the variation in the predictors. Being an extension of PLS, multiple data blocks are analyzed simultaneously to obtain sparse components while at the same time these components should account for the variation in the criterion (A. Tenenhaus et al., 2014). Extracting components that also allow to predict well is similar to SCD-CovR but unlike PCR-SCaDS. However, while SCD-CovR provides a flexible framework to weight reconstruction of the predictors and prediction of the criterion, PLS-based methods tend to lean closer to prediction (Van Deun et al., 2018; Vervloet et al., 2016). This also means that SGCCA may have more difficulties in recovering the underlying processes. Furthermore, methods based on PLS are often more prone to overfitting than those derived from PCovR, which in turn results in a diminished quality of out-of-sample prediction. Finally, SGCCA does not explicitly facilitate the retrieval of common and distinctive processes.

On top of these two methods, SPCovR can also be considered closely related to SCD-CovR. Their only difference is the zero block constraints on the weights for finding the common and distinctive structure. The two methods are expected to yield similar performance with respect to prediction. However, SCD-CovR can be expected to be better at capturing the common and distinctive underlying processes and thus in giving insight into joint and distinctive mechanisms.

Summarizing, the four methods can be expected to perform differently in terms of prediction and recovering the underlying components when administered to the same data. Concerning prediction, PCR-SCaDS is expected to underperform because it would be unable to capture an underlying process that is strongly associated to the criterion but accounts only for a minor portion of the variation in the predictor variables. We anticipate SGCCA to be more prone to overfitting than the other methods. Regarding correct recovery of the component weights, SGCCA would be relatively worse than the other methods due to its stronger focus on the prediction. Lastly, SCD-CovR and PCR-SCaDS are expected to recover the underlying common and distinctive processes more effectively than the other methods as they specifically target these processes through the zero block constraints.

2.3 Simulation study

Although adaptations of PLS, PCR and PCovR have been compared in previous research (Van Deun et al., 2018; Vervloet et al., 2016), they have not been put to test in settings where underlying common and distinctive processes are expected. Also, the effectiveness of the methods may depend on certain data characteristics. Therefore, we have conducted a simulation study in which we examine the performance of the methods with respect to sparse retrieval of the underlying processes, identification of common and distinctive components, and the prediction of the criterion.

2.3.1 Design and procedure

Fixing the number of observations I to 100, two blocks of predictor variables were generated to represent 3 components with a common and distinctive structure. Two components represented processes distinctive to predictor block 1 and 2, respectively. The remaining component reflects a common process involving both of the blocks. We defined the three components such that one of them explains 50% of the true structural variance in the predictors, another one 40% and the remaining one 10%. Adopting the terminology from Vervloet et al. (2016), we refer to the first two components as 'strong' components and call the third one a 'weak' component. On the other hand, the three components also differ in 'relevance' for predicting the criterion, in that one of them explains 66.7% of the true criterion variance and the other two 16.67% each. Finally, 70% of the weights and the loadings were made sparse.

We manipulated five data characteristics which are listed in the overview below. Each level within the manipulated factors is provided in square brackets. For the second and third factor which concern the strength and the relevance of the components, the proportion of variance explained is provided in the following order: [component distinctive to block 1, component distinctive to block 2, and common component].

Study setup

1. Number of predictors J_k in each block: [100], [10]
2. Strength of the three components: [50%, 40%, 10%], [10%, 40%, 50%]; in the first case the common component is weak and in the second case the first distinctive component is.
3. Relevance of the three components: [16.67%, 16.67%, 66.67%], [66.67%, 16.67%, 16.67%]; in the first case the common component is the most relevant and in the second the first distinctive is.
4. Proportion of error in $\mathbf{X}_C$: [10%], [50%]
5. Proportion of error in $\mathbf{y}$: [10%], [50%]

To obtain two predictor blocks that correspond to the settings described above, the following procedure was followed. The true predictor matrix $\mathbf{X}_C^*$ is defined by the model $\mathbf{X}_C^* = \mathbf{X}_C^*\mathbf{W}_C(\mathbf{P}_C^{(X)})^\top T$ where the weights and the loadings are equal and column-orthogonal: $\mathbf{W}_C = \mathbf{P}_C^{(X)}$. First a random column-centered matrix $\mathbf{T}^*$ of size $I \times R$ was generated from a multivariate normal distribution with the identity matrix as covariance matrix. Subsequently, $\mathbf{T}^*$ was centered and column-orthogonalized to yield $\mathbf{T}$. Second, to obtain a sparse and orthogonal weights matrix, we started by generating a random weights matrix of $\mathbf{W}_C^*$ of size $\sum_k J_k \times R$ from a uniform distribution over the interval of [0, 1]. To create one distinctive component for each of the two predictor blocks, the weights of the predictors on this component were set to zero in the other block. In the remaining non-zero parts, randomly chosen elements were replaced by zeros to attain a sparsity level of 70% when computed across the full matrix. The resulting matrix was orthogonalized using a Gram-Schmidt procedure in a manner that the sparse elements are retained to yield the true weights matrix $\mathbf{W}_C$. Furthermore, a diagonal matrix $\mathbf{D}$ was created with the diagonal values representing the relative proportion of variance accounted for by the components (i.e., reflecting their strength). Since $\mathbf{W}_C = \mathbf{P}_C^{(X)}$, the true predictor matrix $\mathbf{X}_C^*$ was then obtained as $\mathbf{X}_C{}^* = \mathbf{T}\mathbf{D}(\mathbf{P}_C^{(X)})^\top = \mathbf{X}_C^*\mathbf{W}_C(\mathbf{P}_C^{(X)})^\top$. Finally, residuals were added generated from a standard normal distribution and scaled such that the predictor blocks contain the desired level of error to yield $\mathbf{X}_C$. The proportion of error is defined as the proportion of total variance in the observed $\mathbf{X}_C$ or $\mathbf{y}$ that is due to error. The scores on the criterion variable were obtained in a similar fashion with the equation $\mathbf{y} = \mathbf{T}\mathbf{D}\mathbf{p}^{(y)} + \mathbf{e}^{(y)} = \mathbf{X}_C^*\mathbf{W}_C\mathbf{p}^{(y)} + \mathbf{e}^{(y)}$. To specify the regression coefficients $\mathbf{p}^{(y)}$, we first fixed the coefficient pertaining to the second component to -0.3. This second component is constantly irrelevant across the conditions. The other two coefficients were specified according to the different levels of strength and relevance.

Fully crossing the conditions and generating 50 replicate datasets per condition, $2 \times 2 \times 2 \times 2 \times 2 \times 50 = 1600$ datasets were produced. Each of the 1600 datasets was subjected to eight different analyses. The different analysis methods resulted from crossing the following four methods with two different numbers of extracted components.

Analysis methods

1. Method: [SCD-CovR], [SPCovR], [PCR-SCaDS], [SGCCA]
2. Number of components extracted: [2], [3]

Although a 3-component model was used for data generation, we varied the extracted number of components because we aim to understand the behaviour and the performance of different methods at identifying the components. When methods extract two components from data generated using a 3-component model, methods can focus on different aspects and thus yield different subsets of components. As the relevance and the strength of the three components are manipulated across the conditions, we can observe how both aspects determine which two components are extracted. For example, as mentioned in Section 2.2.4, we expect PCR-SCaDS to recover the strong components rather than the relevant components.

2.3.2 Model selection

The number of components R extracted for all four methods is fixed by the study design. A few other tuning parameters were fixed such as to correspond to the true model structure. Suitable values for the tuning parameters that were not fixed were found sequentially, for each data set and each analysis method.

For SCD-CovR, using the given R, we first simultaneously tuned the weighting parameter α and the ridge penalty λ_R via 10-fold cross-validation, keeping the lasso penalty λ_L at 0 (which therefore does not induce any sparsity) and the zero block constraints such that no distinctive components are imposed. We adopted the 1 standard error (SE) rule to select a parameter which yields the most general model among the set of parameters with errors within 1 SE from the minimal cross validation error. Usually, generality of models indicate that the model is unsaturated and thus easy to interpret and unlikely to overfit. Since higher α values place more emphasis on criterion prediction and therefore lead to a model more prone to overfitting, we chose the lowest α value via the 1 SE rule. Second, a suitable common and distinctive component structure was determined. When extracting 2 components, the zero block constraints on $\mathbf{W}_C$ that provide the structure of the common and distinctive components was chosen through 10-fold cross-validation. We selected the common and distinctive structure of the two components which led to the smallest cross validation error. The 1 SE rule was not used since it is difficult to define what a general model is with regards to the common and distinctive structure. On the other hand, for retrieving three components, the defined true structure was provided. The lasso parameter was tuned by selecting the value that results in the correct number of zero component weights.

For SPCovR, the set of tuning parameters is the same as SCD-CovR except for the zero block constraints. As α and λ_R were selected without any zero block constraints for SCD-CovR, these values were adopted for SPCovR (note that when the zero block constraints do not impose distinctive components, SCD-CovR is equivalent to SPCovR). Also here λ_L was tuned to return the correct number of zero coefficients.

For PCR-SCaDS the number of common and distinctive components as well as λ_R and λ_L need to be determined. We started the sequential approach by performing 10-fold cross-validation with the 1 SE rule for determining λ_R. Next, the zero block constraints and λ_L were found as previously discussed for SCD-CovR.

Finally, for SGCCA, the λ_L tuning parameter was fixed to yield the same number of zero-coefficients as in the generated data. The ridge penalty in SGCCA was tuned using the default setting the package provides.

2.3.3 Evaluation criteria

The four considered methods serve multiple aims: predicting a criterion, capturing possible common and distinctive underlying processes and providing sparse solutions for better interpretation. To assess the effectiveness of the methods at meeting these aims, we employed two evaluation criteria.

1. Out-of-sample R^2: equivalent to the R^2 measure for OLS, but applied for an independent out-of-sample test set.
2. Correct classification rate: proportion of $\mathbf{W}_C$ coefficients correctly classified as zero and non-zero elements relative to the total number of coefficients.

The independent test set (of 100 observation units) needed for computing the out-of-sample R^2 was generated following the same underlying model and the procedures as the data used for estimation. The out-of-sample R^2 measure is computed by the following equation.

$$R^2{}_{\textit{out-of-sample}} = 1 - \frac{\|\mathbf{y}_{\textit{test}} - \hat{\mathbf{y}}_{\textit{test}}\|_2^2}{\|\mathbf{y}_{\textit{test}}\|_2^2} \tag{2.9}$$

where $\mathbf{y}_{\textit{test}}$ refers to the $\mathbf{y}$ scores from an out-of-sample test set and $\hat{\mathbf{y}}_{test}$ indicates the predicted score that corresponds to $\mathbf{y}_{\textit{test}}$. Therefore, $\frac{\|\mathbf{y}_{\textit{test}} - \hat{\mathbf{y}}_{test}\|_2^2}{\|\mathbf{y}_{\textit{test}}\|_2^2}$ refers to the scaled sum of squared prediction error. Since this scaled sum of prediction error can be larger than one, it is possible for the out-of-sample R^2 to take a negative value. The correct classification rate is computed by comparing the true and the estimated $\mathbf{W}_C$ weights matrices. To handle the permutational freedom and the sign invariance of the estimated components, we calculated Tucker congruence between the columns of the true $\mathbf{W}_C$ matrix and those of the estimated $\mathbf{W}_C$ matrix. After pairing the true and estimated $\mathbf{W}_C$ columns that resulted in the highest Tucker congruence, the correct classification rate is calculated from the matching pairs of true and estimated $\mathbf{W}_C$ columns. This strategy was also used when only two components were extracted: they were matched to those two components of the three true ones that yield the highest Tucker congruence.

2.3.4 Results

2.3.4.1 Out-of-sample R^2

First we consider the performance of the four methods in terms of how well they predict new data. The results are summarized in Figures 2.1, 2.2, 2.3, and 2.4. The first two of these refer to the results obtained when extracting two components only, while the latter two refer to the analyses with three extracted components.

The aggregated results over all conditions, for the analyses with two extracted components, can be found in Figure 2.1. It can be observed that on average PCR-SCaDS has smaller out-of-sample R^2 than the other three methods. The latter show similar performance among each other. To examine whether there is an effect of the design factors and of the used method on out-of-sample R^2, we studied how the out-of-sample R^2 changes according to each of the conditions in the design by observing the boxplots.

Figure 2.2 presents these boxplots of out-of-sample R^2 arranged for each condition, conveying that the proportion of error variance in $\mathbf{y}$ plays an influential role in the performance of the methods. In the conditions where the error variance in $\mathbf{y}$ equals 10%, the four methods have comparable levels of prediction performance in those situations where the strong component is relevant for prediction (the two columns in the middle). In contrast, when the component relevant for prediction is a weak one, the out-of-sample R^2 of PCR-SCaDS decreases considerably. On the other hand, although this trend of underperformance of PCR-SCaDS can also be found in the 50% error on $\mathbf{y}$ conditions, it is not as pronounced.

SGCCA is more sensitive to whether the relevant component is strong or weak: When a strong component is relevant, the method has comparable out-of-sample R^2 with the other three methods. However, for datasets where the weak component is relevant, SGCCA outperforms the other methods. SCD-CovR and SPCovR outperform PCR-SCaDS with respect to prediction in all conditions; they perform similar to or a bit better than SGCCA in terms of prediction when the strong component is also the relevant one but SGCCA has better predictive performance when the relevant component is a weak component. The underperformance of PCR-SCaDS is not a surprising outcome because it only considers the predictor variables in constructing the components. Therefore, the variance explained in $\mathbf{y}$ by a weak but relevant component is not effectively captured by the method, since it extracts the two strong though irrelevant components.

Figure 2.3 summarizes the out-of-sample R^2 obtained when each of the methods extracted three components. SGCCA appears to stand out with a slightly lower out-of-sample R^2 on average, while the other three methods show very sim-

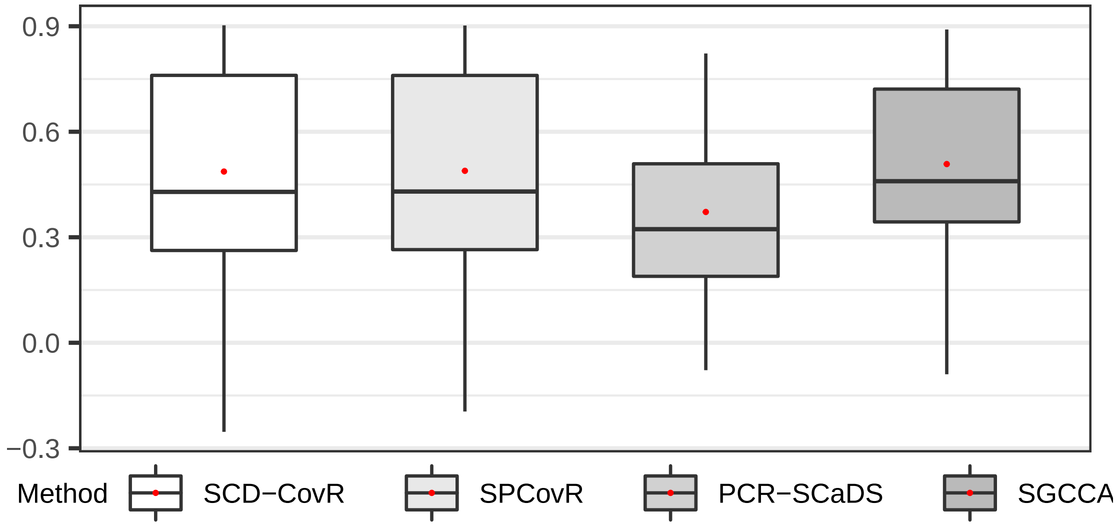


Figure 2.1. Box plots of the out-of-sample R^2 when two components are extracted: Aggregated results. The red dot indicates the mean.

ilar performance. Figure 2.4 shows the results laid out in function of the factors.

In most of the conditions in Figure 2.4 we can observe the trend conveyed in Figure 2.3: SGCCA shows a lower level of out-of-sample R^2 while the other three methods perform comparably. The underperformance of SGCCA is clearer in the conditions in which the proportion of error in $\mathbf{y}$ is 50%. This result can be attributed to overfitting: For these conditions where SGCCA showed low levels of R^2, the residuals (in-sample errors) were considerably smaller than the prediction error computed with the out-of-sample observation of $\mathbf{y}$. On the other hand, the two different types of errors were comparable for the three other methods. In contrast to Figure 2.2 with 2-component models, the prediction quality of PCR-SCaDS is similar with the one shown by SCD-CovR and SPCovR. This is reasonable, as in this setup where all three underlying components are extracted, PCR-SCaDS is able to extract the relevant but weak component.

In conclusion, the results for the out-of-sample R^2 show that SCD-CovR yields a relatively high quality of prediction. When two components are extracted, it outperforms PCR-SCaDS while with three extracted components, the method results greater R^2 than SGCCA. Additionally, the performance of SPCovR is comparable to that of SCD-CovR. It should be noted, however, that when not all components are extracted and there is a weak component that is relevant for prediction, than SGCCA is the prefered method in terms of prediction.

2.3.4.2 Correct classification rate

Figure 2.5 and 2.6 present the results of the correct classification rate. In Figure 2.5 which pertains to the analyses with two extracted components, PCR-SCaDS

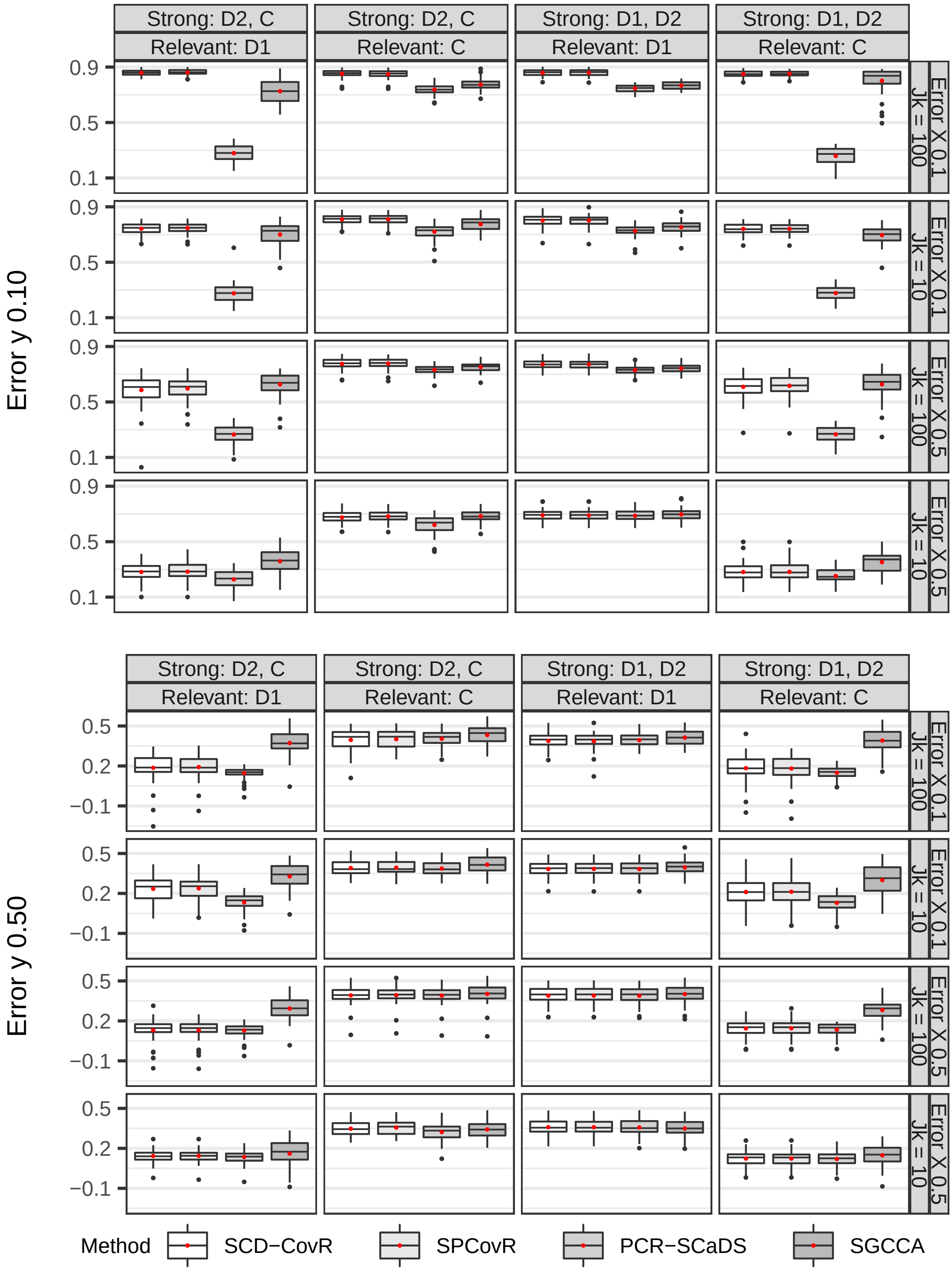


Figure 2.2. Box plots of the out-of-sample R^2 when two components are extracted; each panel corresponds to one of the 16 conditions. The column panels indicate the manipulated strength and relevance of the 3 components; D1 and D2 denote the components distinctive to block 1 and 2 respectively, while C refers to the common component. The row panels indicate the number of variables J_k in each predictor block.

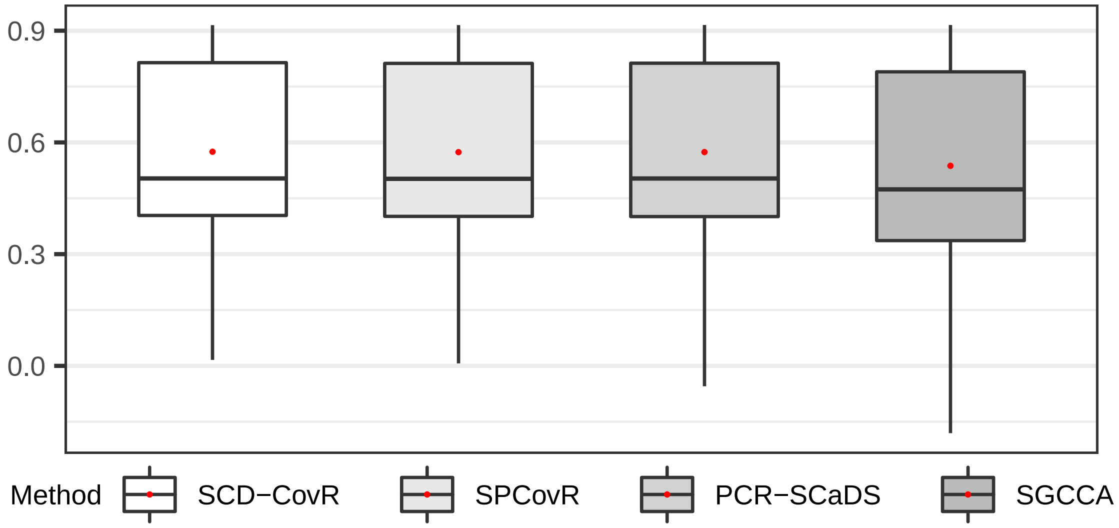


Figure 2.3. Box plots of the out-of-sample R^2 when three components are extracted: Aggregated results. The red dot indicates the mean.

yields the highest rate of weights correctly classified as zero or non-zero, closely followed by SCD-CovR and SPCovR. SGCCA has a considerably lower correct classification rate. SCD-CovR, SPCovR and PCR-SCaDS again show comparable and high correct classification rates also when three components were extracted (Figure 2.6), where SGCCA underperforms again. This general trend seen in Figures 2.5 and 2.6 is largely consistent across conditions.

The outperformance of PCR-SCaDS and SCD-CovR is sensible. On top of the lasso penalty which induces sparsity, these methods also constrain the weights such that an entire set of weights belonging to a predictor block are made sparse. When three components are extracted, the oracle information of the common and distinctive component structure is provided which further eases the correct classification. In contrast, SPCovR and SGCCA do not explicitly cater for capturing common and distinctive processes and thus are expected to show a diminished rate of correct classification. However, SPCovR resulted in a very similar level of performance as SCD-CovR and this can be attributed to the usage of rational starting values based on PCovR. Since the predictor variables were generated with an underlying true unrotated structure of PCA, initializing the convergence with PCovR solutions helps SPCovR in correctly retrieving the weights.

To conclude, the results from the correct classification rate suggest that SCD-CovR and SPCovR return weights that are of similar quality as those obtained with PCR-SCaDS which emphasizes the recovery of the weights more.

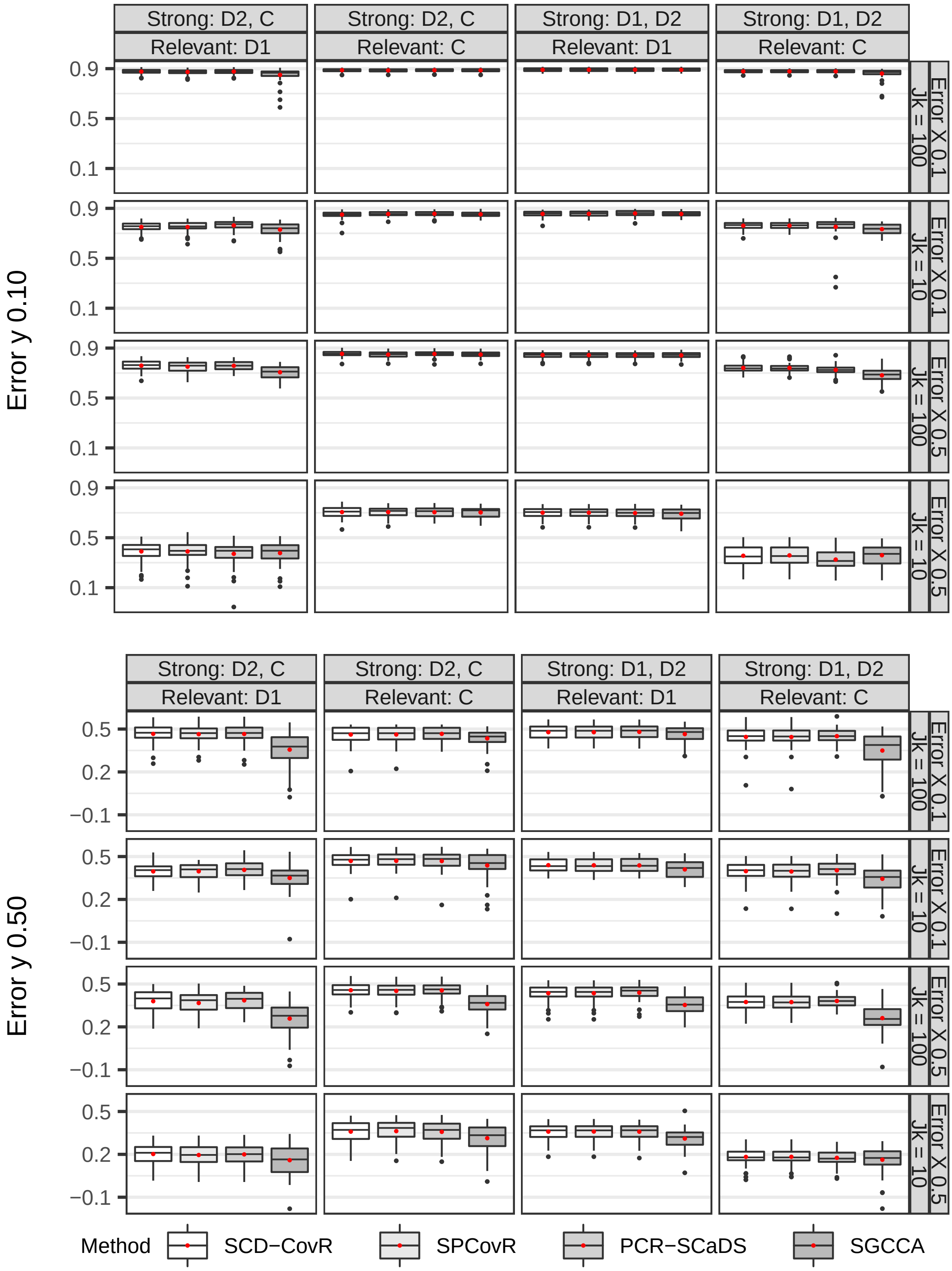


Figure 2.4. Box plots of the out-of-sample R^2 when three components are extracted; each panel corresponds to one of the 16 conditions. The column panels indicate the manipulated strength and relevance of the 3 components; D1 and D2 denote the components distinctive to block 1 and 2 respectively, while C refers to the common component. The row panels indicate the number of variables J_k in each predictor block.

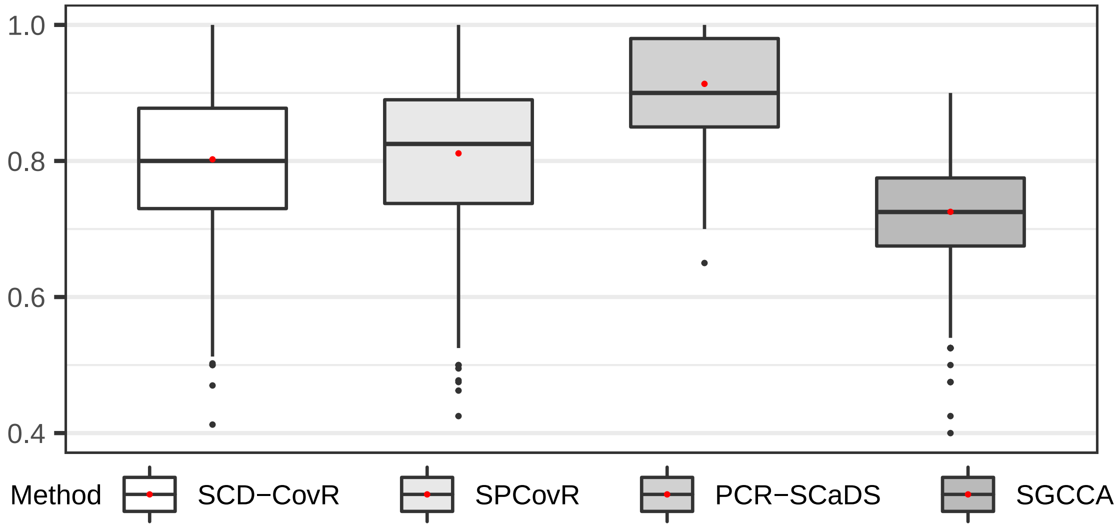


Figure 2.5. Box plots of the correct classification rate when two components are extracted: Aggregated results. The red dot indicates the mean.

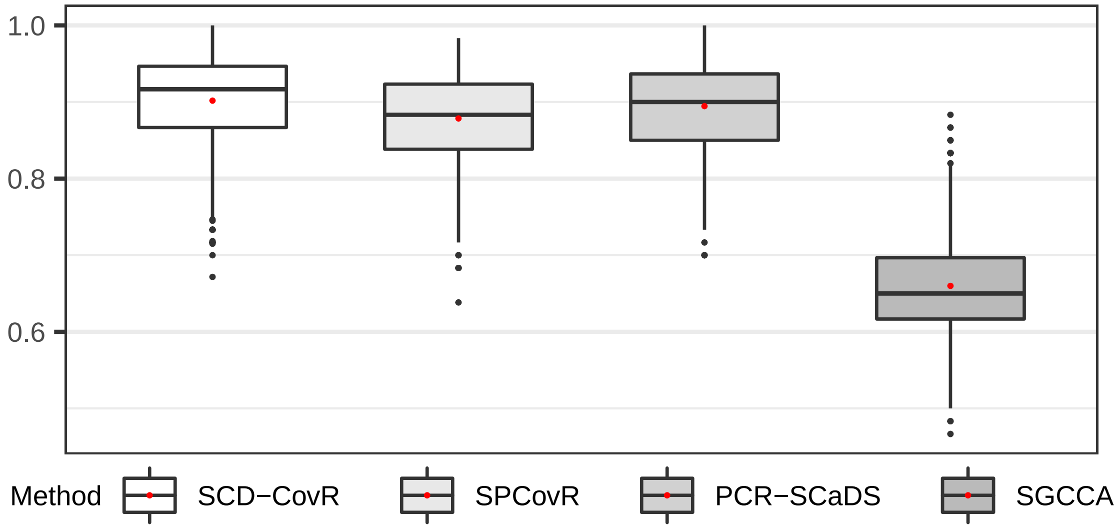


Figure 2.6. Box plots of the correct classification rate when three components are extracted: Aggregated results. The red dot indicates the mean.

2.3.4.3 Capturing common and distinctive components

On top of the prediction quality and the correct retrieval of sparse weights, SCD-CovR also targets another objective, namely to capture common and distinctive predictive processes. For each of the 1600 simulated datasets that the methods were administered to, we counted the number of common and distinctive components found by the methods. Regardless of the presence of zero block constraints, a column of the estimated $\mathbf{W}_C$ matrix that contains only zeroes for a predictor block is considered a distinctive component. Otherwise, when non-zero weights are found for both blocks, the component is a common component. For instances where an entire column is zero, the corresponding component is identified as neither common nor distinctive. Table 2.1 provides the numbers of these components (note that we generated all of the replicate datasets by a 3-component model with two components distinctive to each predictor block and one common component).

Concerning analyses with two components where the zero block constraints are selected via cross validation for SCD-CovR and PCR-SCaDS, it can be seen that almost all of the components found by PCR-SCaDS were distinctive. SCD-CovR identified about 41% of the estimated components as distinctive components. SPCovR and SGCCA which do not impose an explicit constraint for the distinctive components mostly identified common components, naturally. With respect to the 3-component models, SCD-CovR and PCR-SCaDS with the oracle information on the common and distinctive structure returned the components reflecting the structure effectively. However, it can be seen that SCD-CovR provided a few more distinctive components than defined. These are instances where the lasso penalty sparsifies the weights corresponding to an entire predictor block, while the respective component is a common component. Although SPCovR and SGCCA do not provide sufficient numbers of distinctive components, SPCovR derived a lot more of those than in the 2-component setting. Interestingly, the number of components did not appear to influence the effectiveness of SGCCA in capturing the common and distinctive components. Also, a component distinctive to the first predictor block was found much more frequently than the other distinctive component by SGCCA.

These numbers of retrieved common and distinctive components suggest that SCD-CovR is as effective as PCR-SCaDS with heavy emphasis on reconstructing the predictors when the correct common and distinctive structure is given. SPCovR which has similar performance with SCD-CovR at correct classification of the weights falls short at providing enough distinctive components, when the correct number of 3 components are used. This implies in practice that far more components extracted by SPCovR would be interpreted as a common process rather than a distinctive one, than the components derived using SCD-CovR. Evaluating the performance of the methods under 2-component model is less straightforward than 3-component model, because now the methods have to summarize the structural variation governed by three true components by estimating only two. Methods can choose certain favourable components or may create composite components which combine multiple true components. In such cases, simply deriving more distinctive components perhaps does not directly link to outperformance. Although 50% of the replicate datasets were characterized by the common component being a strong component, PCR-SCaDS extracted only distinctive components. This indicates the method's strong inclination towards finding distinctive components. At the same time, while the other 50% of the datasets did not feature the common component being strong, a vast majority of the components retrieved by SPCovR and SGCCA were common components. This implies that these two methods favor common components. In contrast, 59% of the components retrieved by SCD-CovR were common components, and this appears to address the true component structure better than the other methods. To conclude, our results from 2-component models suggest that SCD-CovR is more capable than the other methods in finding an adequate balance between common and distinctive components in reflecting the underlying component structure.

Table 2.1. Number of common and distinctive components considering the weights matrix. D1 and D2 indicate components distinctive to block 1 and 2, respectively and C refers to a common component. There were 1600 replicate datasets, thus the total numbers of estimated components for the analyses with two and three components were 3200 and 4800, respectively.

	SCD-CovR	SPCovR	PCR-SCaDS	SGCCA
2-component model				
D1	666	101	1596	197
D2	641	138	1599	9
C	1893	2961	0	2994
3-component model				
D1	1636	643	1601	200
D2	1601	840	1601	8
C	1563	3317	1595	4592

2.4 Illustrative application

In this section we illustrate SCD-CovR by applying it to an empirical dataset. We also compare with results that are obtained with the related methods to examine the practical effectiveness of SCD-CovR.

2.4.1 Dataset and pre-processing

We analyzed a dataset originally from Thybo, Bechmann, Martens, and Engelsen (2000) regarding texture measurements of potatoes. The dataset consists of 20 potato samples that were analyzed using three measurement platforms: chemical analysis, uniaxial compression and sensory analysis. The chemical analysis block contains 14 variables regarding chemical aspects of the potatoes, such as the chemical composition. The uniaxial compression block with 36 variables provides measurements obtained from administering uniaxial compression at 6 deformation rates on cooked potato samples. The sensory analysis block is comprised of 9 sensory variables reported by trained experts. Here, we conduct SCD-CovR with the aim to predict the sensory experience, while also exploring the underlying common and distinctive predictive processes in the chemical and uniaxial compression blocks.

To this end, we constructed a univariate criterion from the sensory analysis data block by extracting the first principal component. All variables were first centered and scaled to unit sum of squares. Next, in order to account for the differing size of the two predictor data blocks, we scaled these blocks so that the sum of squares of each data block is equal. We administered SCD-CovR along with the three related methods employed in the simulation study to assess the performance of the methods when being applied to an empirical dataset.

2.4.2 Model selection

The model selection strategy for this empirical dataset was largely in line with the strategy used in the simulation study, applying the same tuning sequence. However, the true number of components as well as their status (common, distinctive for block one or two) and the level of sparseness were unknown in this setting. We found the number of components through a residual test where we observe the change of sum of squared residuals $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ (where $\mathbf{y}$ and $\hat{\mathbf{y}}$ indicate the observed criterion and the fitted values respectively) while increasing the number of components. For the test, we fixed the ridge and lasso penalties λ_R and λ_L to 0.01 (to account for high dimensionality) and 0 respectively. As the common and distinctive structures of the model may interact with the number of components needed, we included all the possible combinations of the common and distinctive components in the residual test. Concerning the weighting parameter α, we used the maximum likelihood approach discussed in Vervloet et al. (2013). The following formula was used:

$$\alpha_{ML} = 1 - \frac{\|\mathbf{X}_C\|_2^2}{\|\mathbf{X}_C\|_2^2 + \|\mathbf{y}\|_2^2 \dfrac{\sigma^2_{\mathbf{E}^{(X)}}}{\sigma^2_{\mathbf{e}^{(y)}}}} \tag{2.10}$$

where $\sigma^2_{\mathbf{E}^{(X)}}$ and $\sigma^2_{\mathbf{e}^{(y)}}$ refer to the error variances to be estimated (see Vervloet, Kiers, Van den Noortgate, and Ceulemans (2015) for details). The results from the residual test are shown in Figure 2.9. Within each number of components, models comprised mostly of distinctive components resulted in larger sums of residuals. However, when observing the overall trend, the sum of squared residuals decreases sharply at three components independently of the common and distinctive structures. The sum of residuals then stabilizes with subsequent numbers of components. The residual test using the aforementioned tuning parameters therefore resulted in the choice of three components. In order to make the method comparison fair, we also used three components when applying the other methods.

Given this number of components, we used the same model selection procedure as in the simulation study. This procedure consists of conducting cross-validation for α and λ_R simultaneously, followed by cross-validation for the zero block constraints. Both procedures employed 10 folds. The 1 SE rule was adopted for α and λ_R but not for the zero block constraints. Out of the three different configurations of zero block constraints which resulted in similar levels of cross validation error, (D1,D2,C), (C,C,C) and (D2,C,C), the configuration with the smallest error, (D1, D2, C) was selected (Figure 2.10). We acknowledge that it is hard to tell which of these three structures is the true underlying common and distinctive structure, however. Since the oracle level of sparsity is unavailable for this empirical example, λ_L was determined through 10-fold cross-validation with the 1 SE rule as well. The plots that depict the cross-validation errors and the corresponding standard errors can be found in Appendix 2.C.

With regards to SPCovR, we adopted the same number of components, α and λ_R as used for SCD-CovR. The lasso penalty λ_L was chosen through 10-fold cross-validation with the 1 SE rule. For PCR-SCaDS, the procedures from the simulation study were taken. λ_L was determined through 10-fold cross-validation with the 1 SE rule. Lastly, SGCCA only needs tuning of the lasso penalty governing the level of sparsity, this penalty was tuned via 10-fold cross-validation with the 1 SE rule as well. The plots in Appendix 2.B can be consulted for the cross-validation results.

Table 2.2. Tuning parameters and R^2 per method. 'Block' refers to the zero block constraints

	R	α	Ridge	Lasso	Block	R^2
SCD-CovR	3	0.7	0.005	3.579	C, D1, D2	0.981
SPCovR	3	0.7	0.005	5.477	NA	0.933
PCR-SCaDS	3	NA	0.001	0.011	D1, D2, D2	0.663
SGCCA	3	NA	NA	0.277	NA	0.954

2.4.3 Results

The four methods were administered with the tuning parameters in Table 2.2. The table also provides the R^2 values of each method, calculated by $R^2 = 1 - (\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 \,/\, \|\mathbf{y}\|_2^2)$ where $\mathbf{y}$ and $\hat{\mathbf{y}}$ indicate the observed criterion and the fitted values respectively.

The R^2 values are very high, except for PCR-SCaDS. In order to also test for out-of-sample prediction quality, we conducted 10-fold cross-validation. The results can be seen in Figure 2.7 and are in agreement with those found in our simulation study. SCD-CovR and SPCovR produced less cross-validation errors than PCR-SCaDS and SGCCA; cross-validation error is comparable to prediction error. Inspecting the weights matrix produced by the two outperforming methods, we found that SPCovR produced two common components and one component distinctive to the chemical block, while SCD-CovR found one common component and one distinctive component for each predictor block. It is difficult to determine which of the both solutions is more interpretable, but this finding indicates that SCD-CovR is capable of capturing more distinctive components than SPCovR while providing competitive quality in prediction.

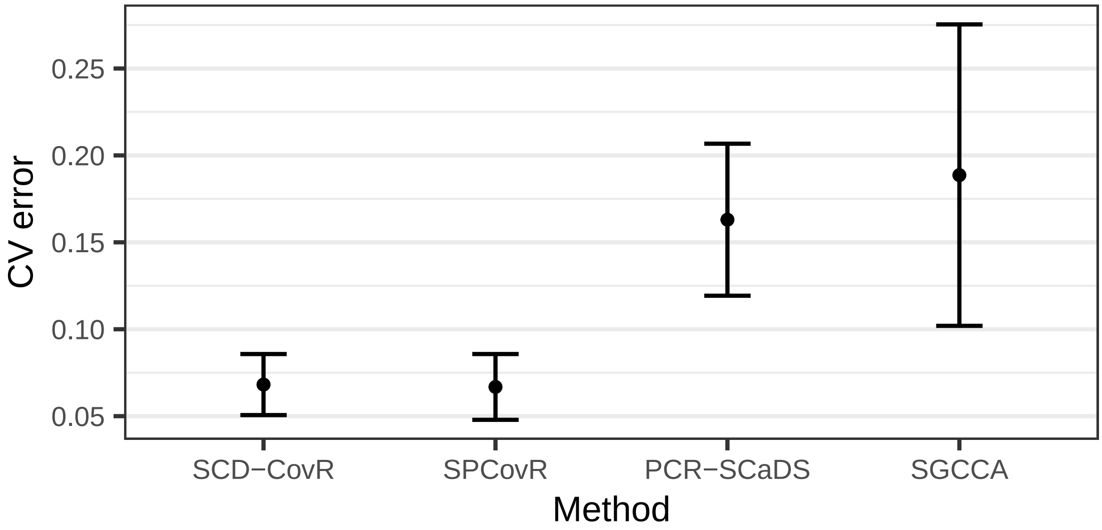


Figure 2.7. Cross-validation error and the corresponding standard error of the four methods.

For interpretation of the final SCD-CovR model, we can first study the retrieved sparse weights matrix (Appendix 2.C). It displays that the resulting weights matrix is very sparse; there are only 7, 5 and 4 non-zero weights that correspond to the three components respectively. As dictated by the tuned zero-block constraints, the weights matrix contains non-zero coefficients from both predictor blocks only in the column that corresponds to the common component.

We further investigated the model by inspecting Figure 2.8. This figure plots the component scores of the potato samples. Out of the 20 potato samples, 12 were grown conventionally and 8 were grown organically. Although this information was not incorporated when fitting the model, the two types can be clearly distinguished using the two distinctive components. Therefore, these components found by SCD-CovR not only are capable of predicting the response variable but also reveal existing structural variation. In summary, the exploration of the final model shows that the method is able to fulfill its aims. It retrieves common and distinctive components that are sparse and thus more interpretable. The components also adequately explain the variance in both response and the predictors.

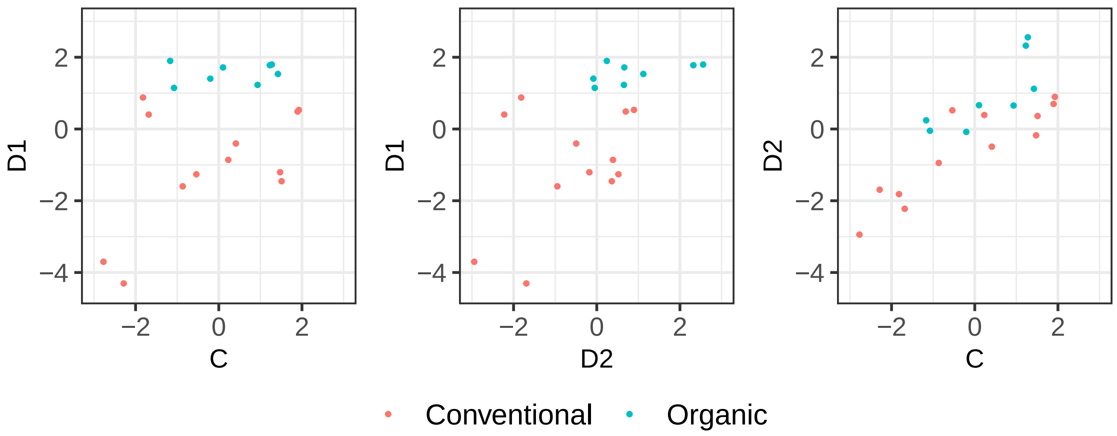


Figure 2.8. Component scores of the potato samples. The two types of potato samples are displayed in different colours. C, D1 and D2 indicate the type of the components (i.e. D1 refers to the component distinctive to the first predictor block which is the chemical analysis).

2.5 Discussion

Data originating from multiple sources can be analyzed with several objectives: prediction of a criterion, selection of relevant variables and uncovering the common and distinctive underlying mechanisms. We proposed SCD-CovR to address these three aims simultaneously.

Through a simulation study incorporating multiple evaluation criteria that reflect these aims, we demonstrated that SCD-CovR outperforms three related methods that serve a subset of these goals; SPCovR, PCR-SCaDS and SGCCA. Our method resulted in better prediction than PCR-SCADS and was also more effective than SGCCA for prediction under certain conditions. The coefficients retrieved by SCD-CovR better reflected the true underlying coefficients than those found by SGCCA. Lastly, with respect to finding common and distinctive processes, the method outperformed SPCovR and SGCCA in capturing the block structure of common and distinctive components. We further illustrated this comparative advantage of SCD-CovR by re-analyzing a publicly available empirical dataset. The SCD-CovR cross-validation error was lower than that of PCR-SCaDS and SGCCA. At the same time, SCD-CovR retrieved more distinctive components than SPCovR.

These results provide further insight into the strengths of our proposed method. The outperformance in prediction compared to PCR-SCaDS reiterates previous comparisons of PCovR and PCR (Heij, Groenen, & van Dijk, 2007; Vervloet et al., 2016). Deriving components while taking the criterion into account is more effective for prediction, than adopting a two-step approach of first constructing the components and then subsequently using them for prediction. Similarly, PLS methods have been found to be more prone to overfitting than PCovR methods (Van Deun et al., 2018) and our outcome of the simulation study shows the same, with SCD-CovR yielding better out-of-sample prediction under several conditions. Moreover, SPCovR and SCD-CovR being more effective than SGCCA exhibits the benefits of the weighting parameter α. It enables a good balance between focusing on the predictors or the criterion, while SGCCA emphasizes the criterion more strongly. Our results are all based on α values established through cross-validation and thus indicate the effectiveness of the weighting parameter even within a data-driven approach. Lastly, concerning the identification of common and distinctive components, the simulation results from the three-component models illustrate the outperformance of SCD-CovR when the zero block constraints are correctly specified. This implies that the method can be especially effective when supported by an adequate model selection strategy.

Our proposed method also comes with some weaknesses. Model selection is an obvious challenge. As the method is devised to serve multiple aims, it involves many parameters to be tuned. The weighting parameter α, the number of components, the common and distinctive component structure and the penalization parameters are all influential and the retrieved model heavily depends on the choice of these parameters. Furthermore, identifying and discerning common and distinctive processes for data fusion methods is a very complicated task as it often interacts with other aspects such as the number of components (Måge et al., 2019). In the same vein, the weighting parameter α involved with PCovR is also difficult to tune (Vervloet et al., 2016). However, as the current paper focuses more on the proposal and the illustration of the new SCD-CovR method, this intricate problem of model selection has not been extensively addressed.

The examples presented in the current study only concern a scenario with two data blocks, but it is possible to extend our method to a situation with more blocks. In that case, a component that is constructed by predictors from a single data block would be defined as a distinctive component. Components pertaining to predictors from multiple but not all blocks would be called partially or locally common, as opposed to globally common components that involve predictors from all of the data blocks. These terminologies are in line with the previous literature such as Måge et al. (2019). In such data circumstances, the challenge of model selection would involve heavy computational burden because our method caters for capturing of common and distinctive underlying processes by means of the pre-specified zero block constraints. Given K data blocks and R components, no less than $\binom{(2^K-1)+R-1}{R}$ different zero block constraints should be evaluated. Considering that the method also involves several other parameters for retrieving the sparse solutions, the model selection procedure becomes a particularly intensive task.

As it holds for many other methods that rely on the lasso and elastic net penalties to attain sparsity, SCD-CovR is not free from the shortcoming that non-zero coefficients may be overly shrunken towards zero. Alternatives have been proposed, including the adaptive lasso (Zou, 2006) and the SCADS penalty (Fan & Li, 2001) which apply different degrees of shrinkage depending on the value of the coefficients. Stability selection (Meinshausen & Bühlmann, 2010) is another effective method for variable selection that does not shrink the non-zero coefficients. However, some degree of shrinkage of the non-zero coefficients may be beneficial in terms of bias-variance tradeoff as it helps to stabilize the OLS estimates (Breiman, 1995).

There are several future directions that the method can extend towards. Handier solutions to retrieve the distinctive components such as the Group lasso penalty can be adopted to greatly relieve the computational demand of the zero block constraints. Gu and Van Deun (2019) have implemented the Group lasso to find distinctive components within the multi-block sparse PCA setting and this could be one of the possible future directions in extending the SCD-CovR method. Another natural extension is to allow multiple criterion variables, as the current method only entails the univariate regression problem. Furthermore, the method can be adapted to incorporate more diverse structures of underlying processes. The current simulation study assumes that the data generating model follows the properties of PCA where the weights and the loadings are equal. However, true structures where this equality does not hold may exist. It would be interesting to examine the applicability of the method within such circumstances, as both weights and loadings would need to be considered for interpretation. Similarly, our proposed method only enforces sparsity in the weights, but the true structure may also include sparse loadings. Looking further into these other possible models where loadings or both weights and loadings are sparse can also be a plausible direction in devising a predictive method that is more interpretable, in a modern multiblock setting.

Appendix

2.A Alternating least squares for SCD-CovR

As given in Section 2.2, the objective function to be minimized is:

$$L(\mathbf{W}_C, \mathbf{P}_C^{(X)}, \mathbf{p}_C^{(y)}) = \alpha \frac{\left\| \mathbf{y} - \mathbf{X}_C \mathbf{W}_C \mathbf{p}_C^{(y)\top} \right\|_2^2}{\|\mathbf{y}\|_2^2} + (1-\alpha) \frac{\left\| \mathbf{X}_C - \mathbf{X}_C \mathbf{W}_C \mathbf{P}_C^{(X)\top} \right\|_2^2}{\|\mathbf{X}_C\|_2^2} \\ + \lambda_L \left|\mathbf{W}_C\right|_1 + \lambda_R \left\|\mathbf{W}_C\right\|_2^2 \tag{2.11}$$

such that $(\mathbf{P}_C^{(X)})^\top \mathbf{P}_C^{(X)} = \mathbf{I}_R$, $\lambda_L, \lambda_R \geq 0$, $\alpha \geq 0$ and zero block constraint on $\mathbf{W}_C$.

The solutions are found through an alternating procedure where the objective is minimized with regards to $\mathbf{P}_C^{(X)}$ and $\mathbf{p}^{(y)}$ conditional on a fixed value of $\mathbf{W}_C$ and vice versa. The procedure iterates until a convergence criterion is met. Many methods which attain sparse solutions from PCA through regularization penalty have adopted this approach to find the solutions (de Schipper & Van Deun, 2018; Van Deun et al., 2018; Zou et al., 2006). The procedure for SCD-CovR is similar to these methods, but the minimization with respect to $\mathbf{P}_C^{(X)}$ and $\mathbf{p}^{(y)}$ given $\mathbf{W}_C$ is slightly different. The loadings $\mathbf{P}_C^{(X)}$ are obtained via an analytical solution; $\mathbf{P}_C^{(X)} = \mathbf{U}\mathbf{V}^\top$ where $\mathbf{U}$ and $\mathbf{V}$ are found through singular value decomposition of $\mathbf{X}_C^\top \mathbf{X}_C \mathbf{W}_C = \mathbf{U}\mathbf{D}\mathbf{V}^\top$. The regression coefficients $\mathbf{y}$ are given by the ridge regression estimates; $\mathbf{p}^{(y)} = (\mathbf{X}_C^\top \mathbf{X}_C + \lambda_R \mathbf{I})^{-1} \mathbf{X}_C^\top \mathbf{y}$, where $\mathbf{I}$ is a $(\sum_k^K J_k) \times (\sum_k^K J_k)$ identity matrix and λ_R is a ridge penalty. Conditional on these values, the weights $\mathbf{W}$ are found through the coordinate descent algorithm. The zero block constraint specifies the elements that will be put to zero to encourage the common and distinctive processes. The details on the conditional estimation of $\mathbf{W}$ given $\mathbf{P}_C^{(X)}$ and $\mathbf{p}^{(y)}$ can be found in de Schipper and Van Deun (2018).

2.B Model selection for the illustrative application

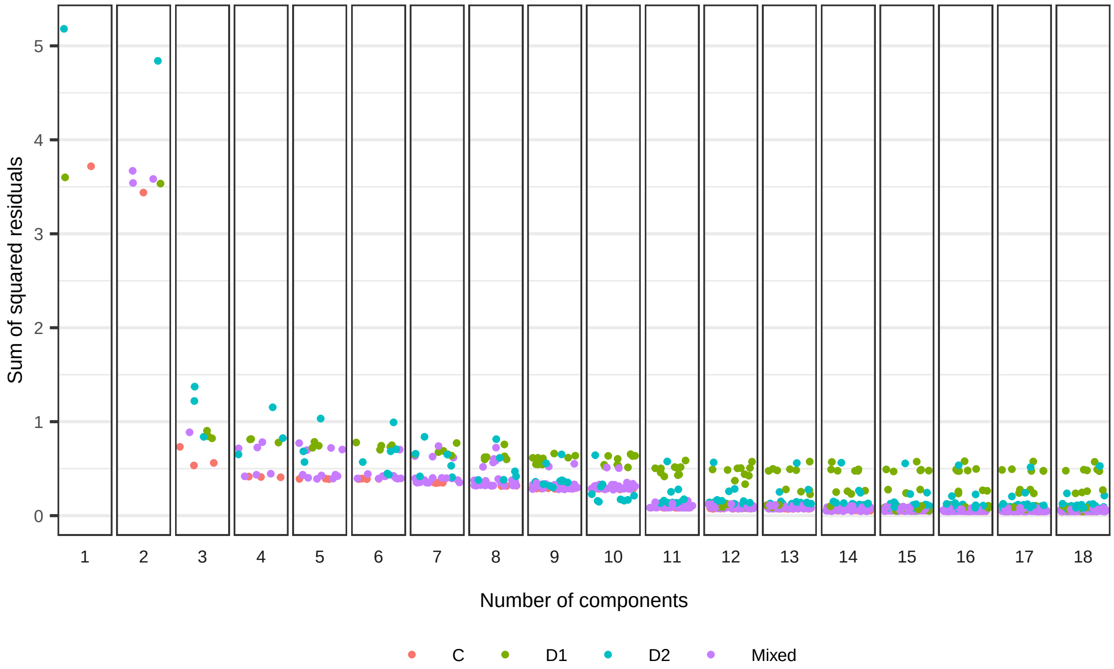


Figure 2.9. SCD-CovR: residual plot for determining the number of components. Each dot represents one model with a certain common and distinctive component structure. The colours indicate the type of component that occupies more than 65% of the total number of components in a model (e.g. when common components make up more than 65% of the total set of components, the model is coloured red). When one particular type of component does not dominate the model, it is indicated by purple. D1 and D2 denote models dominated by components distinctive to block 1 and 2 respectively, while C refers to the common component.

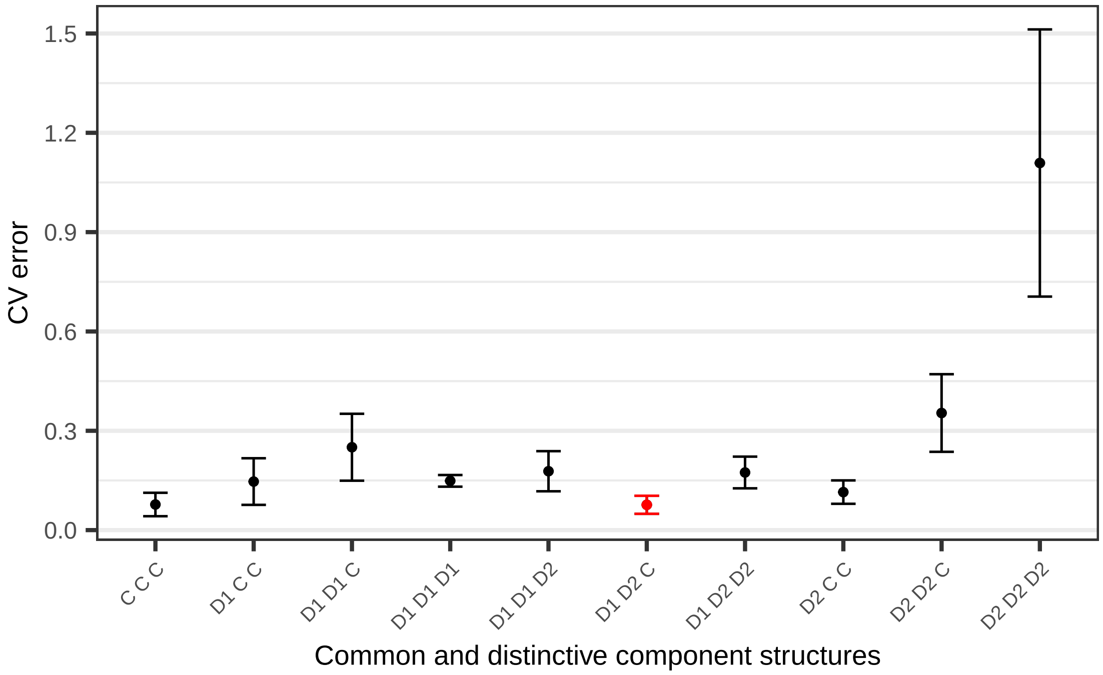


Figure 2.10. SCD-CovR: cross-validation error and corresponding standard error for zero block constraint for common and distinctive structure. D1 and D2 indicate components distinctive to block 1 and 2 while C denotes the common component. The selected zero block constraint with the smallest cross-validation error is displayed in red.

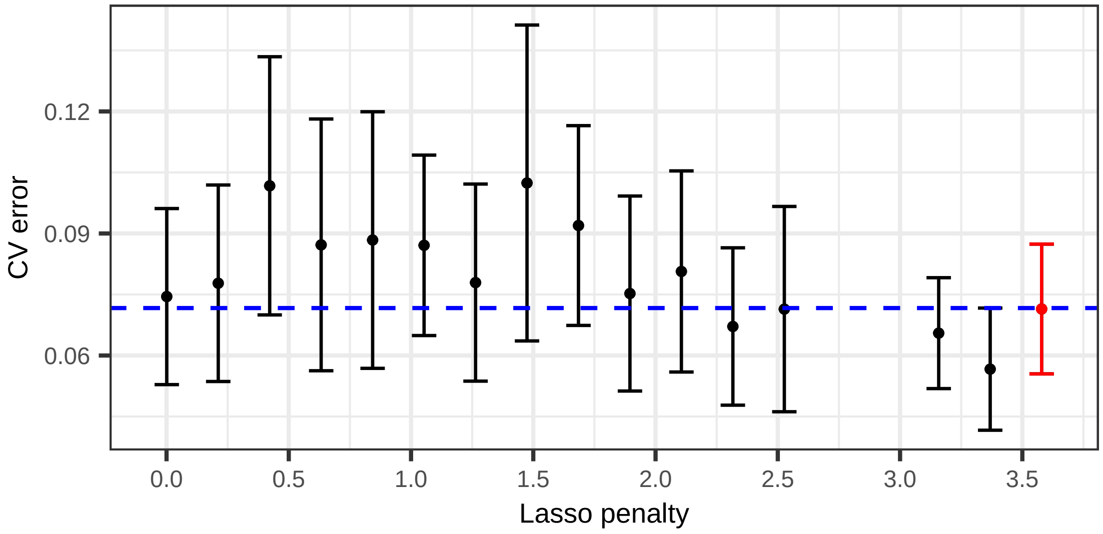


Figure 2.11. SCD-CovR: cross-validation error and corresponding standard error for the lasso penalty. The blue dashed line indicates the bound used for the 1 SE rule and the selected lasso value is shown in red.

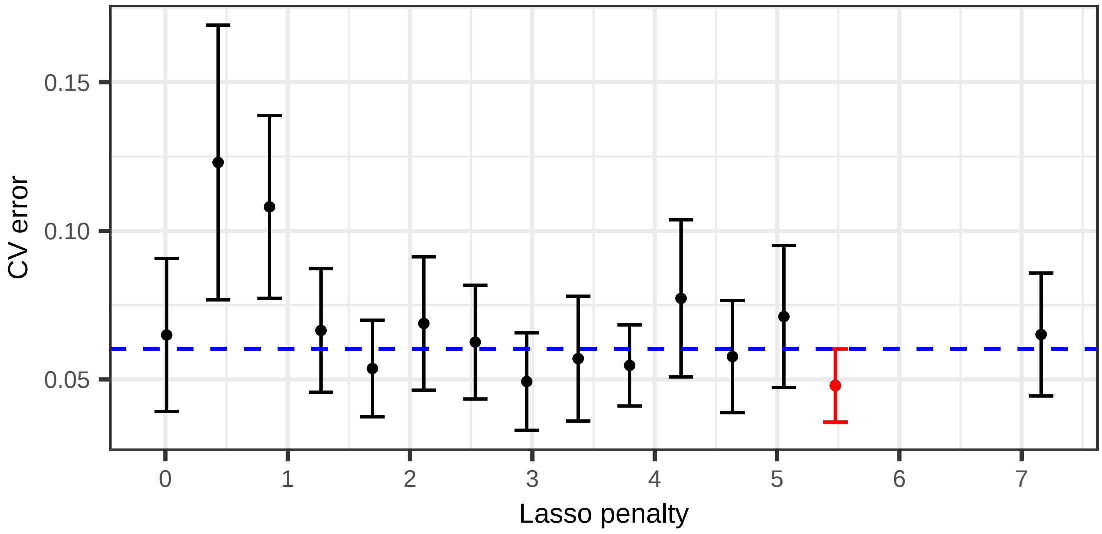


Figure 2.12. SPCovR: cross-validation error and corresponding standard error for the lasso penalty. The blue dashed line indicates the bound used for the 1 SE rule and the selected lasso value is shown in red.

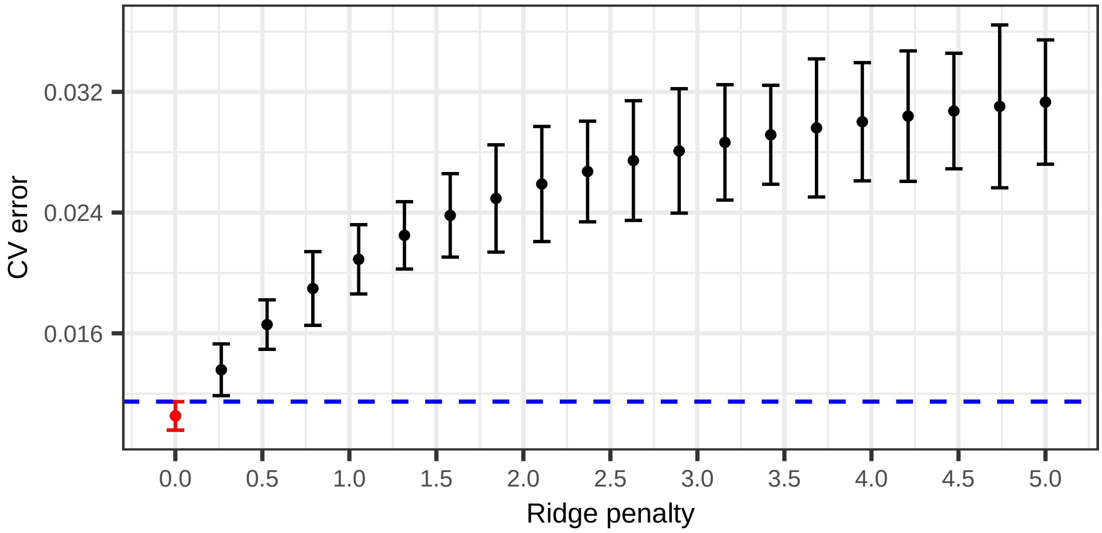


Figure 2.13. PCR-SCaDS: cross-validation error and corresponding standard error for the ridge penalty. The blue dashed line indicates the bound used for the 1 SE rule and the selected ridge value is shown in red.

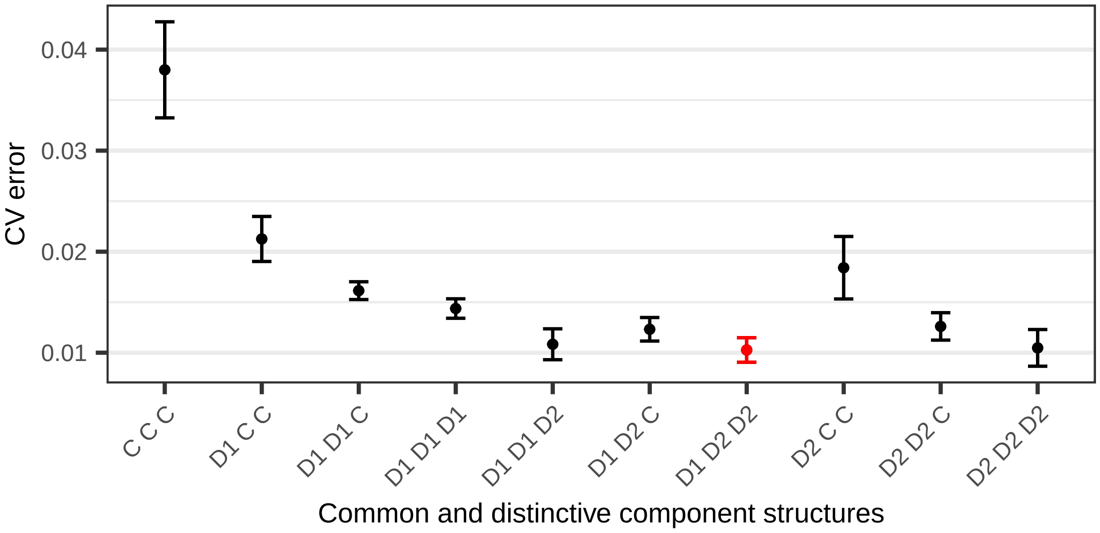


Figure 2.14. PCR-SCaDS: cross-validation error and corresponding standard error for zero block constraint for the common and distinctive structure. D1 and D2 indicate components distinctive to block 1 and 2 while C denotes the common component. The selected zero block constraint with the smallest cross-validation error is displayed in red.

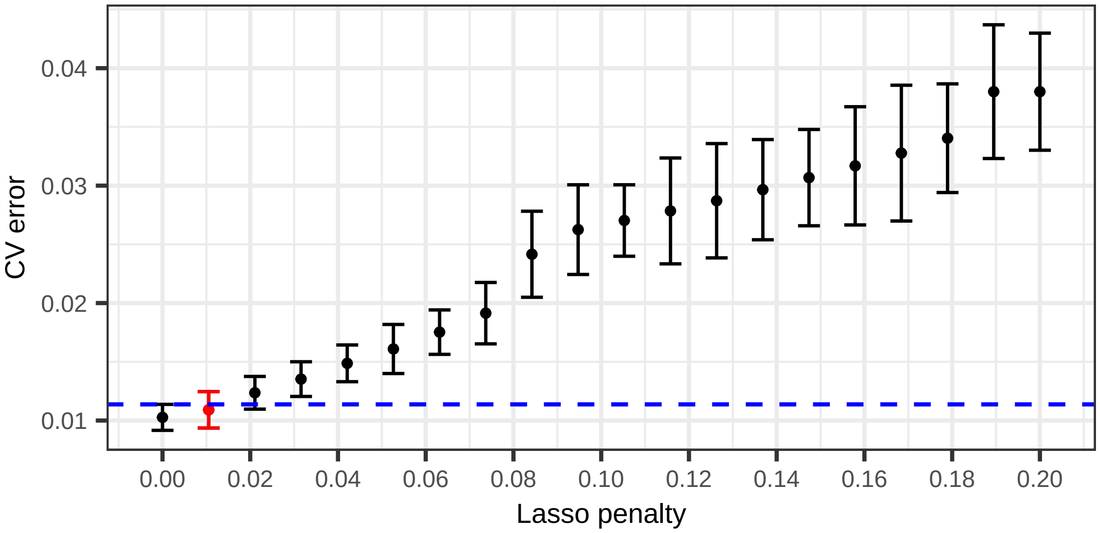


Figure 2.15. PCR-SCaDS: cross-validation error and corresponding standard error for the lasso penalty. The blue dashed line indicates the bound used for the 1 SE rule and the selected lasso value is shown in red.

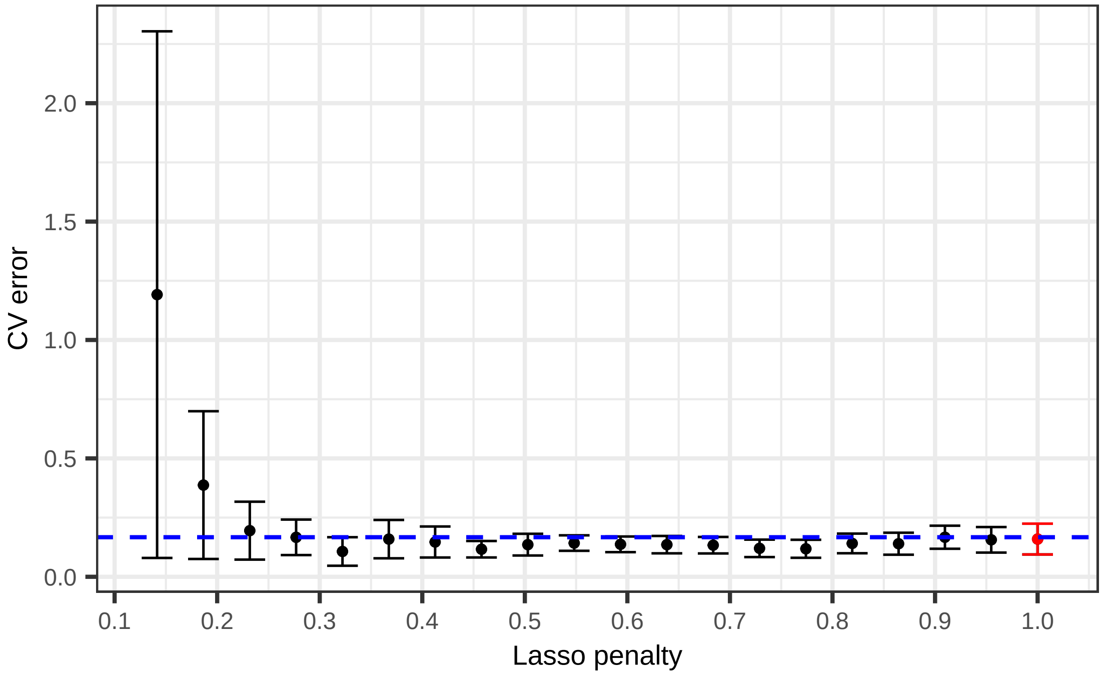


Figure 2.16. SGCCA: cross-validation error and corresponding standard error for the lasso penalty. The blue dashed line indicates the bound used for the 1 SE rule and the selected lasso value is shown in red.

2.C Illustrative application retrieved weights

Table 2.3. Weights retrieved by the final SCD-CovR model from the illustrative application. The table on the left presents weights corresponding to the chemical analysis block, the one on the right corresponding to the uniaxial compression block.

	C	D1	D2
Chemical analysis			
PEU	0	0	0
starch	0	-6.004	0
TotalN	0	0	0
phytic	0	-0.024	0
Ca	0	-0.494	0
Mg	0	0	0
Na	0	-0.003	0
K	-2.519	0	0
his1	0	0	0
his2	0.138	0	0
his3	0.306	0	0
his4	0	1.130	0
his5	0.480	0	0
his6	0	0	0

	C	D1	D2
Uniaxial compression			
FractureWork20	0	0	0
BreakWork20	0	0	4.947
stressT20	0	0	0
strainH20	0	0	0
modulus20	0	0	0
slope20	0	0	0
FractureWor100	0	0	0
BreakWork100	0	0	0
stressT100	0	0	0
strainH100	0.133	0	0
modulus100	0	0	0
slope100	0	0	0
FractureWor250	0	0	0
BreakWork250	0	0	0
stressT250	0	0	0
strainH250	0	0	0
modulus250	0	0	1.766
slope250	0	0	0
FractureWor500	0	0	0
BreakWork500	0	0	0
stressT500	0	0	0
strainH500	0	0	0
modulus500	0	0	0
slope500	0	0	0
FractureWor750	0	0	0
BreakWork750	0	0	0
stressT750	0	0	0
strainH750	0	0	0
modulus750	0	0	1.663
slope750	0	0	0
FractureWor1000	0	0	0
BreakWork1000	5.758	0	0
stressT1000	0	0	0
strainH1000	0	0	0
modulus1000	0	0	1.104
slope1000	0.155	0	0

Chapter 3

Logistic Regression with Sparse Common and Distinctive Covariates

Having large sets of predictor variables from multiple sources concerning the same individuals is becoming increasingly common in behavioural research. On top of the variable selection problem, predicting a categorical outcome using such data gives rise to an additional challenge of identifying the processes at play underneath the predictors. These processes are of particular interest in the setting of multi-source data because they can either be associated individually with a single data source or jointly with multiple sources. Although many methods have addressed the classification problem in high dimensionality, the additional challenge of distinguishing such underlying predictor processes from multi-source data has not received sufficient attention. To this end, we propose the method of Sparse Common and Distinctive Covariates Logistic Regression (SCD-Cov-logR). The method is a multi-source extension of principal covariates regression that combines with generalized linear modeling framework to allow classification of a categorical outcome. In a simulation study, SCD-Cov-logR resulted in outperformance compared to related methods commonly used in behavioural sciences. We also demonstrate the practical usage of the method under an empirical dataset.

Keywords: Multiblock data, Principal covariates regression, Common and distinctive processes, Data integration, Classification, Logistic regression

Park, S., Ceulemans, E., & Van Deun, K. (2023). Logistic regression with sparse common and distinctive covariates. *Behavior Research Methods*, 1-32.

3.1 Introduction

In behavioural research, it is often of interest to classify subjects, e.g., by constructing a logistic regression model. For example, in mental health research scores on various tests are used to classify subjects into having versus not having a disorder such as alcoholism (Babor, Higgins-Biddle, Saunders, & Monteiro, 2001), dementia (Mioshi, Dawson, Mitchell, Arnold, & Hodges, 2006), and eating disorders (Botella, Huang, & Suero, 2015; Hill, Reid, Morgan, & Lacey, 2010). By constructing a classification model, the factors predicting class membership can be investigated. For example, Barnes et al. (2009) studied the importance of various measures such as genotype, fMRI and cognitive tests in predicting dementia among older adults through logistic regression. As a result, a risk index that stratifies older adults into different risk groups depending on their scores on certain risk factors was put forward.

Many studies in behavioural sciences of today involve datasets comprised of multiple blocks of predictor variables obtained for the same individuals, with each block of variables originating from different measurement instruments. Examples of such blocks include demographic data, social media, genetic profiling, and questionnaires. These joint datasets are referred to as multiblock data (more details on the conceptual framework are given in Van Mechelen & Smilde, 2010). A unique feature of multiblock data is that they can reveal two different kinds of sources of interindividual variation; those that concern single individual data blocks and those that jointly encompass multiple blocks. These sources of variation are referred to as distinctive and common, respectively, and they are used to reveal the processes underlying the emergence of particular conditions. To explain more concretely, let us consider a block of genotype data and another block of self-reported health behaviour data collected from two groups of children; ADHD-diagnosed and healthy. Studying the onset of ADHD by adopting this multiblock dataset, processes that only underlies the genotype data may be found. For example, a dompaninergic pathway involving dopamine transporter gene (DAT1) and a serotonergic pathway incorporating serotonin transporter gene (5HTTT) have been reported to play a role in ADHD (Gizer, Ficks, & Waldman, 2009). These biological pathways would be considered as distinctive processes as they entail only the genotype data block. On the other hand, the multiblock data could also reveal a process that involves both blocks of genotype and health behaviour. Kahn, Khoury, Nichols, and Lanphear (2003) found the combination of maternal prenatal smoking with a DAT1 genotype leading to ADHD, while in another study, maternal stress during pregnancy together with dopamine receptor 4 gene (DRD4) were associated with severity of ADHD symptoms (Grizenko et al., 2012). Such cases of gene-environment interplay are examples of common processes as they involve multiple data blocks.

Methods based on PCA have been actively proposed to disentangle the common and distinctive processes from multiblock data, but without considering the prediction problem of an outcome variable (e.g. simultaneous component analysis with distinctive and common components, DISCO-SCA; Schouteden et al., 2013). As multiblock datasets are often characterized by a large number of variables, these PCA based methods have been further extended. The presence of many variables complicates the interpretation of the components derived by SCA as they are associated with a large set of variables. The introduction of sparseness penalties - limiting the number of variables associated with a component - yields interpretable components that represent common and distinctive processes (e.g. sparse common and distinctive SCA (SCaDS); de Schipper & Van Deun, 2018).

Recently, a method that identifies common and distinctive processes from a multiblock dataset in the context of a regression problem for a continuous outcome has been proposed (Sparse Common and Distinctive Covariates Regression (SCD-CovR); S. Park, Ceulemans, & Van Deun, 2020). The method is an extension of Principal Covariates Regression (PCovR) which finds summary variables that explain variance in both predictors and outcome by combining PCA and linear regression (De Jong & Kiers, 1992). SCD-CovR incorporates SCaDS into the PCovR framework to obtain sparse common and distinctive predictor processes. In order to address the classification problem, the current paper extends the SCD-CovR method to logistic regression; this means that here we develop sparse common and distinctive covariates logistic regression method (SCD-Cov-logR). SCD-Cov-logR reveals the common and distinctive predictor processes that play a role in classification of the outcome and does so in an interpretable/insightful way by relying on sparse representations.

The paper is arranged as follows. First, we provide the methodological background and mathematical details of SCD-Cov-logR. Then, the results from simulation studies that comparatively demonstrate the performance of SCD-Cov-logR against an existing method with a similar set of objectives are presented. After further illustration of the current method on an empirical multiblock dataset, the paper is concluded by formulating some limitations and directions for future research. The implementation of SCD-Cov-logR was done in R and Rcpp, which can be found on Github: `https://github.com/soogs/SCD-Cov-logR`, along with the code used to generate the results reported in the paper.

3.2 Methods

3.2.1 Notation

The following notation is used throughout the paper: scalars, vectors and matrices are denoted by italic lowercase, bold lowercase and bold uppercase letters respectively. Transposing is indicated by the superscript $^\top$. Lowercase subscripts running from 1 to corresponding uppercase letters denote indexing: $i \in \{1, 2, \ldots, I\}$. Subscript $_C$ indicates concatenation of multiple data blocks, while superscripts $^{(X)}$, $^{(y)}$ and $^{(g)}$ highlight affiliation with predictor, continuous outcome and binary outcome variables, respectively. To denote estimates, a $\hat{}$ over the symbol denoting the population parameter is used (i.e. $\hat{\mathbf{b}}$ is the estimated logistic regression coefficients). $\mathbf{X}$ refers to a matrix containing the standardized scores of J predictors corresponding to I observation units (that is, each column has mean zero and variance equal to one). In the context of multiple predictor blocks, $\mathbf{X}_k$ (with size $I \times J_k$) indicates a kth predictor block matrix with its predictors column-scaled and standardized; with $k \in \{1, 2, \ldots, K\}$. $\mathbf{X}_C = [\mathbf{X}_1, \ldots, \mathbf{X}_K]$ (of size $I \times \sum_{k=1}^{K} J_k$) denotes the supermatrix that concatenates the predictor blocks. g indicates a dummy vector of size I containing the scores on the binary outcome variable, while $\mathbf{y}$ is a vector of size I of a continuous outcome. In the context of an outcome variable with multiple categories, $\mathbf{G}$ (with size $I \times M$) refers to a dummy matrix for the categorical outcome with M total categories. For the ith observation unit, $g_{im} = 1$ if the response is in the mth category and $g_{im} = 0$ otherwise. Lastly, $\mathbf{I}_a$ denotes a $a \times a$ identity matrix where the subscript a indicates the size of the matrix.

3.2.2 Model and objective function

SCD-Cov-logR is a classification method for a categorical outcome. The method is particularly suitable when multiple large blocks of predictor variables are available as it allows to take the block structure into account and to limit the number of variables contributing to the predictive processes. SCD-Cov-logR constructs two types of summary covariates: distinctive covariates based on a linear combination of the predictor variables of one single data block and common covariates that combine variables of multiple data blocks. Identification of different types of predictor processes helps understanding of processes that play important roles in the classification of the outcome. To further facilitate the interpretation of these processes, SCD-Cov-logR introduces regularization penalties to select a subset of the predictor variables in constructing the common and distinctive covariates. Taken together, an effective classification method results where common and distinctive predictor processes are identified in a sparse and therefore interpretable manner; the method is also flexible in the sense that it includes several other methods as a special case such as logistic regression and PCovR for categorical outcomes. We start with a brief description of the building blocks, namely logistic regression and PCovR, before moving onto SCD-Cov-logR. While the current method allows classification of both binary and multiclass outcome variables via logistic regression, we focus on binary logistic regression in the following subsections in describing our method. The multiclass classification via multinomial logistic regression will be discussed thereafter, as it is a straightforward extension of the binary problem.

3.2.2.1 Logistic regression

Logistic regression assumes that the log-odds (logit) of the binary outcome are linearly dependent on the predictor variables. Let $\mathbf{x}_i$ be the vector of predictor scores for subject i and g_i the score on the outcome (either 0 or 1). The log-odds for subject i is modelled by:

$$\log\left(\frac{p(g_i = 1)}{1 - p(g_i = 1)}\right) = \mathbf{x}_i^\top \mathbf{b} + b_0 \tag{3.1}$$

where $p(g_i = 1)$ denotes the probability that the ith subject would fall under the category represented by a 1. The vector $\mathbf{b}$ indicates the logistic regression weights and the scalar b_0 the intercept. From this model it follows that

$$\begin{aligned} p(g_i = 1) =& \frac{1}{1 + e^{-(\mathbf{x}_i^\top \mathbf{b} + b_0)}} \\ p(g_i = 0) =& 1 - p(g_i = 1), \end{aligned} \tag{3.2}$$

which can be used to set up the likelihood equation. The estimates of the logistic regression parameters can then be obtained by maximizing the log-likelihood or minimizing the negative log-likelihood; here, the latter will be used for integration with the PCovR objective. The following negative log-likelihood is minimized:

$$L(\mathbf{b}, b_0) = -\sum_{i}^{I}(g_i(b_0 + \mathbf{x}_i^\top \mathbf{b}) - \log(1 + e^{(b_0 + \mathbf{x}_i^\top \mathbf{b})})). \tag{3.3}$$

Typically, the minimum of this function is found via a numerical procedure as it has no closed form. A popular approach is the Newton-Raphson method for finding the root of the first derivative which amounts to iteratively reweighted least squares. It boils down to formulating local quadratic approximations of the negative log-likelihood in an iterative scheme that, after initialization, uses the minimum of the quadratic approximation for updating in the next iteration.

3.2.2.2 PCovR

In a setting with a large set of predictor variables, the ordinary (least-squares) approach to linear regression involves several drawbacks. It is difficult to interpret the large set of regression coefficients corresponding to each of the predictors. Also, in the case of multicollinearity (highly correlated predictors), the estimates are instable. When the number of predictors exceeds the number of observations (high-dimensionality), the method has no unique solution. In order to alleviate these difficulties, Principal Covariates Regression (PCovR; De Jong & Kiers, 1992) was put forward by combining PCA with linear regression. PCovR introduces summary variables, the so-called 'principal covariates', in modelling the predictor and outcome variables. The covariates summarize the predictors by a linear combination of the original variables that is obtained in such a way that they account for variation in both predictor and outcome variables. Regression coefficients are found for these limited number of covariates instead of for each of the original predictor variables, resolving the challenges of finding a unique and stable regression model in the setting of a large number of predictors. Since the covariates summarize the predictors, they can be understood to represent the predictor processes behind the outcome. Let R be the pre-specified number of covariates to be derived. PCovR then assumes the following models for the predictor and outcome variables:

$$\begin{aligned}\mathbf{y} =& \mathbf{X}\mathbf{W}\mathbf{p}^{(y)} + \mathbf{e}^{(y)} \\ \mathbf{X} =& \mathbf{X}\mathbf{W}(\mathbf{P}^{(X)})^\top + \mathbf{E}^{(X)}.\end{aligned} \tag{3.4}$$

Both the models for the outcome $\mathbf{y}$ and for the predictor variables $\mathbf{X}$ rely on the same summary predictor scores $\mathbf{XW}$ with $\mathbf{W}$ refering to the weights matrix of size $J \times R$. The weights prescribe the linear combination of the predictors to compose the principal covariates (namely, $\mathbf{T} = \mathbf{XW}$). The first line of Equation (3.4) shows the model underlying the outcome; in that model $\mathbf{p}^{(y)}$ indicates a vector of R regression coefficients while $\mathbf{e}^{(y)}$ denotes the residuals pertaining to the outcome. The second line of Equation (3.4) gives the model for the predictors. $\mathbf{P}^{(X)}$ indicates the loadings matrix of size $J \times R$. Similar to the regression coefficients $\mathbf{p}^{(y)}$ for the outcome variable in the first line, the loadings matrix linearly combine the covariates to reconstruct back the predictors. It can be seen as regression coefficients obtained from regressing the predictor variables on the principal covariates. Note that this model formulation also underlies the methods of principal components regression (PCR; see Jolliffe, 1982) and partial least squares (PLS; H. Wold, 1982; S. Wold et al., 1983).

The aim of PCovR to find covariates that effectively reconstruct $\mathbf{X}$ and simultaneously predict $\mathbf{y}$ is expressed by the following joint loss function (De Jong & Kiers, 1992):

$$L(\mathbf{W}, \mathbf{P}^{(X)}, \mathbf{p}^{(y)}) = \alpha \frac{\left\|\mathbf{y} - \mathbf{X}\mathbf{W}\mathbf{p}^{(y)}\right\|_2^2}{\|\mathbf{y}\|_2^2} + (1-\alpha)\frac{\left\|\mathbf{X} - \mathbf{X}\mathbf{W}(\mathbf{P}^{(X)})^\top\right\|_2^2}{\|\mathbf{X}\|_2^2}, \tag{3.5}$$

with $0 \leq \alpha \leq 1$, a known constant which specifies the balance between fitting the outcome and the predictors. With α set at 0, the method is the same as PCR where the outcome variable is regressed on the principal components found by PCA. On the other hand, with $\alpha = 1$, the method is equivalent to linear regression [1]. The solution of Equation (3.5) is not identifiable without imposing constraints. Therefore, the covariates are often constrained to be orthonormal ($\mathbf{T}^\top\mathbf{T} = \mathbf{I}_R$) to identify the solution (De Jong & Kiers, 1992).

The principal covariates in the PCovR model are used to represent the processes that underlie both the predictor and outcome variables. Therefore, it is important to interpret the derived covariates to understand the nature of these processes. There are two ways of interpreting the covariates. Firstly, the loadings matrix $\mathbf{P}^{(X)}$ can be studied. When the principal covariates are scaled to variance equal to one ($\mathbf{T}^\top\mathbf{T} = I\mathbf{I}_R$) and the predictor variables have been centered and scaled to variance equal to one, the loadings are equal to the correlation between the principal covariates and the predictor variables. Therefore, $\mathbf{P}^{(X)}$ can be conveniently interpreted in two ways; regression coefficients that reconstruct the predictors (namely, $\mathbf{T}(\mathbf{P}^{(X)})^\top = (\mathbf{X}\mathbf{W})(\mathbf{P}^{(X)})^\top = \mathbf{X}$) and covariate-predictor correlations. The loadings derived within PCA are also commonly studied in the same manner. On the other hand, the second way to understand the covariates is by observing the weights matrix $\mathbf{W}$. The weights are used in combining the predictors to construct the covariates, and therefore they describe the composition of the covariates. They also play an important role in applying the model to new data, in the context of prediction for new observations, as they are used to transform the new predictor variables to covariate scores. Studying the loadings or the weights are both valid ways to understand the nature of the covariates and the two estimates can both be inspected in a complementary manner. However,

[1] $\hat{y}_i = \sum_r \hat{p}_r^{(y)} \hat{t}_{ir} = \sum_r (\sum_j \hat{p}_r^{(y)} x_{ij} \hat{w}_{jr}) = \sum_j (\sum_r \hat{p}_r^{(y)} \hat{w}_{jr}) x_{ij}$, with $\sum_r \hat{p}_r^{(y)} \hat{w}_{jr}$ as a regression coefficient for the jth predictor, where r is an index for each covariate.

if one of the estimates should be chosen for inspection, the choice should depend on the research aim of interest; loadings reflect the strength of association of the predictor variables with the principal covariates while weights prescribe how the covariates are constructed. We refer to Guerra-Urzola, Van Deun, Vera, and Sijtsma (2021) for a thorough discussion of the issue of loadings versus weights in the context of sparse PCA.

3.2.2.3 SCD-Cov-logR

Here, we propose a method for binary classification that is suitable for multiblock data where several blocks of predictor variables are available: besides the fact that the method can handle many predictors or even high-dimensional data, it yields particular insight in the data by revealing common and distinctive predictor processes in a sparse and therefore interpretable manner.

Model

We make use of a model formulation that integrates the logistic regression and PCovR models in Equations (3.2) and (3.4). More specifically, the model for the outcome variable is adapted. Let the vector $\mathbf{x}_{Ci}$ denote the ith row of the supermatix $\mathbf{X}_C$ resulting from the concatenation of the predictor blocks and let $\mathbf{W}_C$ of size $\sum_{k=1}^{K} J_k \times R$ denote the corresponding weights matrix, then the log-odds of the binary outcome can be modelled by the principal covariates as follows:

$$\begin{aligned} \log\left(\frac{p(g_i = 1)}{1 - p(g_i = 1)}\right) =& \mathbf{x}_{Ci}^{\top}\mathbf{W}_C\mathbf{p}^{(g)} + p_0^{(g)} \\ \mathbf{x}_{Ci} = & \left[\mathbf{x}_{Ci}^{\top}\mathbf{W}_C(\mathbf{P}_C^{(X)})^{\top}\right]^{\top} + \mathbf{e}_i^{(X)}, \end{aligned} \tag{3.6}$$

where $\mathbf{p}^{(g)}$ in the first line of the equation denotes the vector of R regression coefficients and $p_0^{(g)}$ the intercept. As in the PCovR model (3.4), the weights matrix dictates the composition of the covariates ($\mathbf{T}_C = \mathbf{X}_C\mathbf{W}_C$). In the second line, $\mathbf{P}_C^{(X)}$ indicates the loadings matrix of size $\sum_{k=1}^{K} J_k \times R$. They recover the predictor variables from the covariates, as done in the PCovR model. Therefore, the covariates in this model explain both the variance of predictor variables and the log-odds of the binary outcome variable.

The model in Equation (3.6) includes all predictor variables in constructing the principal covariates while often it is of interest to find the subset of variables that are relevant for the predictor processes represented by the principal covariates. Hence, our proposed model is subject to a sparsity inducing penalty that limits the number of predictor variables contributing to the covariates. SCD-Cov-logR therefore imposes the sparsity on the weights, as we are interested in finding a subset of predictors that together make up the predictor processes. In this way, understanding the covariates becomes much easier as they are based on a smaller subset of predictors.

To understand the composition of the covariates not only at the level of the individual variables but also at the level of the blocks, sparsity is imposed in two ways: On the one hand at the level of the blocks (blockwise sparsity) and, on the other hand, at the level of the individual variables (elementwise sparsity). Blockwise sparsity refers to forcing the weights corresponding to an entire set of predictors in a data block to zero. By doing so, distinctive covariates which are only comprised of predictors from a single data block can be obtained. If more than one predictor blocks but not all make up a covariate, that would be referred to as a locally common covariate, as opposed to a globally common covariate where all of the predictor blocks are involved in deriving the covariate (Måge et al., 2019). Elementwise sparsity indicates dropping individual predictors out of the model. Combining these two types of sparsity encouraged at different levels, only a subset of predictors within the blocks that are chosen by blockwise sparsity would be left in the model to make up a covariate. Common and distinctive covariates that are comprised of a small interpretable subset of predictors can therefore be found to represent the underlying predictor processes.

Objective function

In setting up the objective function of SCD-Cov-logR, the objectives for logistic regression and PCovR are combined. As discussed, for a binary outcome the log-odds are regressed on the covariates. Hence, the squared error pertaining to the outcome (the left term in (3.5)) is replaced by a negative log-likehood function based on the PCovR logistic regression model (first line in (3.6)). Furthermore, the two types of sparsity on the weights $\mathbf{W}_C$ are accomplished by imposing two different penalties. We employ the group lasso penalty (M. Yuan & Lin, 2006) which shrinks and sparsifies the weights at the block level, and the lasso penalty (Tibshirani, 1996) that does the same but for individual weights. This combination of penalties is also known as the sparse group lasso (Friedman, Hastie, & Tibshirani, 2010a; Simon, Friedman, Hastie, & Tibshirani, 2013). The objective of SCD-Cov-logR is to minimize the following loss function,

$$
\begin{aligned}
L(\mathbf{W}_C, \mathbf{P}_C^{(X)}, \mathbf{p}^{(g)}, p_0^{(g)}) = \frac{\alpha}{l_0} & \left[-\sum_i^I (g_i(p_0^{(g)} + \mathbf{x}_{Ci}^\top \mathbf{W}_C \mathbf{p}^{(g)}) - \log(1 + e^{(p_0^{(g)} + \mathbf{x}_{Ci}^\top \mathbf{W}_C \mathbf{p}^{(g)})})) \right] \\
& + \frac{1-\alpha}{\|\mathbf{X}_C\|_2^2} \sum_i^I \left\| \mathbf{x}_{Ci}^\top - \mathbf{x}_{Ci}^\top \mathbf{W}_C (\mathbf{P}_C^{(X)})^\top \right\|_2^2 \\
& + \sum_r^R \lambda_{Lr} \left| \mathbf{w}_{Cr} \right|_1 + \sum_r^R \sum_k^K \lambda_{Gr} \sqrt{J_k} \left\| \mathbf{w}_r^{(k)} \right\|_2 + \lambda_R \left\| \mathbf{p}^{(g)} \right\|_2^2
\end{aligned}
\tag{3.7}
$$

where the loadings associated with the predictors $\mathbf{P}_C^{(X)}$ are constrained to be column-orthogonal ($(\mathbf{P}_C^{(X)})^\top \mathbf{P}_C^{(X)} = \mathbf{I}_R$) in order to identify the solution (and to avoid an ill-posed problem resulting in ever-decreasing weights compensated by ever-increasing loadings). l_0 refers to the negative log-likelihood of the null model fitted without any predictors $l_0 = -\sum_i^I (g_i \text{log}(\bar{p}) + (1-g_i)\text{log}(1-\bar{p}))$, where $\bar{p} = \frac{1}{I}\sum_i^I g_i$ is the proportion of observations in the first category. The terms with λ_{Gr} and λ_{Lr} refer to the group lasso and the lasso penalties corresponding to the rth covariate. $\mathbf{w}_r^{(k)}$ indicates the weights corresponding to the covariate r and the predictor block k. The last term denotes the ridge penalty imposed on the regression coefficients $\mathbf{p}^{(g)}$ to prevent divergence occuring due to covariates being correlated.

The first term of the loss function represents the negative log-likelihood function based on (3.6). It is in the same format as the negative log-likehood function commonly used for logistic regression, except that it has been adapted according to the multiblock PCovR model structure. This term is divided by the log-likelihood of the null model[2] l_0, while the second term of sum of squared predictor errors is divided by the total sum of squared predictor scores. The two types of losses are therefore placed within a comparable scale between 0 and 1. With respect to the penalties on the weights, it can be seen that the group lasso penalty $\|\cdot\|_2$ concerns a group of weights connecting the predictors in the kth predictor block with the rth covariate, while the lasso penalty $|\cdot|_1$ is imposed on all of the $\sum_{k=1}^K J_k$ individual weights corresponding to rth covariate. The two penalties together make up the sparse group lasso.

It is possible to re-express the objective function by scaling the α parameter such that it already takes account of the negative log-likelihood of the null model l_0 and the sum of squared predictor scores $\|\mathbf{X}_C\|_2^2$. The scaled weighting parameter β is defined by:

[2] This ratio of negative log-likelihoods is used in computation of McFadden's pseudo R^2 (McFadden et al., 1973) that provides insight on explained variance in the context of logistic regression.

$$\beta = \frac{\alpha \left\| \mathbf{X}_C \right\|_2^2}{\alpha \left\| \mathbf{X}_C \right\|_2^2 + (1-\alpha) l_0} \tag{3.8}$$

β can then replace $\frac{\alpha}{l_0}$ in the objective function (3.7) while $(1-\beta)$ replaces $\frac{(1-\alpha)}{\|\mathbf{X}_C\|_2^2}$, leading to a different expression of the same objective. Such rescaling of the weighting parameter has been shown in Vervloet et al. (2013).

Relation to existing methods

Several existing methods rely on objective functions that are similar to the objective introduced here in (3.7). A method called Sparse Principal Component Regression (SPCR; Kawano, Fujisawa, Takada, & Shiroishi, 2018) has been proposed and combined with generalized linear modelling. SPCR and SCD-Cov-logR are characterized by similar objective functions; our method can be viewed as an extension of SPCR for the setting of multiple predictor blocks. Likewise, several other methods can be seen as a special case of the objective function in (3.7). First, if the balancing parameter α is fixed at zero, common and distinctive sparse covariates would be found only optimizing the fit to the predictor variables. This solution would be equivalent to that of SCaDS (de Schipper & Van Deun, 2018), which finds common and distinctive sparse components from multiblock data. For this reason, and also because the algorithm for SCD-Cov-logR is infeasible when α is equal to exactly zero, we rely on SCaDS to find the solutions when $\alpha = 0$. Second, if the negative log-likelihood term is replaced by squared error pertaining to a continuous outcome ($\left\| \mathbf{y} - \mathbf{X}_C \mathbf{W}_C \mathbf{p}^{(y)} \right\|_2^2 / \left\| \mathbf{y} \right\|_2^2$), the objective function becomes that of SCD-CovR (S. Park et al., 2020), which shares the same aims as SCD-Cov-logR except it targets a continuous outcome. Third, starting from the SCD-CovR formulation, if the group lasso parameter is fixed at zero and only a single block of predictors are employed, the problem boils down to SPCovR (Van Deun et al., 2018) which finds sparse covariates. As these methods serve as the basis for the current SCD-Cov-logR, further details of these directly related methods are provided in Appendix (3.A). Finally, fixing the lasso and group lasso parameters at zero such that weights are found without sparsity, the problem can be seen as an extension to PCovR to account for a binary classification problem.

Algorithm

The minimizing solution of Equation (3.7) can be found by an alternating procedure where the loadings $\mathbf{P}_C^{(X)}$ and the regression coefficients $\mathbf{p}^{(g)}$ and $p_0^{(g)}$ are solved for conditional upon fixed values for the weights $\mathbf{W}_C$ and vice versa. Such an alternating approach has been effective for SCaDS, SCD-CovR and SPCovR. To treat the minimization of (3.7) which is complicated by the negative log-likelihood term, we make use of a local quadratic approximation, similar to the iteratively reweighted least squares approach that is usually taken to solve the logistic regression problem (Friedman, Hastie, & Tibshirani, 2010b). The alternating routine continues until the algorithm converges to a stationary point, usually a local minimum. Since the iteratively reweighted least squares procedure is known to sometimes lead to divergence, we also employ the maximum number of iteration of 5000 as another form of stopping criterion. As the objective function (3.7) is not a convex problem, it is subject to local minima. We recommend to use multiple random starting values, along with rational starting values based on PCovR (administered by treating the binary outcome as a continuous variable). Furthermore, employing multiple starting values is particularly important because the estimation of $\mathbf{W}_C$ is often a high dimensional regression problem prone to instable estimates (Guerra-Urzola et al., 2021; Jia & Yu, 2010), meaning that different starting values may result in different estimates. The sparse group lasso problem for $\mathbf{W}_C$ is treated via coordinate descent (Friedman et al., 2010a), while closed-form solutions exist for the conditional updates of $\mathbf{P}_C^{(X)}$, $\mathbf{p}^{(g)}$ and $p_0^{(g)}$. Further details on the algorithm for minimizing the objective function can be found in the Appendix (3.B), including the schematic outline of the algorithm and the derivation of solutions to the conditional updates (3.C, 3.D).

3.2.2.4 Multiclass classification

Our method can be slightly adapted to address a classification problem in the presence of more than two categories. The method is posed in the same manner as the binary problem, except it relies on multinomial logistic regression. The logit model in (3.6) is generalized to a 'baseline-category logit model' (Agresti, 2003) which is a common approach to extend logistic regression to a multiclass problem. Let $p(g_{im} = 1)$ and $p(g_{iM} = 1)$ denote the probability that subject i would fall under the category m and the last category M, respectively. Treating the last category as the baseline, the log-odds of the ith observation being in category m as opposed to being in the baseline category is modelled:

$$\begin{aligned} \log\left(\frac{p(g_{im}=1)}{p(g_{iM}=1)}\right) &= \mathbf{x}_{Ci}^{\top}\mathbf{W}_C\mathbf{p}_m^{(g)} + p_{0m}^{(g)}, \text{ for } m = 1, \dots, M-1 \\ \mathbf{x}_{Ci} &= \left[\mathbf{x}_{Ci}^{\top}\mathbf{W}_C(\mathbf{P}_C^{(X)})^{\top}\right]^{\top} + \mathbf{e}_i^{(X)}, \end{aligned} \tag{3.9}$$

where $\mathbf{p}_m^{(g)}$ and $p_{0m}^{(g)}$ refer to the regression coefficients and the intercept that correspond to category m. By calculating $M-1$ sets of the regression coefficients, the log-odds of any pairs of response categories can be determined. As for the objective function, the negative log-likelihood function based on the baseline-category logit model replaces the negative log-likelihood concerning the binary classification provided in (3.7):

$$
\begin{aligned}
&L(\mathbf{W}_C, \mathbf{P}_C^{(X)}, \mathbf{p}_m^{(g)}, {p_0}_m^{(g)}) \\
&= \frac{\alpha}{l_0}\left[-\sum_i^I \left\{\sum_m^{M-1} g_{im}({p_0}_m^{(g)} + {\mathbf{x}_C}_i^\top \mathbf{W}_C \mathbf{p}_m^{(g)}) - \log(1 + \sum_m^{M-1} e^{({p_0}_m^{(g)} + {\mathbf{x}_C}_i^\top \mathbf{W}_C \mathbf{p}_m^{(g)})})\right\}\right] \\
&+ \frac{1-\alpha}{\|\mathbf{X}_C\|_2^2} \sum_i^I \left\| {\mathbf{x}_C}_i^\top - {\mathbf{x}_C}_i^\top \mathbf{W}_C (\mathbf{P}_C^{(X)})^\top \right\|_2^2 \\
&+ \sum_r^R \lambda_{Lr} \left|\mathbf{w}_{Cr}\right|_1 + \sum_r^R \sum_k^K \lambda_{Gr} \sqrt{J_k} \left\|\mathbf{w}_r^{(k)}\right\|_2 + \lambda_R \left\|\mathbf{p}^{(g)}\right\|_2^2
\end{aligned}
\tag{3.10}
$$

where the loadings $\mathbf{P}_C^{(X)}$ are constrained to be column-orthogonal ($(\mathbf{P}_C^{(X)})^\top \mathbf{P}_C^{(X)} = \mathbf{I}_R$) as done for the binary problem (3.7). Other quantities and penalty terms are also defined the same. l_0 here refers to the negative log-likelihood of the null model $l_0 = -\sum_i^I \left[\sum_m^{M-1} g_{im}\log(\bar{p}_m) + g_{iM}\log(\bar{p}_M)\right]$ where $\bar{p}_m = \frac{1}{I}\sum_i^I g_{im}$ is the proportion of observations in the mth category. Hence, the negative log-likelihood and the sum of squared errors are also scaled in this objective function. The weighting parameter α can be rescaled to β in the same manner as for the binary classification problem (see (3.8)). Furthermore, note that both the model (3.9) and the objective function (3.10) become equal to those of the binary problem (3.6), (3.7) when the total number of categories M are set at two. To find the minimizing solution of (3.10), an alternating algorithm very similar to that for the binary problem is employed. The only difference is that the negative log-likelihood term with multiple categories is treated with partial quadratic approximation with respect to the category m where only $\mathbf{p}_m^{(g)}$ and ${p_0}_m^{(g)}$ are allowed to vary at a time. This partial quadratic approximation has been used for treating a penalized multinomial logistic regression problem (Friedman et al., 2010b). Details on the algorithm are provided in the Appendix (3.E).

3.2.3 Toy example

In order to provide a clearer picture of the goals that the method targets and the estimates it provides, we showcase the method on a toy example dataset for a binary classification problem in this section. We generated the dataset according to one of the conditions of the simulation study which follows later. The dataset is composed of two data blocks and its underlying model assumes three covariates. Two of these covariates represent processes that are distinctive to the first and the second data blocks respectively, while the third covariate is a common process, affiliated with both data blocks. In addition, the model was defined such that the covariate distinctive to the second block is not relevant in the classification of the outcome variable. Each of the two data blocks consists of 15 predictors concerning the same set of 100 observation units. There is one binary outcome variable. Details of the data generation setup can be found in the simulation study section.

A few technicalities come with the application of the SCD-Cov-logR to data. First, it is important to note that the solution is influenced by several tuning parameters that need to be fine-tuned via model selection. Second, also different starting values may yield different solutions because the algorithm can converge to a local minimum. The model selection procedure we adopted to find the solutions presented in the following will be discussed in the next section, along with our consideration regarding multiple starting values. Third, a pre-processing step precedes method application. All of the predictor variables are centered and scaled to unit sum of squares. Subsequently, the different predictor blocks are weighted such that the sum of squares are equal across the blocks, in order to account for the differing block sizes.

The estimates retrieved by the method along with the population parameters used to generate the dataset are provided in Table 3.1. It first shows that the weights $\hat{\mathbf{W}}_C$ are found sparse and correctly reflect the population weights zero-nonzero structure. Most of the estimated weights are smaller in magnitude than the population weights because the lasso and group lasso penalties not only enforce sparsity but also shrink the coefficients towards zero. The weights are interpreted as the coefficients in the linear combination that forms the covariates from the predictor variables; $t_{ir} = \sum_j w_{jr} x_{ij}$. Therefore, the weights correctly represent that the first two covariates are distinctive for each of the data blocks while the third is common. The logistic regression coefficients and the intercept $\hat{\mathbf{p}}^{(g)}$ and $\hat{p}_0^{(g)}$ are also obtained and are in agreement with the population parameters; the covariate distinctive to the second data block is much less relevant than the other covariates in the classification problem. These coefficients can be combined with the covariates to yield the predicted log-odds; $\sum_r (\hat{p}_r^{(g)} \hat{t}_{ir}) + \hat{p}_0^{(g)} = \hat{y}_i$. The inverse-logistic function (3.2) is used to transform the $\hat{y}_i$ log-odds into predicted probabilities for the categories of the outcome variable; if the probability is larger than 0.5, the class predicted by the model is 1. Let us take an example of the first observation $\mathbf{x}_{C1}$, the covariate scores of this observation $\hat{\mathbf{t}_1} = {\mathbf{x}_{C1}}^\top \hat{\mathbf{W}}_C = [2.875, 0.046, 3.384]^\top$ are combined with the regression coefficients to get the predicted log odds $\sum_r (\hat{p}_r^{(g)} \hat{t}_{1r}) + \hat{p}_0^{(g)} = \log\left(\frac{p(\hat{g_1}=1)}{1-p(\hat{g_1}=1)}\right) = 0.862$. Applying the inverse logistic function, the predicted probablity for this observation to be classified as 1 is $\frac{1}{1+e^{-0.862}} = 0.703$. Since this probability is larger than 0.5, we predict the observation as being in class 1, which is indeed true for the first observation in our toy example dataset.

Table 3.1. Population weights, and the solution found by SCD-Cov-logR from the toy example dataset: weights and logistic regression coefficients. The column names D1, D2 and C indicate that the corresponding covariate is defined as being distinctive to block 1, distinctive to block 2 and common.

$\mathbf{W}_C$

	D1	D2	C
Block 1			
x1	0.5	0	0
x2	0.5	0	0
x3	0.5	0	0
x4	0.5	0	0
x5	0	0	0.354
x6	0	0	0.354
x7	0	0	0.354
x8	0	0	0.354
x9	0	0	0
x10	0	0	0
x11	0	0	0
x12	0	0	0
x13	0	0	0
x14	0	0	0
x15	0	0	0
Block 2			
x16	0	0	0.354
x17	0	0	0.354
x18	0	0	0.354
x19	0	0	0.354
x20	0	0.5	0
x21	0	0.5	0
x22	0	0.5	0
x23	0	0.5	0
x24	0	0	0
x25	0	0	0
x26	0	0	0
x27	0	0	0
x28	0	0	0
x29	0	0	0
x30	0	0	0

$\hat{\mathbf{W}}_C$

	D1	D2	C
Block 1			
x1	0.358	0	0
x2	0.391	0	0
x3	0.463	0	0
x4	0.475	0	0
x5	0	0	0.359
x6	0	0	0.319
x7	0	0	0.276
x8	0	0	0.233
x9	0	0	0
x10	0	0	0
x11	0	0	0
x12	0	0	0
x13	0	0	0
x14	0	0	0
x15	0	0	0
Block 2			
x16	0	0	0.358
x17	0	0	0.401
x18	0	0	0.342
x19	0	0	0.307
x20	0	0.483	0
x21	0	0.415	0
x22	0	0.381	0
x23	0	0.453	0
x24	0	0	0
x25	0	0	0
x26	0	0	0
x27	0	0	0
x28	0	0	0
x29	0	0	0
x30	0	0	0

Logistic regression coefficients

Population	
D1	-0.600
D2	-0.010
C	0.800
intercept	0
Estimated	
D1	-0.735
D2	-0.072
C	0.907
intercept	-0.090

Altogether, examining this solution, it would be concluded that there are two underlying predictor processes that exclusively involve predictor variables of only one of the two data blocks and one process that involves predictors from both data blocks. Predictors x9 to x15 and x24 to x30 are filtered out of the model; they are not related with any of these processes. Only two processes out of the three are important in classifying the binary outcome variable. The predictor process distinctive to the second data block is irrelevant for the classification problem. Concerning the performance of classifying the outcome, the method classified 92 in-sample observations. To gauge the quality of predicting the classes of unseen data, we applied the fitted model to 100 observations of out-of-sample data that were generated from the same population as the in-sample observations. The method was able to classify 92 out-of-sample observations correctly.

3.2.4 Model selection

The SCD-Cov-logR method involves several (usually) unknown parameters that govern the characteristics of the derived model; the number of covariates R, the weighting parameter α, the lasso and group lasso parameters $\lambda_{Lr}, \lambda_{Gr}$ for the sparse weights and the ridge parameter λ_R for the logistic regression coefficients. These parameters are usually tuned in accordance to a certain optimality criterion such as prediction error. Several model selection strategies can be used for different model parameters, while we adopt cross-validation for all of the parameters except for the number of covariates. A straightforward way to administer cross-validation is the grid search that exhaustively compares all possible combinations of the ranges of values for the different parameters in optimizing the criterion of cross-validation error. However, as the current method entails many parameters to be tuned, such a scheme involves a very heavy computational load. Instead, a sequential approach where sets of parameters are tuned in turn can be considered as it was demonstrated to work well for model selection for PCovR (Vervloet et al., 2016) and also for SCD-CovR (S. Park et al., 2020). In the following, we propose a sequential cross-validation model selection procedure and demonstrate it with the toy example dataset.

The first step of the sequential approach is to determine the number of covariates. This was recommended in a study that compares model selection strategies for PCovR (Vervloet et al., 2016). S. Park et al. (2020) also selected the number of covariates first and obtained models with good performance in SCD-CovR. For finding the number of covariates in SCD-Cov-logR, we first perform PCA on the predictor variables with varying number of principal components. Instead of the well-known scree test that manually looks for an ‘elbow’ in the plot of eigenvalues (representing the amount of variance explained by each principal component) which involves an element of subjectivity, the acceleration factor technique proposed by Raîche, Walls, Magis, Riopel, and Blais (2013) is adopted. It finds the elbow by computing at which point the slope of the graph of eigenvalues change most sharply. The technique retains the principal components that derived prior to the principal component where the sharp change in slopes occurs. The R package "nFactors" is employed for this purpose (Raiche, Magis, & Raiche, 2020).

With the number of covariates fixed, cross-validation is administered to simultaneously select the optimal values of α and λ_R. For each combination of values, the mean of squared residuals is computed. These residuals are discrepancies between the binary outcome scores of the observations in held-out samples and their corresponding predicted probabilities computed by:

$$\frac{1}{n}\sum_{i}^{n}\left(g_i - 1/\left(1 + e^{-({\mathbf{x}_C}_i^\top \hat{\mathbf{W}}_C \hat{\mathbf{p}}^{(g)} + \hat{p}_0^{(g)})}\right)\right)^2$$

where n denotes the size of the held-out samples. In the case of the multiclass problem, the following equation:

$$\frac{1}{n(M-1)}\sum_{m}^{M-1}\sum_{i}^{n}\left[g_{im} - e^{{\mathbf{x}_C}_i^\top \hat{\mathbf{W}}_C \hat{\mathbf{p}}_m^{(g)} + \hat{p_0}_m^{(g)}} / \left(1 + \sum_{m}^{M-1} e^{{p_0}_m^{(g)} + {\mathbf{x}_C}_i^\top \mathbf{W}_C \mathbf{p}_m^{(g)}}\right)\right]^2$$

is employed to compute the residuals. The one standard error rule (Friedman, Hastie, Tibshirani, et al., 2001) is adopted, which selects the least complex model within one standard error of the best-performing model. For α, higher values are associated with model complexity and overfitting because it places a heavier emphasis on the prediction problem of the outcome which becomes prone to overfitting with increasing number of predictor variables (Babyak, 2004; McNeish, 2015). Similarly, lower values of λ_R are related with overfitting as it leads to high variance of parameter estimates across samples. Therefore, the one standard error rule aims to select the models with the lowest α and the highest λ_R values. When the two parameters are not in agreement, the model with lower α is preferred over the model with higher λ_R as the former is seen to exert more impact on the final model. Note that the rescaled parameter β can be tuned instead of directly tuning for α. Higher values of β are related to overfitting, in the same manner as for α. The one standard error rule would thus choose the models comprised with the lowest β and the highest λ_R values in this case.

We tune the sparsity parameters for the weights at the final stage of the model selection procedure because they exert relatively small influences on the fit of the model with respect to both classification or reconstruction of the blocks of predictor variables (de Schipper & Van Deun, 2021; S. Park et al., 2020). In a paper that examined the efficacy of various model selection strategies for sparsity penalty parameters in sparse PCA that retrieves sparse weights like SCD-Cov-logR, it was reported that even a very sparse model yields good recovery of summary component scores (de Schipper & Van Deun, 2021). The authors advise using cross-validation with the one standard error rule to select the parameters, when the aim of the analysis includes understanding of underlying processes. For our proposed method, the one standard error rule is set up such that the model with the highest values of λ_{Lr} and λ_{Gr} are chosen within models with minimal cross-validation error. Between the two parameters, the model with higher λ_{Lr} is preferred over the model with higher λ_{Gr} because λ_{Lr} encourages the sparse solution in a more direct manner than λ_{Gr}. While different values of the parameters can be specified concerning the weights corresponding to each of the rth covariate, we usually adopt the same values across multiple covariates to ease the computational burden. Additionally, in choosing the ranges of sparsity parameters to be considered for model selection, values separated by a reasonable interval can be selected between a near-zero value and another value that leads to complete sparsity. One way to choose such an interval is by selecting a sequence of equally spaced values on the log scale, as done in de Schipper and Van Deun (2021) and recommended in Friedman et al. (2010b).

Model selection for the toy example

We demonstrate the model selection procedure by applying it on the toy example dataset. First, PCA is administered to the concatenated set of centered and standardized predictor variables with various numbers of principal components. Figure 3.4 in Appendix 3.F depicts the variance explained by each component. With the acceleration factor technique, the number of covariates is chosen to be three because the sharpest change in the slopes occurs at the fourth principal component. With the number of covariates fixed, we administered a 5-fold cross-validation, simultaneously varying the values of β and λ_R. Instead of directly controlling the values for α, we varied the values for its rescaled version β. The parameters λ_{Lr} and λ_{Gr} were fixed at zero for the cross-validation. We considered the values of [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] for β and [0.1, 0.5, 1, 3, 5, 10, 30, 50] for λ_R. With the one standard error rule, a β value of 0.2 and λ_R of 1 was selected. Given these parameters, we finally conducted another 5-fold cross-validation for λ_{Lr} and λ_{Gr}. The range of [0.5, 1, 5, 7, 10, 15, 30, 45, 100] was employed for λ_{Lr} and [0.1, 0.5, 1, 2, 5, 10] for λ_{Gr}. The one standard error rule selected the model with $\lambda_{Lr} = 45$ and $\lambda_{Gr} = 2$. The solution provided above in Table 3.1 was obtained by adopting these values for the analysis of the data. It is worth noting that using an exhaustive approach to cross-validation that considers all combinations of these ranges of parameters also resulted in models that are similar to this reported model. The results from this exhaustive approach can be found in Appendix (3.G).

In the above model selection procedures, rational starting values (i.e., the PCovR solution) were used in initializing the SCD-Cov-logR algorithm. To account for the problem of local minima, 20 different sets of random starting values were generated. Using each set of starting values, we conducted the same model selection procedures to find the tuning parameters and the final model estimates. We found that the solution resulted from the rational starting values were associated with the lowest minimum, compared with the other starting values. Comparing the estimates obtained by different starting values, although some starting values yielded estimates that are quite different from those of the rational starting values, the starting values that resulted in smaller loss led to estimates that are very similar to those of the rational starting values. These estimates also correctly classified the same numbers of in-sample and out-of-sample observations as the estimates from the rational starting values. Since the rational starting values led to the lowest minimum, we reported these estimates in the previous section. It also seems sensible that the rational starting values from PCovR finds a lower minimum because the data was generated from a clear PCovR model structure (as seen in the Simulation Study section). However, in practice, it is recommended to adopt multiple random starting values and the rational values to initialize the algorithm and subsequently choose the solution that attains the lowest minimum. This applies especially if the underlying true model structure is unknown, unlike for the current toy example.

3.2.5 Related methods

SCD-Cov-logR is a classification method with three main objectives. It (a) classifies a categorical outcome, (b) recovers the underlying common and distinctive predictor processes via dimension reduction, and (c) derives sparse weights and therefore interpretable covariates. The method offers a solution that achieves all of these objectives in a flexible manner such that the user can emphasize one goal over another according to the research aim. In this section, we will present two methods that are related to SCD-Cov-logR, in the sense that they target a similar set of goals. Alongside, regularized logistic regression is also discussed as a benchmark method for classification with a large set of predictors.

3.2.5.1 PCR (logistic regression)

A commonly used method that aims both at classification and modeling the variation in the block of predictors is based on principal component regression (PCR; see Jolliffe, 1982). This method first performs PCA on the predictors and then, in a second and separate step, builds a classification model using the retrieved components as the predictor variables. In order to derive common and distinctive processes from multiblock data, the PCA step can be conducted with

SCaDS (de Schipper & Van Deun, 2018). We will refer to this two-step approach of SCaDS followed by logistic regression by SCaDS-logR. As discussed above, this is the special case of SCD-Cov-logR with the weighting parameter α is specified at zero. Hence, it addresses the same research goals of SCD-Cov-logR, except that it does not take the outcome variable into consideration when deriving the components. Due to this, the underlying processes that play important roles for the outcome variable rather than the predictor variables may be omitted (Vervloet et al., 2016).

3.2.5.2 DIABLO

Data Integration Analysis for Biomarker discovery using a Latent component method for Omics (DIABLO; Singh et al., 2016) is a partial least squares (PLS)-based framework that addresses the multiple aims of prediction and sparse modeling of the variation in the predictors. PLS (H. Wold, 1982; S. Wold et al., 1983) is a widely used method that has the same model structures as PCovR; it finds components that represent the underlying processes among the predictors while predicting the outcome variable. PLS can also be seen as an approach to Structural Equation Modelling (SEM) when complex models are built without being mainly guided by theory (M. Tenenhaus, Tenenhaus, & Groenen, 2017). DIABLO is an extension of PLS that jointly analyzes multiple predictor blocks and obtains sparse components. Simultaneously, these sparse components explain the variation in the outcome variable. Therefore, DIABLO meets all of the research aims of SCD-Cov-logR. While our proposed method treats the multiblock problem by concatenating the predictor matrix to construct a single model that covers several data blocks, DIABLO derives one model separately for each data block; predictions from each model are accumulated via majority voting to give the overall classification. Therefore, DIABLO can be seen to only find components that are distinctive to each block. However, it is possible to specify how correlated these components built on each block would be. This would encourage capturing of the variance accounted for by common predictor processes, although they may not be explicitly obtained. Singh et al. (2016) demonstrated that when building a classification model for breast cancer subtypes with predictors from multiple data blocks (mRNA, miRNA, methylation and proteins) from The Cancer Genome Atlas (TCGA), DIABLO was able to select more variables that are strongly correlated with each other than elastic net regression.

Another core difference between SCD-Cov-logR and DIABLO lies with the parameter α that balances between reconstruction of the predictors and prediction of the outcome variable. PLS-based methods do not offer such an option and tend to lean closer to a PCovR model emphasizing prediction, this is α close to one

(Van Deun et al., 2018; Vervloet et al., 2016). Furthermore, methods based on PLS are often more prone to overfitting than those derived from PCovR, which in turn results in a diminished quality of out-of-sample prediction. The results from S. Park et al. (2020) demonstrated this pattern of results in a multiblock regression setting.

Moreover, DIABLO does not adopt a generalized linear model framework to treat the classification of categorical outcome variables. Instead, when constructing a classification model, DIABLO adopts a simple heuristic where the categorical outcome is coded into a binary matrix with each column indicating the membership of the observation unit in a certain class. The classification model is then estimated in the same manner as the regression model by treating the binary matrix as continuous outcome variables. Among the fitted values given for each of the classes, the class that corresponds to the largest fitted value is the class determined by the DIABLO model. This approach of administering PLS for a classification problem has also been shown to be equivalent to performing discriminant analysis (Barker & Rayens, 2003). There are PLS methods that are formulated in combination with the generalized linear model framework such that a logistic regression model can be constructed (Chung & Keles, 2010; Ding & Gentleman, 2005), but these methods are only suitable for the analysis of a single data block. Additionally, Lê Cao, Boitard, and Besse (2011) reported that this approach performs comparatively with the binary indicator matrix approach of DIABLO.

3.2.5.3 Regularized logistic regression

Regularized logistic regression is a logistic regression method that performs variable selection (Friedman et al., 2010b). Due to the regularization penalties, the method can also be applied to high dimensional datasets. Hence, it can be considered as a benchmark method for classification in the setting of many predictors, being actively applied in behavioural sciences; for example to detect psychological symptom patterns from large-scale questionnaires (Tutun et al., 2019) and to classify different emotions using EEG signal patterns (D.-W. Chen et al., 2020). However, since it does not extract covariates or factors, the method does not meet all of the aims of SCD-Cov-logR such as identifying the underlying processes governing the predictors.

3.2.5.4 Toy example illustration

In order to compare the two related methods that share the goals of SCD-Cov-logR, we administered them along with the benchmark of regularized logistic regression on the toy example dataset. As the population model parameters are known, we configured the methods such that they return the solutions that reflect the population model structure as closely as possible. For regularized logistic regression, the lasso penalty parameter was tuned by cross-validation, as it is not possible for the method to derive the covariate structures. For principal component (logistic) regression, we administered SCaDS (de Schipper & Van Deun, 2018) on the predictor matrix with three components. Lasso and group lasso parameters were chosen such that they reflect the population model. The outcome variable was regressed on the derived sparse principal components via logistic regression.

In order to fit the DIABLO model in accordance with the population model such that the common and distinctive predictor processes can be explicitly found, we fitted a 1-component model separately from each of the two data blocks which would match the two distinctive covariates generated. For the common covariate, we constructed a 1-component model from a supermatrix that concatenates the two data blocks. These components across the blocks were specified to be uncorrelated, as the true covariates were defined to be uncorrelated. As DIABLO allows the users to specify the number of non-zero weights per component, we specified these in correspondence with the number of non-zero weights in the true weights matrix.

Table 3.2 presents the estimates resulting from the different methods. The table shows that only the two-step principal component logistic regression approach of SCaDS-logR finds the covariates that perfectly represent the population model structure. DIABLO can find the distinctive covariates, but does not perform well at correctly finding the non-zero parameters. It is difficult to interpret the regularized logistic regression coefficients as they do not go hand-in-hand with the population model. However, it can be seen that the predictors that do not have any relations with the covariates were filtered out, yet, also some of the predictors that do have a relation with the covariates were also filtered out.

With respect to the performance to classify the outcome variable, the number of correctly classified in-sample and out-of-sample observations for each of the methods are provided in Table 3.3. The results pertaining to SCD-Cov-logR are also given to offer comparison. It appears that SCD-Cov-logR and SCaDS-logR lead to comparable and good predictive performances, although the four methods don’t exhibit large differences.

Extending this comparative evaluation of the related methods and SCD-Cov-logR to a simulation study requires comparison of the methods on all criteria that reflect the multiple research aims of SCD-Cov-logR. The benchmark regularized logistic regression does not meet this requirement since it fails to meet all of the research aims; it does not uncover underlying predictor processes via structures such as covariates. While both PCR (SCaDS-logR) and DIABLO address the aims,

Table 3.2. Estimates provided by PCR, DIABLO and regularized logistic regression. The true weights $\mathbf{W}_C$ is also provided as a reference.

	$\mathbf{W}_C$			SCaDS-logR			DIABLO			LogR
	D1	D2	C	D1	D2	C	D1	D2	C	b
Block 1										
x1	0.5	0	0	0.392	0	0	0	0	0	-0.198
x2	0.5	0	0	0.399	0	0	0	0	0	-0.304
x3	0.5	0	0	0.430	0	0	0	0	-013	-0.262
x4	0.5	0	0	0.496	0	0	0	0	0	-0.112
x5	0	0	0.354	0	0	0.328	0.606	0	0.480	0.265
x6	0	0	0.354	0	0	0.291	0.411	0	0.330	0.336
x7	0	0	0.354	0	0	0.262	0.636	0	0.502	0.333
x8	0	0	0.354	0	0	0.217	0.242	0	0.200	0.221
x9	0	0	0	0	0	0	0	0	0	0
x10	0	0	0	0	0	0	0	0	0	0
x11	0	0	0	0	0	0	0	0	0	0
x12	0	0	0	0	0	0	0	0	0	0
x13	0	0	0	0	0	0	0	0	0	0
x14	0	0	0	0	0	0	0	0	0	0
x15	0	0	0	0	0	0	0	0	0	0
Block 2										
x16	0	0	0.354	0	0	0.357	0	0.537	0.364	0.180
x17	0	0	0.354	0	0	0.370	0	0.533	0.353	0.189
x18	0	0	0.354	0	0	0.311	0	0.525	0.335	0.232
x19	0	0	0.354	0	0	0.281	0	0.389	0	0
x20	0	0.5	0	0	0.443	0	0	0	0	0
x21	0	0.5	0	0	0.424	0	0	0	0	0
x22	0	0.5	0	0	0.419	0	0	0	0	0
x23	0	0.5	0	0	0.479	0	0	0	0	0
x24	0	0	0	0	0	0	0	0	0	0
x25	0	0	0	0	0	0	0	0	0	0
x26	0	0	0	0	0	0	0	0	0	0
x27	0	0	0	0	0	0	0	0	0	0
x28	0	0	0	0	0	0	0	0	0	0
x29	0	0	0	0	0	0	0	0	0	0
x30	0	0	0	0	0	0	0	0	0	0

Table 3.3. Number of correctly classified observations provided by PCR, DIABLO and regularized logistic regression.

	SCD-Cov-logR	SCaDS-logR	DIABLO	LogR
In-Sample	92	91	83	87
Out-of-Sample	92	92	84	88

PCR has been compared in previous works against PCovR and showed underperformance in discovering the true covariate structure (Vervloet et al., 2016) and also in prediction of the outcome (Heij et al., 2007; Tu & Lee, 2019); the reason that PCR falls short is because its components are found without considering the outcome. Moreover, in the setting of multiple predictor blocks, PCovR resulted in better prediction of the outcome when some of the underlying predictor processes important for predicting the outcome only account for a small amount of variance in the predictors (S. Park et al., 2020). Therefore, in the simulation study section below, we evaluate the performance of our current method against the only competitor that accounts for all criteria, this is DIABLO.

3.2.6 Toy example multiclass problem

As an additional demonstration for our current method under a multiclass classification problem, we generated a toy example dataset again with a categorical outcome variable with 3 categories. The characteristics of the data and the underlying model were kept the same as the toy example above, except for the definition of the regression parameters and the number of observation units ($I = 1000$). Appendix (3.H) provides further details on the data generating setup. Out of the 3 categories, the third category was taken as the baseline category in forming the log-odds models. We administered the sequential model selection procedure as done for the binary problem, employing 5-fold cross-validation considering the same ranges of parameters as for the binary problem again (see section 3.2.4). The following model parameters were selected: $R = 3, \beta = 0.1, \lambda_R = 0.1, \lambda_{Lr} = 100$ and $\lambda_{Gr} = 10$. Table 3.4 shows the solution together with the defined population parameters used to generate the data. It can be seen that the estimated weights correctly represent the true underlying weights. The logistic regression coefficients found are also in agreement with the population parameters; two covariates important for discerning the categories from the third (baseline) category are correctly picked out. Moreover, the constructed model classified 842 in-sample observations and 845 out-of-sample observations correctly (both out of 1000 total observations).

Table 3.4. Population parameters and the solution found by SCD-Cov-logR from the toy example multiclass dataset. The column names D1, D2 and C indicate that the corresponding covariate is defined as being distinctive to block 1, distinctive to block 2 and common. The third category is chosen as the baseline category; the regression coefficients construct the log-odds of the first or the second category as opposed to the third.

$\mathbf{W}_C$

	D1	D2	C
Block 1			
x1	0.5	0	0
x2	0.5	0	0
x3	0.5	0	0
x4	0.5	0	0
x5	0	0	0.354
x6	0	0	0.354
x7	0	0	0.354
x8	0	0	0.354
x9	0	0	0
x10	0	0	0
x11	0	0	0
x12	0	0	0
x13	0	0	0
x14	0	0	0
x15	0	0	0
Block 2			
x16	0	0	0.354
x17	0	0	0.354
x18	0	0	0.354
x19	0	0	0.354
x20	0	0.5	0
x21	0	0.5	0
x22	0	0.5	0
x23	0	0.5	0
x24	0	0	0
x25	0	0	0
x26	0	0	0
x27	0	0	0
x28	0	0	0
x29	0	0	0
x30	0	0	0

$\hat{\mathbf{W}}_C$

	D1	D2	C
Block 1			
x1	0.485	0	0
x2	0.485	0	0
x3	0.475	0	0
x4	0.476	0	0
x5	0	0	0.345
x6	0	0	0.344
x7	0	0	0.348
x8	0	0	0.338
x9	0	0	0
x10	0	0	0
x11	0	0	0
x12	0	0	0
x13	0	0	0
x14	0	0	0
x15	0	0	0
Block 2			
x16	0	0	0.350
x17	0	0	0.345
x18	0	0	0.348
x19	0	0	0.349
x20	0	0.482	0
x21	0	0.475	0
x22	0	0.480	0
x23	0	0.482	0
x24	0	0	0
x25	0	0	0
x26	0	0	0
x27	0	0	0
x28	0	0	0
x29	0	0	0
x30	0	0	0

Logistic regression coefficients

	1	2
Population		
D1	0.600	0.950
D2	0.010	0.312
C	-0.800	0.010
intercept	0	0
Estimated		
D1	1.843	2.865
D2	-0.026	0.941
C	-1.966	0.015
intercept	0.033	-0.025

3.3 Simulation study

Through a simulation study, we study the performance of the SCD-Cov-logR and DIABLO with respect to retrieval of the underlying processes and the classification of a binary outcome variable. We focus on the binary classification problem as the multiclass problem is a direct extension of the binary problem; it is expected that the insights obtained from the binary problem to be applicable for the multiclass problem. We hypothesize that SCD-Cov-logR would be better at out-of-sample classification than DIABLO as it is less susceptible to overfitting. SCD-Cov-logR would also provide models that better reflect the true underlying predictor processes as it allows a good balance between explaining the predictors and the outcome via the weighting parameter.

3.3.1 Design and procedure

We relied on the data generating setup presented by Chung and Keles (2010) which was used for examining the performance of several variants of sparse PLS that were set up to address the classification problem. Fixing the number of observations I to 100, the setup was modified such that two blocks of predictor variables were generated from three underlying covariates. One distinctive covariate per each predictor block was defined, while the remaining covariate reflected a common process involving both of the blocks. The three covariates were defined to differ in relevance for predicting the outcome variable, in that only two of them were defined as being relevant. We generated $J = 200$ predictor variables (100 per data block) for the high dimensional setting and $J = 30$ (15 per data block) for the low dimensional. The following setup was used:

$$\begin{aligned}
&\mathbf{T} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{\Sigma} = 50^2\mathbf{I}_3)\\
&\mathbf{E} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{\Sigma}_E = \sigma^2\mathbf{I}_J)\\
&\mathbf{X}_C \leftarrow \mathbf{T}\mathbf{W}_C^\top + \mathbf{E}\\
&\mathbf{z} \leftarrow 1/(1 + exp(-\mathbf{T}\mathbf{p}^{(g)}))\\
&g_i \sim Bernoulli(z_i)
\end{aligned} \tag{3.11}$$

$\mathbf{T}$ is a $I \times 3$ covariate scores matrix drawn from a multivariate normal distribution defined with the mean vector fixed to $\mathbf{0}$ and a diagonal covariance matrix $\mathbf{\Sigma}$ with all of its diagonal elements fixed at 50^2. The three covariates are therefore uncorrelated. The columns of the $J \times 3$ weights matrix $\mathbf{W}_C$ is defined such that they reflect the defined common or distinctive nature of the corresponding covariates. For example, weights corresponding to a covariate distinctive to the first predictor block, are non-zero only for predictors in the first block while the remaining weights corresponding to predictors in the second block are all zero. Likewise, for a common covariate, non-zero weights are defined for predictors in both blocks. On top of these zero weights that determine the common or distinctive nature of the covariates, further sparsity is added by defining more elements of $\mathbf{W}_C$ as zeros. The sparsity levels of the weights matrix is fixed at 82% and 85%

for low and high dimensional settings, respectively. It is important to note that the weights matrix was constructed such that it is column-orthogonal: $\mathbf{W}_C^\top \mathbf{W}_C = \mathbf{I}_R$. Together with the covariates $\mathbf{T}$ which are orthogonally defined, this model corresponds to the well-known PCA decomposition where the weights are equal to the loadings (for discussion; Guerra-Urzola et al., 2021). This is why the weights $\mathbf{W}_C^\top$ in (3.11) linearly combine the covariates $\mathbf{T}$ to generate the predictors $\mathbf{X}_C$ in the same manner as loadings in PCA decomposition. An example of the population weights matrix in a low dimensional setting is presented in section 3.2.3 (Table 3.2) along with the toy example dataset, and the weights are defined in a similar manner for a high dimensional setting.

The predictors $\mathbf{X}_C$ are generated by multiplying the covariate scores matrix with the weights matrix and adding random error on top. The residual matrix $\mathbf{E}$ is generated from a multivariate normal distribution with zero mean vector and a diagonal covariance matrix $\mathbf{\Sigma}_E$ such that the residuals are uncorrelated with each other and also with the covariate scores. The variance of the error variables are adjusted according to one of the manipulated design factors of the simulation study: proportion of variance in $\mathbf{X}_C$ explained by the underlying covariates. $\mathbf{p}^{(g)}$ indicates the regression coefficients. g_i is sampled from a Bernoulli distribution with the probability defined by the linear combination of $\mathbf{T}$ and $\mathbf{p}^{(g)}$ transformed by the inverse-logitic function (see Equation (3.2)).

Based on this data generating model, we manipulated three data characteristics which are listed in the overview below. The different levels taken by these manipulated factors are provided between square brackets.

Study setup

1. Number of predictors J_k in each block: [100], [15]
2. Covariates relevant to the response $\mathbf{g}$: [D1, D2], [D1, C]
3. Proportion of variance in $\mathbf{X}_C$ explained by the covariates: [0.8], [0.5], [0.2]

The number of predictors manipulated by the first design factor determines whether the dataset would be low or high-dimensional. The second design factor indicates which covariates are relevant for the classification of the binary outcome with D1, D2 and C denoting the two distinctive and the common covariate, respectively. The relevance of the covariates is manipulated by specification of regression coefficients $\mathbf{p}^{(g)}$, which equals $[0.60, -0.80, 0.01]$ and $[0.60, 0.01, -0.80]$ for the two levels respectively. For the first level, the two distinctive covariates are made relevant in explaining the outcome variable, while the covariate distinctive to the first block and the common covariate are relevant in the second level. As stated above, the proportion of variance in the predictors accounted for by the covariates is controlled by the variance of the error variables $\mathbf{E}$. Fully crossing these factors and generating 50 datasets per condition, $2 \times 2 \times 3 \times 50 = 600$ datases were produced.

Two different analyses were administered to each of these datasets: SCD-Cov-logR and DIABLO. As done for DIABO for the toy example dataset, a 1-component model was fitted for each of the two data blocks to match the two distinctive covariates generated. For the common covariate, we constructed a 1-component model from a supermatrix that concatenates the two data blocks.

3.3.2 Model selection

As the true underlying structure of the datasets is already known, several tuning parameters were tailored to correspond to the true structure. For SCD-Cov-logR, the number of covariates was fixed at three. The weighting parameter α and the ridge penalty parameter λ_R that regularizes the logistic regression coefficients were tuned together via 5-fold cross-validation. As done in the toy example in section 3.2.4, we used the rescaled weighting parameter β instead of α. The ranges of [0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] and [0.5, 1, 5, 10, 30, 50] respectively were used for β and λ_R. We adopted the 1 standard error (SE) rule to select a set of parameters which provides the most general model among the set of parameters yielding errors within 1 SE from minimum cross-validation error. We chose the lowest β and the highest λ_R. For the toy example, the lasso λ_{Lr} and the group lasso λ_{Gr} parameters were fixed at zero while tuning β and λ_R. Instead, for the simulation study, they were fixed differently for various conditions of the simulation study to encourage retrieval of one common and two distinctive covariates (Appendix 3.I).

Finally, with values of β and λ_R fixed, in order to find the parameters λ_{Lr} and λ_{Gr} that match the population weights structure the closest, we fitted the method with a range of values for λ_{Lr} and λ_{Gr}. The ranges of [3, 5, 10, 15, 20, 30, 50, 80] and [0.5, 1, 2, 3, 5, 10] were adopted respectively for λ_{Lr} and λ_{Gr}. As in the toy example dataset, the datasets have been generated such that a PCovR model underlies the true sparse model structure. This means that the rational starting values are likely to provide a more optimal solution than random starting values. Therefore, we only employed the rational starting values based on PCovR.

For DIABLO, we specified the number of nonzero weights according to the defined model structure. As done for the toy example dataset, the components from different blocks were fitted such that they are not correlated. This is sensible because the true covariates are generated to be uncorrelated from each other.

3.3.3 Evaluation criteria

Because the methods have several objectives, including recovery of the underlying processes and classification of a binary outcome, two measures are used to study performance of the methods in relation to each of these objectives. The performance measures are:

1. Out-of-sample balanced error rate (BER): (false positive rate + false negative rate)/2.
2. Correct weights classification rate: proportion of the weights correctly classified as zero and non-zero elements relative to the total number of coefficients.

An independent test set (of 100 observation units) needed for computing the out-of-sample BER was generated following the same data generating procedures as the data used for model-fitting. A BER equal to zero indicates a perfect classification. The correct weights classification rate represents the method's ability in retrieving the underlying processes. SCD-Cov-logR provides weights matrix $\hat{\mathbf{W}}_C$ of size $\sum_{k=1}^{2} J_k \times R$ which covers the entire set of the multiblock predictors. For the weights provided by SCD-Cov-logR, we first computed Tucker congruence (L. R. Tucker, 1951) between the columns of the true $\mathbf{W}_C$ matrix and those of the estimated $\hat{\mathbf{W}}_C$ matrix. After matching the columns that resulted in the highest Tucker congruence to account for the permutational freedom of the covariates, the correct classification rate was calculated from the matching pairs of true and estimated $\mathbf{W}_C$ columns.

On the other hand, for DIABLO, one component each was estimated for the two predictor blocks and the concatenated supermatrix. Components derived from the individual predictor blocks naturally correspond to the true distinctive covariates. In order to calculate the correct classification rate, the weights estimated for these estimated components were compared against true weights that correspond to the true distinctive covariates. Likewise, the weights found from the concatenated supermatrix were compared against the true weights corresponding to the common covariate.

3.3.4 Results

3.3.4.1 Out-of-sample BER

We first examine the performance of the two methods concerning the prediction for new data. The estimates obtained by the methods from the training dataset are applied on the out-of-sample test set generated under equal conditions. The results from our simulation study arranged for each condition are displayed in Figure 3.1. It can first be seen that SCD-Cov-logR resulted in the smaller out-of-sample BER in almost all of the conditions. With regards to the manipulated design factors, the relevance of the covariates seems to have played an important role in different performances among the methods. When the two distinctive covariates are defined as being relevant, the discrepancy in the methods is smaller, but with the covariate distinctive to the first block and the common covariate relevant, the outperformance of SCD-Cov-logR stands out more prominently. The proportion of variance in $\mathbf{X}_C$ accounted for by the covariates resulted in the 'main effect' - with smaller proportion leading to higher BER for all of the methods. Finally, it appears that the discrepancy in the performance of the methods is larger when the dataset is high-dimensional. Overall, we conclude that SCD-Cov-logR outperforms DIABLO at predicting the classes of new observations. However, the methods present more comparable performance when the processes relevant for classification are distinctive, under low dimensionality.

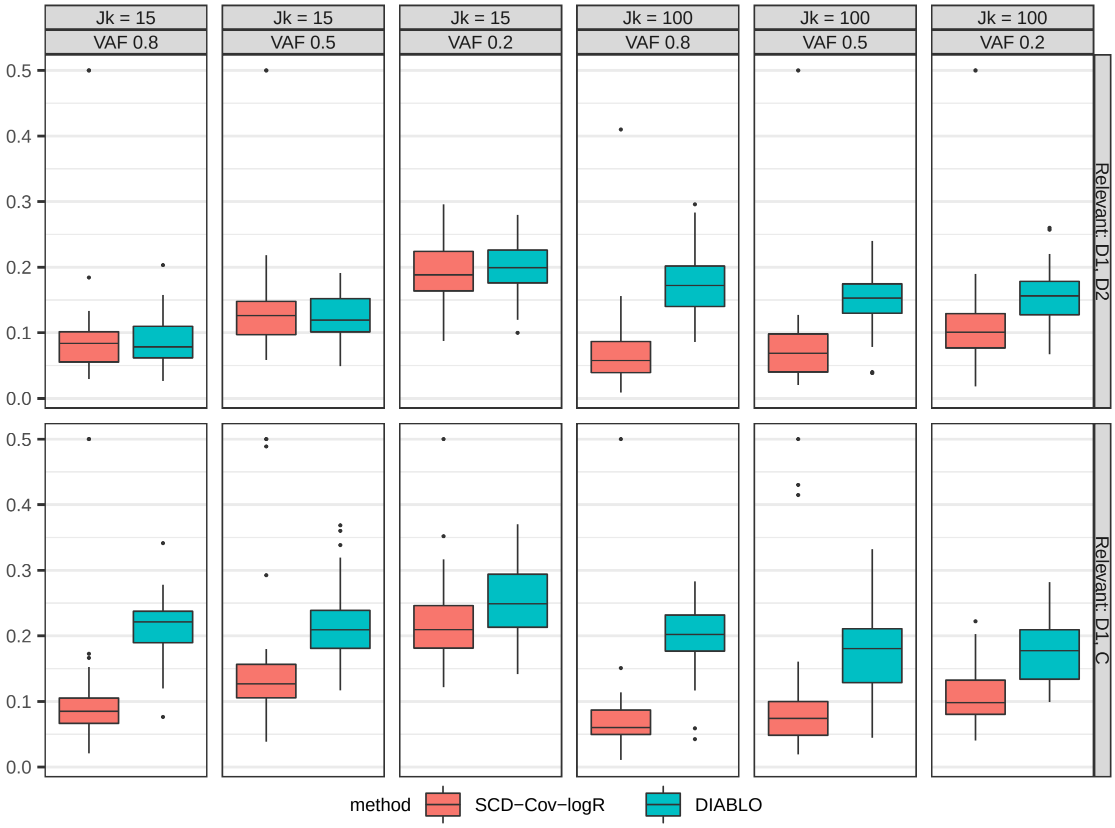


Figure 3.1. Box plots of the out-of-sample BER; each panel corresponds to one of the 12 conditions. The column panels indicate the number of predictors in each data block and the proportion of variance accounted for by the underlying processes. The row panels indicate the two covariates relevant for the outcome variable; "D1", "D2" and "C" refer to the covariate distinctive to the first block, the covariate distinctive to the second block and the common covariate, respectively.

3.3.4.2 Correct weights classification rate

Figure 3.2 presents the outcome of the correct weights classification rate. Across all of the conditions, SCD-Cov-logR resulted in the higher of correct classification. It is also noteworthy that the classification rate for the method is mostly above 0.95. The figure shows the influence of the relevance of the underlying covariates and its interaction with the other manipulated data circumstances. When the two distinctive covariates were relevant, regardless of the dimensionality, SCD-Cov-logR resulted in a much higher classification rate than DIABLO. On the other hand, when the covariate distinctive to the second data block was defined irrelevant, DIABLO's performance was closer to SCD-Cov-logR's in the conditions with more variance of the predictors explained and with 15 predictor variables per block. In conclusion, SCD-Cov-logR is better than DIABLO at correctly retrieving the the underlying population weights.

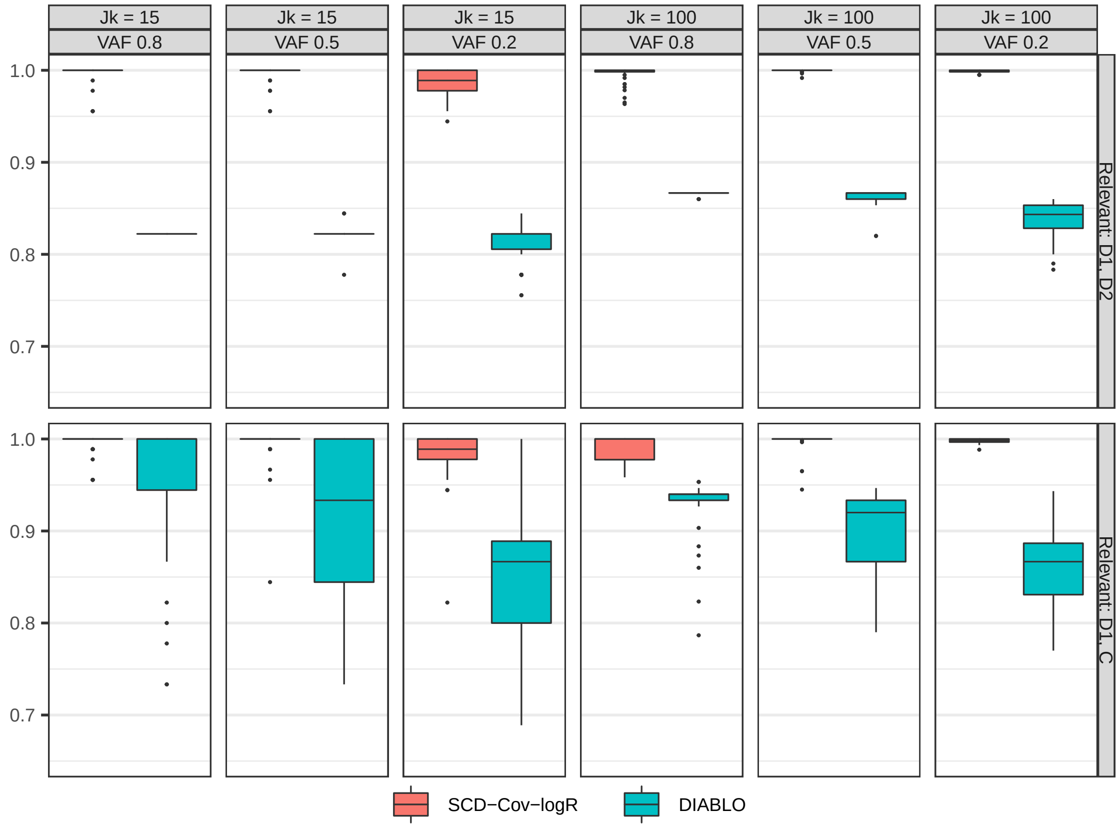


Figure 3.2. Box plots of the correct weight classification rate; each panel corresponds to one of the 12 conditions. The column panels indicate the number of predictors in each data block and the proportion of variance accounted for by the underlying processes. The row columns refer to the two covariates relevant for the outcome variable; "D1", "D2" and "C" refer to the covariate distinctive to the first block, the covariate distinctive to the second block and the common covariate, respectively.

3.4 Illustration: 500 Family Data

3.4.1 Dataset and pre-processing

We demonstrate an example use of SCD-Cov-logR by administering the method on an empirical dataset. We adopted the dataset from the 500 Family Study (Schneider & Waite, 2008) which investigated into how work impacts the well-being of parents and children in American middle-class families. Questionnaire data from different members of the same family were collected. We computed sum scores from questionnaire items that refer to the same construct. These scores concern the feelings of the family members, their recent mutual activities and how they perceive thier relationship. 24 sum score variables were computed and are used as predictors in constructing the SCD-Cov-logR model. They can be found in Table 3.5. Eight of the predictors pertain to responses from the mother, another eight to responses from the father and lastly six predictors are based on the responses of the child. The dataset therefore is comprised of three blocks according to the member of the family, and each observation unit refers to a family. All of the predictors were centered and standardized. Since the blocks have different sizes, they were weighted such that the sum of squares are equal across blocks.

The families are categorized into two groups according to the child's most recent grade at school. The family with the child with a grade B or higher is classified as having academic overachievement (coded as 1), while grade C or lower is classified as underachievement (coded as 0). We excluded the families with missing values on any of the predictor variables, and made a random subset selection of 58 families in order to obtain a balance between the size of two categories. We conducted SCD-Cov-logR to target this classification problem of academic underachievement while simultaneously constructing a model that describes the underlying common and distinctive processes of the three predictor blocks.

3.4.2 Model selection

We employed the sequential cross-validation model selection strategy discussed in section 3.2.4 applied to the toy example dataset. Moreover, 50 sets of random starting values were employed alongside the rational starting values in conducting the model selection and final model fitting.

First, the number of covariates was found by administering PCA on the predictor matrix. By using the acceleration factor technique, we found that when going from 1 to 2 principal components, the amount of variance explained by the principal components changes the most drastically (Figure in the Appendix

3.J). With the number of covariates determined at two, we carry out the cross-validation to select the other tuning parameters. The different sets of starting values were introduced at this stage. The complete process of model selection and model fitting was conducted for each set of starting values. The resulting solutions from 50 random starting values and 1 rational starting value were compared in terms of the value of the loss function: The solution with the smallest loss was retained as the final solution.

The cross-validation procedures administered for each of the starting values were as the following: first, 20-fold cross-validation was conducted with varying values of the rescaled weighting parameter β and λ_R. At this stage, the tuning parameters λ_{Lr} and λ_{Gr} were fixed at zero for the cross-validation. We considered the values of [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] for β and [0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10, 15, 20] for λ_R. Using the one standard error rule, values of β and λ_R are selected. Given these selected values, the second sequence of 20-fold cross-validation for λ_{Lr} and λ_{Gr} was conducted. With the ranges of [0, 0.05, 0.1, 0.3, 0.5, 1, 3, 5, 7, 10, 15, 20, 30, 50] adopted for both parameters, the same parameter value was used concerning the two covariates. We used the one standard error rule again to choose the values of λ_{Lr} and λ_{Gr}, completing the model selection procedure.

Similar to the toy example dataset, a smaller minimum was achieved by the set of rational starting values. The final values for the tuning parameters selected through the sequential procedure were: $\beta = 0.1$, $\lambda_R = 2, \lambda_{Lr} = 10, \lambda_{Gr} = 10$. The final model estimates obtained are presented in Table 3.5.

3.4.3 Results

Table 3.5. Weights and logistic regression coefficients derived by SCD-Cov-logR from the 500 family dataset. The covariate labels heading the columns of the table with weights and the rows of the table with logistic regression coefficients indicate which data blocks the corresponding covariate is associated with.

$\hat{\mathbf{W}}_C$

	Child	Parents
Mother		
Relationship with partners	0	0.276
Argue with partners	0	0.269
Childs bright future	0	0
Activities with children	0	0
Feeling about parenting	0	0.188
Communation with children	0	0.357
Argue with children	0	0.171
Confidence about oneself	0	0.406
Father		
Relationship with partners	0	0.091
Argue with partners	0	0.183
Childs bright future	0	0
Activities with children	0	0
Feeling about parenting	0	0
Communation with children	0	0
Argue with children	0	0.210
Confidence about oneself	0	0.050
Child		
Self confidence/esteem	0.285	0
Social life and extracurricular	0.336	0
Importance of friendship	0.459	0
Self Image	0.381	0
Happiness	0.374	0
Confidence about the future	0.281	0

Logistic regression coefficients

Estimated	
Child	0.288
Parents	0.034
Intercept	-0.007

The estimated weights matrix from Table 3.5 show that there are two predictive processes for the child's academic achievement. The first component is distinctive to the child block and is associated with all of the variables from the data block. It appears that all of the variables in the child block have an impact in the the academic achievement. On the other hand, the second component is locally common, involving several variables from the mother and the father blocks but not from the child block. Observing the weights from the second covariate, it can be seen that parents' high confidence in the child's future and the amount of activities they partake with the child are not important in predicting the child's academic achievement. Also, according to this model, the father's positive feeling about parenting and his level of communication do not exert strong influence in the child's academic achievement. Moreover, the logistic regression coefficients suggest that the Child covariate is much more relevant in predicting child's academic achievement group. It appears that the attitudes that the children themselves have are the most important in leading to academic overachievement.

The covariate scores of the 58 families can be seen in Figure 3.3 which presents a fair separation of the two categories of the families. With the observations separated along the X-axis, It can be seen that the Child covariate plays a more important role in separating the two groups. This is in line with the small magnitude of the coefficient corresponding to the Parents covariate. Out of the 58 families, the final model classifies 43 families correctly. In order to also examine the classification performance of the model on out-of-sample data, we performed a leave-one-out cross-validation which resulted in 40 families being correctly classified. Together, this implies that the model showed about 70% of classification accuracy for both in-sample and out-of-sample observations.

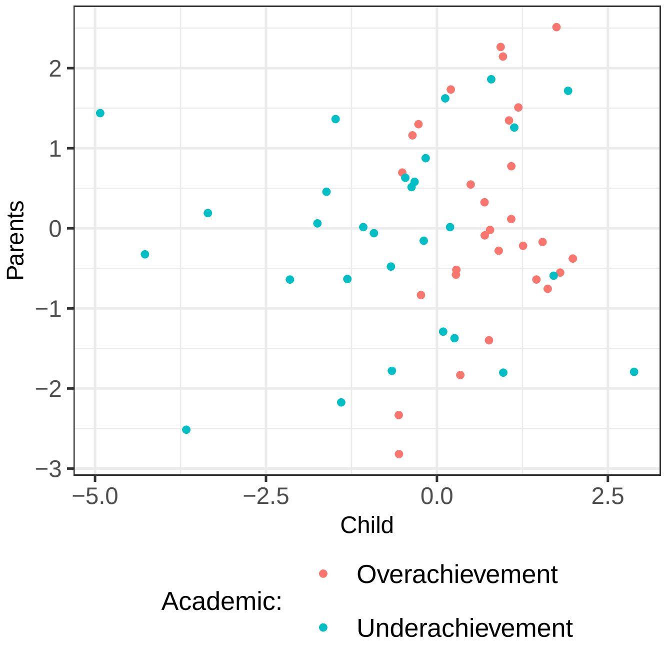


Figure 3.3. Scatterplot of the two covariates found by SCD-Cov-logR. The colours represent the academic achievement of the child.

To obtain more comparative insight about the quality of the method under this empirical dataset, we administered the related methods discussed in the methods section; regularized logistic regression, PCR (SCaDS-logR) and DIABLO. The

PCA step for the PCR was conducted with SCaDS to tackle the multiblock nature of the data, as demonstrated with the toy example dataset in section 3.2.5. The number of components for SCaDS was set at two, so that the model is comparable to the SCD-Cov-logR model constructed with two covariates. The lasso and group lasso parameters governing the sparseness of SCaDS weights were selected with 20-fold cross-validation with the one standard error rule. Similarly, a 2-component model was estimated with DIABLO. The number of non-zero weights to be estimated per component was tuned via 20-fold cross-validation. Lastly, the lasso parameter for regularized logistic regression was also chosen with 20-fold cross-validation. Table 3.6 provides the number of correctly classified in-sample observations from each of the methods. As done for SCD-Cov-logR, leave-one-out cross-validation was conducted to gauge the out-of-sample classification quality. These results are also provided in the table. It can be seen that the four methods led to very comparable performances with respect to prediction. The estimates derived by the methods are provided in Appendix (3.K) and they can be inspected to understand the constructed models. It was found that only SCaDS-logR identified predictive processes concerted by several predictors, akin to the covariates of SCD-Cov-logR. Both regularized logistic regression and DIABLO found a very sparse model with only two non-zero coefficients.

Table 3.6. Number of correctly classified observations (out of the total 58) provided by SCD-Cov-logR, PCR, DIABLO and regularized logistic regression. The out-of-sample classification is computed via leave-out-out cross-validation.

	SCD-Cov-logR	SCaDS-logR	DIABLO	LogR
In-Sample	43	43	44	43
Out-of-Sample (leave-one-out CV)	40	41	38	40

In conclusion, our proposed method is capable in meeting its goals when applied to an empirical dataset. The method identifies common and distinctive covariates and weights that are interpretable. At the same time, the method is able to correctly classify both the samples used for fitting the model and new samples.

3.5 Discussion

A multitude of goals are of interest when building a classification model from a multiblock dataset. The common and distinctive predictor processes need to be identified in an interpretable manner while classifying the outcome variable. We have proposed the method of SCD-Cov-logR that fulfills these goals in a simultaneous manner. We have evaluated the method comparatively against DIABLO; a multiblock variant of PLS. It was found that the proposed method outperforms DIABLO in the objectives that the methods attain: quality of classification and retrieval of weights that are used to understand the underlying processes. Moreover, while DIABLO requires prior information for identifying the common and distinctive processes, our proposed method is able to explore these structures without explicit specification.

In particular, SCD-Cov-logR was found to be considerably better than DIABLO in accurately retrieving the weights matrix. This finding is in line with existing literature that compares between the methodologies of PLS and PCovR. Methods based on PLS tend to place heavier focus on prediction of the outcome variables, as opposed to exploring the structure of the underlying predictor processes. In contrast, the weighting paramter α in the PCovR methods helps to attain a good balance between emphasizing the predictor or the outcome variables. In the current paper, all of the results were based on the rescaled parameter β tuned via cross-validation. This suggests that the parameter can be used effectively in a purely data-driven approach.

SCD-Cov-logR also has weaknesses. Model selection is an inherent challenge since the method requires many parameters to be tuned to meet its multiple research aims. There are in total 5 parameters to be selected and they all play an important role in shaping the retrieved model. Adopting the solution recommended by Vervloet et al. (2016), the current paper suggested a sequential model selection approach where sets of tuning parameters are chosen through cross-validation with the other parameters fixed. Models obtained by this approach led to good results in both simulation experiments and empirical study. We have not visited the model selection problem of our method in great detail as the main purpose of this paper lies within the proposal and illustration of the novel method.

Another remark about the model selection procedure is the optimality criterion used for cross-validation. Throughout the paper, we adopted the sum of squared cross-validation errors concerning the binary outcome variable. This implies that the model selection procedure is conducted only considering the out-of-sample prediction quality. Since our method is not only used for classification of the outcome but also exploring the predictor processes, the optimality criterion for cross-validation can be changed to also include the errors pertaining to the predictor variables. This choice is in the same spirit of the weighting parameter α; if the user is interested more in the exploration of the predictor processes, it may be a viable option to look into such an optimality crterion different from what is used in this paper.

In our illustration of the toy data example and the simulation study, DIA-

BLO was fitted in a peculiar manner to allow for derivation of the distinctive and common covariates. However, in practice, there may be other ways of specifying the method. For example, a supermatrix of concatenated blocks can be provided as the only input dataset and a single DIABLO model can be constructed on it [3]. We have explored into such a specification, and found that it results in consistent underperformance compared to SCD-Cov-logR with respect to prediction and retrieval of population parameters. It also has a tendency to only find common covariates.

Finally, the method and the current paper suggest several future directions of research. It would be a natural extension to broaden the method to encompass generalized linear models. This would allow modelling of outcome variables in diverse nature such as count data. Furthermore, such an extension would allow other related research questions to be addressed. For example, within the high-dimensional multiblock setting, it would be interesting to examine the impact of using a generalized linear model framework to model the categorical outcome, as opposed to the discriminant analysis approach adopted for DIABLO where the categorical outcome variable is simply changed into a dummy matrix and a linear regression model is fit. Although Lê Cao et al. (2008) compared the two approaches and reported that they show comparable performance in practice, the comparison has not been conducted in the multiblock data setting. Our proposed method SCD-Cov-logR can also be easily adapted into the linear regression approach using a dummy outcome matrix, if it is found to be useful in certain data circumstances.

[3] This would then be a single model of sPLS-DA (sparse partial least squarse discriminant analysis).

Appendix

3.A SPCovR, SCaDS and SCD-CovR

3.A.1 SPCovR

For easier interpretation of the principal covariates and consistency of estimates in the high dimensional settings, regularization penalties have been imposed on the weights from Equation (3.5) to lead to sparse PCovR (SPCovR; Van Deun et al., 2018). The method finds sparse weights by minimizing the following objective function:

$$\begin{aligned} L(\mathbf{W}_k, \mathbf{P}_k^{(X)}, \mathbf{p}^{(y)}) = \alpha \frac{\left\|\mathbf{y} - \mathbf{X}_k \mathbf{W}_k \mathbf{p}^{(y)}\right\|_2^2}{\|\mathbf{y}\|_2^2} + (1-\alpha) \frac{\left\|\mathbf{X}_k - \mathbf{X}_k \mathbf{W}_k (\mathbf{P}_k^{(X)})^\top\right\|_2^2}{\|\mathbf{X}_k\|_2^2} \\ + \lambda_L \left|\mathbf{W}_k\right|_1 + \lambda_R \left\|\mathbf{W}_k\right\|_2^2 \end{aligned} \tag{3.12}$$

such that $(\mathbf{P}_k^{(X)})^\top \mathbf{P}_k^{(X)} = \mathbf{I}_R$ and with $\lambda_L \geq 0$, $\lambda_R \geq 0$ and $\alpha \geq 0$. The regularization parameters are the lasso, with $|\mathbf{W}_k|_1 = \sum_{j_k,r} |w_{j_k r}|$, and the ridge $\|\mathbf{W}_k\|_2^2 = \sum_{j_k,r} w_{j_k r}^2$, together forming the elastic net penalty (Zou & Hastie, 2005). The ridge penalty shrinks the magnitude of the estimates and encourages stable estimation for high-dimensional data, while the lasso penalty is involved in variable selection by shrinking and forcing the estimates to exactly zero. When both penalties are defined at 0, it can be seen that the PCovR formulation (3.5) is retrieved.

3.A.1.1 SCA and SCD-CovR

SPCovR only targets data with a single predictor block and hence do not address the questions associated with multiple predictor blocks. A joint analysis of the K predictor blocks can be conducted by imposing a multiblock PCovR model, based on the SCA model (Kiers & ten Berge, 1989):

$$\begin{aligned} \mathbf{X}_C =& \mathbf{X}_C \mathbf{W}_C (\mathbf{P}_C^{(X)})^\top + \mathbf{E}^{(X)} \\ \mathbf{y} =& \mathbf{X}_C \mathbf{W}_C \mathbf{p}^{(y)} + \mathbf{e}^{(y)} \end{aligned} \tag{3.13}$$

where $\mathbf{X}_C = [\mathbf{X}_1, \ldots, \mathbf{X}_K]$ (of size $I \times \sum_{k=1}^{K} J_k$) denotes the supermatrix that concatenates the predictor blocks. Consequently, $\mathbf{W}_C$ and $\mathbf{P}_C^{(X)}$ are weight and loading matrices of size $\sum_{k=1}^{K} J_k \times R$. $\mathbf{p}^{(y)}$ indicates a vector of R regression coefficients.

When SCA is administered to study the processes underlying the variables without considering the regression problem, the concatenated weights matrix $\mathbf{W}_C$ is examined to understand the nature of the components. In order to allow SCA to explicitly distinguish common and distinctive processes and provide a sparse and interpretable solution from high dimensional multiblock datasets, de Schipper and Van Deun (2018) proposed SCaDS. Regularization penalties are imposed upon the weights to force certain elements to zero for handier interpretation, while the $\mathbf{W}_C$ matrix is further constrained such that certain components are a priori fixed as being common or distinctive.

Making use of the multiblock PCovR model (3.13) and also combining with SCaDS, SCD-CovR extends SPCovR to allow multiblock analysis. It predicts the outcome, while providing sparse weights that capture the common and distinctive processes in the predictor blocks. SCD-CovR implies minimizing the following objective function:

$$\begin{aligned} L(\mathbf{W}_C, \mathbf{P}_C^{(X)}, \mathbf{p}^{(y)}) = \alpha \frac{\left\|\mathbf{y} - \mathbf{X}_C \mathbf{W}_C \mathbf{p}^{(y)}\right\|_2^2}{\|\mathbf{y}\|_2^2} + (1-\alpha) \frac{\left\|\mathbf{X}_C - \mathbf{X}_C \mathbf{W}_C (\mathbf{P}_C^{(X)})^\top\right\|_2^2}{\|\mathbf{X}_C\|_2^2} \\ + \lambda_L \left|\mathbf{W}_C\right|_1 + \lambda_R \left\|\mathbf{W}_C\right\|_2^2 \end{aligned} \tag{3.14}$$

such that $(\mathbf{P}_C^{(X)})^\top \mathbf{P}_C^{(X)} = \mathbf{I}_R$, and subject to zero block constraints on $\mathbf{W}_C$ that fix weights that correspond to one or several predictor blocks to zero. This implies that the component is determined only by predictors of those blocks for which the weights have not been fixed to zero. Common components are obtained by not placing such zero block constraints on the component. The elastic net penalty and the constraints concerning the weights are the same as imposed in SCaDS. Also, as in SPCovR, the lasso penalty achieves sparseness within the common and distinctive covariates.

3.B SCD-Cov-logR algorithm

The minimizing solution of (3.7) can be found by iteratively reweighted least squares which involves formulating the quadratic approximation of the negative log likelihood given the current estimates of the parameters (Friedman et al., 2010b). The negative log likelihood part of the objective function is as the following:

$$L_{logr}(\mathbf{W}_C, \mathbf{p}^{(g)}, p_0^{(g)}) = -\sum_{i}^{I} g_i(p_0^{(g)} + \mathbf{x}_{C_i}^{\top} \mathbf{W}_C \mathbf{p}^{(g)}) - \log(1 + e^{(p_0^{(g)} + \mathbf{x}_{C_i}^{\top} \mathbf{W}_C \mathbf{p}^{(g)})}) \tag{3.15}$$

Quadratic approximation of (3.15) given the current estimates of the parameters is as the following.

$$LQ_{logr}(\mathbf{W}_C, \mathbf{p}^{(g)}, p_0^{(g)}) = \frac{1}{2} \sum_{i}^{I} q_i (z_i - p_0^{(g)} - \mathbf{x}_{C_i}^{\top} \mathbf{W}_C \mathbf{p}^{(g)})^2 \tag{3.16}$$

where

$$\begin{aligned} q_i &= \tilde{p}_i(1 - \tilde{p}_i) \\ z_i &= \tilde{p}_0^{(g)} + \mathbf{x}_{C_i}^{\top} \tilde{\mathbf{W}}_C \tilde{\mathbf{p}}^{(g)} + \frac{g_i - \tilde{p}_i}{\tilde{p}_i(1 + \tilde{p}_i)} \\ \tilde{p}_i &= e^{(\tilde{p_0^{(g)}} + \mathbf{x}_{C_i}^{\top} \tilde{\mathbf{W}}_C \tilde{\mathbf{p}^{(g)}})} / (1 + e^{(\tilde{p_0^{(g)}} + \mathbf{x}_{C_i}^{\top} \tilde{\mathbf{W}}_C \tilde{\mathbf{p}^{(g)}})}) \end{aligned} \tag{3.17}$$

The parameters denoted with the ˜ symbol are the current parameters. With the quadratic approximation now replacing the negative log-likelihood in (3.7) and the rescaled weighting parameter β used instead of α (see Equation 3.8), the objective function becomes:

$$\begin{aligned} L(\mathbf{W}_C, \mathbf{P}_C^{(X)}, \mathbf{p}^{(g)}, p_0^{(g)}) = {} & \frac{\beta}{2} \sum_{i}^{I} q_i (z_i - p_0^{(g)} - \mathbf{x}_{C_i}^{\top} \mathbf{W}_C \mathbf{p}^{(g)})^2 \\ & + (1 - \beta) \sum_{i}^{I} \left\| \mathbf{x}_{C_i} - \mathbf{x}_{C_i}^{\top} \mathbf{W}_C (\mathbf{P}_C^{(X)})^{\top} \right\|_2^2 \\ & + \sum_{r}^{R} \lambda_{Lr} \left| \mathbf{w}_{Cr} \right|_1 + \sum_{r}^{R} \sum_{k}^{K} \lambda_{Gr} \sqrt{J_k} \left\| \mathbf{w}_r^{(k)} \right\|_2 + \lambda_R \left\| \mathbf{p}^{(g)} \right\|_2^2 \end{aligned} \tag{3.18}$$

where q_i and z_i are defined as in (3.17). The optimization problem in (3.18) can be solved with an alternating procedure where the loadings $\mathbf{P}_C^{(X)}$ and the regression coefficients $\mathbf{p}^{(g)}, p_0^{(g)}$ are solved for conditional upon fixed values for the weights $\mathbf{W}_C$ and vice versa. The sparse group lasso problem for $\mathbf{W}_C$ is treated via coordinate descent (Friedman et al., 2010a), while closed-form solutions exist for the conditional updates of $\mathbf{p}^{(g)}, p_0^{(g)}$ and $\mathbf{P}_C^{(X)}$. The derivation of these updating rules can be found in Appendix (3.C) and (3.D). After each run of conditional estimation of the parameters, the quadratic approximation in (3.18) is updated with new values of q_i and z_i calculated with the current parameters. To prevent the divergence of the coefficients, when the absolute difference between the current probability $\tilde{p}_i$ and 1 is less or equal to 10^{-5}, $\tilde{p}_i$ is fixed at 1. This follows the recommendation of Friedman et al. (2010b) which proposed a framework of combining regularization with GLM.

A schematic outline of the algorithm is provided in what follows. The optimization procedure that we propose here closely follows those proposed for SCaDS and SPCovR (de Schipper & Van Deun, 2018; Van Deun et al., 2018). This procedure boils down to solving for all components together (unlike deflation methods that solve for each component in turn). The alternating routine continues until the algorithm converges to a stationary point, usually a local minimum. To avoid local minima problems, we recommend to use multiple random and a rational starting value based on PCovR.

Algorithm 3.1 SCD-Cov-logR

```
Inputs:
    $\mathbf{X}_C$ and $\mathbf{g}$, number of components $R$, rescaled weighting parameter $\beta$,
    regularization parameters $\lambda_{Lr}$, $\lambda_{Gr}$ and $\lambda_R$, maximum number of iterations $T$,
    convergence threshold $\epsilon \geq 0$

Initialize:
    $\mathbf{W}_C \leftarrow {\mathbf{W}_C}^{(0)}$, $\mathbf{P}_C^{(X)} \leftarrow {\mathbf{P}_C^{(X)}}^{(0)}$, $\mathbf{p}^{(g)} \leftarrow {\mathbf{p}^{(g)}}^{(0)}$, ${p_0}^{(g)} \leftarrow {{p_0}^{(g)}}^{(0)}$, $L_0 \leftarrow$ Initial loss,
    Loss difference $d \leftarrow 1$, Iteration counter $t \leftarrow 1$

while $t < T$ or $\epsilon < d$ do
    Update of $q_i, z_i$ given ${\mathbf{W}_C}^{(t-1)}$, ${\mathbf{P}_C^{(X)}}^{(t-1)}$, ${\mathbf{p}^{(g)}}^{(t-1)}$, and ${p_0^{(g)}}^{(t-1)}$
    Conditional estimation of ${\mathbf{W}_C}^{(t)}$ given ${\mathbf{P}_C^{(X)}}^{(t-1)}$, ${\mathbf{p}^{(g)}}^{(t-1)}$ and ${p_0^{(g)}}^{(t-1)}$
    Update of $q_i, z_i$ given ${\mathbf{W}_C}^{(t)}$, ${\mathbf{P}_C^{(X)}}^{(t-1)}$, ${\mathbf{p}^{(g)}}^{(t-1)}$, and ${p_0^{(g)}}^{(t-1)}$
    Conditional estimation of ${\mathbf{P}_C^{(X)}}^{(t)}$, ${\mathbf{p}^{(g)}}^{(t)}$ and ${p_0^{(g)}}^{(t)}$ given ${\mathbf{W}_C}^{(t)}$
    $L_u \leftarrow$ updated loss given ${\mathbf{W}_C}^{(t)}$, ${\mathbf{P}_C^{(X)}}^{(t)}$, ${\mathbf{p}^{(g)}}^{(t)}$ and ${p_0^{(g)}}^{(t)}$
    $d \leftarrow L_0 - L_u$
    $t \leftarrow t + 1$
    $L_0 \leftarrow L_u$
end while
```

3.C Estimation of $\mathbf{W}_C$

Conditional estimation of $\mathbf{W}_C$ given the other parameters $\mathbf{P}^{(X)}, \mathbf{p}^{(g)}$ and $p_0^{(g)}$ pertains to a sparse group lasso problem. The SCD-Cov-logR objective function with the quadratic approximation of the negatlive log-likelihood (3.18) is first arranged with respect to the weights corresponding to predictor block k and component r^*:

$$\begin{aligned}
&L(\mathbf{w}_{r^*}^{(k)}, \mathbf{P}_C^{(X)}, \mathbf{p}^{(g)}, p_0^{(g)}) = \\
&\frac{\beta}{2}\sum_i^I q_i(z_i - p_0^{(g)} - \sum_r^R \sum_{l\neq k}^K p_r^{(g)} \mathbf{x}_i^{(l)\top} \mathbf{w}_r^{(l)} - \sum_{r\neq r^*}^R p_r^{(g)} \mathbf{x}_i^{(k)\top} \mathbf{w}_r^{(k)} - p_{r^*}^{(g)} \mathbf{x}_i^{(k)\top} \mathbf{w}_{r^*}^{(k)})^2 \\
&+ (1-\beta)\sum_i^I \left\| \mathbf{x}_{Ci} - \sum_r^R \sum_{l\neq k}^K \mathbf{w}_r^{(l)\top} \mathbf{x}_i^{(l)} \mathbf{p}_{Cr}^{(X)} - \sum_{r\neq r^*} \mathbf{w}_r^{(k)\top} \mathbf{x}_i^{(k)} \mathbf{p}_{Cr}^{(X)} - \mathbf{w}_{r^*}^{(k)\top} \mathbf{x}_i^{(k)} \mathbf{p}_{Cr^*}^{(X)} \right\|_2^2 \\
&+ \lambda_L \left| \mathbf{w}_{r^*}^{(k)} \right|_1 + \lambda_G \sqrt{J_k} \left\| \mathbf{w}_{r^*}^{(k)} \right\|_2
\end{aligned} \tag{3.19}$$

Taking the derivative with respect to $\mathbf{w}_{r^*}^{(k)}$ we get:

$$\begin{aligned}
&-\beta \sum_i^I q_i p_{r^*}^{(g)} (Z_i^{(k)} - p_{r^*}^{(g)} \mathbf{x}_i^{(k)\top} \mathbf{w}_{r^*}^{(k)}) \mathbf{x}_i^{(k)} - 2(1-\beta) \sum_i^I (Y_i^{(k)} - \mathbf{w}_{r^*}^{(k)\top} \mathbf{x}_i^{(k)}) \mathbf{x}_i^{(k)} \\
&+ \lambda_L \partial \left| \mathbf{w}_{r^*}^{(k)} \right|_1 + \lambda_G \sqrt{J_k} \partial \left\| \mathbf{w}_{r^*}^{(k)} \right\|_2
\end{aligned} \tag{3.20}$$

where

$$\begin{aligned}
Z_i^{(k)} &= z_i - p_0^{(g)} - \sum_r^R \sum_{l\neq k}^K p_r^{(g)} \mathbf{x}_i^{(l)\top} \mathbf{w}_r^{(l)} - \sum_{r\neq r^*}^R p_r^{(g)} \mathbf{x}_i^{(k)\top} \mathbf{w}_r^{(k)} \\
Y_i^{(k)} &= \mathbf{x}_{Ci}^{\top} \mathbf{p}_{Cr^*}^{(X)} - \sum_{l\neq k}^K \mathbf{w}_{r^*}^{(l)\top} \mathbf{x}_i^{(l)}
\end{aligned} \tag{3.21}$$

The subdifferential of $\left\| \mathbf{w}_{r^*}^{(k)} \right\|_2$ is defined as the following:

$$\partial \left\| \mathbf{w}_{r^*}^{(k)} \right\|_2 = \begin{cases} \frac{\hat{\mathbf{w}}_{r^*}^{(k)}}{\left\| \hat{\mathbf{w}}_{r^*}^{(k)} \right\|_2}, & \text{if } \hat{\mathbf{w}}_{r^*}^{(k)} \neq \mathbf{0} \\ \in \{\mathbf{u} : \|\mathbf{u}\|_2 \leq 1\}, & \text{if } \hat{\mathbf{w}}_{r^*}^{(k)} = \mathbf{0} \end{cases} \tag{3.22}$$

where $\mathbf{u}$ is a vector of equal length as $\mathbf{w}_{r^*}^{(k)}$.
The jth element of the subdifferential of $\partial \left| \mathbf{w}_{r^*}^{(k)} \right|_1$ is defined as the following:

$$\partial\left(\left|\mathbf{w}_{r^*}^{(k)}\right|_1\right)_j = \begin{cases} \text{sign}\left(\hat{w}_{jr^*}^{(k)}\right), & \text{if } \hat{w}_{jr^*}^{(k)} \neq 0 \\ \in \{v : |v| \leq 1\}, & \text{if } \hat{w}_{jr^*}^{(k)} = 0 \end{cases} \tag{3.23}$$

where v is a scalar.

By equating Equation (3.20) to zero and rearranging, the condition that an optimal solution satisfies with $\hat{\mathbf{w}}_{r^*}^{(k)} = \mathbf{0}$ is the following:

$$\left\| S(\sum_i^I(\beta\; q_i\; p_{r^*}^{(g)}\; Z_i^{(k)} + 2(1-\beta)\; Y_i^{(k)})\; \mathbf{x}_i^{(k)}, \lambda_L) \right\|_2 \leq \lambda_G \sqrt{J_k} \tag{3.24}$$

where S(.) is a element-wise soft-thresholding operator.

In the case that Equation (3.24) is not satisfied and thus $\hat{\mathbf{w}}_{r^*}^{(k)} \neq \mathbf{0}$, we find the conditions for an optimal solution for the hth element of the weights concerning predictor block k and component r^*; $w_{hr^*}^{(k)}$. We first write the objective function with respect to $w_{hr^*}^{(k)}$.

$$\begin{aligned} & L(w_{hr^*}^{(k)}, \mathbf{P}_C^{(X)}, \mathbf{p}^{(g)}, p_0^{(g)}) = \\ & \frac{\beta}{2} \sum_i^I q_i (z_i - p_0^{(g)} - \sum_r^R \sum_l^K \sum_{j \neq h}^{J_k} p_r^{(g)} x_{ij}^{(l)} w_{jr}^{(l)} - \sum_{r \neq r^*}^R \sum_{l \neq k}^K p_r^{(g)} x_{ih}^{(l)} w_{hr}^{(l)} - p_{r^*}^{(g)} x_{ih}^{(k)} w_{hr^*}^{(k)})^2 \\ & + (1-\beta) \sum_i^I \left\| \mathbf{x}_{Ci} - \sum_r^R \sum_l^K \sum_{j \neq h}^{J_k} \mathbf{p}_r^{(X)} x_{ij}^{(l)} w_{jr}^{(l)} - \sum_{r \neq r^*}^R \sum_{l \neq k}^K \mathbf{p}_r^{(X)} x_{ih}^{(l)} w_{hr}^{(l)} - \mathbf{p}_{r^*}^{(X)} x_{ih}^{(k)} w_{hr^*}^{(k)} \right\|_2^2 \\ & + \lambda_L \left| w_{hr^*}^{(k)} \right| + \lambda_G \sqrt{J_k} \left\| \mathbf{w}_{r^*}^{(k)} \right\|_2 \end{aligned} \tag{3.25}$$

Taking the derivative with respect to $w_{hr^*}^{(k)}$:

$$\begin{aligned} & -\beta \sum_i^I q_i\, p_{r^*}^{(g)} x_{ih}^{(k)} (Z_i - p_{r^*}^{(g)} x_{ih}^{(k)} w_{hr^*}^{(k)}) - 2(1-\beta) \sum_i^I x_{ih}^{(k)} (Y_i - x_{ih}^{(k)} w_{hr^*}^{(k)}) \\ & + \lambda_L \partial \left| w_{hr^*}^{(k)} \right| + \lambda_G \sqrt{J_k} \partial \left\| \mathbf{w}_{r^*}^{(k)} \right\|_2 \end{aligned} \tag{3.26}$$

where

$$\begin{aligned} Z_i &= z_i - p_0^{(g)} - \sum_r^R \sum_l^K \sum_{j \neq h}^{J_k} p_r^{(g)} x_{ij}^{(l)} w_{jr}^{(l)} - \sum_{r \neq r^*}^R \sum_{l \neq k}^K p_r^{(g)} x_{ih}^{(l)} w_{hr}^{(l)} \\ Y_i &= \mathbf{x}_{C_i}^\top \mathbf{p}_{Cr^*}^{(X)} - \sum_l^K \sum_{j \neq h}^{J_k} x_{ij}^{(l)} w_{jr^*}^{(l)} - \sum_{l \neq k}^K x_{ih}^{(l)} w_{hr^*}^{(l)} \end{aligned} \tag{3.27}$$

The subdifferential of $\left\|\mathbf{w}_{r^*}^{(k)}\right\|_2$ with respect to $w_{hr^*}^{(k)}$ is provided in Equation (3.22); it is the hth element of $\frac{\hat{\mathbf{w}}_{r^*}^{(k)}}{\left\|\hat{\mathbf{w}}_{r^*}^{(k)}\right\|_2}$. The subdifferential of $\partial\left|w_{hr^*}^{(k)}\right|$ is defined as the following:

$$\partial\left|w_{hr^*}^{(k)}\right| = \begin{cases} \text{sign}\left(\hat{w}_{hr^*}^{(k)}\right), & \text{if } \hat{w}_{hr^*}^{(k)} \neq 0 \\ \in \{v : |v| \leq 1\}, & \text{if } \hat{w}_{hr^*}^{(k)} = 0 \end{cases} \tag{3.28}$$

where v is a scalar.

We can equate the derivate to zero to find the optimality conditions for $\hat{w}_{hr^*}^{(k)}$, which can be summarized by the following:

$$\hat{w}_{hr^*}^{(k)} = \frac{S(\sum_i^I x_{ih}^{(k)}(\beta\, p_{r^*}^{(g)}\; q_i\; Z_i + 2(1-\beta)\; Y_i), \lambda_L)}{\beta\; p_{r^*}^{(g)^2}\;\; \sum_i^I q_i x_{ih}^{(k)^2} + 2(1-\beta)\;\; \sum_i^I x_{ih}^{(k)^2} + \lambda_G \sqrt{J_k} / \left\|\mathbf{w}_{r^*}^{(k)}\right\|_2} \tag{3.29}$$

With these conditions, we can set up the following coordinate descent algorithm.

Algorithm 3.2 Coordinate descent for sparse group lasso

```
for $r^*$ in $1 : R$ do
    for $k$ in $1 : K$ do
        if $\left\|S(\sum_i^I(\beta\; q_i\; p_{r^*}^{(g)}\; Z_i^{(k)} + 2(1-\beta)\; Y_i^{(k)})\; \mathbf{x}_i^{(k)}, \lambda_L)\right\|_2 \leq \lambda_G \sqrt{J_k}$ then
            $\hat{\mathbf{w}}_{r^*}^{(k)} \leftarrow \mathbf{0}$
        for $h$ in $1 : J_k$ do
            $\hat{w}_{hr^*}^{(k)} \leftarrow \frac{S(\sum_i^I x_{ih}^{(k)}(\beta\; p_{r^*}^{(g)}\; q_i\; Z_i + 2(1-\beta)\; Y_i), \lambda_L)}{\beta\; p_{r^*}^{(g)^2}\; \sum_i^I q_i x_{ih}^{(k)^2} + 2(1-\beta)\; \sum_i^I x_{ih}^{(k)^2} + \lambda_G \sqrt{J_k} / \left\|\mathbf{w}_{r^*}^{(k)}\right\|_2}$
```

3.D Estimation of $\mathbf{p}^{(g)}, p_0^{(g)}$ and $\mathbf{P}_C^{(X)}$

Closed-form solutions exist for the regression coefficients and the intercept.

$$\hat{\mathbf{p}}^{(g)} = [(\mathbf{X}_C\mathbf{W}_C)^\top \mathbf{Q}\,\mathbf{X}_C\mathbf{W}_C + (2/\alpha)\,\lambda_R\mathbf{I}_R]^{-1}[(\mathbf{X}_C\mathbf{W}_C)^\top \mathbf{Q}\,\mathbf{z} - p_0^{(g)}(\mathbf{X}_C\mathbf{W}_C)^\top \mathbf{q}] \tag{3.30}$$

$$\hat{p}_0^{(g)} = (\sum_i^I q_i(z_i - \mathbf{x}_{Ci}^\top \mathbf{W}_C\mathbf{p}^{(g)}))\,/\,(\sum_i^I q_i) \tag{3.31}$$

where $\mathbf{Q}$ is a diagonal matrix with the ith diagonal element being q_i. $\mathbf{q}$ and $\mathbf{z}$ are vectors with the elements being q_i and z_i respectively, which are defined in (3.17).

The loadings $\mathbf{P}_C^{(X)}$ are also obtained via a closed-form solution; $\mathbf{P}_C^{(X)} = \mathbf{U}\mathbf{V}^\top$ where $\mathbf{U}$ and $\mathbf{V}$ are found through singular value decomposition of $\mathbf{X}_C^\top\mathbf{X}_C\mathbf{W}_C = \mathbf{U}\mathbf{D}\mathbf{V}^\top$.

3.E SCD-Cov-logR multiclass algorithm

Like for the binary problem, the solution to (3.10) is found by iteratively reweighted least squares. Partial quadratic approximation can be conducted such that only parameters that concern the mth category can vary at a time. With the quadratic approximation replacing the negative log-likelihood in (3.10) and the rescaled weighting parameter β used instead of α (see Equation 8), the objective function becomes:

$$\begin{aligned} L(\mathbf{W}_C, \mathbf{P}_C^{(X)}, \mathbf{p}_m^{(g)}, p{0}_m^{(g)}) =& \frac{\beta}{2}\sum_i^I q_i(z_i - p{0}_m^{(g)} - \mathbf{x}_{Ci}^\top \mathbf{W}_C\mathbf{p}_m^{(g)})^2 \\ &+ (1-\beta)\sum_i^I \left\|\mathbf{x}_{Ci} - \mathbf{x}_{Ci}^\top \mathbf{W}_C(\mathbf{P}_C^{(X)})^\top\right\|_2^2 \\ &+ \sum_r^R \lambda_{Lr}\left|\mathbf{w}_{Cr}\right|_1 + \sum_r^R\sum_k^K \lambda_{Gr}\sqrt{J_k}\left\|\mathbf{w}_r^{(k)}\right\|_2 + \lambda_R\left\|\mathbf{p}_m^{(g)}\right\|_2^2 \end{aligned} \tag{3.32}$$

where

$$\begin{aligned} q_i &= \tilde{p}_i(1-\tilde{p}_i) \\ z_i &= \tilde{p{0}}_m^{(g)} + \mathbf{x}_{Ci}^\top \tilde{\mathbf{W}}_C\tilde{\mathbf{p}}_m^{(g)} + \frac{g_i - \tilde{p}_i}{\tilde{p}_i(1+\tilde{p}_i)} \\ \tilde{p}_i &= e^{(\tilde{p{0}}_m^{(g)} + \mathbf{x}_{Ci}^\top \tilde{\mathbf{W}}_C\tilde{\mathbf{p}}_m^{(g)})}/(1 + \sum_m^{M-1} e^{(\tilde{p{0}}_m^{(g)} + \mathbf{x}_{Ci}^\top \tilde{\mathbf{W}}_C\tilde{\mathbf{p}}_m^{(g)})}) \end{aligned} \tag{3.33}$$

the parameters denoted with the $\tilde{}$ symbol are the current parameters. The loadings are constrained to be column-orthogonal: $(\mathbf{P}_C^{(X)})^\top \mathbf{P}_C^{(X)} = \mathbf{I}_R$. This optimization problem can be solved with an alternating procedure similar to that of the binary classification. In fact, the conditional estimation of the parameters is done in the same way as for the binary problem (shown in Appendix 3.C, 3.D) with a small tweak on the definition of certain quantities. We can first notice that this objective function with quadratic approximation with respect to category m can be considered as a binary problem between category m and the baseline category M. It can be seen that the only difference between the functions for the multiclass (3.32, 3.33) and the binary (3.17, 3.18) problems is the definition of the current parameter $\tilde{p_i}$. Therefore, from the binary objective function (3.18), computing $\tilde{p_i}$ by following (3.33) and replacing the regression coefficients $\mathbf{p}^{(g)}, p_0^{(g)}$ into $\mathbf{p}_m^{(g)}, p_{0m}^{(g)}$ specific for category m would enable us to rely on the same solutions for the conditional updates of the quantities $\mathbf{W}_C, \mathbf{p}_m^{(g)}, p_{0m}^{(g)}$ and $\mathbf{P}_C^{(X)}$. The algorithm for the multiclass problem however cycles over the $M-1$ categories on top of the conditional updates of the quantities. After each run of conditional estimation of the quantities, the quadratic approximation in (3.32) is updated with new values of q_i and z_i calculated with the current parameters.

A schematic outline of the algorithm is provided below. The alternating routine continues until the algorithm converges to a stationary point, usually a local minimum. To avoid local minima problems, we recommend to use multiple random and a rational starting value based on PCovR.

Algorithm 3.3 SCD-Cov-logR for multiclass classification

```
Inputs:
    $\mathbf{X}_C$ and $\mathbf{G}$, number of components $R$, rescaled weighting parameter $\beta$,
    regularization parameters $\lambda_{Lr}$, $\lambda_{Gr}$ and $\lambda_R$, maximum number of iterations $T$,
    convergence threshold $\epsilon \geq 0$
Initialize:
    $\mathbf{W}_C \leftarrow \mathbf{W}_C{}^{(0)}$, $\mathbf{P}_C^{(X)} \leftarrow \mathbf{P}_C^{(X)}{}^{(0)}$, $\mathbf{p}_m^{(g)} \leftarrow \mathbf{p}_m^{(g)}{}^{(0)}$, $p_{0m}^{(g)} \leftarrow p_{0m}^{(g)}{}^{(0)}$, $L_0 \leftarrow$ Initial loss,
    Loss difference $d \leftarrow 1$, Iteration counter $t \leftarrow 1$
while $t < T$ or $\epsilon < d$ do
    for $m \leftarrow 1$ to $M-1$ do
        Update of $q_i, z_i$ given $\mathbf{W}_C{}^{(t-1)}$, $\mathbf{P}_C^{(X)}{}^{(t-1)}$, $\mathbf{p}_m^{(g)}{}^{(t-1)}$, and $p_{0m}^{(g)}{}^{(t-1)}$
        Conditional estimation of $\mathbf{W}_C{}^{(t)}$ given $\mathbf{P}_C^{(X)}{}^{(t-1)}$, $\mathbf{p}_m^{(g)}{}^{(t-1)}$ and $p_{0m}^{(g)}{}^{(t-1)}$
        Update of $q_i, z_i$ given $\mathbf{W}_C{}^{(t)}$, $\mathbf{P}_C^{(X)}{}^{(t-1)}$, $\mathbf{p}_m^{(g)}{}^{(t-1)}$, and $p_{0m}^{(g)}{}^{(t-1)}$
        Conditional estimation of $\mathbf{P}_C^{(X)}{}^{(t)}$, $\mathbf{p}_m^{(g)}{}^{(t)}$ and $p_{0m}^{(g)}{}^{(t)}$ given $\mathbf{W}_C{}^{(t)}$
        $L_u \leftarrow$ updated loss given $\mathbf{W}_C{}^{(t)}$, $\mathbf{P}_C^{(X)}{}^{(t)}$, $\mathbf{p}_m^{(g)}{}^{(t)}$ and $p_{0m}^{(g)}{}^{(t)}$
    $d \leftarrow L_0 - L_u$
    $t \leftarrow t + 1$
    $L_0 \leftarrow L_u$
end while
```

3.F The scree test with acceleration factor conducted to determine the number of covariates for the toy example dataset

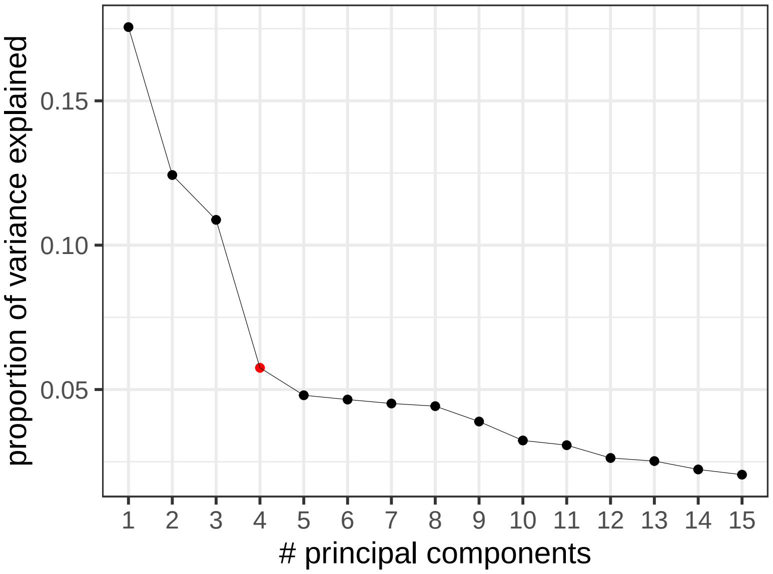


Figure 3.4. It can be seen that the sharpest change of slopes occurs at four principal components. Three components are therefore retained in the model.

3.G Toy example dataset: model selection via exhaustive grid search of all parameters

Instead of the sequential model selection procedure adopted in the toy example dataset (section 3.2.3), we have conducted cross-validation (CV) in which all of the possible parameters are crossed exhaustively. The ranges of the parameters considered were the same as in the sequential procedure:

- β: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
- λ_R: [0.1, 0.5, 1, 3, 5, 10, 30, 50]
- λ_L: [0.5, 1, 5, 7, 10, 15, 30, 45, 100]
- λ_G: [0.1, 0.5, 1, 2, 5, 10]

For the number of covariates R, we adopted the range of [1,2,3,4] because PCA on the predictor data matrix revealed that from the fifth component onwards, the proportion of explained variance is smaller than 5% (this has been depicted in Appendix 3.F). Crossing all of the possible parameters, we administered 5-fold CV to 15552 models in total.

The model with the smallest CV error was characterized by the parameters: $R = 3, \beta = 0.6, \lambda_R = 0.5, \lambda_L = 10, \lambda_G = 5$. The estimated weights and regression coefficients are reported in Table 3.7. It can be seen that the estimates are very similar to the ones found by the model obtained through the sequential approach of CV.

Table 3.7. Weights and regression coefficients provided by the 3-covariate model with the smallest cross-validation error

Weights

Block 1			
x1	0.420	0	0
x2	0.420	0	0
x3	0.439	0	0
x4	0.486	0	0
x5	0	0	0.330
x6	0	0	0.324
x7	0	0	0.288
x8	0	0	0.261
x9	0	0	0
x10	0	0	0
x11	0	0	0
x12	0	0	0
x13	0	0	0
x14	0	0	-0.021
x15	0	0	0
Block 2			
x16	0	0	0.343
x17	0	0	0.361
x18	0	0	0.316
x19	0	0	0.256
x20	0	0.437	0
x21	0	0.429	0
x22	0	0.439	0
x23	0	0.470	0
x24	0	0	0
x25	0	0	0
x26	0	-0.085	0
x27	0	0	0
x28	0	0	0
x29	0	0	0
x30	0	0	0

Logistic regression coefficients

1	-1.272
2	-0.096
3	1.499
intercept	-0.206

If we apply the one standard error rule to select the simplest model among those within 1 SE from the minimum CV error, we would need to make a choice regarding which parameter to look consider first. The number of covariates R can be considered as the most influential parameter, followed by the weighting parameter β. Prioritizing these two parameters, the one standard error rule selects the model: $R = 2, \beta = 0.5, \lambda_R = 0.5, \lambda_L = 30, \lambda_G = 1$. Table 3.8 shows the estimates of this 2-covariate model. It can be seen that the covariate which is distinctive to the second predictor block (D2 in Table 3.1) is excluded from this model. This is sensible because this covariate was defined to have a very small predictive influence on the outcome variable when generating the data: population value of the logistic regression weight was set at -0.01. Hence, it is natural that the exhaustive CV approach that only considers the prediction error could result in omitting this covariate. The two covariates extracted are in agreement to the covariates found by the sequential approach of CV.

Table 3.8. Weights and regression coefficients provided by the 2-covariate model with the one standard error rule

Weights

Block 1		
x1	0.369	0
x2	0.380	0
x3	0.507	0
x4	0.427	0
x5	0	0.345
x6	0	0.311
x7	0	0.265
x8	0	0.212
x9	0	0
x10	0	0
x11	0	0
x12	0	0
x13	0	0
x14	0	0
x15	0	0
Block 2		
x16	0	0.337
x17	0	0.378
x18	0	0.302
x19	0	0.206
x20	0	0
x21	0	0
x22	0	0
x23	0	0
x24	0	0
x25	0	0
x26	0	0
x27	0	0
x28	0	0
x29	0	0
x30	0	0

Logistic regression coefficients

1	-1.263
2	1.481
intercept	-0.199

3.H Data generation for multiclass toy example dataset

The data generating setup employed for our simulation study is adapted slightly such that it can generate more than two categories, in generating the toy example dataset for the multiclass classification problem. As for the simulation study, two blocks of predictor variables wree generated from three underlying covariates; one distinctive covariate per each predictor block and one comon covariate. Each predictor block comprised of 15 variables ($J = 30$ in total), and $I = 1000$ observation units were generated. With the population weights and logistic regression coefficients provied in Table 3.4, the toy example dataset was generated via the following setup:

$$
\begin{aligned}
&\mathbf{T} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{\Sigma} = 50^2\mathbf{I}_3) \\
&\mathbf{E} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{\Sigma}_E = \sigma^2\mathbf{I}_J) \\
&\mathbf{X}_C \leftarrow \mathbf{T}\mathbf{W}_C^\top + \mathbf{E} \\
&\mathbf{z}_m \leftarrow exp(\mathbf{T}\mathbf{p}_m^{(g)})/(1 + exp(\sum_{m'}^{M-1} \mathbf{T}\mathbf{p}_{m'}^{(g)})) \text{ for } m = 1, \ldots, M-1 \\
&\mathbf{z}_M \leftarrow 1/(1 + exp(\sum_{m'}^{M-1} \mathbf{T}\mathbf{p}_{m'}^{(g)})) \\
&g_{im} \sim Multinoulli(z_{im}) \text{ for } m = 1, \ldots, M
\end{aligned}
\tag{3.34}
$$

where $\mathbf{T}, \mathbf{\Sigma}$ and $\mathbf{W}_C$ are all defined in the same manner as in the simulation study (see section 3.3.1). The predictors $\mathbf{X}_C$ are generated by multiplying the covariate scores matrix with the weights matrix and adding random error. The diagonal covariance matrix $\mathbf{\Sigma}_E$ that governs the variance of error variables $\mathbf{E}$ is specified such that the covariates $\mathbf{T}$ account for 50% of variance in $\mathbf{X}_C$. $\mathbf{p}_m^{(g)}$ indicates the logistic regression coefficients for the log-odds of the mth category as opposed to the baseline category $M = 3$. The statements in the fourth and the fifth lines together specify the $(I = 1000 \times M = 3)$ matrix $\mathbf{Z}$; z_{im} denotes the probability of the ith observation belonging to mth category, defined according to the baseline-category logit model (Agresti, 2003). g_{im} is therefore sampled from a Multinoulli distribution defined by the prescribed probabilities z_{im}.

3.I Lasso and Group lasso penalty parameters initially fixed in the simulation study, per each condition

Dimensions	Relevant	VAF	λ_{G1}	λ_{G2}	λ_{G3}	λ_{L1}	λ_{L2}	λ_{L3}
low	D1,D2	0.8	0.5	0.5	0.5	20	10	20
low	D1,D2	0.5	0.5	0.5	0.5	30	15	30
low	D1,D2	0.2	0.5	0.5	0.5	30	15	30
low	D1,C	0.8	0.5	0.5	0.5	30	15	30
low	D1,C	0.5	0.5	0.5	0.5	30	15	30
low	D1,C	0.2	0.5	0.5	0.5	30	15	30
high	D1,D2	0.8	3	3	3	15	7.5	15
high	D1,D2	0.5	2	2	2	30	15	30
high	D1,D2	0.2	1	1	1	20	10	20
high	D1,C	0.8	1	1	1	10	10	10
high	D1,C	0.5	1	1	1	30	15	30
high	D1,C	0.2	1	1	1	10	10	10

3.J The scree test with acceleration factor conducted to determine the number of covariates for the 500 Family dataset

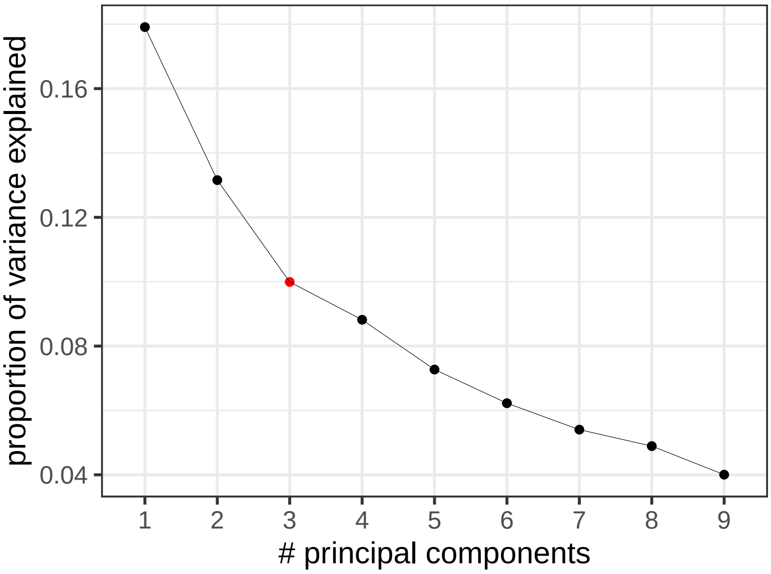


Figure 3.5. It can be seen that the sharpest change of slopes occurs at three components. Two components are therefore retained in the model.

3.K Models constructed from the 500 Family dataset using the related methods

Table 3.9. Estimates provided by PCR (SCaDS-logR), DIABLO and regularized logistic regression for the 500 Family dataset.

	SCaDS-logR		DIABLO		LogR
	1	2	1	2	b
Mother					
Relationship with partners	0	0.243	0	0	0
Argue with partners	0	0.247	0	0	0
Childs bright future	0	0	0	0	0
Activities with children	0	0	0	0	0
Feeling about parenting	0	0.175	0	0	0
Communation with children	0	0.338	0	0	0
Argue with children	0	0.152	0	0	0
Confidence about oneself	0	0.382	0	0	0
Father					
Relationship with partners	0	0.097	0	0	0
Argue with partners	0	0.208	0	0	0
Childs bright future	0	0	0	1	0.058
Activities with children	0	0	0	0	0
Feeling about parenting	0	0	0	0	0
Communation with children	0	0	0	0	0
Argue with children	0	0.255	0	0	0
Confidence about oneself	0	0.047	0	0	0
Child					
Child self confidence/esteem	0.274	0	0	0	0
Social life and extracurricular	0.333	0	0	0	0
Importance of friendship	0.460	0	0	0	0
Self image	0.360	0	1	0	0.278
Happiness	0.371	0	0	0	0
Confidence about the future	0.275	0	0	0	0

Chapter 4

Variable selection for both outcomes and predictors: Sparse Multivariate Principal Covariates Regression

Datasets comprised of large sets of both predictor and outcome variables are becoming more widely used in research. In addition to the well-known problems of model complexity and predictor variable selection, predictive modelling with such large data also presents a relatively novel and under-studied challenge of outcome variable selection. Certain outcome variables in the data may not be adequately predicted by the given sets of predictors. In this paper, we propose the method of Sparse Multivariate Principal Covariates Regression that addresses these issues altogether by expanding the Principal Covariates Regression model to incorporate sparsity penalties on both of predictor and outcome variables. Our method is one of the first methods that perform variable selection for both predictors and outcomes simultaneously. Moreover, by relying on summary variables that explain the variance in both predictor and outcome variables, the method offers a sparse and succinct model representation of the data. In a simulation study, the method performed better than methods with similar aims such as sparse Partial Least Squares at prediction of the outcome variables and recovery of the population parameters. Lastly, we administered the method on an empirical dataset to illustrate its application in practice.

Keywords: Outcome variable selection, Response variable selection, Response selection, Variable selection, Principal Covariates Regression, Dimension reduction

Park, S., Ceulemans, E., & Van Deun, K. (under review).

4.1 Introduction

Following the advancements of technology for data collection, most research disciplines are faced with challenges arising from an abundance of data. In deriving a prediction model, researchers are increasingly encountering a setting where they handle a bulk of data at both ends of predictor and outcome variables. For example, Stein et al. (2010) proposed a model that predicts the volume of each location of the brain (measured by large fMRI data) by numerous predictors from genome-wide association (GWAS) data. Specific genetic polymorphisms that are strongly associated with different parts of the brain were explored and identified therein. Similarly, Mayer, Rahman, Ghosh, and Pal (2018) used genome-wide expression data in predicting responses of cell lines to several types of drugs. The study adopted random forests to find subsets of biologically meaningful associations between transcription rates and responses to drugs. Other examples include image-on-image regression in which an image is employed to predict another image (Guo, Kang, & Johnson, 2020), or multitrait GWAS where multiple correlated phenotypic traits are modelled together by genotypic variables (Kim, Zhang, & Pan, 2016; Oladzad et al., 2019). Studies that investigate associations between genes (M. Y. Park & Hastie, 2008), or across protein and DNA (Zamdborg & Ma, 2009) are also along these lines.

Predictive modelling in the presence of such large amounts of data presents two well-known issues. First, a constructed prediction model with many variables is difficult to interpret due to the sheer number of coefficients; studying the predictor-outcome relationship becomes complicated. Second, certain predictor variables may be redundant. In a setting like the fMRI-GWAS study (Stein et al., 2010) where variables are collected without a specific research question, there is a need to screen out non-essential predictors that do not have any predictive power.

A related but rarely visited issue is that some of the outcome variables may also be redundant. They may not have a substantial relationship with any of the predictors, meaning that they cannot be predicted by the available sets of predictors. Such outcome variables are expected especially in the context of an exploratory research setup. For example, in a multitrait GWAS setup comparable to the aforementioned studies, not all phenotypes may have strong relationships with the available transcription rates and researchers may want to identify a subset of phenotypes that are relevant to the available genetic predictors. Removal of unimportant outcome variables in such cases can also be helpful because these unimportant outcomes may obscure relevant predictive relationships pertaining to other outcome variables (Fowlkes & Mallows, 1983; Steinley & Brusco, 2008). Settings that can benefit from the exclusion of unimportant outcome variables are increasingly common these days with the growing number of investigations that incorporate non-targeted and naturally-occurring sources of data; prior information concerning predictor-outcome relationship is not available. Throughout the paper, we refer to these outcome variables that are not predictable by the given set of predictors as 'inactive outcomes', and otherwise as 'active outcomes'. This terminology is used in other papers that address outcome variable selection (Hu, Huang, Liu, & Liu, 2022; Su, Zhu, Chen, & Yang, 2016).

One way to deal with the two abovementioned well-known problems pertaining to complicated model representation and redundant predictor variables is the method of Principal Covariates Regression (PCovR; De Jong & Kiers, 1992). It is a combination of Principal Component Analysis (PCA) and Ordinary Least Squares (OLS) being applied in fields including chemometrics (Boqué & Smilde, 1999), material science (Helfrecht, Cersonsky, Fraux, & Ceriotti, 2020), health science (Taylor, Sullivan, Ellerbeck, Gajewski, & Gibbs, 2019) and clinical psychology (Nelemans et al., 2019). PCovR introduces 'principal covariates'; a low number of summary variables that condense the information in the large volume of predictor variables, akin to principal components in PCA. The outcome variable is then regressed on the principal covariates, significantly decreasing the number of regression coefficients to be estimated. However, since all of the predictor variables are involved in constructing the principal covariates, a large set of coefficients connecting the predictors with the covariates still has to be estimated. Understanding the nature of the covariates by inspecting these coefficients therefore becomes very cumbersome. To this end, PCovR has been extended to incorporate regularization penalties that induce sparseness in these coefficients (e.g. S. Park et al., 2020; Van Deun et al., 2018). This not only allows the covariates to be easily interpreted, but also discards the predictors that are redundant.

While PCovR and its sparse extensions accommodate for issues arising from a large set of predictors, they are primarily designed to address a single outcome variable. Similarly, whereas methods designed to eliminate redundant predictor variables have been extensively studied (Tibshirani, 1996; M. Yuan & Lin, 2006; Zou & Hastie, 2005), regression problems involving variable selection at the level of outcome variables have not received much attention. There have been many approaches to regress multivariate outcome variables jointly on the predictors instead of modelling the outcomes individually, but most of these works were confined to identifying predictors that are important for predicting all of the outcome variables (Luo, 2020; Obozinski, Taskar, & Jordan, 2006; Peng et al., 2010). Similarly, while multivariate methods such as Partial Least Squares (PLS) and Reduced Rank Regression (RRR) that have their basis on reducing the dimensionality of the variables have been extended to incorporate sparsity, majority of these extensions have only targeted predictor variables (L. Chen & Huang, 2012; Chung & Keles, 2010; Lê Cao et al., 2011). To our knowledge, there has been only a handful of studies that target outcome variable selection; these include regularized regression approaches (An & Zhang, 2017; Hu, Huang, et al., 2022), a sparse RRR method (K. Chen, Chan, & Stenseth, 2012) and a method within a framework of envelope modelling (Su et al., 2016).

In this paper, we propose the method of Sparse Multivariate Principal Covariates Regression (SMPCovR), an extension of PCovR methodology that tackles the variable selection problem for both predictor and outcome variables. Starting from the PCovR model, sparseness is promoted in both sides of the model; in constructing the covariates from the predictors and in predicting the outcome variables based on the covariates. The resulting model is not only sparse and easy to interpret, but also eliminates redundant predictor variables and inactive outcome variables. It contributes to the under-studied problem of variable selection of outcome variables.

The paper is arranged as follows. The next section provides methodological details of SMPCovR. We begin with a discussion of PCovR since it is the basis of our current method. A simulation study that comparatively evaluates SMPCovR along with other methods devised with similar research aims is presented afterwards. The method is also administered to an empirical dataset for an illustrative purpose, as well as to expand upon the comparison against competitive methods in a practical data setting. The paper concludes with a disussion. The R implementation of SMPCovR can be found on Github: `https://github.com/soogs/SMPCovR`. The code for generating the results in this paper is also available therein.

4.2 Methods

4.2.1 Notation

The following notation is used throughout the paper: scalars, vectors and matrices are denoted by italic lowercase, bold lowercase and bold uppercase letters respectively. Transposing is indicated by the superscript $^\top$. Lowercase subscripts running from 1 to corresponding uppercase letters denote indexing (i.e., $i \in \{1, 2, \dots, I\}$). Superscripts $^{(X)}$ and $^{(Y)}$ highlight affiliation with predictor and outcome variables, respectively. To denote estimates, a $\hat{}$ over the symbol denoting the population parameter is used. $\mathbf{X}$ refers to a $I \times J$ matrix containing the standardized scores of J predictors obtained from I observation units (that is, each column has mean zero and variance equal to one). $\mathbf{Y}$ denotes a $I \times L$ matrix of L continuous outcome variables that are mean-centered and scaled to variance equal to one, also observed on the same I observation units.

4.2.2 PCovR

We begin by discussing the method of PCovR and show how the method extends to the current method of SMPCovR. PCovR (De Jong & Kiers, 1992) is a combination of PCA and OLS. It models the predictor and outcome variables by using principal covariates which can be understood as summary variables. These covariates are linear combination of the predictors which are obtained such that they explain the variance in the predictor and outcome variables simultaneously. PCovR decomposes the predictors $\mathbf{X}$ and the outcome variables $\mathbf{Y}$ as follows:

$$\begin{aligned}\mathbf{Y} =&\mathbf{X}\mathbf{W}\mathbf{P}^{(Y)^\top} + \mathbf{E}^{(Y)}\\ \mathbf{X} =&\mathbf{X}\mathbf{W}\mathbf{P}^{(X)^\top} + \mathbf{E}^{(X)}\end{aligned} \tag{4.1}$$

where $\mathbf{W}$ denotes the weights matrix of size $J \times R$: the predictor variables are multiplied by the weights to construct principal covariates $\mathbf{T} = \mathbf{X}\mathbf{W}$ with w_{jr} is the weight corresponding to the jth predictor variable and the rth covariate. It can be seen that both $\mathbf{Y}$ and $\mathbf{X}$ are modelled on the basis of the covariates $\mathbf{X}\mathbf{W}$. The first line of Equation (4.1) is the model for the outcome variables: $\mathbf{P}^{(Y)}$ refers to the regression coefficients matrix of size $L \times R$ with $p_{rl}^{(Y)}$ is the regression coefficient linking the rth covariate with the lth outcome variable. The residuals pertaining to the outcome variables are denoted by $\mathbf{E}^{(Y)}$. On the other hand, the second line of the equation gives the model for the predictors. $\mathbf{P}^{(X)}$ indicates the loadings matrix of size $J \times R$; $p_{rj}^{(X)}$ is the loading that connects the rth covariate with the jth predictor variable.

The following loss function is minimized when estimating the model parameters:

$$L(\mathbf{W}, \mathbf{P}^{(X)}, \mathbf{P}^{(Y)}) = \alpha \frac{\left\|\mathbf{Y} - \mathbf{X}\mathbf{W}\mathbf{P}^{(Y)^\top}\right\|_2^2}{\|\mathbf{Y}\|_2^2} + (1-\alpha)\frac{\left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^{(X)^\top}\right\|_2^2}{\|\mathbf{X}\|_2^2}, \tag{4.2}$$

where $0 \leq \alpha \leq 1$ is a user-specified tuning parameter that expresses the balance between focussing on the reconstruction of predictors or the prediction of the outcome variables in deriving the covariates. With α specified as 0, the method boils down to PCA where principal components are found by only considering the predictors. When $\alpha = 1$, the method becomes equivalent to reduced rank regression (Izenman, 1975; Kiers & Smilde, 2007). Constraints are needed to identify a unique solution from (4.2); an orthonormality constraint is usually placed upon the covariates ($\mathbf{T}^\top\mathbf{T} = (\mathbf{XW})^\top(\mathbf{XW}) = \mathbf{I}$).

The principal covariates can be understood as underlying processes that explain the relation of the outcome variables to the predictor variables. Thus, it is often of research interest to interpret the constructed covariates. All of the parameter sets $\mathbf{W}$, $\mathbf{P}^{(X)}$ and $\mathbf{P}^{(Y)}$ can be studied as they offer insights from different angles. The weights matrix $\mathbf{W}$ provides the composition of the covariates as it prescribes how the predictor variables are combined to form the covariates. The loadings matrix $\mathbf{P}^{(X)}$ shows how the covariates recover back the predictors. Additionally, if the covariates are scaled to variance equal to one ($\mathbf{T}^\top\mathbf{T} = I\mathbf{I}$), the loadings are equivalent to the correlation between the covariates and the predictors. Lastly, the regression coefficients $\mathbf{P}^{(Y)}$ represent how the covariates are used to predict the outcome variables. Unlike the weights and the loadings matrices, the regression coefficients concern the link between the covariates and the outcome variables.

4.2.3 SMPCovR

When large sets of predictor variables and outcome variables are present, inspecting the PCovR estimates to understand the nature of the covariates becomes difficult. Also, the dataset may present redundant predictors and inactive outcomes. The novel method of SMPCovR induces sparseness in the weights $\mathbf{W}$ and regression coefficients $\mathbf{P}^{(Y)}$ so that these issues are resolved within the context of PCovR.

4.2.3.1 Model and objective function

SMPCovR models the predictor and the outcome variables in the same manner as the PCovR model above yet with the additional constraint that only few variables make up the covariates and that not all outcome variables are predictable by (all) covariates. Such a sparse model can be attained by adding penalties to the objective expressed in (4.2):

$$\begin{aligned} L\left(\mathbf{W}, \mathbf{P}^{(X)}, \mathbf{P}^{(Y)}\right) = {} & \frac{\alpha}{\|\mathbf{Y}\|_2^2} \left\|\mathbf{Y} - \mathbf{XW}\mathbf{P}^{(Y)^\top}\right\|_2^2 + \frac{1-\alpha}{\|\mathbf{X}\|_2^2} \left\|\mathbf{X} - \mathbf{XW}\mathbf{P}^{(X)^\top}\right\|_2^2 \\ & + \sum_r^R \lambda_{Lr} \|\mathbf{w}_r\|_1 + \sum_r^R \lambda_{Rr} \|\mathbf{w}_r\|_2^2 \\ & + \sum_r^R \gamma_{Lr} \left\|\mathbf{p}_r^{(Y)}\right\|_1 + \sum_r^R \gamma_{Rr} \left\|\mathbf{p}_r^{(Y)}\right\|_2^2 \end{aligned} \tag{4.3}$$

where the loadings associated with the predictors $\mathbf{P}^{(X)}$ are constrained to be column-orthogonal ($\mathbf{P}^{(X)\top}\mathbf{P}^{(X)} = \mathbf{I}$) in order to avoid trivial solutions with very small weights (close to zero) and very large loadings. Just as in the objective criterion for PCovR, the first and the second terms are sum of squares that concern the regression problem and the PCA problem, respectively. The two terms are balanced by specification of the α parameter ($0 \leq \alpha \leq 1$). Note that the constraint on the covariates employed for PCovR is removed for this objective criterion.

The terms with λ_{Lr} and λ_{Rr} respectively refer to the lasso and ridge penalties for the weights corresponding to the rth covariate, while the terms with γ_{Lr} and γ_{R_r} indicate the lasso and the ridge penalties imposed on the regression coefficients. While the lasso penalty enforces the coefficients to zero and discards the variables from the model, the incorporation of the ridge penalty prevents divergence occurring due to covariates being correlated. This combination of the lasso and ridge penalties is also known as the elastic net penalty (Zou & Hastie, 2005). When all of the regression coefficients corresponding to an outcome variable are forced to zero, this outcome variable is modelled by zero and excluded from the model. Likewise, all of the weights corresponding to a predictor being penalized to zero removes the predictor variable from the model.

4.2.3.2 Algorithm

Estimates of the SMPCovR parameters can be obtained by alternating least squares. In turn, one of the parameter sets among $\mathbf{W}$, $\mathbf{P}^{(X)}$ and $\mathbf{P}^{(Y)}$ is estimated conditionally upon fixed values of the others. The elastic net problems for $\mathbf{W}$ and $\mathbf{P}^{(Y)}$ are convex problems, and they are both tackled via coordinate descent (Friedman et al., 2010a). On the other hand, the conditional problem for $\mathbf{P}^{(X)}$ is known as an Orthogonal Procrustes Problem (Schönemann, 1966); it is not convex, but has a closed-form solution (ten Berge, 1993). Since each of the estimation problems for $\mathbf{W}$, $\mathbf{P}^{(X)}$ and $\mathbf{P}^{(Y)}$ can converge at the global optimum of the conditional (penalized) least squares problem, the resulting alternating least squares procedure is monotonic. However, there is no guarantee of convergence to the global optimum for the combined problem (4.3), due its non-convexity. To avoid local minima, we recommend to use multiple random starting values, along with rational starting values based on PCovR. Further details on the algorithm for minimizing the objective function can be found in Appendix 4.A, including the schematic outline of the algorithm and the derivation of solutions to the conditional updates (Appendices 4.B, 4.C, 4.D).

4.2.3.3 Model selection

The SMPCovR method entails the following list of tuning parameters that shape the model construction.

- Number of covariates R
- Weighting parameter α
- Lasso parameters concerning weights $\boldsymbol{\lambda}_L$
- Ridge parameters concerning weights $\boldsymbol{\lambda}_R$
- Lasso parameters concerning regression coefficients $\boldsymbol{\gamma}_L$
- Ridge parameters concerning regression coefficients $\boldsymbol{\gamma}_R$

We employ k-fold cross-validation (CV) as a standard model selection method for all of the tuning parameters except for the number of covariates R. Although a conventional model selection scheme with CV would consider all possible combinations of different values for all of the tuning parameters involved, such an exhaustive strategy would be computationally intensive, considering that the method is devised to cater for large sets of both predictor and outcome variables. Therefore, a sequential approach is adopted where the number of covariates is determined prior to tuning for the remaining other tuning parameters with CV. Such a sequential approach has been shown to be a suitable model selection strategy for the methods that precede SMPCovR: PCovR and sparse PCovR (S. Park et al., 2020; Vervloet et al., 2016).

The number of covariates R is therefore tuned as the first step of the sequential approach. PCA is performed on the concatenated data matrix $[\mathbf{Y} \;\; \mathbf{X}]$ to find a suitable number of principal components. This number of components would be adopted as the number of covariates R for SMPCovR. A typical approach is the use of scree plot in which an 'elbow' is searched for from a plot that illustrates the amount of variance each principal component explains. However, since this location of the elbow can involve a subjective opinion, the acceleration factor technique (Raîche et al., 2013) is employed instead. It is an objective method that finds at which principal component the amount of explained variance changes most abruptly. It is along the same line as other strategies devised to objectively search for the elbow, such as the Convex Hull method (Wilderjans, Ceulemans, & Meers, 2013). We make use of the implementation in the R package "nFactors" (Raiche et al., 2020).

The subsequent step is to determine the values of α, $\boldsymbol{\gamma}_L$ and $\boldsymbol{\lambda}_L$ simultaneously via CV. In doing so, the number of covariates found in the previous step is used. Also, the ridge parameters $\boldsymbol{\lambda}_R$ and $\boldsymbol{\gamma}_R$ for weights and regression coefficients respectively are fixed at a small value. This is because the ridge penalties primarily play the role of preventing divergence, rather than actively shaping the model structure. For this reason, we also recommend having the parameters fixed at a small value for the final model-fitting step of SMPCovR. Hence, with $R, \boldsymbol{\lambda}_R$ and $\boldsymbol{\gamma}_R$ fixed, different ranges for the three tuning parameters α, $\boldsymbol{\gamma}_L$ and $\boldsymbol{\lambda}_L$ are crossed and all resulting combinations are considered for the CV.

R^2_{cv} criterion for model evaluation via cross-validation

In conducting the CV, we employ an adapted form of R^2 to cater for outcome variable selection. The conventional R^2 incorporates the entire set of outcome variables. Instead, we define a new measure, R^2_{cv}, that computes the R^2 from the *CV test set* only on the basis of the outcome variables that are active in the model fitted from the *CV training set*:

$$R^2_{\text{cv}} = 1 - \frac{\left\| \mathbf{Y}^{test}_{L^*} - \mathbf{X}^{test}\hat{\mathbf{W}}\hat{\mathbf{P}}^{(Y)^\top}_{L^*} \right\|^2_2}{\left\| \mathbf{Y}^{test}_{L^*} \right\|^2_2} \tag{4.4}$$

where $\mathbf{Y}^{test}$ and $\mathbf{X}^{test}$ refer to the outcome and predictor variables in the CV test set. The subscript $_{L^*}$ denotes a subset within the sequence of indices for outcome variables $L^* \subseteq \{1, 2, \ldots, L\}$. It comprises of indices corresponding to outcomes included in the SMPCovR model fitted on the CV training set $(\mathbf{Y}^{train}, \mathbf{X}^{train})$. Since an outcome variable is removed from the model if its corresponding row in the estimated regression coefficients matrix $\hat{\mathbf{P}}^{(Y)}$ is a zero-vector, the indices of non-zero rows of $\hat{\mathbf{P}}^{(Y)}$ make up the set L^*. $\hat{\mathbf{P}}^{(Y)}_{L^*}$ denotes the submatrix of $\hat{\mathbf{P}}^{(Y)}$ with non-zero rows.

The use of the R^2_{cv} criterion therefore omits the outcome variables deemed inactive by the model fitted on the CV training set. In our experiments, incorporating the entire set of outcome variables in the computation of the R^2 from CV often resulted in the set of tuning parameters that include all of the outcome variables; instead, the R^2_{cv} performs well in identifying the active outcome variables (see below Section 4.2.3.4 which provides an illustrative example). One may argue that this criterion is not suitable for comparing different tuning parameters, because models characterized by different tuning parameters would comprise of different sets of active outcome variables. These different sets of active outcomes are then used to calculate the R^2_{cv} measure to conduct evaluation across different models. However, we believe that this practice is not very far away from the general rationale behind CV for variable selection. In a common setting with lasso regression where variable selection is performed only on the predictors, models comprised of different sets of predictors are compared against each other. We consider the use of R^2_{cv} to be a suitable way to expand upon this practice to compare different models.

Model selection after the cross-validation procedure

We rely on the one standad error rule (1SE rule; Friedman et al., 2001) to select the final model after cross-validation. The 1SE rule would favour the model with the lowest model complexity within 1SE of the model that resulted in the highest R^2_{cv}. After the CV procedure, we propose to use the sets of tuning parameters within the 1SE region to fit the SMPCovR model once again on the entire dataset, in order to evaluate the model complexity[1]. The set of tuning parameters that resulted in the smallest number of total non-zero coefficients (weights and regression coefficients together) is selected within the 1SE region. If multiple models are characterized by the same number of non-zero coefficients, models with a smaller set of regression coefficients are preferred over those with a smaller set of weights. This is because we consider outcome variable selection as a distinct feature offered by SMPCovR; selection of predictor variables can be achieved with many other tools, including sparse PCovR.

In addition to model complexity, if there are prior expectations concerning certain outcome variables or research aims regarding the number of active outcomes, these can also be taken into account. To be concrete, there may be an outcome variable of particular interest that must be included in the model, or the research goal may pertain to finding the smallest subset of active outcome variables. We believe that it is sensible to allow incorporation of such research aims along with model complexity for the model selection within the 1SE region, because all of the models that fall within the 1SE region can be seen as adquate model candidates.

4.2.3.4 Toy example for model selection

In this section we make use of a toy example dataset to demonstrate the model selection procedure. The dataset is generated according to one of the conditions of the simulation study that follows below. It comprises of 200 predictor and 20 outcome variables concerning 100 observation units. While three covariates underlie the variables, only 90 predictor and 12 outcome variables are linked with the covariates. Other remaining predictors and outcomes are redundant; the

[1] When tuning parameters governing regularization penalties are selected, it is common that the tuning parameters that are associated with lower levels of complexity are directly chosen, instead of the additional model-fitting step on the entire dataset. Namely, for lasso regression, models with higher lasso values are selected within the 1SE region, since it is the only tuning parameter that defines complexity. However, SMPCovR has three such tuning parameters ($\boldsymbol{\lambda}_L$, $\boldsymbol{\gamma}_L$ and α) which is difficult to translate into a single measure of complexity. Therefore, this additional model-fitting step is proposed.

covariates do not explain the variance in them. More details of the data generation setup can be found in the simulation study section.

Prior to the model selection procedure, both predictor and outcome variables were mean-centered and scaled to variance 1. As the first step of the sequential model selection approach, PCA was administered to the concatenated set of predictor and outcome variables. Figure 4.6 in Appendix 4.F displays the variance explained by each component. The acceleration factor technique revealed that the rate of change in the variance is the largest at the fourth component, and hence the number of SMPCovR covariates was determined as three.

With the number of covariates fixed at three, a 5-fold cross-validation was conducted by crossing the ranges of values of α, $\boldsymbol{\gamma}_L$ and $\boldsymbol{\lambda}_L$. For this illustrative example, we employed the ranges of [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] for α, all equally distanced sequence of size 20 from 10^{-5} to 0.5 on the natural log scale for $\boldsymbol{\gamma}_L$ and another equally distanced sequence of size 15 from 10^{-5} to 0.5 on the natural log scale for $\boldsymbol{\lambda}_L$. Ridge parameters $\boldsymbol{\lambda}_R$ and $\boldsymbol{\gamma}_R$ were fixed constant at 10^{-7} for the cross-validation. Crossing these ranges for the three tuning parameters, $9 \times 20 \times 15 = 2700$ models were put forward. The model with the highest R^2_{cv} value was characterized by the tuning parameters: $R = 3, \alpha = 0.1, \boldsymbol{\lambda}_L = 0.00779, \boldsymbol{\gamma}_L = 0.00744, \boldsymbol{\lambda}_R = 10^{-7}, \boldsymbol{\gamma}_R = 10^{-7}$. The model includes 12 outcome variables out of the total 20, which correctly represents the true underlying model. Figure 4.1 shows the number of active outcome variables picked up by the model with varying values of tuning parameters. It can also be seen that the three tuning parameters together impact the number of active outcomes included, implying that they should be tuned simultaneously via CV.

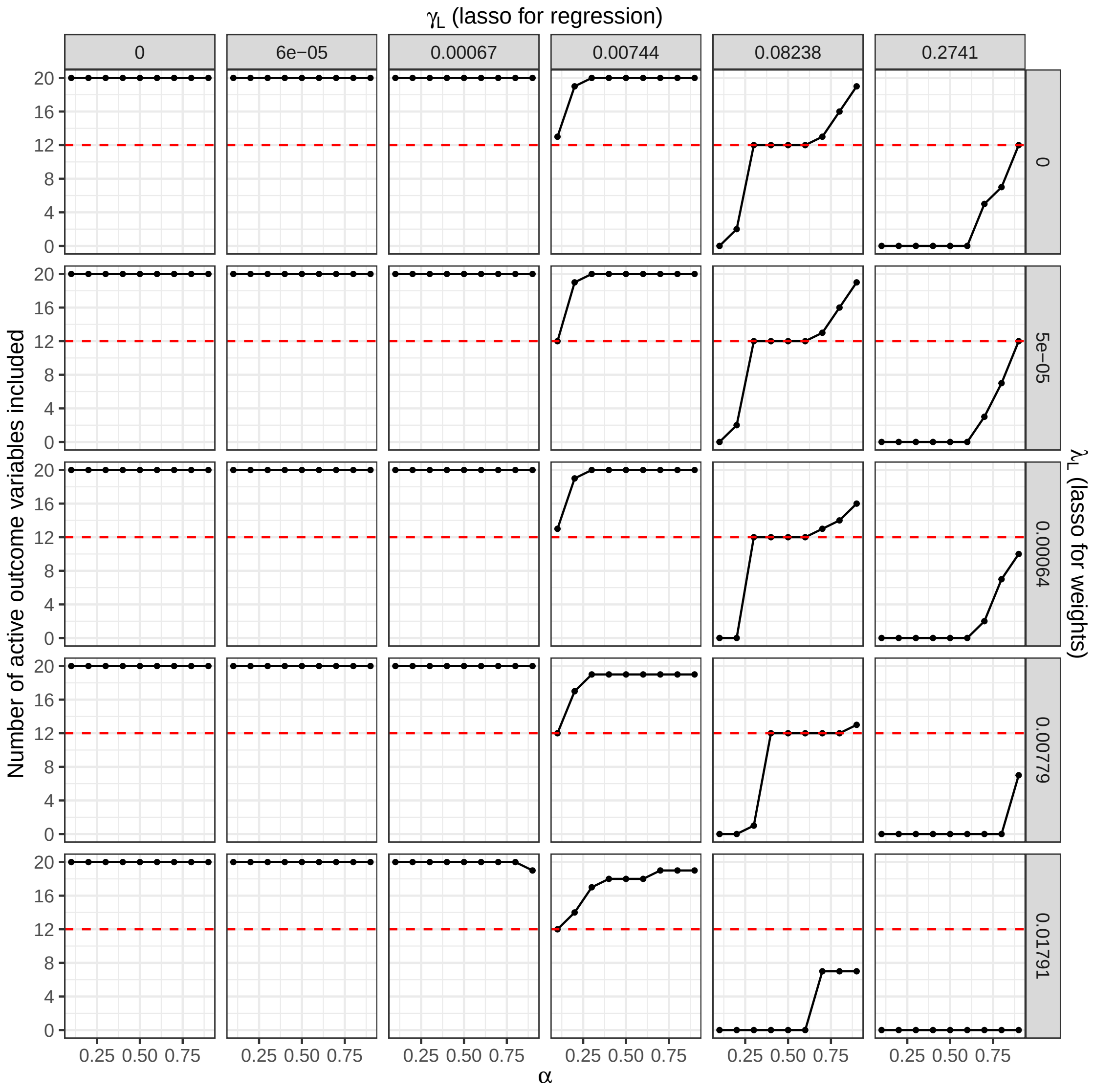


Figure 4.1. The number of outcome variables included by different model configurations. The dashed line indicates that the true covariates underlie 12 outcome variables out of the total 20.

There were no other models which resulted in R^2_{cv} within the 1SE. In practice, the aforementioned tuning parameters would then be selected. However, for the sake of demonstration of the model selection procedure, we have considered models that led to R^2_{cv} higher than *2 standard errors* below the maximal R^2_{cv}. The following table provides the tuning parameters of the models within the 2SE.

Table 4.1 shows that the four models do not differ much in the numbers of outcome variables included. Considering the model complexity, the numbers of estimated nonzero regression coefficients were also comparable to each other among the models. However, the number of nonzero weights was much lower in model 4. Hence, either model 4 or model 3 would be chosen depending on the research aim; if the goal is to find the least complex model, model 4 with

Table 4.1. The configurations of the models that fall within the 2 SE region from the maximum R^2_{cv}. SE denotes the standard error of R^2_{cv}, while 'Outcome included' refers to the number of outcome variables included. The models are arranged in a descending order of R^2_{cv}.

Model	α	$\boldsymbol{\gamma}_L$	$\boldsymbol{\lambda}_L$	R^2_{cv}	SE	Nonzero weights	Nonzero reg.	Outcome included
1	0.1	0.00744	0.00779	0.506	0.016	80	14	12
2	0.8	0.15027	0.00147	0.483	0.032	82	14	12
3	0.2	0.01357	0.00779	0.482	0.014	75	14	12
4	0.3	0.01357	0.01791	0.477	0.030	44	15	13

the smallest number of nonzero parameters would be the choice. However, if one aims to filter out as many inactive outcome variables as possible, model 3 would be selected; it is the least complex model among the models with 12 outcome variables.

4.2.3.5 Related methods

Our proposed method of SMPCovR accommodates three goals: (a) it is a prediction method for multiple continuous outcome variables, (b) it represents underlying predictive processes by covariates, and (c) it provides sparse coefficients at both sides of predictor and outcome variables. This section compares SMPCovR to other methods that are devised with a similar set of aims.

Regularized multivariate regression

Regularized multivariate regression is a regularized regression method which handles multiple outcome variables. This could be achieved by fitting regularized regression models with a group lasso penalty, where all of the regression coefficients pertaining to one predictor variable are grouped (e.g. implemented in the R package 'glmnet'; Friedman et al., 2021). The method drops out predictor variables completely from the model, providing a subset of predictors that are important in predicting all of the outcome variables jointly. Expanding on this group lasso idea to also conduct variable selection of outcome variables, An and Zhang (2017) proposed a regression method with double group lasso penalties: one penalty for the predictors and another for the outcomes. While these methods impose the penalties on the coefficients, Hu, Huang, et al. (2022) devised a formulation in which the penalty is placed on an indicator matrix that denotes the inclusion of the outcome variables. In this particular method, variable selection is only performed for the outcome variables. Compared against the SMPCovR method, while these approaches provide sparse coefficients for a problem with multiple outcome variables, they do not employ covariate or factor structures which capture shared explained variance across multiple predictor or outcome variables.

Sparse PCovR

Sparse PCovR (SPCovR; Van Deun et al., 2018) is an immediate predecessor of SMPCovR. The method finds sparse weights, but sparseness is not imposed to the regression coefficients $\mathbf{P}^{(Y)}$. In fact, SMPCovR without the lasso penalty on the regression coefficients boils down to SPCovR. Although the covariates can be found considering the multiple outcomes, an entire set of regression coefficients $\mathbf{P}^{(Y)}$ is estimated which is burdening for model interpretation. This also implies that inactive outcome variables are not filtered out from the model.

PCovR2

Developed as an extension of PCovR that caters for multiple outcome variables, PCovR2 (Gvaladze, Vervloet, Van Deun, Kiers, & Ceulemans, 2021) adds an additional type of covariates to the PCovR model. These extra covariates are summary variables that compress the multiple outcome variables, and they are regressed on the original PCovR covariates which summarize the predictor variables. By introducing these new covariates that combine the outcome variables, the method provides a succinct representation of predictor-outcome relationship. Like SMPCovR, PCovR2 is an extension of PCovR targeting for a multivariate outcome problem. However, the two methods address different research goals concerning multiple outcome variables. Whereas SMPCovR imposes sparseness and removes inactive outcomes, PCovR2 condenses the outcome variables into a more concise representation. Within PCovR2, sparseness is also not induced in the weights that combine predictor variables into covariates.

Sparse Principal Component Regression

A method called Sparse Principal Component Regression (SPCR; Kawano, 2021; Kawano, Fujisawa, Takada, & Shiroishi, 2015) has also been proposed. It relies on an objective function that is very similar to SMPCovR. However, SPCR is devised as a univariate regression problem; it targets only one outcome variable. While a separate model can be fitted on each outcome variable, this implies that different sets of covariates would be derived concerning each outcome variable. Although SPCR imposes sparseness in weights much like SPCovR, the univariate setup is not in line with the aim of SMPCovR which tries to construct interpretable covariates that account for a large set of predictor and outcome variables at the same time.

Sparse PLS

Sparse Partial Least Squares (sPLS; Chung & Keles, 2010; Lê Cao et al.,

2008) is a sparse extension to PLS, which is a well-known method in the same spirit as PCovR; it models predictor and outcome variables simultaneously by introducing summary variables (H. Wold, 1982; S. Wold, Ruhe, Wold, & Dunn, 1984). Just like in PCovR, these summary variables account for variance in both predictor and outcome variables. However, PLS does not incorporate the balancing parameter α. Although sPLS can model multiple outcome variables and performs variable selection for the predictors, it has not been extended to also enforce sparseness on coefficients that connect the summary variables with the outcome variables. However, outcome variable selection has been addressed within the framework of envelope modelling (Su et al., 2016). Envelope modelling[2] has been shown to be connected with PLS; the two methods target the same population parameters, but they differ in the method of estimation (Cook, Helland, & Su, 2013). Yet, the method in Su et al. (2016) is only designated for variable selection for the outcomes, and not for the predictor variables; the authors suggest a prior subset selection of predictor variables in the case of high-dimensionality. Therefore, similarly to sPLS, the method does not address the complete set of goals of SMPCovR.

4.3 Simulation study

While all of the above methods address a subset of aims targeted by SMPCovR, we selected SPCovR and sPLS as competing methods to assess the performance of our novel method via a simulation study. Regularized multivariate regression is not comparable to SMPCovR since it does not include dimension reduction. PCovR2 does not impose any sparseness in the coefficients; it does not perform variable selection to either predictor or outcome variables, which is one of the main goals of SMPCovR. SPCR is not devised for multiple outcome variables, so it is infeasible to include SPCR in our experiment. Lastly, to the best of our knowledge, the envelope method for outcome variable selection (Su et al., 2016) does not have a publicly available software implementation.

We have therefore conducted a simulation study in which we examine the performance of SMPCovR, SPCovR and sPLS with respect to the retrieval of underlying processes and the prediction of the multiple outcome variables. These underlying processes are specified by covariates that underlie the simulated data. Similar to the toy example in Section 4.2.3.4, the covariates only explain the variance in subsets of predictor and outcome variables.

[2] Envelope modelling (Cook, Li, & Chiaromonte, 2010) is a recent branch of methods that identifies ‘material’ and ‘immaterial’ parts of predictor and outcome variables. A linear model is constructed only on the basis of the useful ‘material’ parts, which allows efficient estimation and overcomes problems such as collinearity.

Owing to the sparsity penalty imposed upon the regression coefficients, we expect SMPCovR to outperform the other two methods in prediction when some outcome variables are inactive. By filtering out the inactive variables that are not related with the underlying covariates, overfitting of these inactive variables would be avoided. As a consequence, this would result in better prediction quality of the outcome variables overall, compared to SPCovR and sPLS.

Since the defined covariates underlie both predictor and outcome variables, the quality of retrieval of the underlying processes can be studied from two angles: (1) covariate-predictor relationships and (2) covariate-outcome relationships. With respect to the covariate-predictor relationships, it is anticipated that SMPCovR and SPCovR would show comparable performance because they are equipped with the same set of sparsity penalties on the weights. In contrast, sPLS is hypothesized to underperform as PLS-based methods have shown to be less effective in recovering the weights that prescribe the relationships between covariates and predictors (S. Park et al., 2020). On the other hand, we expect SMPCovR to provide better recovery of regression coefficients that represent the covariate-outcome relationships than the other two methods. Owing to the sparsity penalty imposed on the regression coefficients, SMPCovR would be able to discern between the important and unimportant covariate-outcome associations.

4.3.1 Design and procedure

Fixing the number of observations I to 100, the predictor variables were generated from an underlying model comprised of three covariates. While varying the number of outcome variables $\mathbf{Y}$ to be at either $L = 5$ or $L = 20$, we generated $J = 200$ predictor variables for the high dimensional setting and $J = 30$ for the low dimensional setting. The following setup was used.

$$\begin{aligned}
&\mathbf{T} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{\Sigma} = 50^2\mathbf{I}) \\
&\mathbf{E}^{(X)} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{\Sigma}_{E^{(X)}} = \sigma^2\mathbf{I}) \\
&\mathbf{E}^{(Y)} \sim \mathcal{MVN}(\mathbf{0}, \mathbf{\Sigma}_{E^{(Y)}} = \sigma^2\mathbf{I}) \\
&\mathbf{X} \leftarrow \mathbf{T}\mathbf{W}^\top + \mathbf{E}^{(X)} \\
&\mathbf{Y} \leftarrow \mathbf{T}\mathbf{P}^{(Y)^\top} + \mathbf{E}^{(Y)}
\end{aligned} \tag{4.5}$$

$\mathbf{T}$ (size 100×3) is the covariate scores matrix which is generated from multivariate normal distribution characterized by the mean vector $\boldsymbol{\mu} = \mathbf{0}$ and the diagonal covariance matrix $\mathbf{\Sigma}$ with diagonal elements fixed at 50^2. Therefore, the three covariates are the same size in variance and are uncorrelated. The weights matrix $\mathbf{W}$ (size $J \times 3$) is defined with 82% and 85% level of sparsity for low and high dimensional setups, respectively. Furthermore, it is ensured that the columns of the weights matrix are orthogonal to each other ($\mathbf{W}^\top\mathbf{W} = \mathbf{I}$; this constraint is not included in our objective function; it is used specifically for the data generation here). Since the covariates are defined to be uncorrelated, the model we use here can be seen as a PCA decomposition where the weights are equal to loadings. This is how $\mathbf{X}$ is defined by multiplying $\mathbf{T}$ and $\mathbf{W}$. The weights matrix defined for a low dimensional setup can be seen in Table 4.2.

Table 4.2. Weights defined for the low-dimensional setup.

	W	
1	2	3
0.5	0	0
0.5	0	0
0.5	0	0
0.5	0	0
0	0.354	0
0	0.354	0
0	0.354	0
0	0.354	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0.354	0
0	0.354	0
0	0.354	0
0	0.354	0
0	0	0.5
0	0	0.5
0	0	0.5
0	0	0.5
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

It can be seen that out of the 30 predictors in the low dimensional setting, 14 predictors are redundant; they are not related with any covariates. In the high dimensional setting, 110 predictors out of the 200 are defined as being redundant. Similarly, in specifying the regression coefficients $\mathbf{P}^{(Y)}$ (size $L \times 3$), 40% of the outcome variables are always defined as inactive; more details regarding the regression coefficients follow below.

$\mathbf{E}^{(X)}$ (size $100 \times J$) and $\mathbf{E}^{(Y)}$ (size $100 \times L$) denote the residual matrices corresponding to the predictor and outcome variables, respectively. They are drawn from multivariate normal distribution with zero mean vector and diagonal covariance matrices $\mathbf{\Sigma}_{E^{(X)}}$ and $\mathbf{\Sigma}_{E^{(Y)}}$, respectively. The two residual matrices are generated such that they are uncorrelated with each other, and also with the covariate scores. The variance of the residual matrices are governed by the design factors of the simulation study (given below): proportion of variance in $\mathbf{X}$ and $\mathbf{Y}$ explained by the underlying covariates. Four data characteristics were manipulated, based on the data generating model provided above. The different levels of the manipulated factors are given by square brackets.

Study setup

1. Number of predictors J: [200], [30]
2. Number of outcome variables: [5], [20]
3. Proportion of variance in $\mathbf{X}$ and $\mathbf{Y}$ explained by the covariates: [0.9], [0.5]

The first and the second design factors concern the dimensionality of the predictor and outcome variables, respectively. The $\mathbf{P}^{(Y)}$ matrices created by the third design factor are shown below. We show the matrices corresponding to 5 outcome variables; the coefficients were defined in a similar manner for the case with 20 outcome variables, (provided in Appendix 4.E).

$$\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

The columns indicate the regression coefficients corresponding to each covariate. As aforementioned, 40% of the outcome variables (2 out of 5) are not linked with any of the covariates. Fully crossing the design factors and generating 20 datasets per condition, $2 \times 2 \times 2 \times 50 = 400$ datases were produced. Three different analyses were administered to each of these datasets: SMPCovR, SPCovR and sPLS.

4.3.2 Model selection

The model selection procedure for SMPCovR in the simulation study follows the procedure detailed in Section 4.2.3.4, except for the number of covariates which is fixed at three, by following the true covariate structure. The tuning parameters $\alpha, \boldsymbol{\lambda}_L$ and $\boldsymbol{\gamma}_L$ are then chosen at the same time by cross-validation. The ranges of [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] was adopted for α, and equally distanced sequence of size 7 from 10^{-5} to 0.5 on the natural log scale for both $\boldsymbol{\lambda}_L$ and $\boldsymbol{\gamma}_L$. The ridge parameters $\boldsymbol{\gamma}_R$ and $\boldsymbol{\lambda}_R$ are fixed at 10^{-7}. Crossing these ranges for the three tuning parameters, $9 \times 7 \times 7 = 441$ models were considered for each replicate dataset. The R^2_{cv} measure employed for the toy example dataset was used to evaluate each model by 5-fold CV. As discussed in Section 4.2.3.3, the 1SE rule was used to choose the model. We assumed a research scenario of selecting the least complex (least number of total non-zero coefficients) model, among the models that include the least number of outcome variables. The selected tuning parameters were applied in the final model estimation step, where the ridge parameters were again fixed at 10^{-7}.

With regards to SPCovR, the number of covariates was fixed at three following the true number of covariates. Then, the α parameter and the lasso parameter concerning the weights were selected together by 5-fold CV. The same range as used in SMPCovR was considered for α, while equally distanced sequence of size 15 from 10^{-5} to 0.5 was employed for the lasso parameter. Similar to the procedure for SMPCovR, the ridge parameters for the weights and the regression coefficients were fixed constant at (10^{-7}) for both model selection and estimation. In total, $9 \times 15 = 135$ models were evaluated. The 1SE rule was employed in such a way that the model with the smallest α and the largest lasso parameter for the weights was chosen, as they encourage a more sparse model to be found.

Lastly, the number of covariates for sPLS was fixed at three again. Unlike SMPCovR and SPCovR, the sparsity of the model for sPLS can be directly specified by providing the number of non-zero coefficients (linking the predictor variables with the covariates) as input. The number of non-zero coefficients to be included in the sPLS model was chosen through 5-fold CV. The range of [4, 8, 12, 16, 20, 28] non-zero coefficients per covariate was considered for the low dimensional setup ($6^3 = 216$ models in total), while the range of [25, 40, 50, 75, 80, 100, 120, 125, 150, 160, 175, 200] was employed for the high dimensional setup ($12^3 = 1728$ models in total). We used the 1SE rule to pick out the model with the least number of non-zero coefficients.

4.3.3 Evaluation criteria

The following three measures were employed to study the performance of the methods:

1. R^2_{out}: proportion of explained variance in the outcome variables in the out-of-sample test dataset.
2. Correct weights classification rate: proportion of the elements in $\mathbf{W}$ correctly classified as zero and non-zero elements relative to the total number of coefficients.
3. Correct regression coefficients classification rate: proportion of the elements in $\mathbf{P}^{(Y)}$ correctly classified as zero and non-zero elements relative to the total number of coefficients.

An independent test set (of 100 observation units) needed for computing the out-of-sample R^2 was generated following the same data generating procedures as the data used for model-fitting. Unlike the R^2_{cv} measure which was computed only on the basis of outcome variables included in the model, the R^2_{out} measure was calculated with respect to all of the outcome variables in the out-of-sample dataset. This is because the sPLS and SPCovR simply include all of the outcome variables, unlike SMPCovR:

$$R^2_{\text{out}} = 1 - \frac{\left\| \mathbf{Y}^{\text{out}} - \mathbf{X}^{\text{out}} \hat{\mathbf{W}} \hat{\mathbf{P}}^{(Y)^\top} \right\|_2^2}{\left\| \mathbf{Y}^{\text{out}} \right\|_2^2} \tag{4.6}$$

where $\mathbf{Y}^{\text{out}}$ and $\mathbf{X}^{\text{out}}$ indicate the outcome and predictor variables, respectively, from the out-of-sample data. The correct classification rates concerning the weights and the regression coefficients represent the method's ability in retrieving the underlying processes.

4.3.4 Results

4.3.4.1 Out-of-sample R^2_{out}

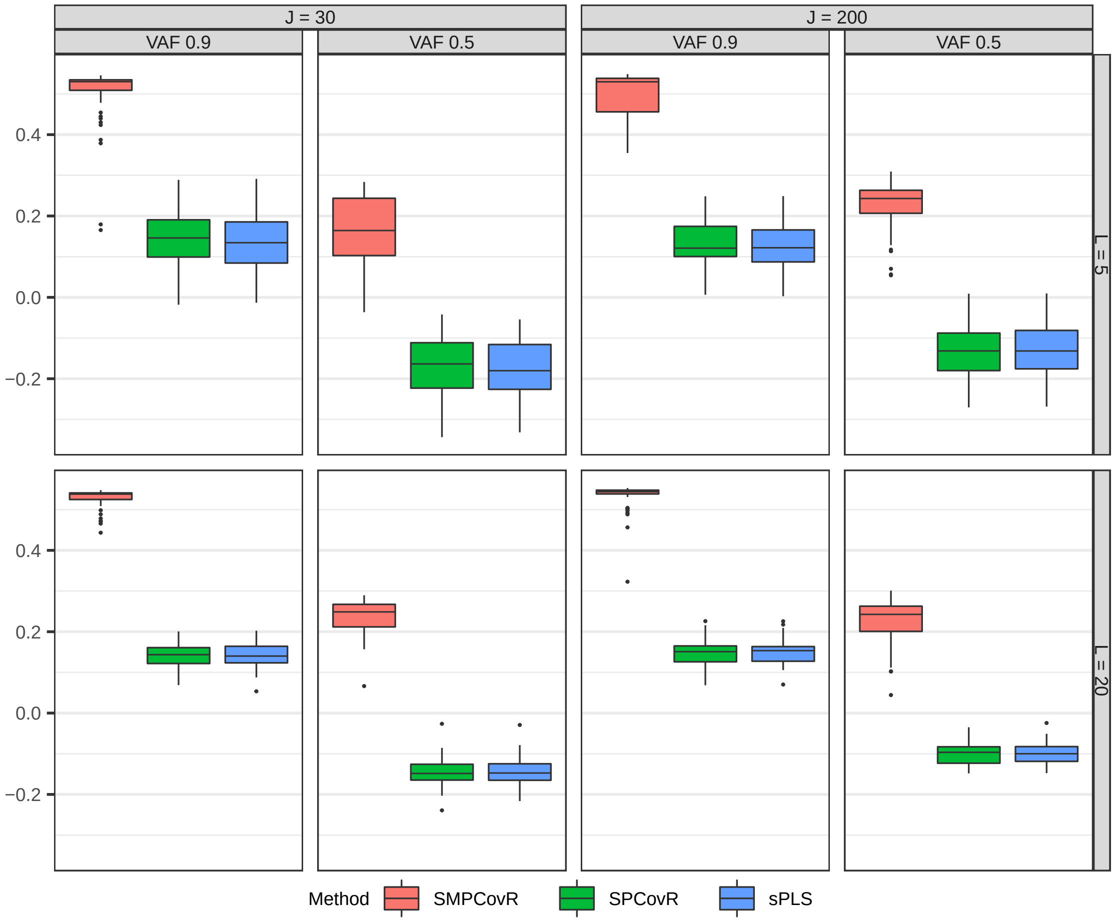


Figure 4.2. Boxplots of the out of sample R^2_{out}. Each panel corresponds to one of the 8 conditions.

The figure clearly shows the outperformance of SMPCovR over the other methods. None of the study design factors led to results pointing in another direction. The proportion of variance in data explained by the covariates resulted in an intuitive 'main effect'; all of the methods performed better with greater proportion of explained variance.

The outperformance of SMPCovR comes from the fact that the method screens out the inactive outcome variables, while the other methods include these outcome variables. This can be understood as a case of overfitting, since the other methods are modelling inactive outcomes which are only comprised of error variance. Appendix 4.G reports the R^2_{out} values computed only on the basis of active outcome variables; it can be seen that the three methods result in similar quality of prediction for the active outcomes. When the covariates explain 50% of variance in the variables, SMPCovR shows slight underperformance. Hence, the strength of SMPCovR originates from correct identification of active and inactive outcome variables.

4.3.4.2 Correct weights classification rate

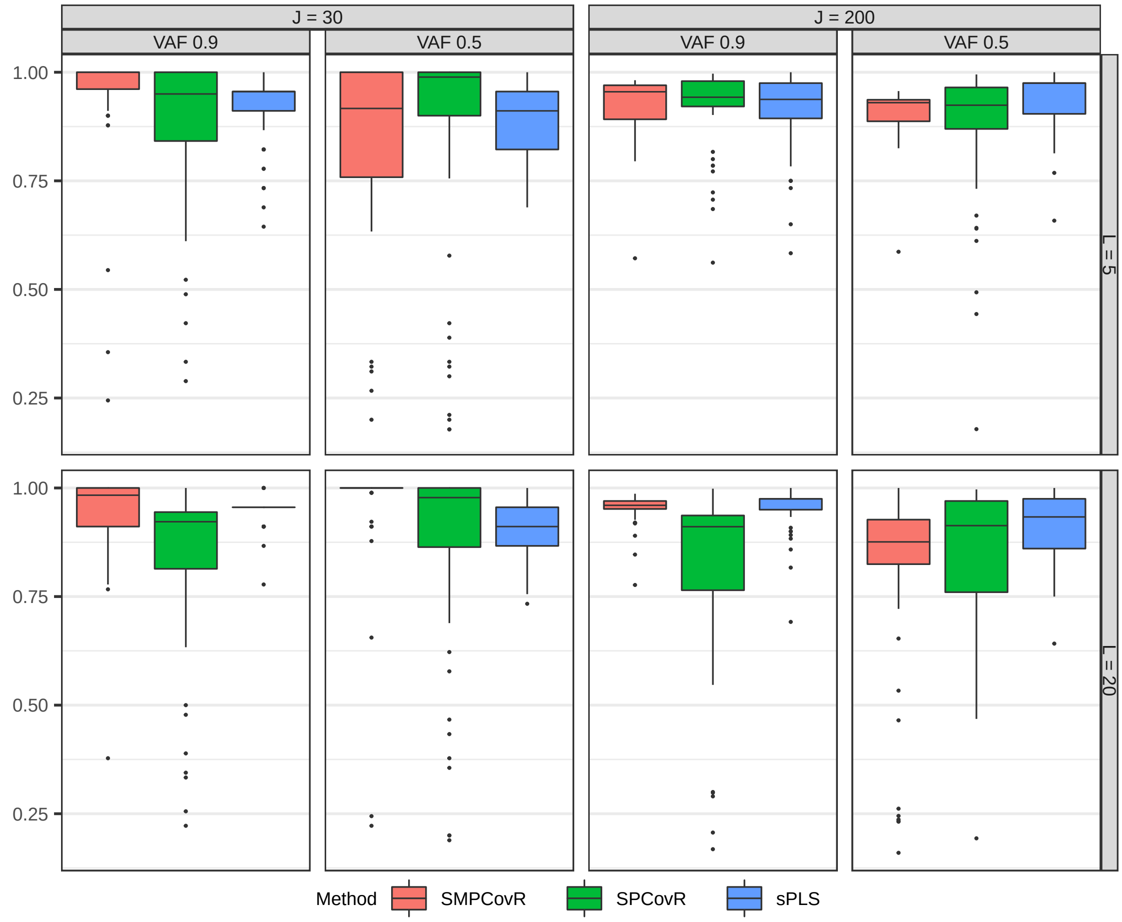


Figure 4.3. Boxplots of the correct classification rate for the **W**. Each panel corresponds to one of the 8 conditions.

Figure 4.3 portrays that the most impactful design factor in the comparative performance with respect to correct identification of the zero versus non-zero weights is the dimensionality of the predictors. In the low dimensional setting, SMPCovR and SPCovR resulted in comparable levels of correct classification rate which are higher than that of sPLS. In contrast, when the number of predictor variables exceeds the number of observations, the three methods have resulted in similar classification rates. Nevertheless, across most of the data conditions, it can be seen that similar levels of classification rates were obtained between the three methods.

4.3.4.3 Correct classification rate for regression coefficients

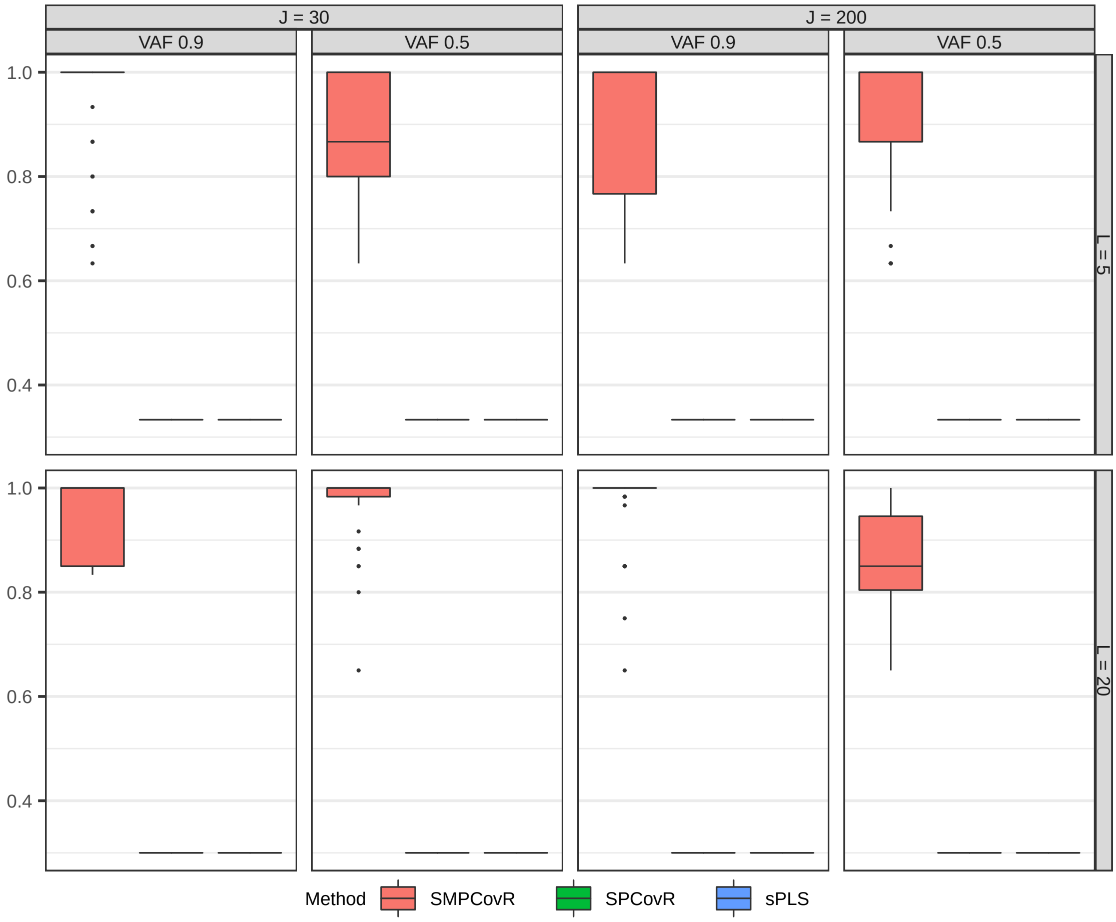


Figure 4.4. Boxplots of the correct classification rate for the $\mathbf{P}^{(Y)}$. Each panel corresponds to one of the 8 conditions.

As SMPCovR is the only method among the three with a sparsity penalty on regression coefficients, it correctly classified the regression coefficients far better than the other two methods which only provided non-zero regression coefficients. It appears that the true structure of the regression coefficients is recovered well in most conditions. In addition, Appendix 4.H shows the rate of correctly classified outcome variables. It can be seen that the method perfectly discerns between active and inactive in most of the replicate datasets.

The results conveyed in Figure 4.4 is natural because SPCovR and sPLS do not impose sparsity on the regression coefficients. However, we inspected the coefficients that the two methods provided for the true zero regression coefficients. The mean absolute discrepancy of the estimated coefficients from zero are reported in Appendix 4.I. It can be observed that the coefficients from the two methods are quite far away from zero under low dimensionality. For high dimensional data, while the mean discrepancy of SPCovR becomes near-zero, sPLS shows high discrepancy. This finding supports the use of a sparsity penalty on the regression coefficients, because without it, the methods struggle to derive near-zero values.

4.4 Empirical illustration: Pittsburgh Cold Study

We illustrate the use of SMPCovR by administering the method to an empirical dataset. We also apply SPCovR and sPLS on the same dataset to evaluate the effectiveness of our proposed method in a pratical setting.

4.4.1 Dataset and pre-processing

We adopted the dataset from the third wave of the Pittsburgh Cold Study (PCS) which took place from 2007 to 2011[3]. Healthy participants were invited and administered nasal drops of rhinovirus that causes symptoms of common cold. Severity of 16 types of symptoms related to cold and flu were self-reported each day up to five days after the virus exposure. Out of the 16, there were 8 symptoms that were known to comprise the common cold: headache, sneezing, chills, sore throat, runny nose, nasal congestion, cough and malaise (Jackson, DOWLING, SPIESMAN, & BOAND, 1958). Among the other symptoms, fever, muscle ache, joint ache and poor appetite have been identified as symptoms of flu (Monto, Gravenstein, Elliott, Colopy, & Schweinle, 2000), while there were 4 other related symptoms such as chest congestion, sinus pain, earache and sweating. Hence, it can be expected that the participants are more likely to develop the 8 cold symptoms than the other symptoms, as they were exposed to rhinovirus. Furthermore, 187 variables regarding the participants were also collected under various themes including blood chemistry, health practices and psychosocial states.

The participants are categorized into two groups according to the diagnosis of cold infection. This diagnosis was conducted by combining the serological testing of blood and illness criteria, and most of the participants were not diagnosed of cold infection. Therefore, we selected a subset of 46 participants by excluding the observations with missing values in the variables and to obtain a balance between the size of two diagnosis groups. Using the symptom variables as the outcome and the other variables as the predictors, we conduct SMPCovR to target the regression problem of symptom severity while constructing a model that de-

[3] The data were collected by the Laboratory for the Study of Stress, Immunity, and Disease at Carnegie Mellon University under the directorship of Sheldon Cohen, PhD; and were accessed via the Common Cold Project website (www.commoncoldproject.com; grant number NCCIH AT006694).

scribes the underlying predictive processes characterized by subsets of important predictor and outcome variables.

4.4.2 Model selection

Prior to the model selection and estimation, both predictor and outcome variables were centered and standardized such that the variance of each variable was equal to 1. We followed the model selection strategy outlined in Section 4.2.3.4. First, with the acceleration factor technique, the number of covariates was determined to be two. Appendix 4.J shows the proportion of variance explained with increasing number of components. The tuning parameters $\alpha, \boldsymbol{\lambda}_L$ and $\boldsymbol{\gamma}_L$ were selected via 5-fold cross-validation where the ridge parametrs $\boldsymbol{\lambda}_R$ and $\boldsymbol{\gamma}_R$ were fixed at 10^{-7}. The following ranges of values for the tuning parameters were employed.

Ranges of tuning parameters for cross-validation

- α: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
- $\boldsymbol{\lambda}_L$: 0 and equally distanced sequence of size 19 from 10^{-5} to 0.5 on the natural log scale
- $\boldsymbol{\gamma}_L$: 0 and equally distanced sequence of size 19 from 10^{-5} to 0.5 on the natural log scale

Crossing these ranges of the tuning parameters, we administered the cross-validation for $9 \times 20 \times 20 = 3600$ different models. As done in model selection for the toy example dataset and in the simulation study, the R^2_{cv} measure was used to determine the quality of prediction for each model. The 1 SE rule included 17 model configurations which are presented in the Table 4.3.

As discussed in 4.2.3.3, we selected the model using the 1SE rule. We assumed here a setup with the aim to select a model that is the least complex, among the models that contain the least number of outcome variables. Between models 1, 2 and 4 that are comprised of 10 outcomes, model 4 was selected since the number of non-zero weights was the lowest.

4.4.3 Results

Table 4.4 presents the weights and regression coefficients found by the chosen model. It first shows that only the first covariate is able to predict the cold symptoms; the model has excluded the second covariate in predicting the outcome variables. Out of the 187 predictor variables, 18 predictor variables compose the first covariate. IL-6, IL-8, IL-10 and TNF alpha are concentrations of

Table 4.3. The configurations of the models that fall within the 1 SE region from the maximum R^2_{cv}. SE denotes the standard error of R^2_{cv}, while 'Outcome included' refers to the number of outcome variables included. The models are arranged in a descending order of R^2_{cv}.

Model	α	$\boldsymbol{\gamma}_L$	$\boldsymbol{\lambda}_L$	R^2_{cv}	SE	Nonzero weights	Nonzero reg.	Outcome included
1	0.8	0.274	0.000	0.160	0.060	124	10	10
2	0.7	0.150	0.001	0.135	0.030	55	10	10
3	0.8	0.150	0.001	0.134	0.049	62	15	14
4	0.8	0.150	0.004	0.118	0.029	21	10	10
5	0.4	0.000	0.002	0.116	0.017	64	32	16
6	0.5	0.025	0.001	0.113	0.026	91	23	16
7	0.5	0.045	0.004	0.110	0.039	34	15	14
8	0.7	0.082	0.007	0.107	0.018	15	11	11
9	0.9	0.274	0.001	0.105	0.031	66	12	12
10	0.4	0.014	0.001	0.105	0.024	107	23	16
11	0.8	0.082	0.007	0.103	0.032	16	12	12
12	0.4	0.007	0.002	0.103	0.040	65	27	16
13	0.3	0.000	0.004	0.103	0.027	43	32	16
14	0.7	0.082	0.004	0.102	0.032	30	12	12
15	0.3	0.000	0.001	0.102	0.033	122	32	16
16	0.9	0.045	0.007	0.101	0.020	40	17	15
17	0.3	0.000	0.001	0.101	0.040	120	32	16

nasal cytokine. These concentrations were measured each day for five days after the viral exposure and summed. Among the total 7 variables present in the data concerning cytokine, these 4 were picked out by the model. The model also selected Corpuscular Hgb conc (hemoglobin concentration), Non-fasting glucose and Urea nitrogen among the 29 blood chemistry variables measured before the viral exposure. Whereas lower levels of hemoglobin appears to result in more cold-related symptoms, glucose and nitrogen levels seem to have the opposite effect. # weekdays alcohol refers to the amount of alcohol usually consumed during weekdays. The alcohol consumption appears to be positively associated with the cold-related symptoms. This was the only variable chosen among 17 variables regarding health practices such as smoking, sleeping and physical activity. The next 6 variables concern measures from various psychosocial assessment scales measured before the viral exposure. Sadness and fatigue were found to be related with cold symptoms from the 13 PANAS (Positive and Negative Affect Schedule; Watson, Clark, & Tellegen, 1988) measures that target mood and affect. While the ECR (Experiences in Close Relationships; Fraley, Waller, & Brennan, 2000) scale concerns adult attachment types, TSC (Tucker Social Control Scale; J. S. Tucker, 2002) is about how health behaviours are encouraged by the social environment.

Table 4.4. Weights and regression coefficients derived by SMPCovR from the PCS dataset. The weights are only provided for the predictors chosen by the model out of the total 187. $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$ indicate the weights corresponding to the first and second covariates respectively. The regression coefficients corresponding to all of the outcome variables in the dataset are provided.

$\hat{\mathbf{w}}_1$

	1
IL-6	0.363
IL-8	0.677
IL-10	0.408
TNF alpha	3.262
Corpuscular Hgb conc	-0.602
Non-fasting glucose	0.175
Urea nitrogen	0.308
# weekdays alcohol	0.882
PANAS: sadness	1.305
PANAS: fatigue	0.698
ECR: anxiety	0.171
TSC: network size	-0.007
Social participation	-0.371
Loneliness	0.032
Daily negative affect	0.070
Daily fatigue subscale	1.080
Daily fatigue	0.624
Daily anger	0.396

$\hat{\mathbf{w}}_2$

	2
PANAS: joviality	-0.036
PANAS: positive	-0.084
Psych well-being	-1.900

$\hat{\mathbf{P}}^{(Y)}$

	1	2
Sneezing	0.066	0
Runny nose	0.049	0
Nasal congestion	0.050	0
Cough	0.081	0
Sore throat	0.047	0
Headache	0	0
Chills	0	0
Malaise	0.023	0
Chest congestion	0.053	0
Sinus pain	0.061	0
Earache	0	0
Muscle ache	0	0
Joint ache	0	0
Sweating	0	0
Fever	0.045	0
Poor appetite	0.063	0

Similarly, social participation and loneliness were results of a self-report before the viral exposure. Predictors originating from several other assessment scales such as Perceived Stress Scale (Cohen, Kamarck, Mermelstein, et al., 1994), Emotion Regulation Questionnaire (Gross & John, 2003) and Family Environment Scale (Moos, 1990) were excluded by the SMPCovR model. Lastly, the daily negative affect, fatigue and anger variables come from daily interviews conducted prior to the viral exposure. Altogether, the first covariate represents the combined effect of these physiological and behavioural elements in leading to the various cold symptoms.

Ten symptoms out of the total 16 were indicated to be in relation with the first covariate. Six out of 8 symptoms characterizing the common cold according to Jackson et al. (1958)[4] were included in the model; it excluded headache and chills. It is also interesting to see that symptoms typically associated with flu such as fever and poor appetite are also included (Monto et al., 2000), while the participants were not exposed to an influenza virus known to cause flu.

[4] headache, sneezing, chills, sore throat, runny nose, nasal congestion, cough and malaise

The second covariate which is not relevant in predicting the symptoms is constructed with three predictor variables that measure positive mood and psychological well-being. However, we found that it explains much more variance in the predictor variables than the first covariate comprised of 18 variables. While the two covariates together explained 7.1% of variance in the predictors, the first covariate took account of only 0.9% while the second covariate explained the remainnig 6.2%.

To evaluate the quality of this model in outcome variable prediction, the R^2 measures were computed. We have calculated four different types of R^2 measures: $R^2_{\text{fit}_{all}}$, $R^2_{\text{fit}_{sub}}$, $R^2_{\text{loocv}_{all}}$ and $R^2_{\text{loocv}_{sub}}$. The first two measures were computed on the basis of in-sample data while the next two measures were results from leave-one-out CV. The measures with the subscript 'all' were computed with respect to all of the outcome variables in the dataset, while the others with the subscript 'sub' were derived on the basis of the subset of 10 outcome variables selected by the SMPCovR model. Appendix 4.K provides the formulae for these measures. To obtain a comparative insight about the quality of the SMPCovR method under the PCS dataset, we also computed the R^2 values using SPCovR and sPLS that were employed in the simulation study. We extracted two covariates for both methods in order to match the SMPCovR model. As done in the simulation study, 5-fold CV and the 1SE rule were employed to select the α and the lasso parameters for SPCovR and the number of non-zero coefficients for sPLS. Appendix 4.L provides the ranges of tuning parameters adopted to generate the models for the two methods. Table 4.5 reports the four different types of R^2 measures computed for the three methods.

Table 4.5. R^2 measures attained from the three methods from the PCS data.

	SMPCovR	sPLS	SPCovR
fit$_{all}$	0.167	0.206	0.266
fit$_{sub}$	0.267	0.280	0.376
loocv$_{all}$	0.119	0.052	0.115
loocv$_{sub}$	0.159	0.028	0.135

It can be seen that SMPCovR resulted in the highest R^2_{loocv} measures which represent the quality of out-of-sample prediction. While SPCovR showed comparable results with SMPCovR, sPLS fell short by a big margin. While both SPCovR and sPLS performed well for in-sample prediction with high R^2_{fit} values, the large discrepancy in the values compared to the R^2_{loocv} measures signal possible occurrence of overfitting. The models constructed by SPCovR and sPLS can be found in Appendix 4.L. While the SPCovR model found considerably more non-zero weights (43 and 1 for the two covariates, respectively), the sPLS model was comprised of

6 and 1 non-zero coefficients, leading to a more sparse model than the SMPCovR model.

Lastly, we inspected the SMPCovR model by plotting the covariate scores with the additional grouping information of diagnosis of cold infection (diagnosed using serological testing and illness criteria). Although this grouping information was not provided as a predictor, the two groups of cold and no cold can be fairly distinguished. As portrayed by the regression coefficients shown in Table 4.4, it appears that the first covariate is much more related with cold diagnosis than the second covariate. To conclude, the SMPCovR method was able to meet its goals when analyzing the PCS dataset. It derived a predictive model where some of the inactive outcome variables are filtered out while summarizing the predictor processes into interpretable covariates comprised of a small subset of predictor variables.

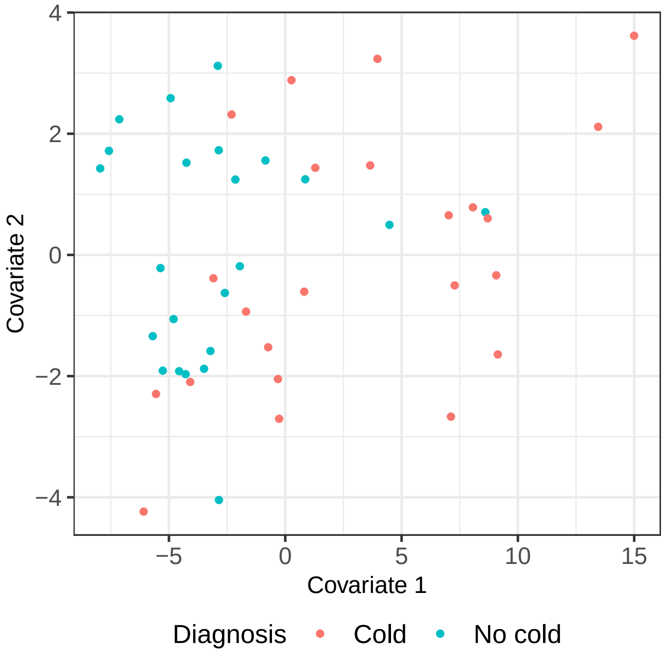


Figure 4.5. Scatterplot of the two covariates found by SMPCovR. The colours represent the cold diagnosis.

4.5 Discussion

Predictive modelling in the presence of large numbers of predictor and outcome variables presents multiple challenges. Constructed models feature a huge number of estimated coefficients, rendering the interpretation infeasible. Moreover, there may be subsets of both predictor and outcome variables that are not important. Certain predictor variables may be redundant in predicting any of the outcome variables, while some outcome variables may not at all be adequately predicted by the available predictors.

In this paper, we proposed the method of SMPCovR that accommodates for these issues by relying on PCovR methodology and incorporating sparsity penalties at both sides of predictors and outcomes. Comparative assessment of the method against SPCovR and sPLS resulted in SMPCovR showing outperformance in outcome prediction, achieved by correct exclusion of inactive outcome variables. Our method also performed comparatively well at retrieving the coefficients that represent how processes underneath data underlie the predictor and outcome variables.

The PCovR methodology provides an advantageous position in the settings with large numbers of predictor and outcome variables. The predictors and outcomes are linked with the reduced dimensions of the covariates, instead of being directly connected with each other. This reduces the number of estimated coefficients by far. In total, $(J+L) \times R$ coefficients need to be found by SMPCovR, while $J \times L$ coefficients need estimation in a regularized regression setup with predictors and outcomes directly connected. Using the example of the PCS dataset in section 4.4, SMPCovR model would comprise of $(187 + 16) \times 2 = 406$ coefficients at maximum, while a regression model can consist of $187 \times 16 = 2992$ coefficients. By imposing further sparsity penalties on the coefficients, SMPCovR can derive an even more sparse and concise model representation. Furthermore, the reduction of the number of coefficients also implies that less number of coefficients need to be forced to zero to exclude a variable (both predictor and outcome) altogether from the model. As a consequence, SMPCovR is a prediction method with multivariate outcomes that conducts variable selection in an effective manner. These strengths also apply generally to other regression methods based on dimension reduction such as PLS.

There are limitations to our proposed method. Being characterized with 6 different tuning parameters, model selection is a natural complication. To reduce the compuational burden of CV, we fixed the ridge parameters to a small value and employed a sequential model selection approach where the number of covariates is first chosen prior to tuning for the sparsity parameters. The ridge parameters were kept small because they play a role of preventing divergence; they do not exert a big influence in shaping the final model. The sequential approach has been shown suitable for PCovR and SPCovR (S. Park et al., 2020; Vervloet et al., 2016). This model selection strategy resulted in good results in both the simulation and empirical studies. However, we did not conduct an extensive investigation focused on the model selection approaches due to the scope of our paper.

In a similar vein, the optimality criterion we employed for the CV in the context of outcome variable selection could be a subject of further research. In our study, we used the R^2_{cv} measure that only employs the active outcome variables included in the model fitted from the CV training set. This approach was effective in finding the correct subset of active outcome variables[5]. To the best of our knowledge, the choice on the optimality criterion for a prediction method that performs outcome variable selection has not yet been addressed in the literature.

Our proposed method is one of the first regression methods that conducts variable selection in both predictor and outcome variables. With growing availability of large datasets and increasing use of data collected without specific research aims, we believe such methods are becoming more relevant. The literature also seems to be steering towards this direction, with Hu, Liu, Liu, and Xia (2022) hinting at an adaptation to the objective criterion to allow predictor variable selection on top of the outcome variable selection offered in Hu, Huang, et al. (2022). We expect that PCovR and other multivariate methods that leverage from dimension reduction to bear great potential in taking the lead in this under-studied research problem.

[5] In our experiments, we found that the R^2_{cv} also works well even when all of the outcome variables were defined as being active; the final model chosen included all of the outcomes.

Appendix

4.A SMPCovR algorithm

The SMPCovR loss (4.3) can be minimized by an alternating least squares procedure. A schematic outline of the algorithm is provided in what follows. It is similar to the procedures proposed to solve SCaDS (de Schipper & Van Deun, 2018), SPCovR (Van Deun et al., 2018) and SSCovR (S. Park et al., 2020). The algorithm involves solving for all covariates together (unlike the deflation approach in which one covariate is solved in turn). The routine continues until the algorithm converges into a stationary point, usually a local minium. To avoid local minima problems, we recommend to use multiple random and a rational starting value based on PCovR.

Algorithm 4.1 SMPCovR

```
Inputs:
    $\mathbf{X}$ and $\mathbf{Y}$, number of covariates $R$, weighting parameter $\alpha$, regularization
    parameters for $\mathbf{W}$ $\lambda_{Lr}$ and $\lambda_{Rr}$, regularization parameters for $\mathbf{P}^{(Y)}$ $\gamma_{Lr}$, and $\gamma_{Rr}$,
    maximum number of iterations $T$, convergence threshold $\epsilon \geq 0$

Initialize:
    $\mathbf{W} \leftarrow \mathbf{W}^{(0)}$ $L_0 \leftarrow$ Initial loss,
    Loss difference $d \leftarrow 1$, Iteration counter $t \leftarrow 1$

while $t < T$ or $\epsilon < d$ do
    Conditional estimation of ${\mathbf{P}^{(X)}}^{(t)}$, ${\mathbf{P}^{(Y)}}^{(t)}$ given $\mathbf{W}^{(t)}$
    Conditional estimation of $\mathbf{W}^{(t+1)}$ given ${\mathbf{P}^{(X)}}^{(t+1)}$ and ${\mathbf{P}^{(Y)}}^{(t+1)}$
    $L_u \leftarrow$ updated loss given $\mathbf{W}^{(t+1)}$, ${\mathbf{P}^{(X)}}^{(t+1)}$ and ${\mathbf{P}^{(Y)}}^{(t+1)}$
    $d \leftarrow L_0 - L_u$
    $t \leftarrow t + 1$
    $L_0 \leftarrow L_u$
end while
```

4.B Estimation of W

Conditional estimation of $\mathbf{W}$ given the other parameters $\mathbf{P}^{(X)}, \mathbf{P}^{(Y)}$ pertains to an elastic net regression problem. The SMPCovR objective function (4.3) is first arranged with respect to the the hth element of the weights corresponding to the covariate component r^*: w_{hr^*}.

$$
\begin{aligned}
L\left(w_{hr^*}\right) = {} & \frac{\alpha}{\|\mathbf{Y}\|_2^2} \sum_{i}^{N} \left\| \mathbf{y}_i - \sum_{r}^{R} \sum_{j \neq h}^{J} x_{ij} w_{jr} \mathbf{p}_r^{(Y)} - \sum_{r \neq r^*}^{R} x_{ih} w_{hr} \mathbf{p}_r^{(Y)} - x_{ih} w_{hr^*} \mathbf{p}_{r*}^{(Y)} \right\|_2^2 \\
& + \frac{1-\alpha}{\|\mathbf{X}\|_2^2} \sum_{i}^{N} \left\| \mathbf{x}_i - \sum_{r}^{R} \sum_{j \neq h}^{J} x_{ij} w_{jr} \mathbf{p}_r^{(X)} - \sum_{r \neq r^*}^{R} x_{ih} w_{hr} \mathbf{p}_r^{(X)} - x_{ih} w_{hr^*} \mathbf{p}_{r^*}^{(X)} \right\|_2^2 \\
& + \lambda_{Lr^*} \left| w_{hr^*} \right| + \lambda_{Rr^*}\, w_{hr^*}^2 + \gamma_{Lr^*} \left| \mathbf{p}_{r^*}^{(Y)} \right|_1 + \gamma_{Rr^*} \left\| \mathbf{p}_{r^*}^{(Y)} \right\|_2^2
\end{aligned}
\tag{4.7}
$$

Taking the derivative with respect to w_{hr^*} we get:

$$
\begin{aligned}
& \frac{-2\alpha}{\|\mathbf{Y}\|_2^2} \sum_{i}^{N} \mathbf{p}_{r^*}^{(Y)\top} \left(\mathbf{r}_{ih} - x_{ih} w_{hr^*} \mathbf{p}_{r^*}^{(Y)} \right) x_{ih} - \frac{2(1-\alpha)}{\|\mathbf{X}\|_2^2} \sum_{i}^{N} \left(s_{ih} - x_{ih} w_{hr^*} \right) x_{ih} \\
& + \lambda_{Lr^*} \partial \left| w_{hr^*} \right| + 2\lambda_{Rr^*}\, w_{hr^*}
\end{aligned}
\tag{4.8}
$$

where

$$
\begin{aligned}
\mathbf{r}_{ih} &= \mathbf{y}_i \sum_{r}^{R} \sum_{j \neq h}^{J} x_{ij} w_{jr} \mathbf{p}_r^{(Y)} - \sum_{r \neq r^*}^{R} x_{ih} w_{hr} \mathbf{p}_r^{(Y)} \\
s_{ih} &= \mathbf{p}_{r^*}^{(X)\top} \mathbf{x}_i - \sum_{j \neq h}^{J} x_{ij} w_{jr^*}
\end{aligned}
\tag{4.9}
$$

We can equate the derivative to zero to satisfy the optimality conditions for $\hat{w}_{hr^*}$, which can be summarized by the following:

$$
\hat{w}_{hr^*} = \frac{S\left(\sum_i^N \left[\frac{2\alpha}{\|\mathbf{Y}\|_2^2} \left(\mathbf{p}_{r^*}^{(Y)\top} \mathbf{r}_{ih} + \frac{2(1-\alpha)}{\|\mathbf{X}_C\|_2^2} s_{ih} \right) x_{ih} \right], \lambda_{Lr^*} \right)}{\sum_i^N \left(\frac{2\alpha}{\|\mathbf{Y}\|_2^2} \left\| \mathbf{p}_{r^*}^{(Y)} \right\|_2^2 + \frac{2(1-\alpha)}{\|\mathbf{X}_C\|_2^2} \right) {x_{ih}}^2 + 2\lambda_{Rr^*}}
\tag{4.10}
$$

where S(.) is a element-wise soft-thresholding operator. With these conditions, we can set up the following coordinate descent algorithm.

Algorithm 4.2 Coordinate descent for the weights

```
for $r^*$ in $1 : R$ do
    for $h$ in $1 : J$ do
        $\hat{w}_{hr^*} \leftarrow \frac{S\left( \sum_i^N \left[ \frac{2\alpha}{\|\mathbf{Y}\|_2^2} \left( \mathbf{p}_{r^*}^{(Y)\top} \mathbf{r}_{ih} + \frac{2(1-\alpha)}{\|\mathbf{X}_C\|_2^2} s_{ih} \right) x_{ih} \right], \lambda_{Lr^*} \right)}{\sum_i^N \left( \frac{2\alpha}{\|\mathbf{Y}\|_2^2} \left\| \mathbf{p}_{r^*}^{(Y)} \right\|_2^2 + \frac{2(1-\alpha)}{\|\mathbf{X}_C\|_2^2} \right) {x_{ih}}^2 + 2\lambda_{Rr^*}}$
```

4.C Estimation of $\mathbf{P}^{(Y)}$

Conditional estimation of $\mathbf{P}^{(Y)}$ given the other parameters $\mathbf{W}_C, \mathbf{P}_C^{(X)}$ is an elastic net regression problem. The SMPCovR objective function (4.3) is first arranged with respect to the regression coefficients corresponding to hth outcome variable and r^*th covariate:

$$\begin{aligned} L\left(p_{hr^*}^{(Y)}\right) = \frac{\alpha}{\|\mathbf{Y}\|_2^2} \sum_i^N \left(y_{ih} - \sum_{r \neq r^*}^R \mathbf{x}_i{}^\top \mathbf{w}_r {p_{hr}}^{(Y)} - \mathbf{x}_i{}^\top \mathbf{w}_{r^*} p_{hr^*}^{(Y)} \right)^2 \\ + \gamma_{Lr^*} \left| p_{hr^*}^{(Y)} \right| + \gamma_{Rr^*} \, p_{hr^*}^{(Y)2} \end{aligned} \tag{4.11}$$

Taking the derivative with respect to ${p_{hr^*}}^{(Y)}$:

$$\frac{-2\alpha}{\|\mathbf{Y}\|_2^2} \sum_i^N \mathbf{x}_i{}^\top \mathbf{w}_{r^*} \left(t_{ih} - \mathbf{x}_i{}^\top \mathbf{w}_{r^*} p_{hr^*}^{(Y)} \right) + \gamma_{Lr^*} \partial \left| {p_{hr^*}}^{(Y)} \right| + 2\gamma_{Rr^*} \, p_{hr^*}^{(Y)} \tag{4.12}$$

where

$$t_{ih} = y_{ih} - \sum_{r \neq r^*}^R \mathbf{x}_i{}^\top \mathbf{w}_{r^*} p_{hr}^{(Y)} \tag{4.13}$$

We can equate the derivative to zero to satisfy the optimality conditions for $\hat{p}_{hr^*}^{(Y)}$, which can be summarized by the following:

$$\hat{p}_{hr^*}^{(Y)} = \frac{S\left(\sum_i^N \left(\mathbf{x}_i{}^\top \mathbf{w}_{r^*} \right) t_{ih}^{(r^*)}, \frac{\|\mathbf{Y}\|_2^2 \gamma_{Lr^*}}{2\alpha} \right)}{\sum_i^N \left(\mathbf{x}_i{}^\top \mathbf{w}_{r^*} \right)^2 + \left(\|\mathbf{Y}\|_2^2 / \alpha \right) \gamma_{Rr^*}} \tag{4.14}$$

With these conditions, we can set up the following coordinate descent algorithm.

Algorithm 4.3 Coordinate descent for the regression coefficients $\mathbf{P}^{(Y)}$

```
for $r^*$ in $1 : R$ do
    for $h$ in $1 : L$ do
        $\hat{p}_{hr^*}^{(Y)} \leftarrow \frac{S\left( \sum_i^N \left( \mathbf{x}_i{}^\top \mathbf{w}_{r^*} \right) t_{ih}^{(r^*)}, \frac{\|\mathbf{Y}\|_2^2 \gamma_{Lr^*}}{2\alpha} \right)}{\sum_i^N \left( \mathbf{x}_i{}^\top \mathbf{w}_{r^*} \right)^2 + \left( \|\mathbf{Y}\|_2^2 / \alpha \right) \gamma_{Rr^*}}$
```

4.D Estimation of $\mathrm{P}^{(X)}$

The loadings $\mathbf{P}^{(X)}$ such that $\mathbf{P}^{(X)\top}\mathbf{P}^{(X)} = \mathbf{I}_R$ are obtained via a closed-form solution; $\mathbf{P}^{(X)} = \mathbf{U}\mathbf{V}^\top$ where $\mathbf{U}$ and $\mathbf{V}$ are found through singular value decomposition of $\mathbf{X}^\top\mathbf{X}\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$.

4.E Regression coefficients $\mathrm{P}^{(Y)}$ defined for the case with 20 outcome variables

$$\begin{array}{c}
\begin{matrix} 1 & 2 & 3 \end{matrix} \\
\begin{pmatrix}
1 & 0 & 0 \\
1 & 0 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 1 & 0 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
0 & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}
\end{array}$$

4.F The scree test with acceleration factor conducted to determine the number of covariates for the toy example dataset

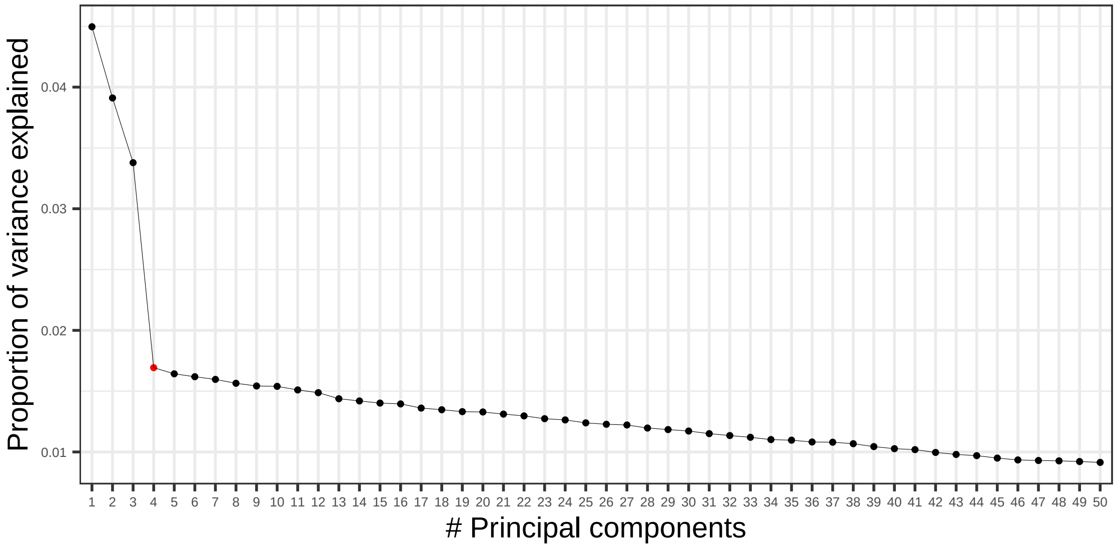


Figure 4.6. For readability, the plot only displays the proportion of variance for the first 50 components. It can be seen that the sharpest change of slopes occurs at the fourth principal component. Therefore, the number of SMPCovR covariate is determined as three.

4.G Simulation study: R^2_{out} measure computed only on the basis of active outcomes

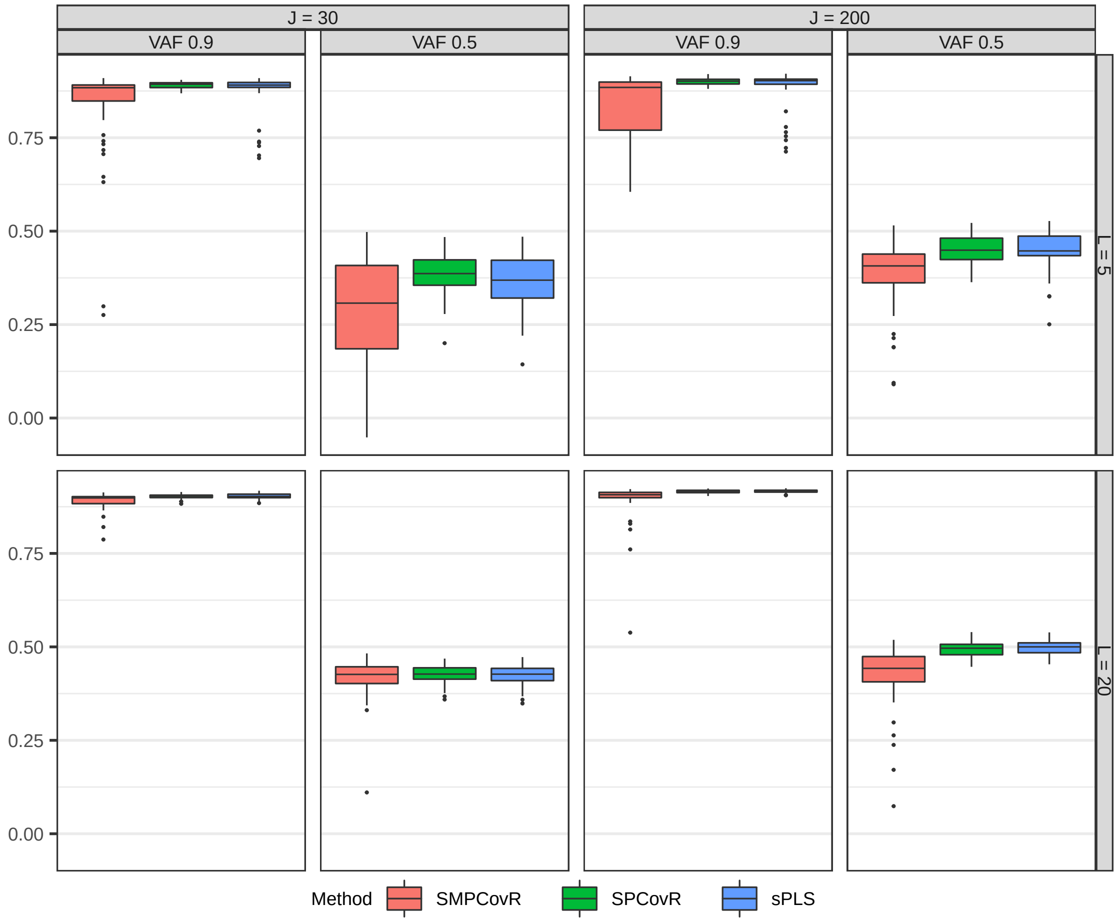


Figure 4.7. R^2_{out} only on the basis of active outcomes. Each panel corresponds to one of the 8 conditions.

4.H Simulation study: proportion of outcomes correctly identified as active and inactive by SMPCovR

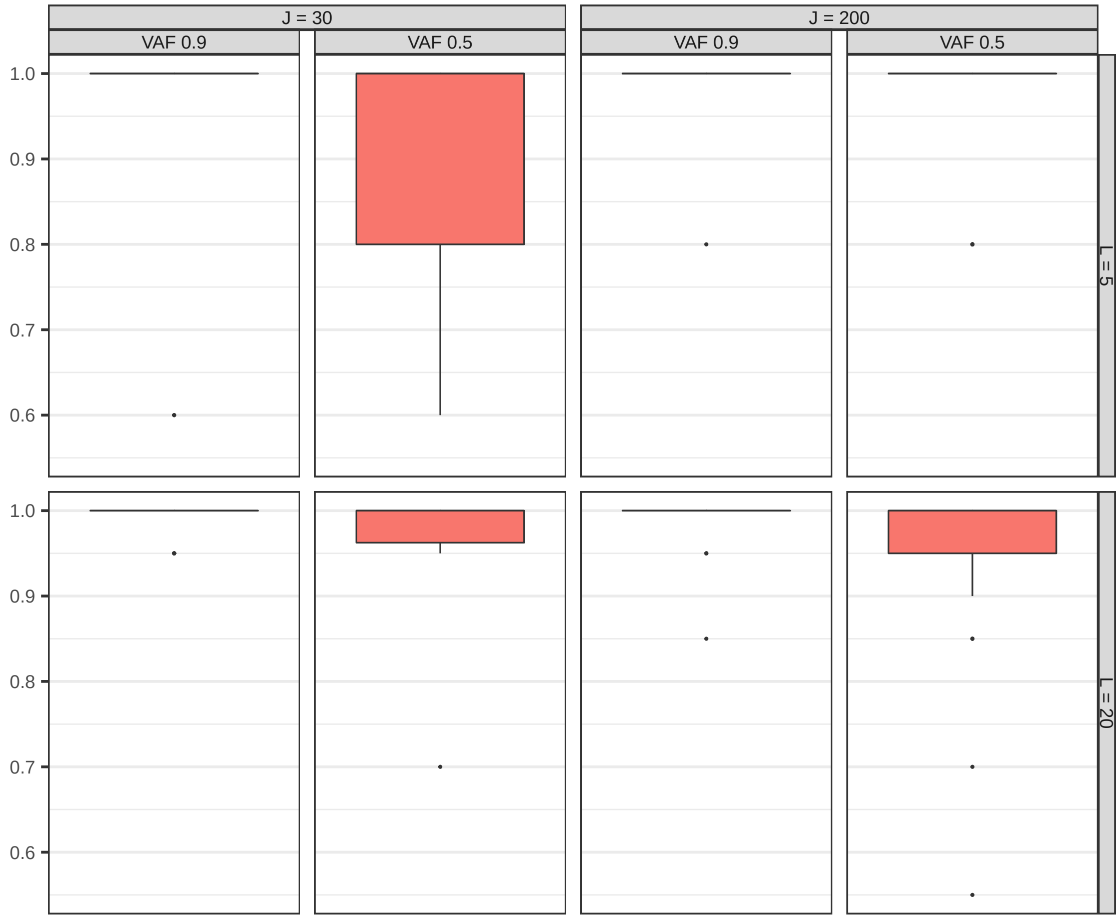


Figure 4.8. Proportion of outcomes correctly identified as active and inactive by SMPCovR. Each panel corresponds to one of the 8 conditions.

4.I Simulation study: discrepancy from zero regression coefficients from SPCovR and sPLS

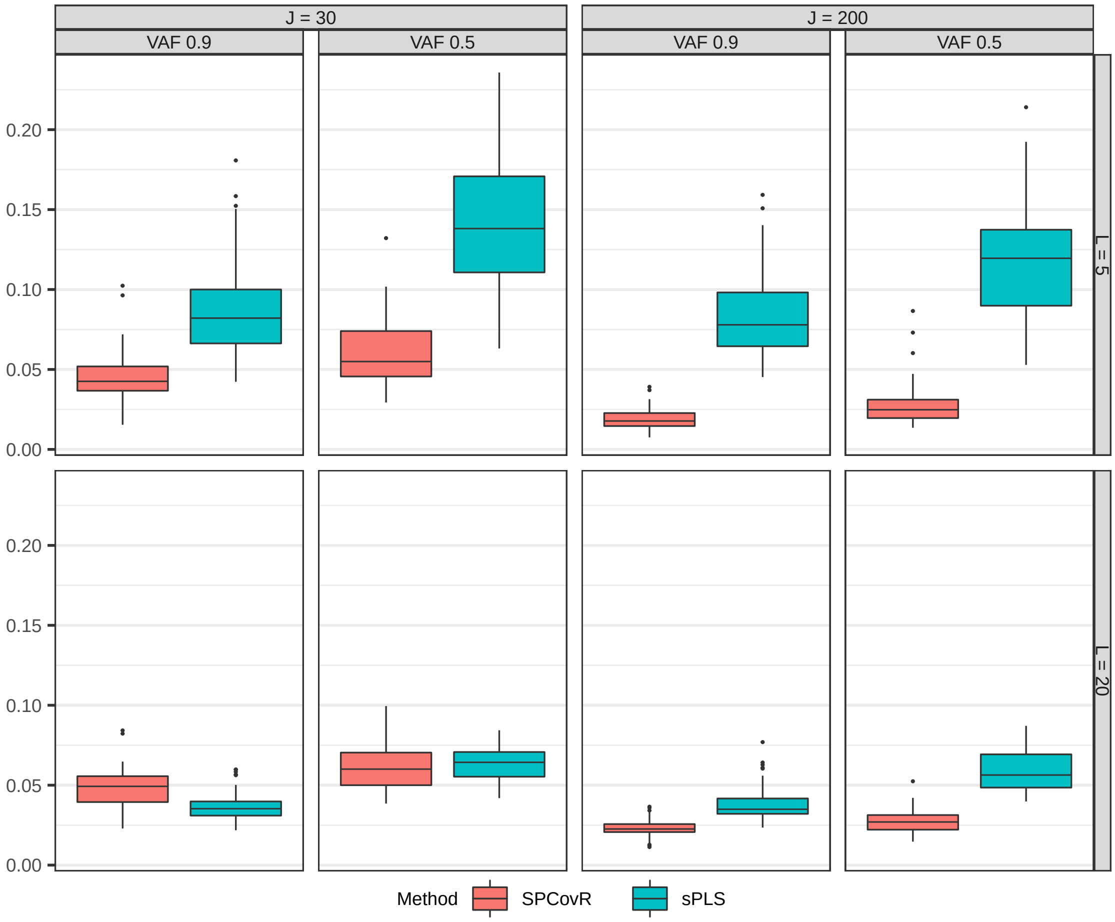


Figure 4.9. Mean absolute discrepancy of the zero regression coefficients. Each panel corresponds to one of the 8 conditions.

4.J The scree test with acceleration factor conducted to determine the number of covariates for the PCS dataset

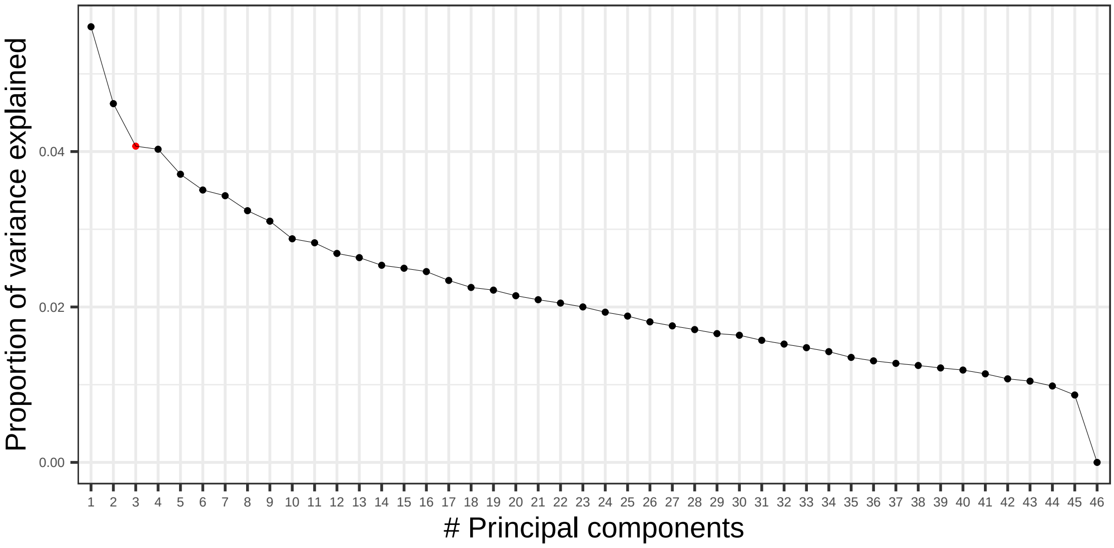


Figure 4.10. It can be seen that the sharpest change of slopes occurs at the third principal component. Therefore, the number of SMPCovR covariate is determined as two.

4.K The R^2 measures computed on the PCS dataset

$R^2_{\text{fit}_{all}}$, $R^2_{\text{fit}_{sub}}$, $R^2_{\text{loocv}_{all}}$ and $R^2_{\text{fit}_{sub}}$ employed to evaluate the models fitted on the PCS dataset were calculated by the following equations.

$R^2_{\text{fit}_{all}}$ is the R^2 measure computed on the in-sample data on the basis of all of the outcome variables. This can be considered as the conventional R^2 measure:

$$R^2_{\text{fit}_{all}} = 1 - \frac{\left\|\mathbf{Y} - \mathbf{X}\hat{\mathbf{W}}\hat{\mathbf{P}}^{(Y)\top}\right\|_2^2}{\|\mathbf{Y}\|_2^2} \tag{4.15}$$

The $R^2_{\text{fit}_{sub}}$ measure is computed on the in-sample data, however on the basis of outcome variables selected as being active by the SMPCovR model:

$$R^2_{\text{fit}_{sub}} = 1 - \frac{\left\|\mathbf{Y}_{L^*} - \mathbf{X}\hat{\mathbf{W}}\hat{\mathbf{P}}^{(Y)\top}_{L^*}\right\|_2^2}{\|\mathbf{Y}_{L^*}\|_2^2} \tag{4.16}$$

with the subscript $_{L^*}$ indicating a subset within the sequence of indices for outcome variables $L^* \subseteq \{1, 2, \ldots, L\}$. It comprises of indices corresponding to the active outcomes selected by SMPCovR. As reported in Table 4.4, $\mathbf{Y}_{L^*}$ would comprise of the 10 following outcomes: sneezing, runny nose, nasal congestion, cough, sore throat, malaise, chest congestion, sinus pain, fever and poor appetite. Since an outcome variable is removed from the SMPCovR model if its corresponding row in the estimated regression coefficients matrix $\hat{\mathbf{P}}^{(Y)}$ is a zero-vector, the indices of non-zero rows of $\hat{\mathbf{P}}^{(Y)}$ make up the set L^*. $\hat{\mathbf{P}}^{(Y)}_{L^*}$ denotes the submatrix of $\hat{\mathbf{P}}^{(Y)}$ with non-zero rows.

$R^2_{\text{loocv}_{all}}$ is calculated via leave-one-out CV. All of the outcome variables in the PCS dataset are incorporated:

$$R^2_{\text{loocv}_{all}} = 1 - \frac{\left\| \mathbf{y}^{\text{test}} - \mathbf{x}^{\text{test}\top} \hat{\mathbf{W}} \hat{\mathbf{P}}^{(Y)\top} \right\|_2^2}{\|\mathbf{y}^{\text{test}}\|_2^2} \tag{4.17}$$

where $\mathbf{y}^{test}$ and $\mathbf{x}^{test}$ refer to the outcome and predictor variables in the CV test set (it is a vector, since leave-one-out CV uses one observation unit for each test set).

Lastly, $R^2_{\text{loocv}_{sub}}$ is also calculated via leave-one-out CV, but on the basis of active outcome variables selected by the SMPCovR model:

$$R^2_{\text{loocv}_{sub}} = 1 - \frac{\left\| \mathbf{y}^{\text{test}}_{L^*} - \mathbf{x}^{\text{test}\top} \hat{\mathbf{W}} \hat{\mathbf{P}}^{(Y)\top}_{L^*} \right\|_2^2}{\|\mathbf{y}^{\text{test}}_{L^*}\|_2^2} \tag{4.18}$$

As for the formula for $R^2_{\text{loocv}_{all}}$, $\mathbf{y}^{test}$ and $\mathbf{x}^{test}$ refer to the outcome and predictor variables in the CV test set. As for the formula for $R^2_{\text{fit}_{sub}}$, the subscript $_{L^*}$ denotes a subset within the sequence of indices for outcome variables $L^* \subseteq \{1, 2, \ldots, L\}$. It comprises of indices corresponding to the active outcomes selected by SMPCovR. As reported in Table 4.4, $\mathbf{Y}_{L^*}$ would comprise of the 10 following outcomes: sneezing, runny nose, nasal congestion, cough, sore throat, malaise, chest congestion, sinus pain, fever and poor appetite.

4.L Model selection for SPCovR and sPLS for the PCS dataset

For SPCovR, the α parameter and the lasso parameter for the weights were tuned by 5-fold CV. We adopted the sequence [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] for α. For the lasso parameter, 0 and equally distanced sequence of size 49 from 10^{-5} to 0.5 on the natural log scale was employed as the range. Crossing the two ranges, $9 \times 50 = 450$ different models were evaluated by CV. With regards to sPLS, the range considered for the number of non-zero coefficients per component was the multiples of 6 running from 6 to 180 along with 1 and 187 (minimal and maximal number of non-zero coefficients). With the number of components fixed at two, the 5-fold CV was performed for $32^2 = 1024$ total models. After the CV, the 1SE rule was used to select the final model for both methods.

Table 4.6. Weights derived by SPCovR from the PCS dataset. $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_1$ (cont.) indicate the weights corresponding to the first covariate, while $\hat{\mathbf{w}}_2$ refers to the weights concerning the second covariate.

$\hat{\mathbf{w}}_1$

	1
IL-6	0.100
TNF alpha	1.339
Red blood cells	0.025
Absolute neutrophil count	-0.053
Corpuscular Hgb	-0.118
Corpuscular Hgb conc	-0.046
Potassium	0.200
Calcium	0.006
Alkaline phosphatase	0.014
Non-fasting glucose	0.161
Urea nitrogen	0.067
# weekdays alcohol	0.324
# weekend days alcohol	0.122
# drinks on weekdays	0.006
PSQI: too hot	-0.117
FES: Expressiveness	0.006
Parental social participation	-0.104
ReCAPS 15	0.007
Neighbourhood physical	0.209
Neighbourhood social	-0.001
Perceived SES Mom	-0.087
PANAS:guilt	-0.178
PANAS: sadness	0.316
PANAS: fatigue	0.296
IPIP: extraversion	0.171
Communal orientation	0.105
TSC: network size	-0.133
TSC: indirect social control	0.182
GS-ISEL	0.194
Negative aspects of relationships	-0.055
Social participation	-0.233
Perceived community score	0.052
Loneliness	0.072
Perceived stress scale	0.117

$\hat{\mathbf{w}}_1$ (cont.)

	1
Leisure activities at home	-0.190
# days working	0.000
Time spent in bed awake	-0.084
# days exercise	0.047
# drinks total	0.004
# days alcohol	0.021
Fatigue	0.526
Lively	-0.300
Anger subscale	0.170

$\hat{\mathbf{w}}_2$

	2
Psych well-being	-1.889

Table 4.7. Regression coefficients derived by SPCovR from the PCS dataset. Each column corresponds to the coefficients for each covariate.

$\hat{\mathbf{P}}^{(Y)}$

	1	2
Sneezing	0.335	-0.044
Runny nose	0.305	-0.041
Nasal congestion	0.327	-0.033
Cough	0.370	0.029
Sore throat	0.289	-0.103
Headache	0.187	-0.044
Chills	0.069	0.082
Malaise	0.232	-0.004
Chest congestion	0.265	0.049
Sinus pain	0.321	-0.053
Earache	0.049	0.003
Muscle ache	0.173	-0.074
Joint ache	0.027	-0.086
Sweating	0.119	0.128
Fever	0.267	0.001
Poor appetite	0.285	0.020

Table 4.6 and 4.7 present the weights and regression coefficients from the SPCovR model. The model is comprised with 43 and 1 non-zero weights for each covariate. While only 2 variables were selected from those that concern nasal cytokine, SPCovR retrieved many more non-zero weights than SMPCovR for the various within other themes: blood chemistry, health practices, psychosocial assessment scales and daily interviews. Moreover, whereas SMPCovR excluded all of the predictors regarding childhood experiences, SPCovR included them (such as Family Environment Scale: Expressiveness). Only 1 predictor was found corresponding to the second covariate. This is also in line with the model derived by SMPCovR that only found 3 predictors. Lastly, similar to SMPCovR that forced all of the regression weights concerning the second covariate to zero, SPCovR also found near-zero values for these coefficients.

Table 4.8. Weights derived by sPLS from the PCS dataset. $\hat{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_2$ indicate the weights corresponding to the first and second covariates respectively.

$\hat{\mathbf{w}}_1$

	1
TNF alpha	1

$\hat{\mathbf{w}}_2$

	2
Daily loneliness	0.033
Daily negative affect & fatigue	0.411
Daily negative affect	0.019
Daily fatigue subscale	0.608
Daily tiredness	0.436
Daily fatigue	0.518

Table 4.9. Regression coefficients derived by sPLS from the PCS dataset. Each column corresponds to the coefficients for each covariate.

$\hat{\mathbf{P}}^{(Y)}$

	1	2
Sneezing	-0.200	-0.259
Runny nose	-0.245	-0.247
Nasal congestion	-0.240	-0.285
Cough	-0.341	-0.398
Sore throat	-0.316	-0.264
Headache	-0.088	-0.224
Chills	-0.047	-0.222
Malaise	-0.090	-0.308
Chest congestion	-0.328	-0.259
Sinus pain	-0.404	-0.237
Earache	-0.022	-0.394
Muscle ache	-0.198	-0.104
Joint ache	0.126	-0.086
Sweating	-0.125	0.062
Fever	-0.354	0.059
Poor appetite	-0.376	-0.253

The weights and regression coefficients found by sPLS are provided in Table 4.8 and 4.9. The model constructed by sPLS is largely different from the SPCovR and SMPCovR models that are similar among each other. It was also comprised of much smaller number of non-zero weights. The first covariate only consisted of TNF alpha, one of the 7 variables regarding nasal cytokine. The second covariate was associated with 6 variables from daily interviews. With respect to the regression coefficients, most of them were far away from zero. This was also in line with our finding in the simulation study, where sPLS did not provide near-zero coefficients as estimates for the true zero regression coefficients (see Appendix 4.I).

Part II

Novel results in sparse Principal Component Analysis relevant for extending Principal Covariates Regression

Chapter 5

A critical assessment of sparse PCA (research): Why (one should acknowledge that) weights are not loadings

Principal component analysis (PCA) is an important tool for analyzing large collections of variables. It functions both as a pre-processing tool to summarize many variables into components and as a method to reveal structure in data. Different coefficients play a central role in these two uses. One focuses on the *weights* when the goal is summarization, while one inspects the *loadings* if the goal is to reveal structure. It is well known that the solutions to the two approaches can be found by singular value decomposition; weights, loadings, and right singular vectors are mathematically equivalent. What is often overlooked, is that they are no longer equivalent in the setting of sparse PCA methods which induce zeros either in the weights or the loadings. The lack of awareness for this difference has led to questionable research practices in sparse PCA. First, in simulation studies data is generated mostly based only on structures with sparse singular vectors or sparse loadings, neglecting the structure with sparse weights. Second, reported results represent local optima as the iterative routines are often initiated with the right singular vectors. In this paper we critically re-assess sparse PCA methods by also including data generating schemes characterized by sparse weights and different initialization strategies. The results show that relying on commonly used data generating models can lead to over-optimistic conclusions. They also highlight the impact of choice between sparse weights versus sparse loadings methods and the initialization strategies. The practical consequences of this choice are illustrated with empirical datasets.

Keywords: Sparse principal component analysis, Exploratory data analysis, Dimension reduction, Sparse weights, Sparse loadings

Park, S., Ceulemans, E., & Van Deun, K. (2023). A critical assessment of sparse PCA (research): why (one should acknowledge that) weights are not loadings. *Behavior Research Methods*, 1-20.

5.1 Introduction

“Principal component analysis (PCA) is probably the most popular multivariate statistical technique and it is used by almost all scientific disciplines. It is also likely to be the oldest multivariate technique." (Abdi & Williams, 2010, p.433). Often referred to as the basis for multivariate data analysis (S. Wold, Esbensen, & Geladi, 1987), the central idea of PCA is to reduce a possibly large set of variables to a few derived variables - usually called components - which preserve a maximum amount of information in the data (Jolliffe, 2002). The resulting low-dimensional representations are mainly used in two ways: they are either used as a data *pre-processing step* where the constructed summary scores are subsequently adopted for regression or classification or as an *exploratory tool* to detect patterns and to create attractive visualizations of the data (Gabriel, 1971). PCA produces two types of coefficients that serve these aims: variable ‘weights’ that define the transformation from the raw data to the summary scores and ‘loadings’ which reflect the strength of association of the raw variables with the low-dimensional representations. Although PCA has been presented in several ways, it is well known that the different PCA solutions are equivalent and that weights and loadings can both be obtained from the singular value decomposition (SVD) of the data matrix.

With the advent of big data, especially those in which the number of variables largely exceeds the number of observation units, the use of PCA to reduce the dimensionality of the data has become more widespread. However, there are several issues with using PCA in the high-dimensional setting. First, computation of weights and loadings may suffer from a problem of statistical inconsistency in high-dimensional data settings (Johnstone & Lu, 2009; D. Shen, Shen, & Marron, 2016). Furthermore, interpreting the summarized scores via inspection of weights and loadings becomes difficult as PCA computes these coefficients for the entire set of variables. Traditionally, the burden of interpretation that arises from studying all of the coefficients had been addressed by rotation to simple structure (Jolliffe, 2002). Yet, rotation followed by neglecting coefficients with small magnitude has been pointed out to be a rather arbitrary and suboptimal way of selecting variables (Cadima & Jolliffe, 1995; Trendafilov & Adachi, 2015).

In response to these issues, sparse versions of PCA that reduce the number of variables involved in the PCA representation have been proposed (Jolliffe, Trendafilov, & Uddin, 2003; H. Shen & Huang, 2008; Witten, Tibshirani, & Hastie, 2009; Zou et al., 2006). Incorporation of sparsity allows much easier interpretation of the components and restores statistical consistency of the coefficients. Several other benefits are gained through sparsity including that it addresses the need - in substantive research - of selecting those variables that are important for further investigation (Rasmussen & Bro, 2012) and economic aspects associated to the cost of measuring (many) variables (d'Aspremont, Ghaoui, Jordan, & Lanckriet, 2004).

Sparseness of PCA weights and loadings can be obtained in multiple ways. They can be constrained with respect to the number of estimated non-zero elements, or penalty terms such as the lasso can be added to them. Several such constrained or penalized PCA formulations have been proposed in line with different objectives for PCA. Yet, whereas the different PCA problems can all be solved via the SVD, this does not hold for the different sparse PCA methods. Importantly, weights, loadings and right singular vectors are no longer mathematically equivalent to each other in the sparse setting. However, the difference among these structures has been largely overlooked, leading to questionable practices in the sparse PCA literature. First, most simulation studies in the literature restrict themselves to data generating schemes with sparseness residing in the right singular vectors or the loadings, instead of also incorporating models with sparseness in the weights. Second, it is a common practice to adopt the right singular vectors as initial values while local optimization procedures are employed for methods that impose sparsity on weights or loadings. These practices seem to ignore the fact that these quantities represent different model structures.

Our current paper aims to create awareness for the fact that weights and loadings are truly different model structures with different roles. We conduct a simulation study and employ empirical datasets in doing so. In our simulation study, the focus is on comparing the performance of sparse PCA methods in terms of criteria that matter for data analysis and explicitly taking into account that this difference between weights and loadings also resides at the level of the data generating model. Such a comparison has been made elsewhere (Guerra-Urzola et al., 2021; Van Deun et al., 2011) but with a somewhat different focus. The contribution of this work is to shed light on the performance of sparse loadings versus sparse weights in different data generating contexts by employing sparse PCA methods that are based on the same model formulation, allow exact control over the level of sparsity and account for local optima. Moreover, the difference between sparse loadings and sparse weights methods are also discussed in a more practical manner using empirical data.

The paper is arranged as follows: in the next section we detail formulations of PCA and sparse PCA. We highlight that the equality between weights, loadings and right singular vectors that exists within PCA is lost as the methods transition into sparse PCA. This is clearly illustrated by a toy example. In a simulation study we compare the performance of the sparse weights versus sparse loading methods under different data generation schemes and for different algorithm initialization strategies. By employing two different empirical datasets, we present the practical impact of the choice of sparse PCA formulation and the initialization strategy. The paper concludes with a discussion. Code used to generate the results reported in the paper is available on Github: `https://github.com/soogs/Sparse-PCA-Critical-Assessment`.

5.2 Methods

After introducing the notation, we will first present the PCA decomposition and objective with special attention for the different roles played by component weights and loadings. We will highlight how the model structures (weights, loadings, singular vectors) are equal to one another. Then, we will discuss some commonly used sparse PCA methods that result from penalizing or constraining the PCA objective. It will be shown that the model structures are no longer subject to such equality within sparse PCA.

5.2.1 Notation

Throughout the paper, vectors and matrices will be denoted by bold lowercase and bold uppercase letters respectively. Lowercase subscripts that run from 1 to the corresponding uppercase letters will be used for indexing: $i \in (1, 2, \ldots, I)$. I, J and R denote the total numbers of observation units, variables and components, respectively. For example, we will use $\mathbf{X}$ to denote the data matrix, in which the J columns represent the variables and the I rows the observation units; note that the variables are assumed to be mean centered. Transposed vectors and matrices will be indicated by the superscript $^\top$, therefore $I^{-1}\mathbf{X}^\top\mathbf{X}$ is the covariance matrix. The Frobenius norm for matrices is denoted as $\|.\|_F$ and the squared Frobenius norm $\|\mathbf{X}\|_F^2 = \sum_{i,j} x_{i,j}^2$. Vector norms are defined as: $\|.\|_1$ for ℓ_1 norm ($\|\mathbf{x}\|_1 = \sum_i |x_i|$) and $\|.\|_2$ for ℓ_2 norm ($\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$). $\mathbf{Card}(.)$ indicates the cardinality of a matrix or a vector: this is the number of non-zero elements in the matrix or the vector. The addition of a subscript R to a matrix indicates the first R columns of the matrix.

5.2.2 Principal Component Analysis

PCA has been presented in several ways that are mathematically equivalent. (Guerra-Urzola et al., 2021; Jolliffe, 2002). The formulation incorporating both loadings and weights relies on the following decomposition of the data (Gabriel, 1978; Whittle, 1952; S. Wold et al., 1987):

$$\mathbf{X} = \mathbf{T}_R\mathbf{P}_R^\top + \mathbf{E}, \tag{5.1}$$
$$\text{subject to } \mathbf{P}_R^\top\mathbf{P}_R = \mathbf{I}_R \text{ and } \mathbf{t}_r^\top\mathbf{t}_{r'} = 0 \text{ for } r \neq r',$$

with $\mathbf{T}_R$ $(I \times R)$ denoting the principal component scores of the observation units and $\mathbf{P}_R$ $(J \times R)$ the *loadings* of the variables on the components. $\mathbf{E}$ is the matrix of residuals which is assumed to be orthogonal to $\mathbf{T}_R$. These parameters $\mathbf{T}_R$ and $\mathbf{P}_R$ are not unique; $\mathbf{T}_R\mathbf{A}$ and $\mathbf{P}_R\mathbf{A}^{-1}$ with an invertible matrix $\mathbf{A}$ also suffice the model equation. The principal component scores are often written out as linear combination of the variables ($\mathbf{T}_R = \mathbf{X}\mathbf{W}_R$) where $\mathbf{W}_R$ $(J \times R)$ matrix is referred to as *weights* which are understood analogously to regression weights in regression analysis[1]. This can be explicitly expressed in the model: $\mathbf{X} = \mathbf{X}\mathbf{W}_R\mathbf{P}_R^\top + \mathbf{E}$ with the same constraints as in (5.1). To obtain the PCA decomposition of the data, a least squares criterion is used:

$$\hat{\mathbf{T}}_R, \hat{\mathbf{P}}_R = \underset{\mathbf{T}_R, \mathbf{P}_R}{\operatorname{argmin}} \; ||\mathbf{X} - \mathbf{T}_R\mathbf{P}_R^\top||_F^2 \tag{5.2}$$
$$\text{subject to } \mathbf{P}_R^\top\mathbf{P}_R = \mathbf{I}_R \text{ and } \mathbf{t}_r^\top\mathbf{t}_{r'} = 0 \text{ for } r \neq r',$$

Since $\mathbf{T}_R = \mathbf{X}\mathbf{W}_R$, the solution for the weights $\hat{\mathbf{W}}_R$ falls directly from the solution for the component scores $\hat{\mathbf{T}}_R$ via least squares. In addition, the PCA objective (5.2) expresses the sum of squared errors between the observed data $\mathbf{X}$ and its reconstruction $\mathbf{T}_R\mathbf{P}_R^\top$. Hence, the proportion of variance accounted for (VAF) by the estimated PCA model is computed by: $1 - ||\mathbf{X} - \hat{\mathbf{T}}_R\hat{\mathbf{P}}_R^\top||_F^2 / ||\mathbf{X}||_F^2$. This VAF measure is commonly used as a measure of model fit for PCA or sparse PCA solutions, and adopted throughout the current paper.

Mathematical equivalence of weights, loadings, and singular vectors

As for the model (5.1), the problem in (5.2) is also not uniquely defined. Usually this issue is resolved by requiring a principal axis orientation of the principal components; they are found such that they successively explain maximum variance (Hotelling, 1933; Jolliffe, 2002). It is well known that the optimization problem in (5.2) can be solved via the singular value decomposition (SVD; see for example Jolliffe (2002)): Let $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$ with column orthogonal left singular vectors $\mathbf{U}$ and right singular vectors $\mathbf{V}$ ($\mathbf{U}^\top\mathbf{U} = \mathbf{I}_I$ and $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_J$) and $\mathbf{S}$ a $I \times J$ rectangular diagonal matrix with singular values in a decreasing order

[1] The component weights function as the regression weights in the regression of the rth component score vector on the J observed variables. Adachi and Trendafilov (2016) also refers to the $\mathbf{W}_R$ matrix as the weights matrix following the same rationale.

($s_{11} \geq s_{22} \geq \ldots \geq 0$), then the rank R approximation $\mathbf{X} \approx \mathbf{U}_R\mathbf{S}_R\mathbf{V}_R^\top$ is optimal in the least squares sense. Hence, adopting $\hat{\mathbf{T}}_R = \mathbf{U}_R\mathbf{S}_R$, $\hat{\mathbf{P}}_R = \mathbf{V}_R$ and $\hat{\mathbf{W}}_R = \mathbf{V}_R$ provides the solution to the least squares problem in (5.2) under the set constraints. The weights and loadings are both provided by the right singular vectors and therefore are numerically equal to each other as they have the same value: $p_{jr} = v_{jr} = w_{jr}$.

Conceptual difference between weights and loadings

Despite weights and loadings being numerically equivalent, the two structures have different conceptual roles in the decomposition of data. The component weights w_{jr} represent the weight that is given to a variable in the linear combination used to construct the component scores: $t_{ir} = \sum_j w_{jr}x_{ij}$. On the other hand, the loadings p_{jr} represent the strength of association of the components with the observed variable: $x_{ij} \approx \sum_r t_{ir}p_{jr}$; note that this strength of association is not influenced by the other components because of their orthogonality. Under proper normalization constraints[2], the loadings are equal to the correlation between the observed variable and the component scores. Although both weight and loading matrices are equal to each other in the numerical sense, understanding their conceptual difference is important as their mathematical equivalence is lost for PCA decompositions relying on other constraints and optimization criteria than the ones presented in (5.1) and (5.2).

5.2.3 Sparse Principal Component Analysis

Sparse forms of PCA can be obtained by imposing sparseness either on the weights or the loadings in the PCA decomposition (5.1). Several sparse PCA methods have been proposed that rely on this idea (e.g., Erichson et al., 2020; H. Shen & Huang, 2008; Witten et al., 2009; Zou et al., 2006). Here, we focus on two well-known sparse PCA methods that rely on the least-squares approach to the decomposition, that extract all components simultaneously, and that allow exact control over the number of zero loadings or weights. These methods are SPCA (Zou et al., 2006) for the setting with sparse weights and USLPCA (Adachi & Trendafilov, 2016) for the setting with sparse loadings. Focusing on these two sparse PCA methods has the benefit that any observed differences in performance can be attributed to the choice for either sparse weights or loadings, ruling out alternative explanations such as algorithmic differences or differences in the level of sparsity.

[2] This is scaling both the variables and the component scores to unit variance which is common practice in the social and behavioral sciences and implemented in the PCA procedure of the SPSS software (IBM Corp., 2020).

SPCA

Zou et al. (2006) proposed the SPCA criterion, where an elastic net penalty is placed on the weights from the PCA objective (5.2) in which weights are explicitly written out:

$$(\hat{\mathbf{W}}_R, \hat{\mathbf{P}}_R) = \underset{\mathbf{W}_R, \mathbf{P}_R}{\operatorname{argmin}} \ \left\| \mathbf{X} - \mathbf{X}\mathbf{W}_R\mathbf{P}_R^\top \right\|_F^2 + \lambda \sum_{r=1}^{R} \|\mathbf{w}_r\|_1 + \lambda_2 \sum_{r=1}^{R} \|\mathbf{w}_r\|_2^2 \qquad (5.3)$$
$$\text{subject to } \mathbf{P}_R^\top \mathbf{P}_R = \mathbf{I}_R,$$

with $\lambda \geq 0$ a tuning parameter for the lasso penalty; the effect of the penalty is that it shrinks the weights to zero, some/many of them even exactly so. In addition to the lasso, also a ridge penalty has been added. Its function is to obtain stable estimates in case of highly correlated predictors and to allow for more non-zero coefficients than I (in the setting with $J > I$).

The estimation of the weights and loadings is based on an alternating routine that updates the weights conditional upon the loadings and vice versa. The updating step of the sparse weights is based on the elastic net regression of the components on the variables. The SPCA procedure treats this problem with LARS-EN algorithm (Efron et al., 2004; Zou & Hastie, 2005) which allows the desired number of zero coefficients per component to be exactly specified in computing the weights[3]. It is important to note that elastic net regression problems are known to have difficulties in identifying the true sparse model in the high-dimensional setting (Jia & Yu, 2010).

USLPCA

Sparsity can also be imposed to the loadings matrix in (5.2). Adachi and Trendafilov (2016) proposed a sparse PCA method by imposing a cardinality constraint on the loadings, leading to the USLPCA criterion[4]:

$$(\hat{\mathbf{T}}_R, \hat{\mathbf{P}}_R) = \underset{\mathbf{T}_R, \mathbf{P}_R}{\operatorname{argmin}} \ \left\| \mathbf{X} - \mathbf{T}_R\mathbf{P}_R^\top \right\|_F^2 \qquad (5.4)$$
$$\text{subject to } \mathbf{T}_R^\top \mathbf{T}_R = \mathbf{I}_R \text{ and } \mathbf{Card}(\mathbf{p}_r) = k.$$

USLPCA is also based on an alternating optimization procedure between the

[3] The entire solution path for elastic net can be generated when solved by the LARS algorithm and thus when the desired number of non-zero predictors are included in the model, the iteration procedure can be stopped.

[4] A similar version with sparseness of the loadings obtained by adding a penalty has also been proposed (sPCA-rSVD; H. Shen & Huang, 2008). In this approach the orthogonality of the components is not imposed as the approach uses deflation to extract more than one component.

loadings and the component scores. The update of the loadings is a constrained univariate regression where each variable is regressed on each of the components. This implies that the estimates for the loadings do not suffer from stability issues in case of high correlations or high-dimensional data. Also, the estimation of the loadings easily allows the incorporation of a cardinality constraint in a computationally efficient way.

Local optimality

Unlike PCA formulations that have closed-form solutions based on SVD, iterative estimation procedures are adopted by the SPCA and USLPCA. As both methods are based on non-convex problems that are solved via an alternating procedure, the obtained solutions are prone to local optima. In their experiment of USLPCA, Adachi and Trendafilov (2016) reported that the method is sensitive to local optima characterized by solutions that are distant from the optimal solution. In order to aim for the global optimum, multiple random starting values should therefore be considered. Initializing these sparse PCA algorithms only with the right singular vectors can be problematic since it encourages the convergence to a local optimum near $\mathbf{V}_R$.

Loss of mathematical equivalence

Under both sparse PCA formulations (5.3, 5.4), the equality among weights, loadings and right singular vectors is lost since SVD is no longer adopted as a direct solution. While SPCA finds sparse weights and non-sparse loadings, USLPCA finds sparse loadings. Weights from USLPCA, although not explicitly estimated, are non-sparse (they can be inferred by regressing the component scores $\mathbf{T}_R$ on data $\mathbf{X}$). Hence, within each sparse PCA formulation, the weights and loadings are different to each other and they are no longer equal to the right singular vectors of $\mathbf{X}$. Across the two formulations, the weights and loadings estimated via the SPCA are different from the weights and loadings from the USLPCA.

5.2.3.1 Sparse PCA properties: toy example

In order to clearly illustrate the loss of equality among weights, loadings and right singular vectors for sparse PCA formulations, we make use of a toy example in this section. We created a 5×3 data matrix $\mathbf{X}$ of rank two so the data can be perfectly reconstructed with $R = 2$ components. A 2-component PCA model with sparse loadings underlies $\mathbf{X}$ and therefore, the component scores $\mathbf{T}$ are column-orthogonal ($\mathbf{T}^\top \mathbf{T} = \mathbf{I}$) and the loadings $\mathbf{P}$ are sparse as in (5.4):

$$\underset{\mathbf{X}}{\begin{bmatrix} 0.63 & 0.52 & 0.11 \\ -1.56 & -0.88 & 0.30 \\ 0.04 & 0.83 & 1.14 \\ 1.07 & 0.80 & 0.06 \\ -0.18 & -1.27 & -1.61 \end{bmatrix}} = \underset{\mathbf{T}}{\begin{bmatrix} 0.31 & 0.05 \\ -0.78 & 0.15 \\ 0.02 & 0.57 \\ 0.54 & 0.03 \\ -0.09 & -0.81 \end{bmatrix}} \times \underset{\mathbf{P}^\top}{\begin{bmatrix} 2 & 0 \\ 1.41 & 1.41 \\ 0 & 2 \end{bmatrix}^\top} \tag{5.5}$$

On this toy example dataset, we administered PCA, together with SPCA and USLPCA. Note that PCA weights and loadings are obtained by the right singular vectors. The two sparse PCA methods were applied such that one coefficient is returned sparse per component; this corresponds to the true sparse structure of the loading matrix in (6.10). The weights for USLPCA were calculated by regressing the estimated components on the variables. As aforementioned, these methods adopt the right singular vectors $\mathbf{V}_R$ of $\mathbf{X}$ by default as initial values for the iterative procedures. However, to account for the issue of local optima, a set of solutions stemming from 100 random initial values (with elements drawn from $\mathcal{U}(-1,1)$) were considered. The solution with the lowest value of the least squares loss was accepted as the final solution. We refer to the default approaches initialized by $\mathbf{V}_R$ by SPCA-svd and USLPCA-svd, while the multistart versions are denoted by SPCA-multi and USLPCA-multi. Table 5.1 presents the solutions provided by the PCA and sparse PCA. The first column provides the loss values of each solution. Since the methods are characterized by different objective criteria, these values are only comparable across different initialization strategies within the same sparse PCA method. In addition, the last column concerns the model fit: VAF by each of the components $(1-||\mathbf{X}-\hat{\mathbf{t}}_r\hat{\mathbf{p}}_r^\top||_F^2/||\mathbf{X}||_F^2)$ and the total VAF $(1-||\mathbf{X}-\hat{\mathbf{T}}_R\hat{\mathbf{P}}_R^\top||_F^2/||\mathbf{X}||_F^2)$. The VAF values for the sparse PCA methods are computed in the same manner by replacing the PCA estimates with sparse PCA estimates.

Table 5.1. Solutions for the PCA and sparse PCA methods. The VAF for each component and in total is indicated by $\text{vaf}_{1,2}$ and $\text{vaf}_{\text{total}}$, respectively.

Method	$\hat{\mathbf{W}}_R$	$\hat{\mathbf{P}}_R$	$\hat{\mathbf{T}}_R$	VAF
PCA (5.2) loss 0	$\begin{bmatrix} 0.50 & 0.71 \\ 0.71 & 0 \\ 0.50 & -0.71 \end{bmatrix}$	$\begin{bmatrix} 0.50 & 0.71 \\ 0.71 & 0 \\ 0.50 & -0.71 \end{bmatrix}$	$\begin{bmatrix} 0.73 & 0.37 \\ -1.25 & -1.32 \\ 1.18 & -0.78 \\ 1.14 & 0.71 \\ -1.80 & 1.01 \end{bmatrix}$	$\text{vaf}_1 = 2/3$ $\text{vaf}_2 = 1/3$ $\text{vaf}_{\text{total}} = 1$
SPCA-svd loss 5.66	$\begin{bmatrix} 0 & 0.71 \\ 1.41 & 0 \\ 0 & -0.71 \end{bmatrix}$	$\begin{bmatrix} 0.50 & 0.71 \\ 0.71 & 0 \\ 0.50 & -0.71 \end{bmatrix}$	$\begin{bmatrix} 0.73 & 0.37 \\ -1.25 & -1.32 \\ 1.18 & -0.78 \\ 1.14 & 0.71 \\ -1.80 & 1.01 \end{bmatrix}$	$\text{vaf}_1 = 2/3$ $\text{vaf}_2 = 1/3$ $\text{vaf}_{\text{total}} = 1$
SPCA-multi loss 5.11	$\begin{bmatrix} 0.83 & 0 \\ 0.56 & 0.02 \\ 0 & 1.14 \end{bmatrix}$	$\begin{bmatrix} 0.82 & -0.28 \\ 0.57 & 0.42 \\ -0.01 & 0.87 \end{bmatrix}$	$\begin{bmatrix} 0.81 & 0.13 \\ -1.78 & 0.33 \\ 0.50 & 1.32 \\ 1.34 & 0.09 \\ -0.86 & -1.87 \end{bmatrix}$	$\text{vaf}_1 = 0.552$ $\text{vaf}_2 = 0.448$ $\text{vaf}_{\text{total}} = 1$
USLPCA-svd loss 1.17	$\begin{bmatrix} 0.30 & 0.26 \\ 0.23 & -0.10 \\ 0.03 & -0.39 \end{bmatrix}$	$\begin{bmatrix} 1.85 & 0.77 \\ 1.85 & 0 \\ 0 & -1.85 \end{bmatrix}$	$\begin{bmatrix} 0.31 & 0.07 \\ -0.66 & -0.44 \\ 0.24 & -0.52 \\ 0.51 & 0.18 \\ -0.39 & 0.71 \end{bmatrix}$	$\text{vaf}_1 = 0.569$ $\text{vaf}_2 = 1/3$ $\text{vaf}_{\text{total}} = 0.902$
USLPCA-multi loss 0	$\begin{bmatrix} 0.38 & -0.13 \\ 0.18 & 0.18 \\ -0.13 & 0.38 \end{bmatrix}$	$\begin{bmatrix} 2 & 0 \\ 1.41 & 1.41 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 0.31 & 0.05 \\ -0.78 & 0.15 \\ 0.02 & 0.57 \\ 0.54 & 0.03 \\ -0.09 & -0.81 \end{bmatrix}$	$\text{vaf}_1 = 1/2$ $\text{vaf}_2 = 1/2$ $\text{vaf}_{\text{total}} = 1$

We first study the solutions from PCA. As the loadings and weights are both derived from the right singular vectors $\mathbf{V}_R$, the two are equal to each other. Now observing the sparse PCA solutions, we can notice that all of the sparse PCA formulations result in different solutions. The variance explained by each component and by both components collectively is also different across the formulations. The loss of equality among weights, loadings and right singular vectors is clear; within and between the sparse PCA formulations, the weight and loading matrices are not equal to each other, or to the right singular vectors $\mathbf{V}_R$. A notable exception is the solution obtained by SPCA-svd which found loadings and component scores identical to those of PCA. This is because $\mathbf{X}$ was generated from a 2-component PCA model without any noise and because SPCA-svd is initialized with the right singular vectors[5]. However, as seen in the loss values, this is not the optimal solution in terms of the optimization criterion.

Our example also illustrates the role of initial values in sparse PCA formulations. Smaller loss was obtained by incorporating multiple starts for initialization. While the total amount of variance captured by the components was equal across the two initial value procedures for SPCA, the multistart approach explained more variance for USLPCA. Moreover, the coefficients attained by different initial value strategies show large discrepancies; this shows that neglecting the problem of local optima is consequential as it may result in lower VAF and inconsistency of the estimated weights and loadings. Unless the true model underlying the data is suspected to be characterized by sparse singular vectors, initializing the algorithm with only the right singular vectors may result in a suboptimal results.

5.2.3.2 Sparse PCA properties: Some pitfalls

The loss of equality among weights, loadings and right singular vectors has a non-negligible consequence for simulation studies conducted to evaluate the sparse PCA methods; a data generating model characterized by sparse weights is disparate from another with sparse loadings or sparse singular vectors, and vice versa. It also has an important implication with respect to choice of the method in practical applications. This is evident in the toy example. The true sparse loadings in (6.10) were only recovered correctly by USLPCA-multi. PCA and SPCA, which target the right singular vectors and sparse weights respectively, were unable to recover the true sparse loading structure.

However, in the sparse PCA literature, models comprised of sparse loadings or sparse singular vectors have been predominantly employed for data generation, regardless of the structure (weights, loadings and right singular vectors) being sparsified by the method (e.g. Johnstone & Lu, 2009; H. Shen & Huang, 2008; Wang & Fan, 2017; Zou et al., 2006). The current literature has therefore largely overlooked the data generating model with sparse weights for simulation studies. In papers which propose a sparse PCA formulation with sparse loadings or sparse singular vectors, excluding the model with sparse weights can be seen as an incomprehensive practice. In other works that propose a sparse weights formulation, neglecting the model with sparse weights for data generation can be considered erroneous. A further complicating factor is that data generated with the sparse

[5] For the same reason of generating the data without noise, two zero weights were estimated for the first component.

weights model poses a more difficult challenge for sparse PCA methods in retrieving the true parameters than the other two models. This implies that many of the results in the sparse PCA literature can be expected to be over-optimistic.

5.2.3.3 Data generating models

This section provides the data generating models (DGM) each comprised with sparse singular vectors, sparse loadings and sparse weights. We also discuss why the sparse weights model is more challenging to analyze than the other two models. Data from the model with sparse right singular vectors can be generated from the model $\mathbf{X} = \mathbf{U}_R\mathbf{S}_R\mathbf{V}_R^\top + \mathbf{E}$ where the right singular vectors $\mathbf{V}_R$ are sparse and column-orthogonal with norm equal one. This is equivalent to generating data from a multivariate normal distribution characterized by a zero mean vector and covariance matrix $\mathbf{\Sigma} = \mathbf{V}_R\mathbf{S}_R^2\mathbf{V}_R^\top + \mathbf{\Sigma}_E$ [6] where $\mathbf{S}_R^2$ denotes the covariance matrix among components. Note that all off-diagonal elements are equal to zero as the component scores are orthogonal. This sparse singular vectors model is referred to as the ***spiked covariance model*** (Johnstone, 2001). With the right singular vectors defined sparse, this model simultaneously comprises sparse weights and sparse loadings ($\mathbf{V}_R = \mathbf{W}_R = \mathbf{P}_R$). From the model $\mathbf{X} = \mathbf{T}_R\mathbf{P}_R^\top + \mathbf{E}$ with $\mathbf{P}_R$ the sparse singular vectors, it follows that $\mathbf{X}\mathbf{P}_R = (\mathbf{T}_R\mathbf{P}_R^\top + \mathbf{E})\mathbf{P}_R = \mathbf{T}_R$, so $\mathbf{P}_R$ indeed comprises the weights that make up the component scores $\mathbf{T}_R$.

The DGM with sparse loadings (***sparse loadings model***[7]) is derived from the PCA decomposition (5.1), in which the loadings are defined sparse: $\mathbf{X} = \mathbf{T}_R\mathbf{P}_R^\top + \mathbf{E}$. The USLPCA formulation (5.4) imposes this model. It is closely related with the spiked covariance model because it coincides with generating from the multivariate normal distribution with covariance matrix $\mathbf{\Sigma} = \mathbf{P}_R\mathbf{P}_R^\top + \mathbf{\Sigma}_E$. In fact, if $\mathbf{P}_R$ is further constrained to be column-orthogonal, the sparse loadings model is equal to the spiked covariance model.

Lastly, to obtain the model with sparse weights (***sparse weights model***) the PCA decomposition with the weights written out is adopted ($\mathbf{X} = \mathbf{X}\mathbf{W}_R\mathbf{P}_R^\top + \mathbf{E}$) and sparsity is induced in the weights matrix. This model is implicitly assumed by the SPCA formulation (5.3). It does not coincide with sparse loadings or spiked covariance models. In comparison to the two models, the sparse weights model poses a much more complicated challenge for the sparse PCA methods for two reasons. First, the component scores $\mathbf{X}\mathbf{W}_R$ are post-multiplied by the loadings $\mathbf{P}_R$ in constructing the data. This implies that the sparseness structure in the weights may not be clearly reflected in the observed data. Second, it suffers from the

[6] The eigenvectors of the covariance matrix $\mathbf{\Sigma} = I^{-1}\mathbf{X}^\top\mathbf{X}$ are equal to the right singular vectors of $\mathbf{X}$.

[7] Referred to as 'factor model' in Fan, Liao, and Mincheva (2013)

problem of indeterminacy; different $\mathbf{W}$ matrices can lead to the same component scores $\mathbf{X}\mathbf{W}_R$[8]. Appendix 5.A provides an example illustrating this indeterminacy problem.

The sections above emphasized the difference between weights, loadings and singular vectors within sparse PCA. Models comprised with any of these sparse structures are also disparate from each other. In the following section, we evaluate common sparse PCA methods on the three different sparse PCA models to demonstrate the consequence of neglecting the difference between these structures and the models comprised of them.

5.3 Simulation study

We present a critical assessment of the sparse PCA methods by taking into account 1) the three different data generating models characterized by sparse weights, sparse loadings and sparse singular vectors and 2) that the sparse PCA solutions resulting from SPCA and USLPCA are subject to the the problem of local optima. Each of the generated data sets is analyzed by each of the methods. Both the effectiveness of the methods at retrieving the true underlying model and at reconstructing the data are evaluated.

It is expected to be more difficult for the sparse PCA methods to reveal the underlying model if the data is generated from the sparse weights model, compared to the other two models. We anticipate that the initial value approaches would lead to different results, as demonstrated by the toy example. We also expect that the difference in performance between the SVD-based and multistart approach will be larger for data generated from the sparse weights model, due to the indeterminacy problem; since multiple different weights matrices can be viable solutions given the same component scores, the initial values would play a role in finding the solution that matches with the true parameters. With respect to the methods' quality of capturing the variance in the data which can be quantified by the VAF measure, it is expected that the sparse weights method would perform better than the sparse loadings method. This is because high levels of sparseness in the loadings can result in all variable scores being estimated as zero. This variable would not at all be accounted for by the model (all scores on the variable become equal to zero), resulting in small VAF. Lastly, with regards to retrieving the true parameters, because SPCA estimates sparse weights and USLPCA sparse loadings, one may expect the SPCA weights to better recover the true weights

[8] For example, consider a case where $\mathbf{X}$ consists of two variables which are identical to each other in a one-component setting. For the component scores, as long as the weights sum up to a particular value, there are infinitely many possible values that these weights can take; the linear combination $\mathbf{X}\mathbf{w}$ would always be identical.

than the USLPCA loadings and also the USLPCA loadings to better recover the true loadings than the SPCA weights. For the recovery of the true loadings, this is a reasonable expectation. However, given the indeterminacy of the weights and when these are generated under the spiked covariance model, it is more reasonable to expect the USLPCA loadings to better recover the sparse weights (as in this setting loadings and weights are equal yet estimation of the loadings is stable unlike the weights). Nevertheless, it is in general difficult to hold a clear expectation about SPCA and elastic net as they suffer from problems such as indeterminacy under high dimensionality.

5.3.1 Design and procedure

Along with the three DGMs, various other data characteristics of the datasets were also manipulated in order to study the interaction between sparse PCA methods, data characteristics and DGMs. Fixing the number of components R to two, we generated datasets via the design below. For each manipulated design factor, the levels are provided between square brackets.

Study design

1. Data generating model (DGM): [Spiked covariance], [Sparse loadings], [Sparse weights]
2. Dimensions of $\mathbf{X}$ ($I \times J$): [Low-dimensional (100×50)], [High-dimensional (100×500)]
3. Level of sparsity in the coefficients matrix: [90%], [50%]
4. Proportion of error variance in $\mathbf{X}$ (PEV): [0%], [10%], [50%]

The following provides the scheme used to generate the data from spiked covariance and sparse loadings models. We adapted the setups devised in Johnstone (2001) (spiked covariance) and in Zou et al. (2006) (sparse loadings).

Algorithm 5.1 Spiked covariance and sparse loadings data generation

```
$\mathbf{X}_{init} \sim \mathcal{MVN}(\mathbf{0}_J, \mathbf{I}_J)$ where $\mathbf{0}_J$ is a zero vector with $J$ elements and $\mathbf{I}_J$ is a $J \times J$ identity matrix. $\mathbf{X}_{init} \in \mathbb{R}^{I \times J}$
Mean-center columns of $\mathbf{X}_{init}$
Perform SVD: $\mathbf{X}_{init} = \begin{bmatrix} \mathbf{U}_R & \mathbf{U}_{R^c} \end{bmatrix} \begin{bmatrix} \mathbf{S}_R & \\ & \mathbf{S}_{R^c} \end{bmatrix} \begin{bmatrix} \mathbf{V}_R & \mathbf{V}_{R^c} \end{bmatrix}^\top$
Replace the elements of $\mathbf{V}_R$ with the smallest absolute values by 0, according to the level of sparsity
if DGM = spiked covariance then
    Orthogonalize columns of $\mathbf{V}_R$, preserving the zero elements
else if DGM = sparse loadings then
    Normalize each column of $\mathbf{V}_R$ to a unit vector
$\mathbf{X}_R \leftarrow \mathbf{U}_R \mathbf{S}_R \mathbf{V}_R^\top$
Project $\mathbf{U}_{R^c}$ and $\mathbf{V}_{R^c}$ to spaces orthogonal to $\mathbf{U}_R$ and $\mathbf{V}_R$, respectively
Orthogonalize columns of $\mathbf{U}_{R^c}$ and $\mathbf{V}_{R^c}$
Scale the elements of $\mathbf{S}_{R^c}$ according to the PEV
$\mathbf{E} \leftarrow \mathbf{U}_{R^c} \mathbf{S}_{R^c} \mathbf{V}_{R^c}^\top$
$\mathbf{X} \leftarrow \mathbf{X}_R + \mathbf{E}$
```

$\mathbf{U}_R$ and $\mathbf{V}_R$ refer to the first R columns of $\mathbf{U}$ and $\mathbf{V}$ (left and right singular vectors), whereas $\mathbf{U}_{R^c}$ and $\mathbf{V}_{R^c}$ refer to the remaining $(J - R)$ columns, respectively. Similarly, $\mathbf{S}_R$ and $\mathbf{S}_{R^c}$ are the first $R \times R$ submatrix and the remaining $(J - R) \times (J - R)$ submatrix of $\mathbf{S}$ (diagonal matrix with singular values), respectively. By relying on the SVD formulation, the model part ($\mathbf{X}_R = \mathbf{U}_R \mathbf{S}_R \mathbf{V}_R^\top$) and the error part ($\mathbf{E} = \mathbf{U}_{R^c} \mathbf{S}_{R^c} \mathbf{V}_{R^c}^\top$) of the final data matrix $\mathbf{X}$ can be defined in an uncorrelated manner. $\mathbf{S}_{R^c}$ is scaled such that the ratio between $\|\mathbf{X}_R\|_F^2$ and $\|\mathbf{E}\|_F^2$ reflects the PEV condition. Additionally, for the sparse loadings model, the true component scores matrix and the loadings matrix are defined by the following: $\mathbf{T}_R = \mathbf{U}_R$ and $\mathbf{P}_R = \mathbf{V}_R \mathbf{S}_R$.

The setup used to generate data according to the sparse weights model is provided in the following. Similar setups have been used in the literature (de Schipper & Van Deun, 2018; Guerra-Urzola et al., 2021; Van Deun et al., 2011).

Algorithm 5.2 Sparse weights data generation

```
1: $\mathbf{X}_{init} \sim \mathcal{MVN}(\mathbf{0}_J, \mathbf{I}_J)$ where $\mathbf{0}_J$ is a zero vector with $J$ elements and $\mathbf{I}_J$ is a $J \times J$ identity
        matrix. $\mathbf{X}_{init} \in \mathbb{R}^{I \times J}$
2: Mean-center columns of $\mathbf{X}_{init}$
3: Perform SVD: $\mathbf{X}_{init} = \begin{bmatrix}\mathbf{U}_R & \mathbf{U}_{R^c}\end{bmatrix} \begin{bmatrix}\mathbf{S}_R & \\ & \mathbf{S}_{R^c}\end{bmatrix} \begin{bmatrix}\mathbf{V}_R & \mathbf{V}_{R^c}\end{bmatrix}^\top$
4: $\mathbf{W}_R \leftarrow \mathbf{V}_R$
5: Replace the elements of $\mathbf{W}_R$ with the smallest absolute values by 0, according to the level of sparsity
6: Normalize each column of $\mathbf{W}_R$ to a unit vector
7: Compute $\mathbf{P}_R$ by performing SVD: $\mathbf{P}_R \leftarrow \mathbf{U}\mathbf{V}^\top$ from $\mathbf{X}_{init}^\top \mathbf{X}_{init} \mathbf{W}_R = \mathbf{U}\mathbf{S}\mathbf{V}^\top$
8: $\mathbf{X}_R \leftarrow \mathbf{X}_{init} \mathbf{W}_R \mathbf{P}_R^\top$
9: Project $\mathbf{U}_{R^c}$ and $\mathbf{V}_{R^c}$ to spaces orthogonal to $\mathbf{T}_R = \mathbf{X}_R \mathbf{W}_R$ and $\mathbf{W}_R$, respectively
10: Orthogonalize columns of $\mathbf{U}_{R^c}$ and $\mathbf{V}_{R^c}$
11: Scale the elements of $\mathbf{S}_{R^c}$ according to the PEV
12: $\mathbf{E} \leftarrow \mathbf{U}_{R^c} \mathbf{S}_{R^c} \mathbf{V}_{R^c}^\top$
13: $\mathbf{X} \leftarrow \mathbf{X}_R + \mathbf{E}$
```

$\mathbf{E}$ is defined in the same manner as for Algorithm 1; via the SVD formulation, it is ensured that the model part and the error part are uncorrelated. The non-sparse loadings matrix $\mathbf{P}_R$ is computed by solving the least squares problem with the orthonormality constraint $\mathbf{P}_R^\top \mathbf{P}_R = \mathbf{I}_R$. The solution is given by $\mathbf{P}_R = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ are left and right singular vectors of $\mathbf{X}_{init}^\top \mathbf{X}_{init} \mathbf{W}_R$ (ten Berge, 1993). This closed-form solution is used in several sparse weights estimation methods where the loadings matrix is constrained to be orthonormal, including the SPCA algorithm (Zou et al., 2006).

For all three DGMs, the initial matrix $\mathbf{X}_{init}$ can also be generated with correlation between the variables instead of using the diagonal covariance matrix. The results obtained from data generated with uncorrelated $\mathbf{X}_{init}$ variables are very similar to the results from correlated $\mathbf{X}_{init}$ matrix, which are reported in the Appendix 5.B.

Fully crossing these factors provided in the study design resulted in $3 \times 2 \times 2 \times 3 = 36$ conditions, and 50 datasets were generated through the above schemes according to each condition. For each of the 1800 datasets, 4 analysis methods which resulted from crossing the following factors were administered.

Analysis methods

1. Sparse PCA method: [SPCA (sparse weights)], [USLPCA (sparse loadings)]
2. Initial value approach: [SVD-based], [Multistart]

The SPCA algorithm implemented in the R package 'elasticnet' was slightly adapted such that the algorithm can be initiated with starting values other than the right singular vectors. For the USLPCA procedure we employed our own R implementation. Both SPCA and USLPCA allow to specify the number of desired zero elements in the estimated coefficient matrix. Therefore the information of the true level of sparsity was provided as an input. The right singular vectors $\mathbf{V}_R$ of the data $\mathbf{X}$ were used as the SVD-based initial values. For the multistart approach, a set of the right singular vectors of $\mathbf{X}$ and 19 other sets of randomly drawn values from uniform distribution $\mathcal{U}(-1, 1)$ were incorporated as initial values. Each set was employed separately for estimation and the solution with minimum loss value was selected as the final solution of the multistart approach. In the following section, the four methods are referred to as SPCA-svd, SPCA-multi, USLPCA-svd and USLPCA-multi, respectively.

To examine the performance of the 4 methods with respect to retrieving the true model parameters and to reconstructing the data, three evaluation criteria were adopted:

Evaluation criteria

1. Zero versus non-zero recovery rate: the number of coefficients correctly estimated as zero or non-zero elements, divided by the total number of coefficients.
2. Component scores congruence: Tucker congruence computed between the estimated and the true component scores.
3. Proportion of variance accounted for (VAF) by the derived components.

The zero versus non-zero recovery rate is always calculated between the quantity being defined sparse in the true model and the quantity being estimated sparse by the method. For example, when USLPCA is used to analyze a dataset generated from the sparse weights model, the true sparse weights defined is compared against the sparse loadings estimated by USLPCA.

The congruence between the true component scores $\mathbf{T}_R^{true}$ and the estimated scores $\hat{\mathbf{T}}_R$ is measured by the Tucker congruence statistic ϕ which is defined as:

$$\phi = \frac{\text{vec}(\mathbf{T}_R^{true})^\top \text{vec}(\hat{\mathbf{T}}_R)}{\sqrt{(\text{vec}(\mathbf{T}_R^{true})^\top \text{vec}(\mathbf{T}_R^{true}))(\text{vec}(\hat{\mathbf{T}}_R)^\top \text{vec}(\hat{\mathbf{T}}_R))}}. \tag{5.6}$$

5.3.2 Results

5.3.2.1 Zero versus non-zero recovery rate

Figure 5.1 shows the boxplots of the zero versus non-zero recovery rate. We separated the results according to whether the datasets were generated such that the defined underlying components completely account for the variance in the data (conditions with zero error variance, in the two bottom rows) or not (10% or 50% of error variance, top rows). The figure first shows that datasets generated from the sparse weights model resulted in a much lower quality of zero versus non-zero recovery than the other two DGMs. Even when the defined components completely explain the variance in the data, the methods resulted in poor performance under the sparse weights model. In contrast, all of the methods resulted in perfect recovery of the coefficients under the spiked covariance and the sparse loadings generation schemes when no error variance was added on top of the true model structure. With respect to the performance of sparse weights versus sparse loadings methods, USLPCA showed an overall higher recovery rate than SPCA. Concerning the initial value approaches, the multistart approach yielded a higher recovery rate than SVD-based initial values.

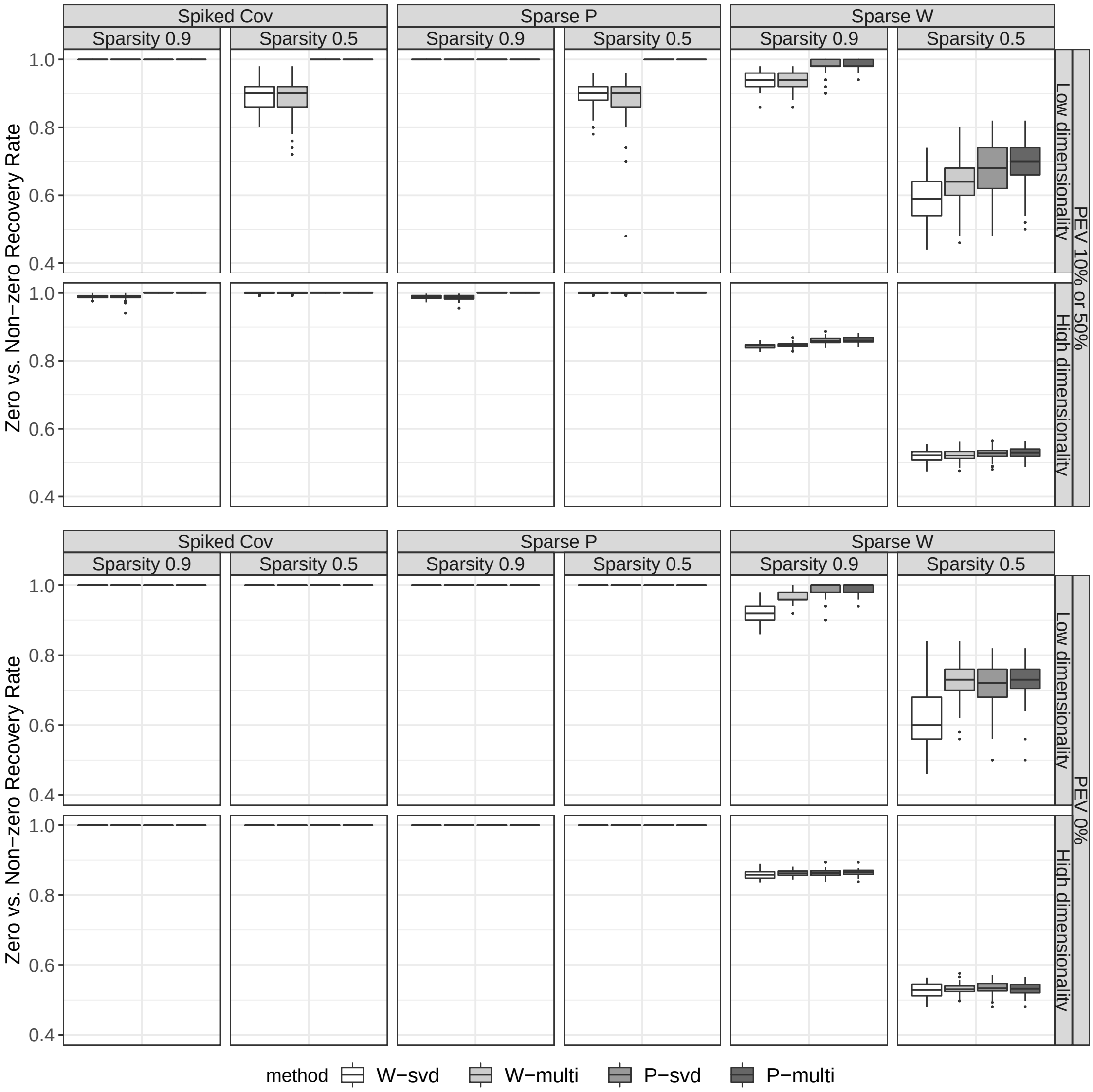


Figure 5.1. Box plots of zero versus non-zero recovery rate. The different columns correspond to the different DGM schemes and sparsity. The rows differ in the level of PEV and whether the data are low- or high-dimensional. The two top rows refer to results concerning datasets in which the defined components do not fully account for the variance in the data (error variance added on top of the DGM), while the bottom rows refer to datasets generated without any error variance.

5.3.2.2 Component scores congruence

Figure 5.2 displays the results on the component scores congruence laid out in the same format as Figure 5.1. The results are largely in agreement with those concerning the zero versus non-zero recovery rate of the coefficients; the DGMs other than the sparse weights model led to good performance, while the methods struggled on datasets generated from the sparse weights model. Concerning the spiked covariance model and the sparse loadings model, all of the methods performed nearly perfectly in retrieving the component scores, except for SPCA which led to some poor outlying results when error variance was added on top of the defined DGM.

Although the methods performed poorly for the sparse weights model compared to the other models, it can be seen that the median Tucker congruence of the methods were in most cases above 0.9. Components with a congruence value in between 0.85 and 0.94 are often seen as fairly similar (Lorenzo-Seva & Ten Berge, 2006). Despite the low zero versus non-zero recovery of true weights shown in Figure 5.1, the component scores were recovered quite well by the methods. This hints back at the indeterminacy problem of the sparse weights model. Although the component scores are well recovered, it is difficult for the methods to retrieve the true weights since there are multiple different weights matrices that can construct very similar component scores. Lastly, the impact of the starting values is also seen; the multistart approach yielded better results than initializing the methods with SVD solutions.

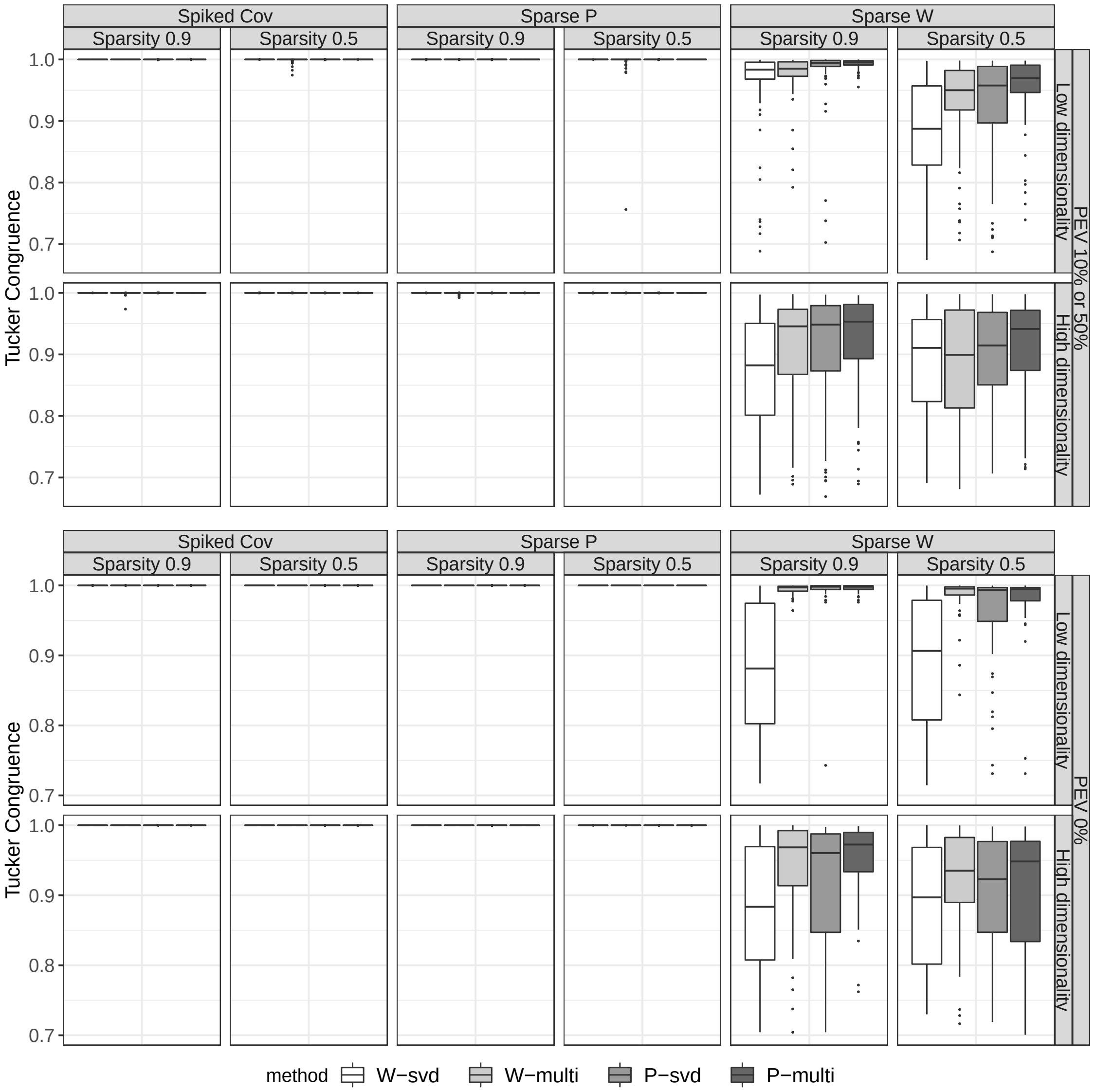


Figure 5.2. Box plots of component scores congruence. The different columns correspond to the different DGM schemes and sparsity. The rows differ in the level of PEV and whether the data are low- or high-dimensional. The two top rows refer to results concerning datasets in which the defined components do not fully account for the variance in the data (error variance added on top of the DGM), while the bottom rows refer to datasets generated without any error variance.

5.3.2.3 Proportion of variance accounted for (VAF)

While the two evaluation criteria above reflect the behaviour of the methods with regards to retrieving the underlying model parameters, VAF pertains to the degree to which the methods restore the observed data. Figure 5.3 presents these results. Dimensionality did not lead to differential results for VAF thus aggregated results are presented. The boxplots show that the most impactful factor for reconstruction quality is the proportion of additional error variance in the observed dataset on top of the true components. As more noise is added to the DGM, the methods exhibit smaller VAF. It is interesting to see that across all of the conditions, the approach of initializing the algorithm did not exert an influence, unlike in the other evaluation criteria above. While all four methods performed comparably for the spiked covariance model and the sparse loadings model, the USLPCA methods underperformed compared to SPCA methods for the datasets generated with sparse weights. Moreover, with this exception of USLPCA being administered to data from the sparse weights model, the methods have succeeded in capturing the correct proportion of variance accounted for by the true components. For the condition with 50% of error variance, the methods have even explained slightly more variance than the true proportion of variance. This can be seen as a case of overfitting; on top of capturing the variance defined by the true underlying components, the methods seem to also explain a small amount of error variance.

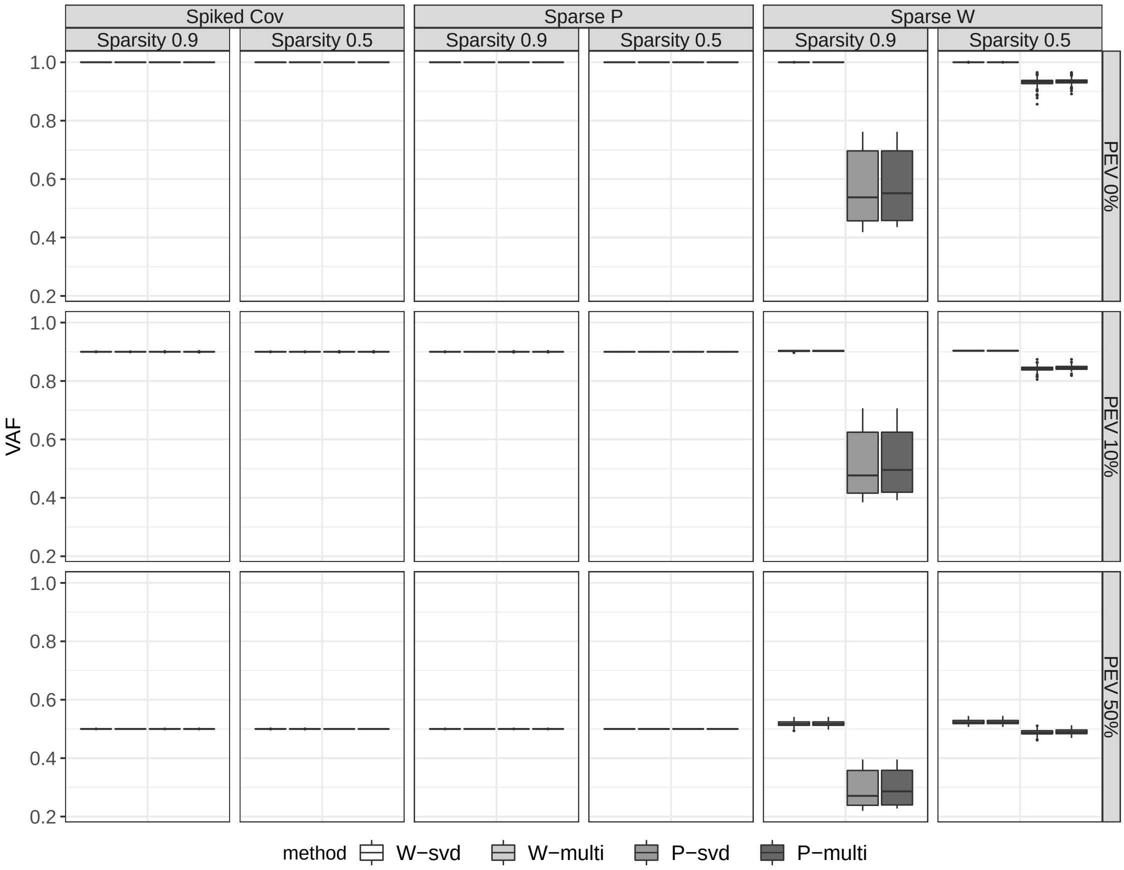


Figure 5.3. Box plots of proportion of variance accounted for. The different columns correspond to the different DGM schemes and sparsity. The rows differ in the level of PEV by the different components.

5.3.3 Discussion of the Results

The results from the simulation study are mainly in line with our expectations. While the sparse PCA methods resulted in near-optimal performance in finding back the true parameters under the spiked covariance and sparse loadings models, they showed much poorer performance under the sparse weights model. The methods struggled particularly with respect to recovering the true weights. This finding is quite alarming because it implies that the conclusions being drawn from the current sparse PCA literature dominated by the spiked covariance and spiked loadings models may be over-optimistic.

Our results also clearly illustrate the impact of initial values for these sparse PCA methods. For both evaluation criteria of zero versus non-zero recovery rate and component scores congruence, the multistart approach resulted in better performance than starting with the right singular vectors. In particular, the difference in the performance between the initial value approaches was large under the sparse weights model. Considering that the true model underlying data is unknown in practice, it is important to adopt multiple starting values for sparse PCA methods that are characterized by local optima, such as SPCA and USLPCA in our simulation study.

Finally, our simulation study highlights the difference between the results obtained by sparse loadings versus sparse weights methods. In our toy example above, it was shown that the methods lead to different estimates and therefore different insights about the same dataset. The simulation study extends this difference towards performance of the methods. It was shown that SPCA with sparse weights was poorer at zero versus non-zero recovery of parameters under all data generation schemes than USLPCA with sparse loadings. SPCA resulted in underperformance compared to USLPCA even when the data was generated from the sparse weights model. However, SPCA was better suited at deriving components that explain a large amount of variance in the data if the data were generated under a sparse weights model and performed nearly as well as USLPCA under the sparse spiked covariance and loading scheme. This finding implies that the methods must be chosen carefully in practice.

5.4 Empirical application

In this section we further illustrate the practical impact of the choice between the sparse weights and sparse loadings methods coupled with initialization strategies in an empirical setting.

5.4.1 Big Five dataset IPIP-NEO-120

We adopted a dataset comprised of 120 items of the IPIP-NEO-120 questionnaire. The IPIP-NEO-120 scale is a set of public domain items designed to measure the Big Five personality traits (Goldberg et al., 1999); each of the personality traits is measured by 24 items. We downloaded the raw data collected with the questionnaire which is publicly available from the online repository:`https://osf.io/wxvth/` (Johnson, 2018). The raw data was collected via a large internet survey conducted for the construction of the questionnaire (Johnson, 2014) in which 619150 subjects participated, and we selected the first 1000 observations of the data for our sparse PCA analysis to ease the computational burden.

As the questionnaire measures five underlying constructs, we fixed the number of components to be five. Also following the design of the questionnaire, the level of sparsity was determined such that 24 non-zero coefficients are estimated per component. In order to examine the impact of choosing between the sparse weights and sparse loadings methods and the initial value approaches, we applied the methods used in the simulation study: SPCA-svd, SPCA-multi, USLPCA-svd and USLPCA-multi.

The weights from SPCA methods and the loadings from USLPCA methods were inspected to interpret the found components. Table 5.2 presents the numbers of items designed for each of the personality traits which have a non-zero coefficient for each of the components.

Table 5.2. Big Five dataset: sparse PCA methods with 24 non-zero coefficients per component (O: openness, C: conscientiousness, E: extraversion, A: agreeableness, N: neuroticism). The columns indicate each component while the rows indicate each personality trait.

SPCA-svd weights

	t1	t2	t3	t4	t5
O	20	2	1	1	0
C	0	18	0	1	4
E	0	4	19	4	1
A	4	0	0	17	1
N	0	0	4	1	18

USLPCA-svd loadings

	t1	t2	t3	t4	t5
O	19	0	1	4	0
C	0	16	5	1	0
E	1	7	11	0	6
A	3	0	2	19	0
N	1	1	5	0	18

SPCA-multi weights

	t1	t2	t3	t4	t5
O	20	2	1	1	0
C	0	18	0	1	4
E	0	4	19	4	1
A	4	0	0	17	1
N	0	0	4	1	18

USLPCA-multi loadings

	t1	t2	t3	t4	t5
O	19	3	1	1	1
C	0	18	1	1	0
E	0	3	17	4	2
A	5	0	0	14	3
N	0	0	5	4	18

Table 5.2 presents the similarities between the models constructed by the sparse weights and the sparse loadings methods. For all four methods, a majority of the non-zero coefficients on each component correspond to items that measure the same personality trait. As the items in the scale operationalize the five-factor model of personality traits (FFM; McCrae & Costa Jr, 2008), the models found by the four methods seem to nicely reflect the true model behind the observed items. This goes together with the results in our simulation study where the sparse weights and the sparse loadings methods were both capable at finding the underlying structure when low-dimensional data was generated from a spiked covariance or sparse loadings structure. The FFM resembles these models as the subsets of 24 items load on the five factors.

The table also shows the role of initial values in these sparse PCA methods. While all of the non-zero weights found by the two SPCA methods corresponded to each other, different initial values have led to different models being constructed for USLPCA. The third component from USLPCA-svd can be interpreted as a mix between extraversion, conscientiousness and neuroticism, while the third component from USLPCA-multi is less diffuse. This demonstrates the impact of initial value strategies in practice where sparse PCA models are used.

Finally, the findings from the simulation study regarding the proportion of variance explained by the sparse PCA methods are also echoed with this dataset. SPCA methods accounted for more variance than USLPCA methods. SPCA-svd and SPCA-multi models explained 32.3% and 32.3% of variance in the data while USLPCA-svd and USLPCA-multi fell short at 25.9% and 27.3%.

5.4.2 Autism gene expression data

This dataset concerns gene expression profiles of three groups: 6 male subjects with autism caused by fragile X syndrome (FMR1-FM), 7 male subjects with autism caused by inherited duplication of 15q11-q13 (dup15q) and 14 non-autistic control male subjects (Nishimura et al., 2007)[9]. The dataset consisted of 43893 probe sets measuring the transcription rates of about 20 thousand genes for each subject. In the original publication, the authors selected a subset of the probes that are important at discerning the three groups by inspecting the *p* values derived by univariate ANOVA. The authors continued on by conducting PCA with 3 components on this subset to explore the mechanism underlying the probes.

Building on the original publication, we also conducted ANOVA to obtain a subset of probes which are strongly related with the group membership of the subjects. Prior to our analysis, each column of the dataset was mean-centered and standardized to unit variance. ANOVA resulted in 107 probes with *p* values smaller than 0.05. As our current paper discusses sparse PCA, we also sampled 1000 other ‘redundant’ probes among the probes with *p* values greater than 0.5 and constructed a dataset of 1107 probes. We did not use the entire set of variables to reduce the computational burden. The four methods SPCA-svd, SPCA-multi, USLPCA-svd and USPLCA-multi were then administered, in order to study these differences between the methods in providing the sparse components comprised of important and redundant probes.

The number of components was fixed at 3 as in Nishimura et al. (2007). As there are 107 important probes, we administered the methods with 107 non-zero coefficients per component and studied the returned coefficients[10]. Unlike the Big Five dataset above, the data generating model underlying the autism gene expression dataset is ambiguous; the nature of the mechanisms governing the gene expressions is unknown. Since it is not possible to compare the retrieved sparse PCA models with the true structure, we compared the results among themselves.

[9] The data is publicly available on the NCBI GEO database with the accession code GSE7329.

[10] As it may also be sensible to distribute the non-zero coefficients evenly across the 3 components, we also administered the methods with 36 non-zero coefficients per component. The results can be found in Appendix 5.C and the conclusions drawn are in line with the results presented here.

Table 5.3 presents the proportion of non-zero coefficients that correspond across each pair of the four methods (above) and the Tucker congruence values between the component scores computed by the methods (below).

Table 5.3. Autism dataset. Above: proportion of corresponding non-zero coefficients out of the total 321 (107 non-zero coefficients ×3 components). Below: Tucker congruence between the component scores.

Proportion of corresponding non-zero coefficients

	SPCA-svd	SPCA-multi	USLPCA-svd
SPCA-multi	0.283		
USLPCA-svd	0.461	0.461	
USLPCA-multi	0.305	0.452	0.573

Component scores Tucker congruence

	SPCA-svd	SPCA-multi	USLPCA-svd
SPCA-multi	0.685		
USLPCA-svd	0.818	0.796	
USLPCA-multi	0.629	0.780	0.728

Table 5.3 conveys that the models derived by the four methods are quite different. Within the same formulation of SPCA, only 28.3% of the non-zero weights found by the multistart approach were also found by employing the starting values based on SVD. Likewise, 57.3% of the non-zero loadings from the two starting value approaches corresponded to each other for USLPCA. This proportion of corresponding non-zero coeffcients is also low when comparing weights from the SPCA methods against loadings from the USLPCA methods. On the other hand, the congruence values among the components were higher; although largely different sets of variables were picked up by the different methods, the estimated components ended up rather correlated. This is in line with our results from the simulation study where the congruence between the estimated and the true components were high for the sparse weights model, despite its low zero versus non-zero recovery rate. Nevertheless, the congruence scores between the methods concerning the current autism dataset were all lower than 0.85 which is a value expected for fairly similar components (Lorenzo-Seva & Ten Berge, 2006). Altogether, these results imply that the four methods would provide components that are understood as being different from each other. They reiterate the findings from the simulation study that showed different performances of the methods depending on the sparse PCA formulation and the initial value strategies of choice.

With respect to the proportion of explained variance, SPCA-svd and SPCA-multi both recovered 36.6% of the data. On the contrary, USLPCA-svd and USLPCA-multi resulted in 16.3% and 16.4%. Like in the simulation study, the choice between sparse weights and loadings led to a considerable difference in the amount of variance explained.

5.5 Conclusion

The contribution provided by this paper concerns an important warning towards the difference between weights and loadings and its implications. Section 5.2 discussed the theoretical difference. It was shown that the weights and the loadings are not equal to each other or to the right singular vectors within sparse PCA, making it important for the quantities to be distinguished carefully. We pointed out that this loss of equality between the quantities is not well reflected in research employing sparse PCA. Namely, a vast majority of simulation studies confine themselves to data generating models characterized by sparse singular vectors or sparse loadings, and most sparse PCA methods initialize the algorithms with SVD solutions. Through a simulation study, we demonstrated that such practices paint a wrong picture about the performance of sparse PCA methods. In fact, reported simulation studies have been dominated by the spiked covariance and sparse loadings model, also to study methods that estimate sparse weights. Based on such studies, over-optimistic conclusions have been drawn about the performance of the sparse weights method in recovering the underlying zero-nonzero structure of the data. A related issue is the combination of generating data under a sparse SVD structure in combination with SVD based starting values; as shown in the toy example, also the sparse loadings method suffers from recovering the underlying sparseness structure when sparseness does not reside in singular vectors. The importance of using methods that implement a multi-start initialization strategy was discussed and shown throughout the paper.

Our paper also touches upon the issue of choosing between sparse weights and sparse loadings PCA. In practice, in making an informed choice between them, we consider the research aim as the first aspect to take account of. If it is expected that the variables are associated to a few underlying components and one wishes to find a sparse representation of the relationships between the components and the variables, sparse loadings PCA is suitable. On the other hand, as reported in our simulation study, if one aims to derive summary scores that account for a great amount of variance in the variables, sparse weights PCA should be the choice.

Besides the aim of the analysis (summarizing versus exploring the component-variable associations), often domain-related beliefs regarding the data may determine the choice between the sparse loadings and sparse weights model. For example, psychological scales are often constructed according to a sparse loadings model in the sense that the variables (items) are designed to measure particular latent constructs: the variables are reflective indicators of a latent variable (Hwang et al., 2021). Personality questionnaire data such as one provided in Section 5.4 would serve as an example, or measurements of the construct IQ.

On the other hand, the sparse weights model may be considered appropriate when the interest is to measure indices of observable constructs (e.g., poverty index) in a context where such indices are not yet known. This is for example of particular interest for the construction of genetic risk scores. Data originating from sparse weights models would be comprised of variables that linearly combine into a component. Economic data containing variables that combine and form an economic index is another example; education level, income, occupation and other variables link up together to form socioeconomic status (e.g. Hauser & Warren, 1997; Thomson, 2018). Here, the observed variables form the component and are therefore often referred to as composite indicators (Hwang et al., 2021).

Our findings from the datasets generated from the sparse weights model can appear counterintuitive to existing literature concerning the consistency of sparse PCA methods. In the data generation, these studies have employed the spiked covariance model. Our findings are therefore in agreement with them, as the sparse PCA methods were very good at correctly revealing the spiked covariance model or the sparse loadings model. The contribution of our work is showcasing that when other plausible sparse PCA models underlie data, the sparse PCA methods may not be as optimal in recovering the data generating model. Note that this does not imply that these previous studies are irrelevant as they indeed demonstrate the effectiveness of sparse PCA in the high-dimensionality-low-sample-size setting where PCA is known to perform poorly.

We conclude with a plea. Simulation studies should not be restricted to sparse singular vectors or sparse loadings structures, as (A) different models may underlie data in practice and (B) certain sparse PCA formulations impose a model structure that do not match these structures. Also, multiple starting values should be considered along with the solutions of SVD. Many sparse PCA formulations are characterized by non-convex problems, and limiting the starting values at the SVD solutions can push the methods to converge into a local optimum. The loss of equality between weights, loadings and right singular vectors in the context of sparse PCA should be carefully acknowledged.

Appendix

5.A Sparse weights indeterminacy problem

Deriving the true sparse weights from the data is a difficult task because of the indeterminacy problem of the weights; different $\mathbf{W}_R$ matrices can construct the same component scores $\mathbf{X}\mathbf{W}_R$. For high-dimensional data, this implies that different sets of variables can be combined to lead to the same component scores. We provide a small example conveying the problem in this section.

We have generated a small data matrix $\mathbf{X}$ of size 3×5 from a multivariate normal distribution characterized by a zero vector of length 5 for the mean and a 5×5 identity matrix for the covariance. After centering and standardizing the variables, we can extract one component that captures the largest amount of variance by performing PCA, which provides the following weights and component scores:

$$\underbrace{\begin{bmatrix} -1.37 & -1.40 & -0.99 & -1.36 & -1.20 \\ 0.37 & 0.52 & 1.37 & 1.01 & 1.25 \\ 0.99 & 0.88 & -0.37 & 0.35 & -0.05 \end{bmatrix}}_{\mathbf{X}} \times \underbrace{\begin{bmatrix} 0.43 \\ 0.45 \\ 0.41 \\ 0.48 \\ 0.46 \end{bmatrix}}_{\hat{\mathbf{w}}} = \underbrace{\begin{bmatrix} -2.83 \\ 2.02 \\ 0.81 \end{bmatrix}}_{\hat{\mathbf{t}}} \tag{5.7}$$

The indeterminacy problem is that there are many solutions for $\mathbf{w}$ different from the one in the above equation that yield the same component scores. This means that when administering a sparse weights PCA method, any linear combination of 3 out of the 5 variables can lead to the same component scores. Below are two such weights:

$$\underset{\mathbf{X}}{\begin{bmatrix} -1.37 & -1.40 & -0.99 & -1.36 & -1.20 \\ 0.37 & 0.52 & 1.37 & 1.01 & 1.25 \\ 0.99 & 0.88 & -0.37 & 0.35 & -0.05 \end{bmatrix}} \times \underset{\hat{\mathbf{w}}}{\begin{bmatrix} 0 \\ 0 \\ -0.49 \\ 1.90 \\ 0.62 \end{bmatrix}} = \underset{\hat{\mathbf{t}}}{\begin{bmatrix} -2.83 \\ 2.02 \\ 0.81 \end{bmatrix}} \tag{5.8}$$

$$\underset{\mathbf{X}}{\begin{bmatrix} -1.37 & -1.40 & -0.99 & -1.36 & -1.20 \\ 0.37 & 0.52 & 1.37 & 1.01 & 1.25 \\ 0.99 & 0.88 & -0.37 & 0.35 & -0.05 \end{bmatrix}} \times \underset{\hat{\mathbf{w}}}{\begin{bmatrix} 0.60 \\ 0.69 \\ 1.05 \\ 0 \\ 0 \end{bmatrix}} = \underset{\hat{\mathbf{t}}}{\begin{bmatrix} -2.83 \\ 2.02 \\ 0.81 \end{bmatrix}} \tag{5.9}$$

In a high-dimensional setting where the number of observations is smaller than the number of variables, retrieving the correct underlying sparse weights from many possible solutions can therefore be a complicated task. Different sets of variables can combine into the same component scores.

5.B Simulation study with data generation where initial data matrix is generated with correlation

This section presents a simulation study employing data generation schemes that start off with an initial data matrix with correlated variables. Section 5.3.1 provides the motivation behind these additional schemes.

5.B.1 Design and procedure

Fixing the number of components R to two, we generated datasets via the following design. The levels for the data generating model factor are the only difference from the study design presented in Section 5.3.1.

Study design

1. Data generating model (DGM): [Spiked covariance (correlated)], [Sparse loadings (correlated)], [Sparse weights (correlated)]
2. Dimensions of $\mathbf{X}$ $(I \times J)$: [Low-dimensional (100×50)], [High-dimensional (100×500)]
3. Level of sparsity in the coefficients matrix: [90%], [50%]
4. Proportion of error variance in $\mathbf{X}$ (PEV): [0%], [10%], [50%]

The following algorithm provides how the initial data matrix is generated with correlated variables. Aside from the generation of the initial matrix, the data generating schemes are the same as the three schemes used in Section 5.3. Hence, after the generation of the initial matrix, the same steps given in Algorithm 1 and 2 are followed to simulate the data.

Algorithm 5.3 Correlated initial data matrix generation

```
Generate initial loadings matrix $\mathbf{P}_{init}$ $(R \times J)$ from the uniform distribution $\mathcal{U}(-1, 1)$.
    $R$ and $J$ refer to the number of components and the number of variables, respectively.
Replace the elements of $\mathbf{P}_{init}$ with the smallest absolute values by 0, according to the
    level of sparsity (either 90% or 50%)
$\mathbf{\Sigma}_1 = \mathbf{P}\mathbf{P}^\top$
$\mathbf{\Sigma}_2 = \mathbf{\Sigma}_2 + \mathbf{D}$, where $\mathbf{D}$ is a diagonal matrix with elements that are very small in
    magnitude. This is added to ensure that all of the eigenvalues of $\mathbf{\Sigma}_2$ are positive.
Standardize $\mathbf{\Sigma}_2$ such that it becomes a correlation matrix $\mathbf{S}_{init}$
$\mathbf{X}_{init} \sim \mathcal{MVN}(\mathbf{0}_J, \mathbf{S}_{init})$ where $\mathbf{0}_J$ is a zero vector with $J$ elements
if DGM = spiked covariance | sparse loadings then
    Proceed to step 2 in Algorithm 1
else if DGM = sparse weights then
    Proceed to step 2 in Algorithm 2
```

Fully crossing the factors in the study design led to $3 \times 2 \times 2 \times 3 = 36$ conditions, and 50 datasets were generated through the above schemes according to each condition. For each of the 1800 datasets, the 4 analysis methods were administered in the same manner as the above simulation study.

5.B.2 Results

5.B.2.1 Zero versus non-zero recovery rate

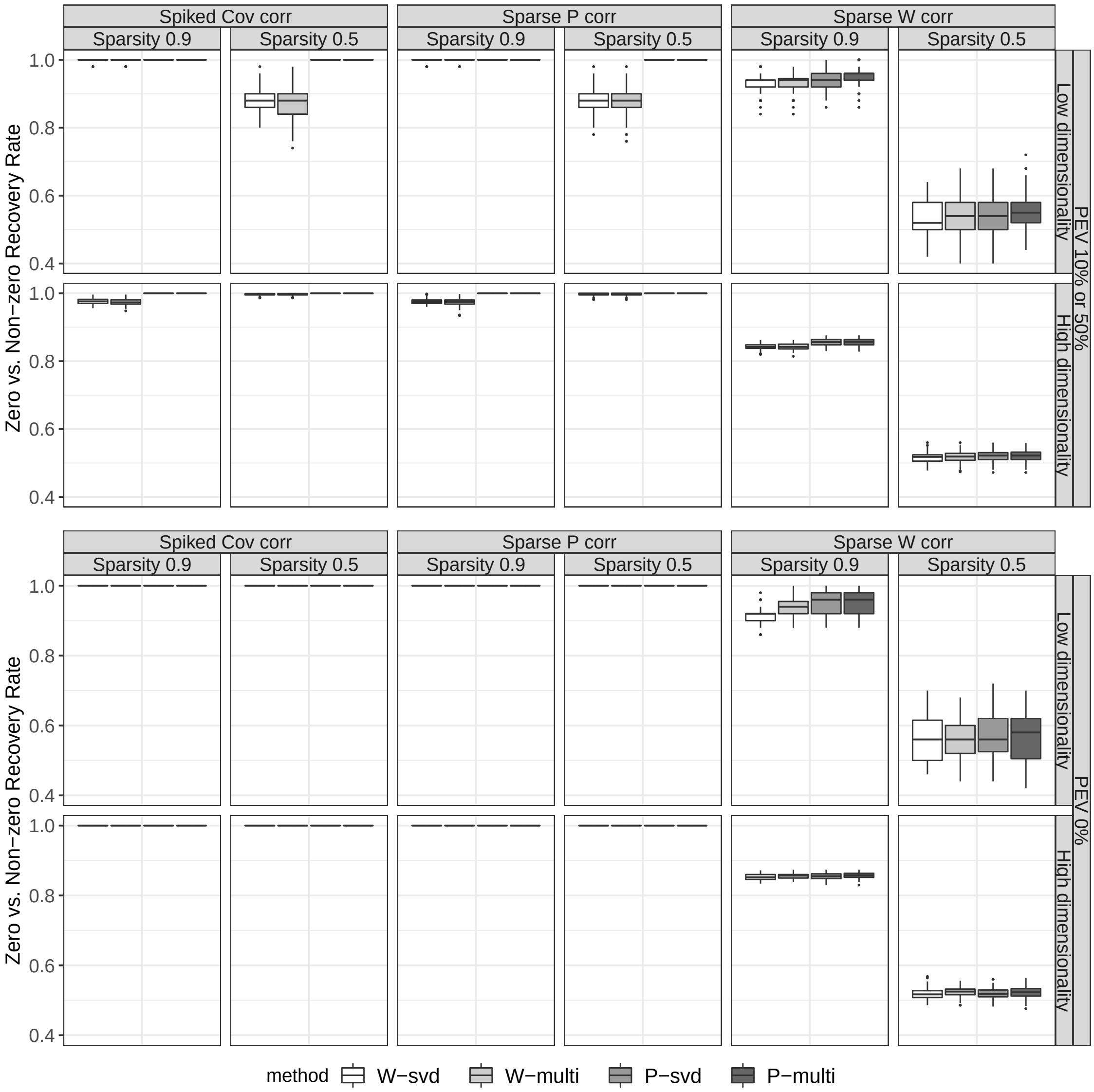


Figure 5.4. Box plots of zero versus non-zero recovery rate. The different columns correspond to the different DGM schemes and sparsity. The rows differ in the level of PEV and whether the data are low- or high-dimensional. The two top rows refer to results concerning datasets in which the defined components do not fully account for the variance in the data (error variance added on top of the DGM), while the bottom rows refer to datasets generated without any error variance.

5.B.2.2 Component scores congruence

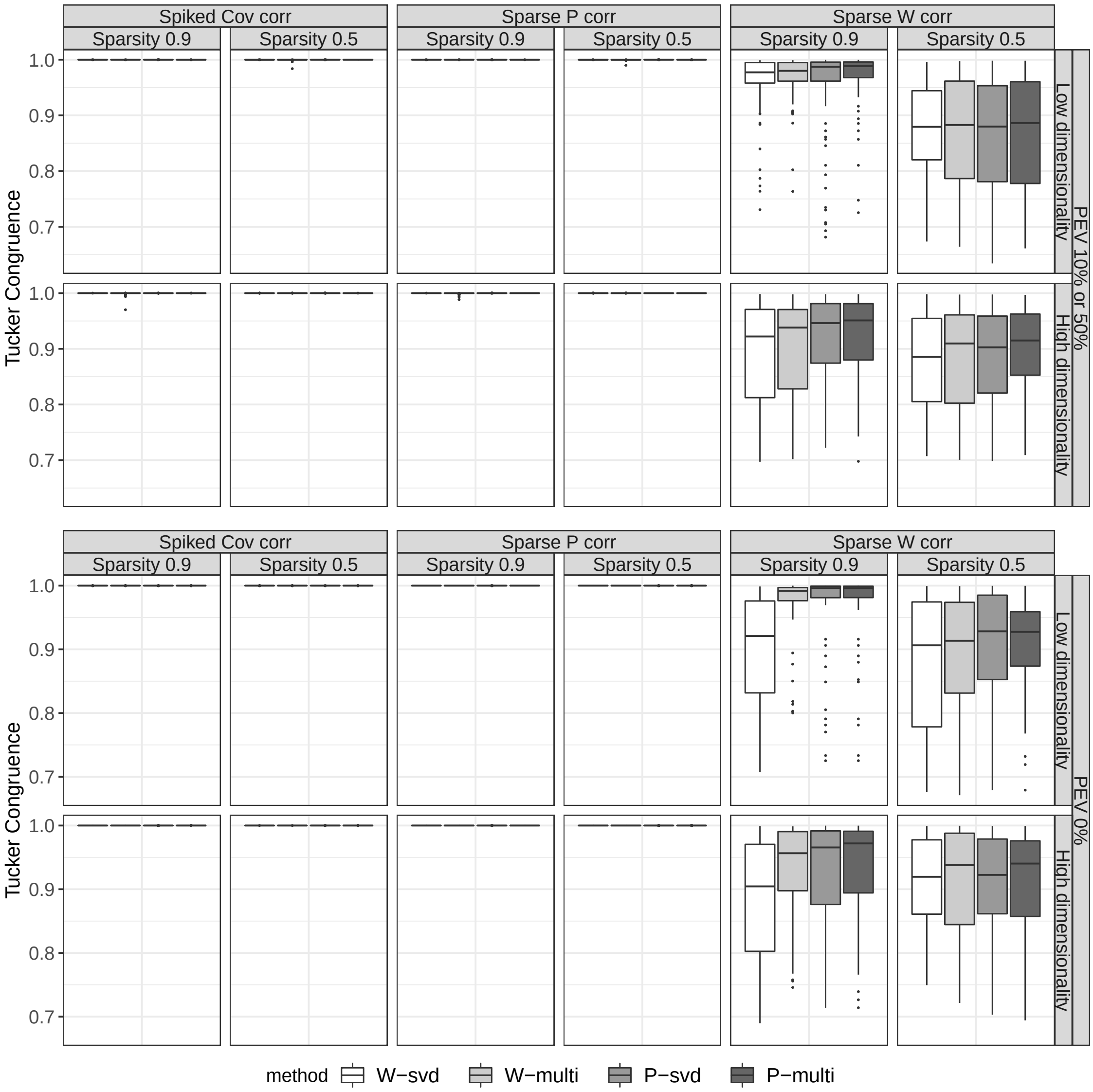


Figure 5.5. Box plots of component scores congruence. The different columns correspond to the different DGM schemes and sparsity. The rows differ in the level of PEV and whether the data are low- or high-dimensional. The two top rows refer to results concerning datasets in which the defined components do not fully account for the variance in the data (error variance added on top of the DGM), while the bottom rows refer to datasets generated without any error variance.

5.B.2.3 Proportion of variance accounted for (VAF)

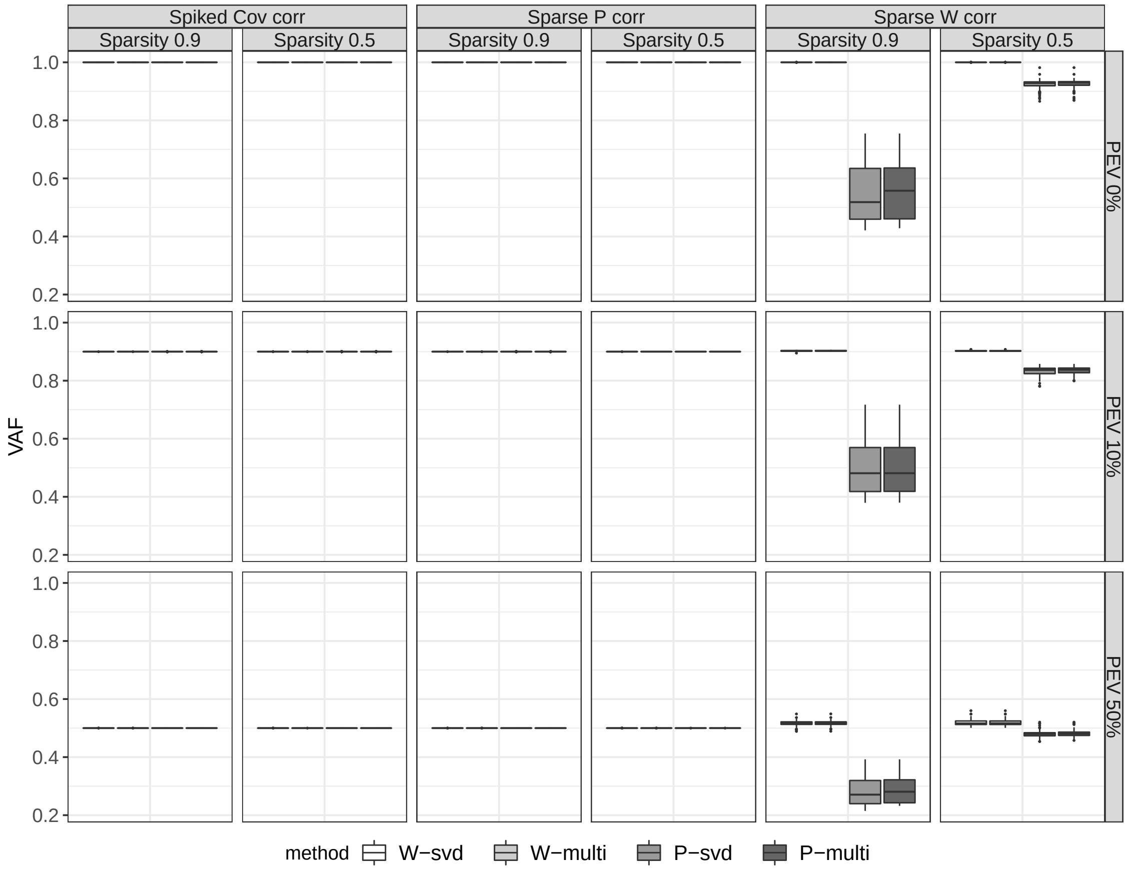


Figure 5.6. Box plots of proportion of variance accounted for. The different columns correspond to the different DGM schemes and sparsity. The rows differ in the level of PEV by the different components.

Compared to Figures 5.1, 5.2 and 5.3 from the simulation study in Section 5.3, it can be seen that throughout the three evaluation criteria that the new scheme with correlated initial data matrix leads to very similar results. The conclusions drawn are identical; the sparse weights model offers a much more complicated challenge for the sparse PCA methods in recovering the underlying true weights. Across all DGMs, SPCA methods underperform compared to USLPCA methods in identifying the zero-nonzero structure in the parameters.

5.C Autism gene expression data analysis with 36 non-zero coefficients per component

Since three components are extracted by sparse PCA from a dataset with 107 important probes out of the total 1107 probes, it may also be sensible to estimate the 36 non-zero coefficients per component. This would evenly distribute the 107 important probes across the three components. As done above, the results obtained from the different sparse PCA methods were compared against each other. Table 5.4 presents the proportion of non-zero coefficients that correspond across each pair of the four methods (above) and the Tucker congruence values between the component scores computed by the methods (below).

Table 5.4. Autism dataset. Above: proportion of corresponding non-zero coefficients out of the total 108 (36 non-zero coefficients $\times 3$ components). Below: Tucker congruence between the component scores.

Proportion of corresponding non-zero coefficients

	SPCA-svd	SPCA-multi	USLPCA-svd
SPCA-multi	0.231		
USLPCA-svd	0.204	0.185	
USLPCA-multi	0.074	0.056	0.417

Component scores Tucker congruence

	SPCA-svd	SPCA-multi	USLPCA-svd
SPCA-multi	0.772		
USLPCA-svd	0.816	0.736	
USLPCA-multi	0.613	0.625	0.817

Table 5.4 conveys a message very much in line with the results obtained from 107 non-zero coefficients per component (Table 5.3). The four models construct different models. Only 23.1% of the non-zero weights found by SPCA-svd and SPCA-multi corresponded to each other. Likewise, 18.5% of the non-zero loadings obtained from USLPCA-svd were also obtained from USLPCA-multi. The proportion of corresponding non-zero coeffcients are also low when considering other pairs of methods. Component scores' congruence values were all lower than 0.85 which is also in line with Table 5.3. The components extracted by the four methods would be interpreted as being different from each other.

With respect to the proportion of explained variance, SPCA-svd and SPCA-multi both recovered 36.6% of the data. Estimating a smaller number of non-zero weights did not decrease the amount of explained variance, compared to the model found above with 107 non-zero weights per component. This is in line with previous findings in the literature that showed that high levels of sparsity in weights can still explain large amount of variance (de Schipper & Van Deun, 2021). In contrast, USLPCA-svd and USLPCA-multi resulted in 6.5% and 7.4%. As in the simulation study and also for the results above, the sparse loadings methods expalin a considerably lower amount of variance than the sparse weights methods.

Chapter 6

Optimal penalized Principal Component Analysis using cardinality as a sparsity-inducing penalty

Sparse principal component analysis (PCA) is well-accepted in many areas as a means to perform dimension reduction in an interpretable and consistent manner from a high-dimensional dataset. Among various sparse PCA approaches, those that rely on penalization to obtain sparse solutions have been widely used due to their computational tractability and scalability. However, one of the main criticisms of these penalized PCA methods in the literature is that their performance is assessed via numerical experiments without a theoretical guarantee of obtaining optimal solutions. This paper considers a penalized PCA problem with cardinality as a sparsity-inducing penalty. A minorization-maximization scheme is proposed to solve the problem, and it is shown theoretically that the resulting solution is a local optimum. While local optimality is guaranteed under the condition that the smallest eigenvalue of the covariance matrix is greater than 1, we provide a simple procedure that safeguards the condition for any dataset, including those in high dimensionality. Numerical experiments involving a synthetic dataset and an empirical dataset are conducted to demonstrate the implication of this condition in practice.

Keywords: Sparse PCA, Penalized PCA, Optimality, Cardinality, Minorization-maximization

Park, S., & Guerra-Urzola, R. (in prepration).

6.1 Introduction

Sparse PCA has been an active topic in the literature to gain interpretability and consistency in PCA solutions, especially in the setting of high-dimensional data. Although imposing a constraint on the cardinality of the solution seems to be a natural choice for attaining the sparse solution, adding a cardinality constraint results in an NP-hard problem which is intractable (Natarajan, 1995). To address this impracticality, relaxations that consider sparsity-inducing penalties have been used to achieve sparsity in the PCA solution. We refer to this kind of methods as *penalized PCA*. Despite the advantages concerning computational tractability, scalability and statistical properties (see e.g. Guerra-Urzola et al., 2022), penalized PCA methods in the literature have a shortcoming that they provide heuristic solutions without a theoretical guarantee of optimality.

Whilst sparse PCA problems have been formulated in different ways, this paper focuses on the formulation with variance maximization where the variance of the derived components are maximized (e.g. d'Aspremont et al., 2004; Journée, Nesterov, Richtárik, & Sepulchre, 2010), rather than the formulation with least squares where the squared error is minimized between the original data and the PCA-reconstructed data (e.g. H. Shen & Huang, 2008; Zou et al., 2006). Under the variance maximization formulation, several penalties have been proposed to induce specific sparse structures in the solution. The most common sparsity-inducing penalties are the l_0 and l_1 norms. Representative works to solve the penalized PCA problem, using the norms l_0 and l_1, include the well-known iterative GPower algorithm (Journée et al., 2010) and an alternating optimization scheme presented by Richtárik, Jahani, Ahipaşaoğlu, and Takáč (2021). On the other hand, Sriperumbudur, Torres, and Lanckriet (2011a) proposed a broad majorization-minimization approach to the sparse generalized eigenvalue problem considering an approximation of the l_0 norm as a sparsity-inducing penalty[1].

This paper studies a penalized PCA problem based on variance maximization and cardinality as a sparsity-inducing penalty. We consider the problem

$$\max_{\mathbf{w}\in\mathcal{B}} \mathbf{w}^\top \mathbf{\Sigma} \mathbf{w} - \alpha \|\mathbf{w}\|_0, \tag{6.1}$$

with $\alpha > 0$ a denoting penalty parameter, $\mathbf{\Sigma} = \mathbf{X}^\top \mathbf{X}$ is the covariance matrix with $\mathbf{X} \in \mathbb{R}^{I\times J}$ is the data set, $\|\mathbf{w}\|_0$ denotes the number of nonzero elements in $\mathbf{w}$, and $\mathcal{B} = \{\mathbf{x} \in \mathbb{R}^J : \|\mathbf{x}\| \leq 1\}$ is the unit Euclidean ball. We use a minorization-maximization (MM) method to solve problem 6.1, and show that it achieves a locally optimal solution to problem (6.1). Local optimality is attained by our method

[1] For a comprehensive review of penalized PCA method see (Guerra-Urzola et al., 2022, 2021)

under the condition that the smallest eigenvalue of $\mathbf{\Sigma}$ is greater than 1. We show that this condition can be met for any data set by employing a simple procedure to transform the $\mathbf{\Sigma}$ matrix. The procedure can also ensure the condition to be met for high-dimensional data sets where $\mathbf{\Sigma}$ is positive semidefinite. To our knowledge, a few methods have studied the necessary optimality conditions (Sriperumbudur et al., 2011a), but none have proved optimality.

The remainder of the paper is as follows. Section 6.2 presents the minorization-maximization method and convergence analysis. In Section 6.3, we illustrate the implications of convergence conditions in a numerical setting. Finally, Section 6.4 provides a conclusion. Next, we collect our notation for the convenience of our readers.

Notation. Matrices are denoted by bold uppercase, the transpose of a matrix by the superscript $^\top$ (e.g., $\mathbf{A}^\top$), vectors by bold lowercase and scalars by lowercase italics, and we use capital letters for the last value of a running index (e.g., j running from 1 to J). Given a vector $\mathbf{x} \in \mathbb{R}^J$, its j-th entry is denoted by x_j. The $\|\mathbf{x}\|_0$ denotes the number of non-zero elements in $\mathbf{x}$. The l_1 norm is defined by $\|\mathbf{x}\|_1 = \sum_{j=1}^{J} |x_j|$, and the Euclidean norm (l_2 norm) by $\|\mathbf{x}\| = (\sum_{j=1}^{J} x_j^2)^{1/2}$. Given a matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$, its rows i and columns j are indicated by $x_{i,j}$, and $\|\mathbf{X}\|_F^2 = \sum_{i=1}^{I}\sum_{j=1}^{J} |x_{i,j}|^2$ denotes the squared Frobenius norm. $\mathbb{N}$ refers to the set of all natural numbers.

6.2 Theoretical Framework

We use a minorization-maximization (MM) scheme in Sect. 6.2.1, the solution of which is given by an iterative thresholding algorithm in Sect. 6.2.2. We present some convergence analysis in Sect. 6.2.3 and show that our method converges to a local optimum solution of the problem (6.1).

6.2.1 Minorization-Maximization (MM)

Suppose that we want to maximize the function F. The MM principle involves minorizing F by a surrogate function G. Consider an iterative algorithm that leads to a sequence $\{\mathbf{x}^t\}_{t\geq 0}$ by the following:

$$\mathbf{x}^{t+1} \in \underset{\mathbf{x}}{\operatorname{argmax}} \; G(\mathbf{x}, \mathbf{x}^t). \tag{6.2}$$

The function G minorizes the objective function F if it satisfies the following two conditions (Lange, Hunter, & Yang, 2000):

$$
\begin{aligned}
F(\mathbf{x}^t) &= G(\mathbf{x}^t, \mathbf{x}^t) \\
F(\mathbf{x}) &\geq G(\mathbf{x}, \mathbf{x}^t),
\end{aligned}
$$

which are known as the tangency condition and the domination condition, respectively.

The MM principle entails iteratively maximizing the minorizing function $G(\mathbf{x}, \mathbf{x}^{t+1})$ instead of the objective function $F(\mathbf{x})$. The solution $\mathbf{x}^{t+1}$ that maximizes $G(\mathbf{x}, \mathbf{x}^t)$ increases the objective: $F(\mathbf{x}^{t+1}) \geq F(\mathbf{x}^t)$. This is the result of the following inequalities.

$$F(\mathbf{x}^{t+1}) \geq G(\mathbf{x}^{t+1}, \mathbf{x}^t) \geq G(\mathbf{x}^t, \mathbf{x}^t) = F(\mathbf{x}^t), \tag{6.3}$$

where the first inequality is the result of the domination condition, and the second inequality holds since $G(\mathbf{x}, \mathbf{x}^t)$ is maximized at $\mathbf{x} = \mathbf{x}^{t+1}$.

The MM principle has seen success in various domains (see Nguyen (2017)). It is also relevant in the PCA setting. Whereas Sriperumbudur et al. (2011a) used an MM algorithm for the penalized PCA problem, the classical power method to solve the largest eigenvalue of a positive semidefinite matrix can also be derived from the MM perspective (Lange, 2016).

6.2.2 MM implementation for Problem (6.1)

For clarity, let us define the objective of problem (6.1) as $C(\mathbf{w}) = \mathbf{w}^\top \mathbf{\Sigma} \mathbf{w} - \alpha \|\mathbf{w}\|_0$. We propose the following minorizing function S over $\mathcal{B} \times \mathcal{B}$ as

$$S(\mathbf{w}, \mathbf{z}) = \mathbf{w}^\top \mathbf{\Sigma} \mathbf{w} - \alpha \|\mathbf{w}\|_0 - (\mathbf{w} - \mathbf{z})^\top (\mathbf{\Sigma} - \mathbf{I})(\mathbf{w} - \mathbf{z}) \tag{6.4}$$

Observe that $S(\mathbf{w}, \mathbf{z}) \leq C(\mathbf{w})$ and $S(\mathbf{w}, \mathbf{w}) = C(\mathbf{w})$ for all $\mathbf{w}, \mathbf{z} \in \mathcal{B}$. Then, the update of $\mathbf{w}$, in iteration $t+1$, is given by

$$\mathbf{w}^{t+1} \in \underset{\mathbf{w} \in \mathcal{B}}{\operatorname{argmax}} \ S(\mathbf{w}, \mathbf{w}^t), \tag{6.5}$$

and stopping when $\mathbf{w}^{t+1} = \mathbf{w}^t$.

6.2.2.1 Iterative Hard Thresholding

We now show that the update presented in Equation (6.5) is equivalent to an iterative hard-thresholding rule. Let us consider the lagrangian of problem (6.5) as

$$\begin{aligned}\mathcal{L}(\mathbf{w},\mu) =& S(\mathbf{w},\mathbf{w}^t) - \mu(\mathbf{w}^\top\mathbf{w} - 1)\\ =& \mathbf{w}^\top\boldsymbol{\Sigma}\mathbf{w} - \alpha\|\mathbf{w}\|_0 - (\mathbf{w}-\mathbf{w}^t)^\top(\boldsymbol{\Sigma}-\mathbf{I})(\mathbf{w}-\mathbf{w}^t) - \mu(\mathbf{w}^\top\mathbf{w} - 1)\\ =& \mathbf{w}^\top\mathbf{w} + 2\mathbf{w}^\top(\boldsymbol{\Sigma}-\mathbf{I})\mathbf{w}^t - \alpha\|\mathbf{w}\|_0 - \mu(\mathbf{w}^\top\mathbf{w} - 1) - \mathbf{w}^{t\top}(\boldsymbol{\Sigma}-\mathbf{I})\mathbf{w}^t\end{aligned}$$

Then, the KKT-conditions are given by:

$$\begin{aligned}(\boldsymbol{\Sigma}_j^\top\mathbf{w}^t - w_j^t) - (\mu-1)w_j = 0, &\qquad \forall j\in[J]\\ \mathbf{w}^\top\mathbf{w} \le 1&\\ \mu \ge 0&\\ \mu(\mathbf{w}^\top\mathbf{w} - 1) = 0&\end{aligned}$$

with the solution

$$\hat{\mathbf{w}} = \frac{(\boldsymbol{\Sigma}-\mathbf{I})\mathbf{w}^t}{\|(\boldsymbol{\Sigma}-\mathbf{I})\mathbf{w}^t\|}. \tag{6.6}$$

It can be observed, by replacing $\hat{\mathbf{w}}$ back in the Lagrangian and analyzing it component-wise, that the maximum is attained at $\hat{\mathbf{w}} = \frac{U_\alpha([\boldsymbol{\Sigma}-\mathbf{I}]\mathbf{w}^t)}{\|U_\alpha([\boldsymbol{\Sigma}-\mathbf{I}]\mathbf{w}^t)\|}$, where U_α is defined component-wise as

$$U_\alpha(\mathbf{y})_j = \begin{cases} 0 & \text{if } \frac{y_j^2}{\|\mathbf{y}\|} < \alpha \\ y_j & \text{if } \frac{y_j^2}{\|\mathbf{y}\|} \ge \alpha \end{cases}. \tag{6.7}$$

We propose the algorithm 6.1 to find an optimal solution to problem (6.1).

Algorithm 6.1 Iterative hard thresholding

```
Inputs:
    $\boldsymbol{\Sigma}, \mathbf{w}_0$
Outputs:
    $\mathbf{w}^*$

while $\mathbf{w}^{t+1} \neq \mathbf{w}^t$ do
    $\mathbf{w}^{t+1} = \frac{U_\alpha([\boldsymbol{\Sigma}-\mathbf{I}]\mathbf{w}^t)}{\|U_\alpha([\boldsymbol{\Sigma}-\mathbf{I}]\mathbf{w}^t)\|}$
end while
```

6.2.3 Convergence Analysis

We now conduct a convergence analysis of the solution achieved using the MM scheme in Equation (6.5). We begin by showing in Lemma 6.2.1 that the sequence generated by Algorithm 6.1 increases and converges in value.

Lemma 6.2.1. *Let* $\mathbf{w}^0 \in \mathcal{B}$. *Let* $\{\mathbf{w}^t\}_{t\geq 1}$ *be the sequence generated using the MM scheme in Equation* (6.5) *starting at* $\mathbf{w}^0$. *Then* $\lim_{t\to\infty} S(\mathbf{w}^t, \mathbf{w}^{t-1})$ *and* $\lim_{t\to\infty} C(\mathbf{w}^t)$ *exist.*

Proof. By the definition of C and S, we have the following:

$$\begin{aligned} C(\mathbf{w}^{t+1}) \geq & C(\mathbf{w}^{t+1}) - \|(\mathbf{\Sigma} - \mathbf{I})^{1/2}(\mathbf{w}^{t+1} - \mathbf{w}^t)\|^2 \\ = & S(\mathbf{w}^{t+1}, \mathbf{w}^t) \\ \geq & S(\mathbf{w}^t, \mathbf{w}^t) \\ = & C(\mathbf{w}^t) \\ \geq & S(\mathbf{w}^t, \mathbf{w}^{t-1}), \end{aligned}$$

where the second inequality is due to the update formula in Equation (6.5), and the last inequality follows the same reasoning as the first equality. Therefore, the sequences $\{S(\mathbf{w}^{t+1}, \mathbf{w}^t)\}_{t\geq 1}$ and $\{C(\mathbf{w}^t)\}_{t\geq 1}$ do not decrease. Additionally, these sequences are bounded above by $\{\max\ \mathbf{w}^\top \mathbf{\Sigma} \mathbf{w}$ s.t. $\mathbf{w} \in \mathcal{B}\}$, the maximum eigenvalue of the matrix $\mathbf{\Sigma}$. This implies the desired result. □

Given the relation $S(\mathbf{w}, \mathbf{w}) = C(\mathbf{w})$, it is natural in the proposed MM scheme to stop when $\mathbf{w}^{t+1} = \mathbf{w}^t$. In Lemma 6.2.2, we show the sufficient condition to guarantee that the use of the MM scheme in Equation (6.5) converges and meets this stopping criterion $\mathbf{w}^{t+1} = \mathbf{w}^t$.

Lemma 6.2.2. *Let* $\mathbf{\Sigma}$ *be such that its minimum eigenvalue is greater than* 1. *Let* $\mathbf{w}^0 \in \mathcal{B}$ *and* $\{\mathbf{w}^t\}_{t\geq 1}$ *be the sequence generated using the MM scheme in Equation* (6.5) *starting at* $\mathbf{w}^0$. *Then,* $\lim_{t\to\infty} \|\mathbf{w}^{t+1} - \mathbf{w}^t\| = 0$.

Proof. Let $\sigma_{min} > 1$ be the minimum eigenvalue of the matrix $\mathbf{\Sigma}$, and $C^* = \lim_{t\to\infty} C(\mathbf{w}^t)$. To show this lemma, we show that the series $\sum_{t=1}^{\infty} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2$ is bounded. To show boundedness, we use that $0 < \sigma_{min} - 1 \leq \frac{\|(\mathbf{\Sigma}-\mathbf{I})^{1/2}(\mathbf{w}^{t+1}-\mathbf{w}^t)\|^2}{\|(\mathbf{w}^{t+1}-\mathbf{w}^t)\|^2}$ for all t. This implies that

$$\|(\mathbf{w}^{t+1} - \mathbf{w}^t)\|^2 \leq \frac{1}{\sigma_{min} - 1}\|(\mathbf{\Sigma} - \mathbf{I})^{1/2}(\mathbf{w}^{t+1} - \mathbf{w}^t)\|^2 \leq \frac{1}{\sigma_{min} - 1}[C(\mathbf{w}^{t+1}) - C(\mathbf{w}^t)].$$

The last inequality comes from the inequalities in the proof of Lemma 6.2.1. Summing up both sides of the previous inequality over t, we have

$$\sum_{t=1}^{\infty} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \leq \frac{1}{\sigma_{min} - 1}[C^* - C(\mathbf{w}^0)]$$

which proves the desired result. □

The main assumption on Lemma 6.2.2 is that $\sigma_{min} > 1$. This assumption seems unrealistic in practice, especially when dealing with high-dimensional data where the matrix $\mathbf{\Sigma}$ is positive semidefinite and thus $\sigma_{min} = 0$. Nevertheless, this complication can be circumvented by implementing Algorithm 6.1 using $\hat{\mathbf{\Sigma}} = \mathbf{\Sigma} + \tau\mathbf{I}$ instead of $\mathbf{\Sigma}$, which has always $\hat{\sigma}_{min} > 1$ when $\tau > 1$. It can be easily observed that when $\mathbf{w}^\top\mathbf{w} = 1$, solving problem (6.1) using $\hat{\mathbf{\Sigma}}$ is equivalent to use $\mathbf{\Sigma}$ as follows.

$$\begin{aligned}
\mathbf{w}^* \in & \underset{\mathbf{w}\in\mathcal{B}}{\operatorname{argmax}}\ \mathbf{w}^\top\mathbf{\Sigma}\mathbf{w} - \alpha\|\mathbf{w}\|_0 + \tau 1 \\
\Leftrightarrow & \underset{\mathbf{w}\in\mathcal{B}}{\operatorname{argmax}}\ \mathbf{w}^\top\mathbf{\Sigma}\mathbf{w} - \alpha\|\mathbf{w}\|_0 + \tau\mathbf{w}^\top\mathbf{w} \\
\Leftrightarrow & \underset{\mathbf{w}\in\mathcal{B}}{\operatorname{argmax}}\ \mathbf{w}^\top(\mathbf{\Sigma} + \tau\mathbf{I})\mathbf{w} - \alpha\|\mathbf{w}\|_0.
\end{aligned} \tag{6.8}$$

This easy 'trick' is frequently used to guarantee that $\mathbf{\Sigma}$ is convex by shifting the eigenvalues to be positive (Journée et al., 2010; G. X. Yuan, Ho, & Lin, 2011).

Let the support $\operatorname{supp}(\mathbf{w}) \equiv \{j | w_j \neq 0\}$ be the set of indexes with a nonzero element in $\mathbf{w}$. Lemma 6.2.2 implies that the support of the sequence generated using Algorithm 6.1 stabilizes, that is, it is the same after some N. This is stated in Corollary 6.2.3.

Corollary 6.2.3. *Let $\mathbf{\Sigma}$ be such that its minimum eigenvalue is greater than 1. Let $\mathbf{w}^0 \in \mathcal{B}$ and $\{\mathbf{w}^t\}_{t\geq 1}$ be the sequence generated using the MM scheme in Equation* (6.5) *starting at $\mathbf{w}^0$. Then there exists $N \in \mathbb{N}$ such that, for all $t > N$,* $\operatorname{supp}(\mathbf{w}^{t+1}) = \operatorname{supp}(\mathbf{w}^t)$.

Proof. Let $\sigma_{max} > 1$ be the maximum eigenvalue of the matrix $\mathbf{\Sigma}$. If $w_j^t \neq 0$, we have from Equation (6.7) that

$$\begin{aligned}
{w_j^t}^2 = \frac{(U_\alpha([\mathbf{\Sigma} - \mathbf{I}]_j^\top \mathbf{w}^{t-1}))^2}{\|U_\alpha([\mathbf{\Sigma} - \mathbf{I}]\mathbf{w}^{t-1})\|^2} = \frac{([\mathbf{\Sigma} - \mathbf{I}]_j^\top \mathbf{w}^{t-1})^2}{\|U_\alpha([\mathbf{\Sigma} - \mathbf{I}]\mathbf{w}^{t-1})\|^2} \geq & \frac{([\mathbf{\Sigma} - \mathbf{I}]_j^\top \mathbf{w}^{t-1})^2}{\|[\mathbf{\Sigma} - \mathbf{I}]\mathbf{w}^{t-1}\|^2} \\
{w_j^t}^2\|[\mathbf{\Sigma} - \mathbf{I}]\mathbf{w}^{t-1}\| \geq & \frac{([\mathbf{\Sigma} - \mathbf{I}]_j^\top \mathbf{w}^{t-1})^2}{\|[\mathbf{\Sigma} - \mathbf{I}]\mathbf{w}^{t-1}\|} \geq \alpha \\
{w_j^t}^2(\sigma_{max} - 1) \geq & \alpha \\
{w_j^t}^2 \geq & \frac{\alpha}{\sigma_{max} - 1}
\end{aligned} \tag{6.9}$$

Now, let us consider any ϵ such that $0 < \epsilon < \alpha/(\sigma_{max} - 1)$. From Lemma 6.2.2, there exists $N \in \mathbb{N}$ such that for any $t > N$, $\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \leq \epsilon$. If $\operatorname{supp}(\mathbf{w}^{t+1}) \neq \operatorname{supp}(\mathbf{w}^t)$, there exists $j \in \operatorname{supp}(\mathbf{w}^t \setminus \operatorname{supp}(\mathbf{w}^{t+1}))$, which implies that $\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \geq \frac{\alpha}{\sigma_{max}-1}$ from Equation (6.9). This is a contradiction. □

From Corollary 6.2.3 and Equation (6.6), it can be observed that when the support stabilizes, algorithm 6.1 is equivalent to applying the Power method on the matrix $\mathbf{\Sigma} - \mathbf{I}$. Then, the desired result follows. We use this to show that the solution provided by Algorithm 6.1 is a local optimum of problem (6.1). This is stated in Theorem 6.2.5.

6.2.3.1 Local Optimizer

To finalize this section, we show that any solution obtained from Algorithm 6.1 is a local optimum of problem (6.1).

Proposition 6.2.4. *Let $\mathbf{\Sigma}$ be such that its minimum eigenvalue is greater than* 1. *Let $\mathbf{w}^0 \in \mathcal{B}$ and $\{\mathbf{w}^t\}_{t\geq 1}$ be the sequence generated using the MM scheme in Equation* (6.5) *starting at $\mathbf{w}^0$ and ending at $\mathbf{w}^*$. Let $\mathbf{d} \in \mathcal{B}$ be a feasible direction of problem* (6.1). *Then* $\operatorname{supp}(\mathbf{w}^*) \subseteq \operatorname{supp}(\mathbf{w}^* + \delta\mathbf{d})$ *for any* $0 < \delta < \sqrt{\alpha/(\sigma_{max} - 1)}$ *with σ_{max} the maximum eigenvalue of $\mathbf{\Sigma}$.*

Proof. Let us consider $j \in supp(\mathbf{w}^*) \setminus \operatorname{supp}(\mathbf{w}^* + \delta\mathbf{d})$. Let us take any $0 < \delta < \sqrt{\alpha/(\sigma_{max} - 1)}$. Then, it follows that

$$\delta^2 = \|\mathbf{w}^* + \delta\mathbf{d} - \mathbf{w}^*\|^2 \geq |w_j^* + \delta d_j - w_j^*|^2 = |w_j^*|^2 \geq \frac{\alpha}{\sigma_{max} - 1}.$$

The second equality comes from the assumption that $j \in supp(\mathbf{w}^*) \setminus \operatorname{supp}(\mathbf{w}^* + \delta\mathbf{d})$ and the last inequality from Equation (6.9). Therefore, there is no $j \in supp(\mathbf{w}^*) \setminus \operatorname{supp}(\mathbf{w}^* + \delta\mathbf{d})$, which implies the desired result. □

Theorem 6.2.5. *Let $\mathbf{\Sigma}$ be such that its minimum eigenvalue is greater than* 1. *Let $\mathbf{w}^0 \in \mathcal{B}$ and $\{\mathbf{w}^t\}_{t\geq 1}$ be the sequence generated using the MM scheme in Equation* (6.5) *starting at $\mathbf{w}^0$ and ending at $\mathbf{w}^*$. There exists $\delta > 0$ such that*

$$C(\mathbf{w}^*) \geq C(\mathbf{w}^* + \delta\mathbf{d})$$

for any feasible direction $\mathbf{d} \in \mathcal{B}$.

Proof. Let σ_{max} be the maximum eigenvalue of the matrix $\mathbf{\Sigma}$. Let us consider any δ such that $0 < \delta < \sqrt{\alpha/(\sigma_{max} - 1)}$. From Proposition 6.2.4, it holds that $\operatorname{supp}(\mathbf{w}^*) \subseteq \operatorname{supp}(\mathbf{w}^* + \delta\mathbf{d})$. If $\operatorname{supp}(\mathbf{w}^*) = \operatorname{supp}(\mathbf{w}^* + \delta\mathbf{d})$, $\mathbf{w}^*$ is the solution of the Power method when the support stabilizes (see Corollary 6.2.3). Then, it is a global optimum, and the result follows.

Now, if $\text{supp}(\mathbf{w}^*) \subset \text{supp}(\mathbf{w}^* + \delta\mathbf{d})$, it holds that $\|\mathbf{w}^* + \delta\mathbf{d}\|_0 > \|\mathbf{w}^*\|_0$. Then, taking

$\delta^2\sigma_{max} + 2\delta\sigma_{max} \leq c$ and $c = \alpha(\|\mathbf{w}^* + \delta\mathbf{d}\|_0 - \|\mathbf{w}^*\|_0)$, we have that

$$\begin{aligned}
\alpha(\|\mathbf{w}^* + \delta\mathbf{d}\|_0 - \|\mathbf{w}^*\|_0) \geq& \delta^2\mathbf{d}^\top\boldsymbol{\Sigma}\mathbf{d} + 2\delta\mathbf{d}^\top\boldsymbol{\Sigma}\mathbf{w}^* \\
\mathbf{w}^{*\top}\boldsymbol{\Sigma}\mathbf{w}^* + \alpha(\|\mathbf{w}^* + \delta\mathbf{d}\|_0 - \|\mathbf{w}^*\|_0) \geq& \mathbf{w}^{*\top}\boldsymbol{\Sigma}\mathbf{w}^* + \delta^2\mathbf{d}^\top\boldsymbol{\Sigma}\mathbf{d} + 2\delta\mathbf{d}^\top\boldsymbol{\Sigma}\mathbf{w}^* \\
\mathbf{w}^{*\top}\boldsymbol{\Sigma}\mathbf{w}^* - \alpha\|\mathbf{w}^*\|_0 \geq& \mathbf{w}^{*\top}\boldsymbol{\Sigma}\mathbf{w}^* + \delta^2\mathbf{d}^\top\boldsymbol{\Sigma}\mathbf{d} + 2\delta\mathbf{d}^\top\boldsymbol{\Sigma}\mathbf{w}^* - \alpha\|\mathbf{w}^* + \delta\mathbf{d}\|_0 \\
\mathbf{w}^{*\top}\boldsymbol{\Sigma}\mathbf{w}^* - \alpha\|\mathbf{w}^*\|_0 \geq& (\mathbf{w}^* + \delta\mathbf{d})^\top\boldsymbol{\Sigma}(\mathbf{w}^* + \delta\mathbf{d}) - \alpha\|\mathbf{w}^* + \delta\mathbf{d}\|_0 \\
C(\mathbf{w}^*) \geq& C(\mathbf{w}^* + \delta\mathbf{d})
\end{aligned}$$

The second inequality is due to the Cauchy–Schwarz inequality:

$$\mathbf{d}^\top\boldsymbol{\Sigma}\mathbf{w}^* = (\mathbf{X}\mathbf{d})^\top(\mathbf{X}\mathbf{w}^*) \leq \|\mathbf{X}\mathbf{d}\|\|\mathbf{X}\mathbf{w}^*\| \leq \sigma_{max},$$

In both cases, we show that δ exists. □

6.3 Numerical Examples

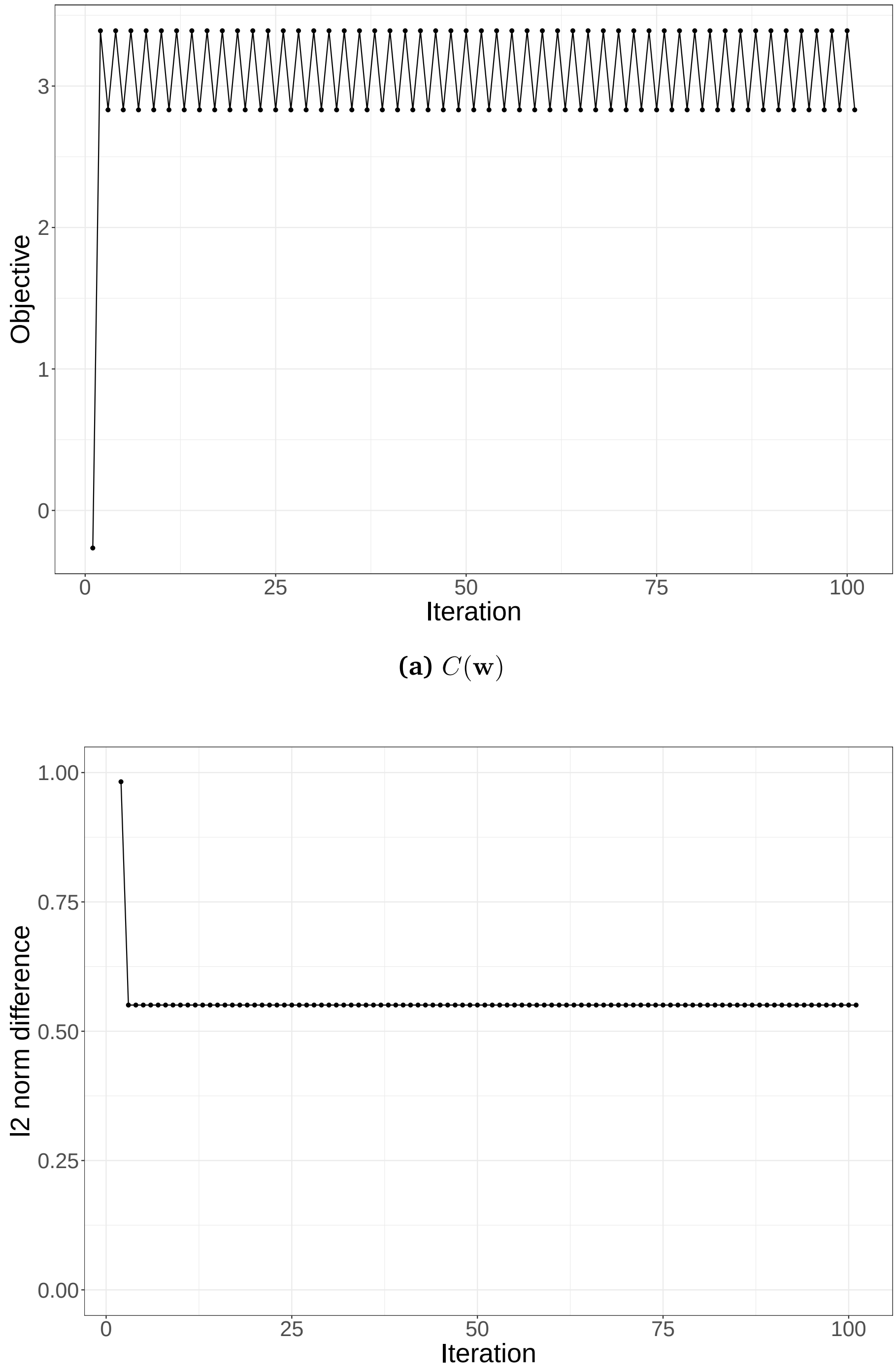
In Lemma 6.2.2, we showed that the step size in Algorithm 6.1 converges when the minimum eigenvalue (σ_{min}) of the matrix $\boldsymbol{\Sigma}$ is larger than 1. Here we illustrate this finding by administering our method on a simulated dataset and an empirical dataset, both for which the condition is not satisfied. We illustrate that Algorithm 6.1 does not converge for these specific datasets, and how the divergence problem can be overcome by the aforementioned transformation of data in Equation (6.8).

6.3.1 Synthetic Dataset

By relying on the eigenvalue decomposition, we generated a $\boldsymbol{\Sigma}$ matrix from one eigenvector with a defined sparse structure:

$$\begin{array}{ccccc} \begin{bmatrix} -0.302 \\ 0 \\ 0 \\ 0.302 \\ -0.905 \end{bmatrix} & \begin{bmatrix} 5 \end{bmatrix} & \begin{bmatrix} -0.302 \\ 0 \\ 0 \\ 0.302 \\ -0.905 \end{bmatrix}^\top & = & \begin{bmatrix} 0.455 & 0 & 0 & -0.455 & 1.364 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.455 & 0 & 0 & 0.455 & -1.364 \\ 1.364 & 0 & 0 & -1.364 & 4.091 \end{bmatrix} \\ \mathbf{v}_1 & \sigma_1 & \mathbf{v}_1^\top & & \mathbf{\Sigma} \end{array} \tag{6.10}$$

By defining only the first eigenvalue ($\sigma_1 = 5$), the remaining 4 eigenvalues are defined as zero. Therefore, the smallest eigenvalue of the matrix $\mathbf{\Sigma}$ would be $\sigma_{min} = 0$. Then, implementing Algorithm 6.1, with penalty parameter $\alpha = 0.7$, in this particular setting the sequence diverges; see Figure 6.1.

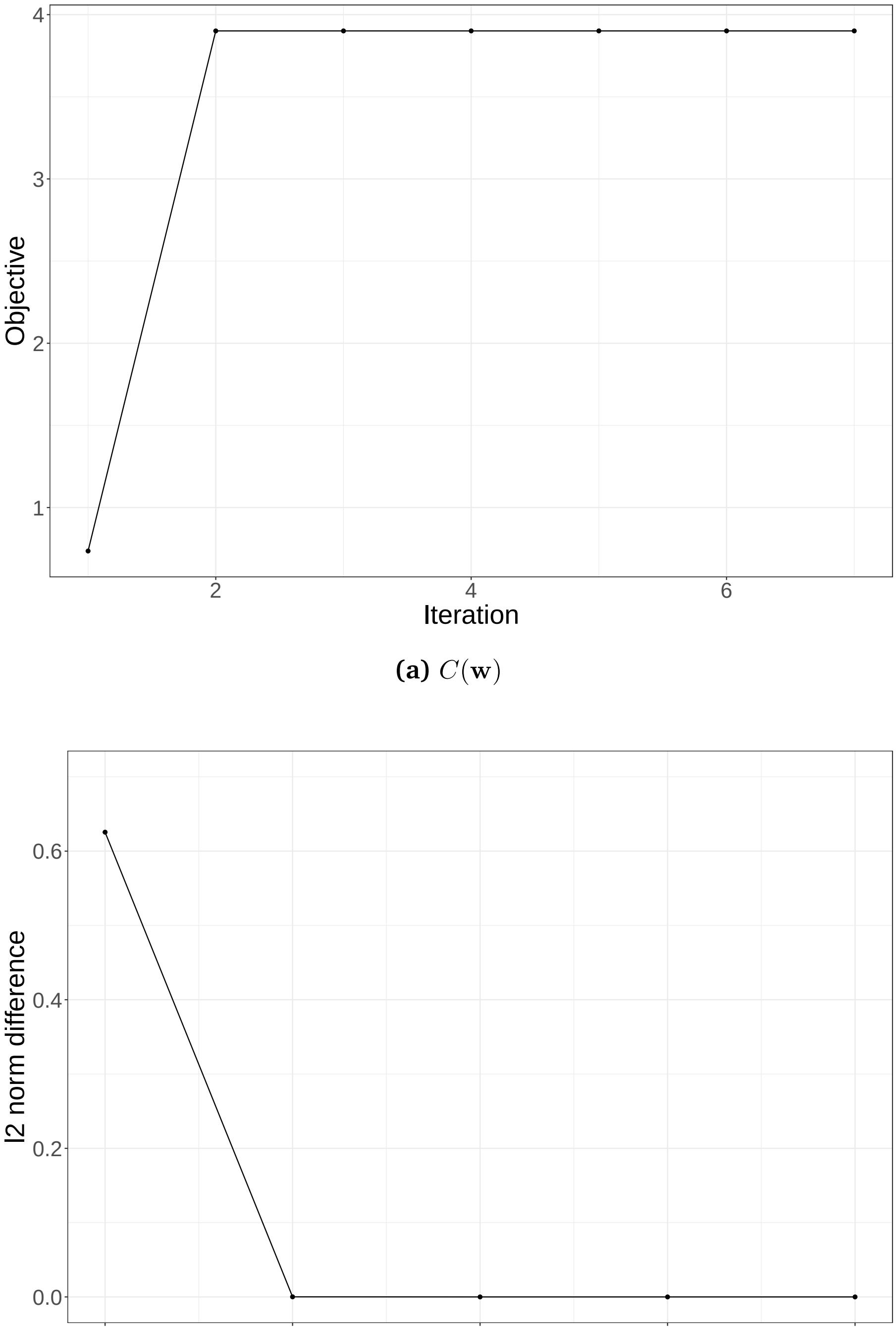


(a) $C(\mathbf{w})$

(b) $\left\|\mathbf{w}^t - \mathbf{w}^{t+1}\right\|$

Figure 6.1. Divergence for the simulated $\mathbf{\Sigma}$. The objective function is displayed in (a), while (b) shows the l_2 norm of the difference between the iterates.

Now, we illustrate that with a transformed matrix $\hat{\mathbf{\Sigma}} = \mathbf{\Sigma} + \tau\mathbf{I}$, with $\tau > 1$, Algorithm 6.1 converges (Figure 6.2) under the same set of parameters. Note that the accumulation point $\mathbf{w}^*$ in this case is also identical to the defined eigenvector $\mathbf{v}_1$ (Figure 6.3).



(a) $C(\mathbf{w})$

(b) $\left\|\mathbf{w}^t - \mathbf{w}^{t+1}\right\|$

Figure 6.2. Convergence with the transformation on the simulated data

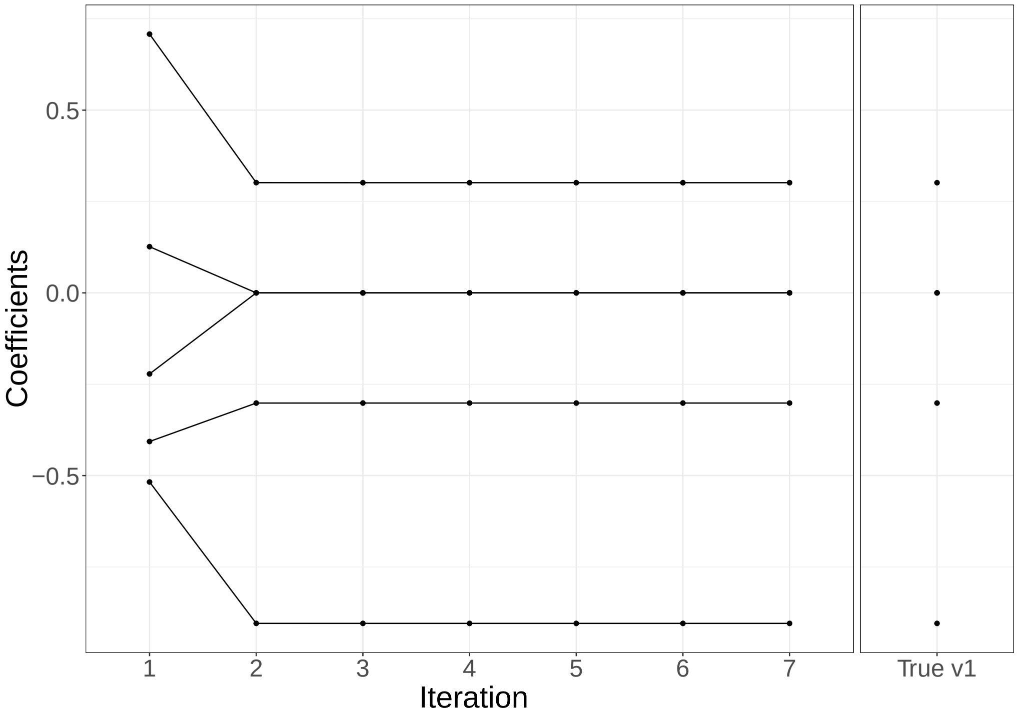
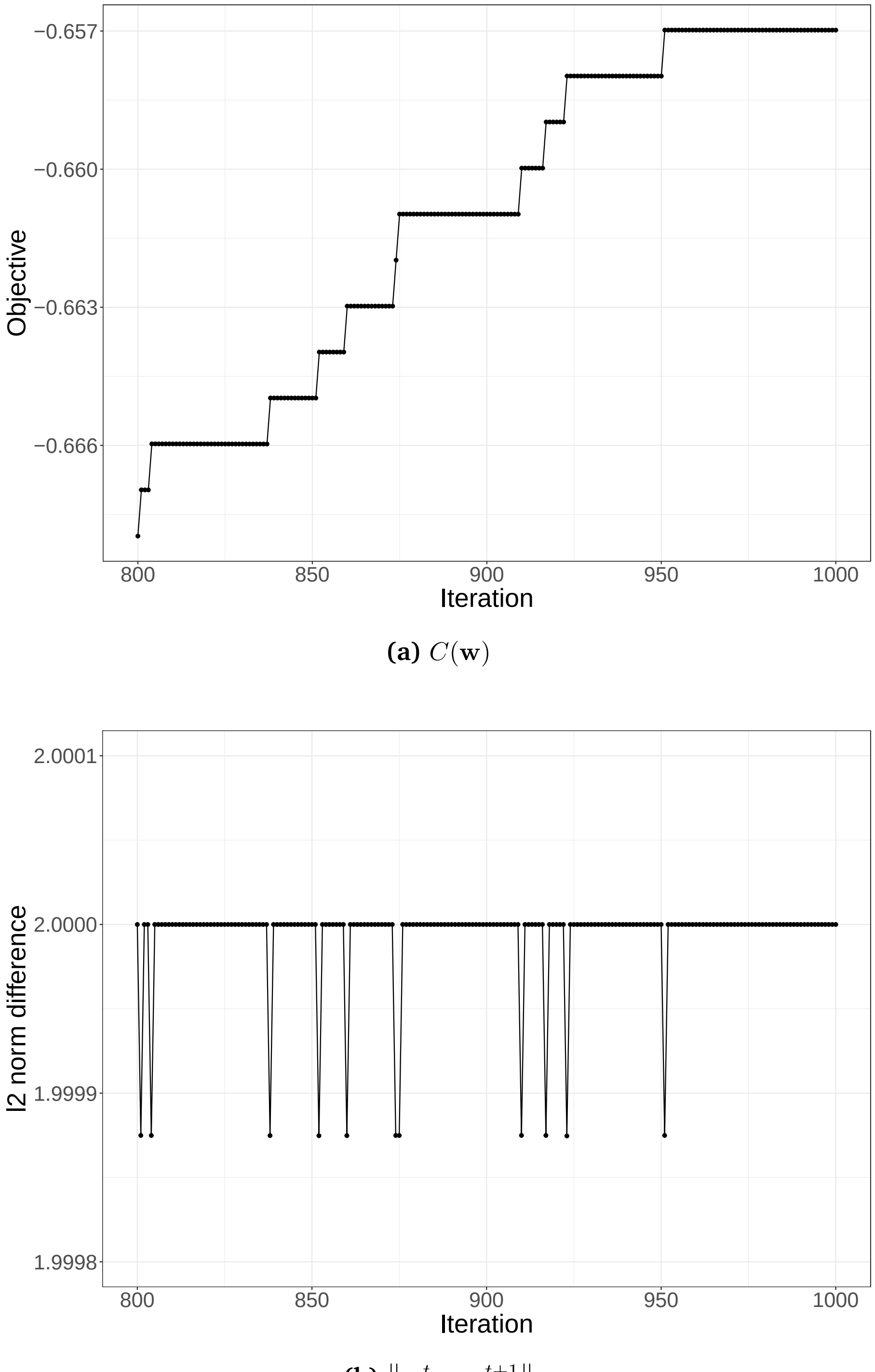


Figure 6.3. Series of $\mathbf{w}^t$ compared to the true eigenvector $\mathbf{v}_1$. Each line in the left panel represents each element of $\mathbf{w}^t$.

6.3.2 Empirical Dataset

We imported the '16S data', which relate to microbiomes in the human body. It refers to measurements from three different regions of the body (namely, oral, skin, and stool) that present the greatest diversity in the microbial community. The dataset is characterized by 1674 measurements from 162 observation units. We imported the dataset from the R package 'mixOmics' (Rohart, Gautier, Singh, & Cao, 2017).

We perform the eigenvalue decomposition on the $\mathbf{\Sigma}$ matrix, which results in $\sigma_{max} = 0.603 < 1$. This implies $\sigma_{min} < 1$. With the penalty parameter $\alpha = 0.001$, we found that Algorithm 6.1 did not converge in 100000 iterations; Figure 6.4 shows the objective in problem (6.1) in iterations from $t = 800$ to $t = 1000$. The plots corresponding to the complete set of the first 1000 iterations are provided in the Appendix 6.A (Figure 6.6).

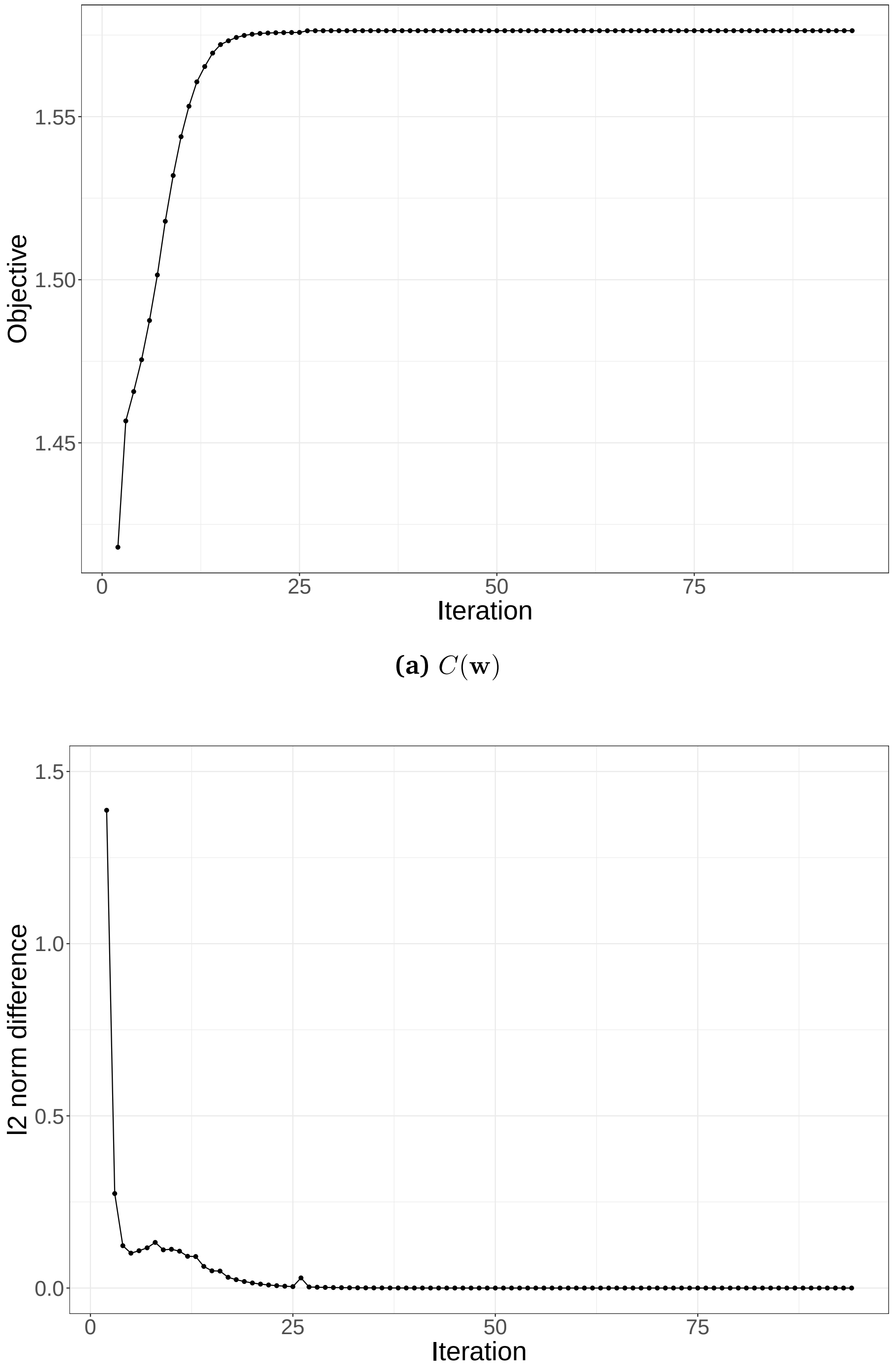


(a) $C(\mathbf{w})$

(b) $\left\|\mathbf{w}^{t} - \mathbf{w}^{t+1}\right\|$

Figure 6.4. Divergence for the 16S data. The plots portray the method from iteration 800 to iteration 1000.

Although Figures 6.4 and 6.6 show that the objective $C(\mathbf{w})$ continues to increase, it can be seen that the l_2 norm between the solutions does not decrease over iterations. On the other hand, the sequence converges successfully when administered to the transformed matrix $\hat{\mathbf{\Sigma}}$, with the same initial vector and penalty parameter. Figure 6.5 shows that convergence is achieved in 96 iterations.



(a) $C(\mathbf{w})$

(b) $\left\|\mathbf{w}^t - \mathbf{w}^{t+1}\right\|$

Figure 6.5. Convergence with the transformation on the 16S data

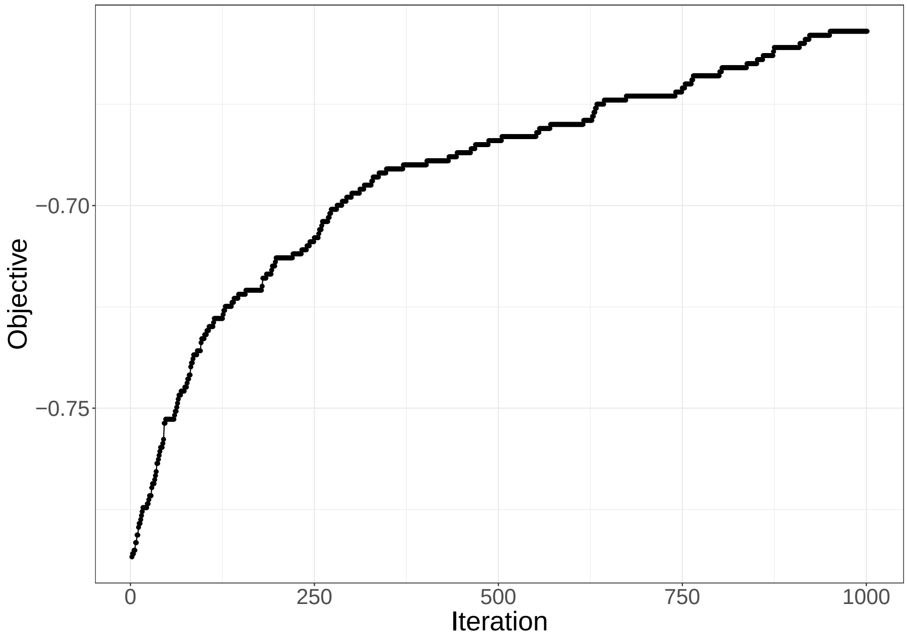
6.4 Conclusion

In this paper, we considered the penalized PCA with the l_0 norm as a sparsity-inducing penalty and proposed a minorization-maximization (MM) scheme that achieves a locally optimal solution to the penalized PCA problem. Although some previous work has proposed methods that meet the necessary optimality conditions (Guerra-Urzola, Van Deun, Vera, & Sijtsma, 2023; Sriperumbudur, Torres, & Lanckriet, 2011b), this is the first to prove optimality in the context of penalized PCA. Based on the MM principle, we derived an iterative method that has convergence guarantees under the condition that the minimum eigenvalue of the covariance matrix is greater than one. We also proposed a simple transformation of the covariance matrix that ensures the condition, illustrating the practical implications of the condition by the use of a synthetic and empirical dataset.

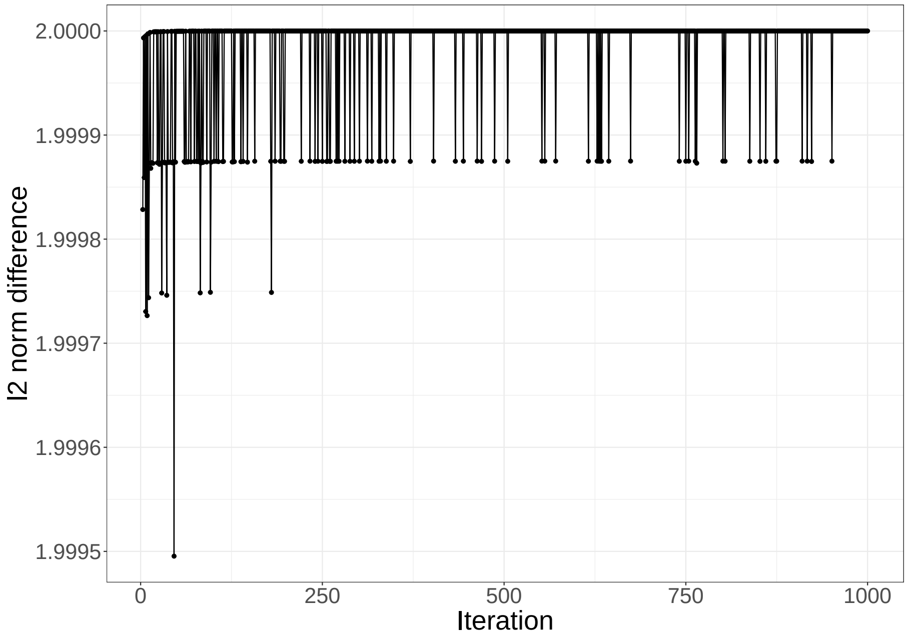
For future work, it would be worthwhile to study the optimality conditions in using other types of penalties. It would be along the lines of the work of Guerra-Urzola et al. (2023) that provided the necessary optimality conditions for a general form of penalty within penalized PCA. Additionally, the condition regarding the minimum eigenvalue of the covariance matrix would also be an interesting topic of research. As the same condition was also found for an alternating method Guerra-Urzola et al. (2023), which is different from our approach, it appears that the condition may be applicable to penalized PCA problems in general.

Appendix

6.A 16S data: divergence plots for 1000 iterations



(a) $C(\mathbf{w})$



(b) $\left\|\mathbf{w}^t - \mathbf{w}^{t+1}\right\|$

Figure 6.6. Divergence for the 16S data

Chapter 7

Discussion

This chapter provides a review on the research conducted within the dissertation. It starts with an overview of the dissertation, followed by a discussion of practical issues to consider when using the proposed methods. Thereafter, drawbacks concerning model selection procedures and data generation strategies employed throughout the dissertation are presented. Lastly, future directions for further extending the PCovR methods with respect to computational feasibility are suggested.

7.1 Overview

Part I of this dissertation presented adaptations of PCovR suitable for large and high-dimensional data from multiple sources. In Chapter 2, we introduced a regression method (SCD-CovR) that effectively represents the predictive common and distinctive processes by a sparse and interpretable covariate model. Its recovery of the predictive processes was better than a preceding PCovR method that does not account for the multiblock data setup. At the same time, compared to the multiblock extension of PLS, SCD-CovR performed better at prediction of the outcome. By fusing with logistic regression, the method was further expanded to address a classification problem in Chapter 3 (SCD-Cov-logR). Our method was found to be substantially better than a classifier based on PLS at both classification of the outcome and capturing of relevant predictive processes. Whereas SCD-CovR identified the common and distinctive covariates by imposing the zero block constraints which inflates the computational load, the group lasso penalty was employed in SCD-Cov-logR instead to find the two types of covariates. While the first two chapters only addressed a single outcome variable, we looked towards a data problem also comprised with multiple outcome variables in Chapter 4. Another PCovR extension (SMPCovR) that filters out outcome variables which cannot be adequately predicted by the available predictor variables was put forward therein. In the presence of such outcome variables, the prediction quality of our novel method was found to be better than previous methods based on PCovR and PLS that do not exclude redundant outcome variables. SMPCovR serves as one of the first tools that performs variable selection for predictor and outcome variables simultaneously. In Part II, we zoomed into problems within sparse PCA in pursuit of potential directions of sparse PCovR research. Due to the close link shared between PCA and PCovR, findings in this part of the dissertation have meaningful implications for PCovR extensions proposed in Part I. Chapter 5 addressed issues concerning the consequences of imposing sparsity - either on the loadings or the weights - in the PCA problem. We suggested that existing experiments on sparse PCA have been incomprehensive. Alongside, we found that PCA with sparse weights is more unstable in identifying the true underlying sparse structure than PCA with sparse loadings, which is a finding directly applicable to sparse PCovR methods. Lastly, an iterative algorithm that solves a sparse PCA problem with guarantees for local optimality was introduced in Chapter 6. To our knowledge, it is the first work that proves local optimality for the sparse PCA formulation at hand. The method presented in Chapter 6 entails a different algorithmic approach from the methods in Part I. The sparse PCA problem in Chapter 6 is a maximization problem tackled by the minorization-maximization principle which results in an iterative thresholding algorithm to find the sparse solution. On the other hand, the objective criteria considered in Part I are minimization problems approached by alternating least squares and eventually the sparse solutions are obtained by coordinate descent algorithms.

7.2 Practical considerations

While the effectiveness of the proposed methods at addressing the challenges of predictive modelling with high-dimensional multiblock data has been shown in the chapters in Part I, there are data settings in practice under which the methods are not expected to be well-behaved or functional. In this section, we provide practical guidelines concerning data in using the methods.

7.2.1 Levels of measurement

The methods cater for both continuous and categorical outcome variables. Chapters 2 and 3 have targeted the two types of measurement, respectively. At the moment, the multivariate outcome setting can only be tackled with continuous outcome variables (Chapter 4). However, extending the classification method in

Chapter 3 to a setting with multiple outcome variables would be a straightforward step.

With respect to the predictor variables, the methods in this dissertation only accommodate for continuous variables. Nevertheless, categorical predictors can be dummy-coded into a indicator matrices. This is the principle behind multiple correspondence analysis (Abdi & Valentin, 2007) and PCAmix (Chavent, Kuentz-Simonet, Labenne, & Saracco, 2014), which are generalizations of PCA for categorical variables and for both categorical and continuous variables, respectively. In this case of using indicator matrices, each column would represent a class within a categorical variable. Therefore, a group lasso penalty can be imposed on the entire set of columns pertaining to the variable, instead of the lasso penalty which would filter a single column.

7.2.2 Number of blocks

While the methods in Chapters 2 and 3 target multiple blocks of predictor variables, Chapter 4 that takes account of multiple outcome variables only aims for a single block of predictors. However, the extension to multiple predictor blocks is a simple step to include a group lasso penalty to the weights. The setting with multiple blocks of outcome variables has not been considered in this dissertation, and it may be an interesting future direction.

The methods can account for a number of predictor blocks that ranges from one to many. In the case with more than two blocks, covariates in relation to a single block would be referred to as being distinctive, while those associated with predictor variables from multiple but not all blocks would be defined as being “locally common”, and lastly covariates that are linked with predictors from all of the blocks would be known as “globally common”. These terminologies were proposed by Måge et al. (2019).

7.2.3 Dimensionality

The numbers of variables and observations impact the efficacy of the methods with regards to how well the true population parameters are recovered by the estimated coefficients. The methods in this dissertation were not examined in an asymptotic context where the number of variables grows towards infinity while the number of observations is fixed, and vice versa. However, there are insights from existing studies on PCA and sparse PCA that can be borrowed to infer about the asymptotic properties of the our methods. These studies employ the concept of “consistency” which represents how close the population parameters and the estimated coefficients are; when the coefficients perfectly reflect the population parameters, the method is considered consistent.

D. Shen et al. (2016) have established a framework of asymptotic properties of PCA that encompassees previous asymptotic results on PCA. They reported that PCA is consistent if (a) the dataset is in low dimensionality or (b) the variance of the first component in population is considerably larger than the following components (even if the dataset is high-dimensional). Asymptotic studies on sparse PCA extend these conditions for consistency. On top of being consistent for the conditions above, it was found that if the number of non-zero population parameters is small, sparse PCA is consistent even if the dataset is high-dimensional and the difference in variance between the first and the following components is small (D. Shen, Shen, & Marron, 2013). To be concrete, a small number of non-zero population parameters refers to a small number of true population PCA weights that are non-zero, which can be examplified by a setting where only a few predictor variables have true linkages with a component.

Although the PCovR extensions in this dissertation are closely related with PCA and sparse PCA, there are issues to be considered when inferring about our methods from these asymptotic results. The literature on asymptotic properties of sparse PCA is focussed on sparse PCA with sparse loadings, while the methods in this dissertation takes an approach with sparse weights. As highlighted in Chapter 5, imposing sparsity on the loadings as opposed to the weights has consequences; one of which is that sparse weights PCA and sparse loadings PCA are disparate methods that derive different results. Computation of sparse weights is a regression problem often under high dimensionality unlike the problem for sparse loadings which is univariate (or low-dimensional). However, an asymptotic study on elastic net regression (used to solve for sparse weights for Chapters 2 and 4) reported that the estimates can be consistent under high dimensionality if the number of observations is large enough in relation to the number of non-zero population parameters (Jia & Yu, 2010).

To sum up, the performance of the methods in this dissertation in recovering the population parameters from high-dimensional data is expected to improve with (a) diminishing number of non-zero population parameters, (b) growing difference in the variance of the first population component as opposed to the following components, (c) growing number of observations and (d) diminishing number of variables. While (a) and (b) seem to have bigger roles, information regarding these aspects is rarely available in practice. Yet, we believe that these studies inform us that there are high-dimensional settings where the methods in this dissertation are anticipated to show good recovery of population parameters. Nevertheless, it should be noted that a clear picture of consistency of our PCovR extensions will only be attained with an asymptotic investigation.

7.3 Drawbacks

7.3.1 Model selection

The PCovR extensions presented in the current dissertation involve many model parameters. Taking the example of Sparse Common and Distinctive Covariates Logistic Regression (SCD-Cov-logR) presented in Chapter 3, there are five different types of parameters to be tuned: number of covariates, lasso and group lasso parameters for weights and ridge parameter for logistic regression coefficients. For each type of parameter, a range of different values must be considered, leading to a very large number of models to be evaluated via cross-validation. This entails a heavy burden of computation. In this dissertation, we employed two strategies as a possible remedy. Firstly, in Chapters 2 and 3, the computational load was reduced by relying on the sequential model selection procedure. Instead of conducting the cross-validation on the exhaustive set composed of all combinations of the model parameters, the parameters were tuned in turn, while fixing the remaining parameters at constant. Having been recommended as a viable approach for PCovR in the previous literature (Vervloet et al., 2016), it has also performed well in our experiments. Secondly, in Chapter 4, in addition to the sequential procedure to first select the number of covariates, we fixed the ridge parameters at a small near-zero value, instead of tuning them. The total number of model parameters considered for model selection was therefore reduced, further decreasing the computational intensiveness. We chose to fix the ridge parameters considering that the role of ridge penalty is to prevent overfitting and divergence, rather than shaping the model structure. This model selection strategy also showed good recovery of the true underlying structures. However, both of these strategies present a risk to miss the optimal model, since they are based on the rationale of not employing the entire set of possible models given the parameter ranges. These strategies can be considered as decisions made amidst an inevitable trade-off between computational load and optimal tuning of the model.

Another weakness with regards to model selection present in this dissertation is that cross-validation is the only method of model selection employed. While there are many other model selection tools applicable to the PCovR methods we proposed, a notable strategy is index of sparseness (Gajjar, Kulahci, & Palazoglu, 2017; Trendafilov, 2014). It jointly takes into account of in-sample model fit and level of sparsity in selecting a model. A big advantage of index of sparseness is that it is not computationally intensive. Unlike n-fold cross-validation that requires n repetitions of estimation per model configuration, the model has to be estimated only once for index of sparseness. Despite this strength, previous research on sparse multiblock component analysis methods found mixed results. Whereas Gu, Schipper, and Van Deun (2019) compared several model selection strategies and concluded with the recommendation for index of sparseness, the component analysis method they focused on imposed the sparsity on loadings. A similar study on model selection has also been carried out for a method with sparse weights, but it was concluded that cross-validation is a better approach than index of sparseness (and other model selection strateigies) for deriving a model that reflects the true underlying processes (de Schipper & Van Deun, 2021). We selected cross-validation since the methods in the current dissertation are based on sparse weights. In considering an alternative model selection strategy, perhaps research in an innovative direction could help achieve the leap in striking a better balance between model optimality and computational intensiveness. A hybrid approach that combines index of sparseness and stability selection (which is a popular alternative to cross-validation) suggested by (Gu et al., 2019) could be an example.

7.3.2 Data generation

One of the core messages conveyed in Chapter 5 is that sparse PCA research should incorporate data generating models with sparse weights and sparse loadings, rather than being confined to models based on sparse singular vectors. Yet, the simulation studies conducted in Chapters 2, 3 and 4 have solely employed the data generating model with sparse singular vectors. Admittedly, this practice is incomprehensive and ignores other relevant data generating models. However, it was a choice made considering the scope of the papers; which was to propose the novel methods. An extensive study that focuses on testing the proposed PCovR methods on these other data generating models would be well-fitting as the next step. This investigation could be more valuable if it includes other variants of PCovR, as previous research on PCovR has only incorporated the model with sparse singular vectors.

7.4 Future directions

7.4.1 Computational feasiblity and sparse loadings

As the methods proposed in this dissertation are methods suited for large and high-dimensional datasets, they involve heavy computational burden. To provide an indication, a laptop equipped with a four-core Intel i5-10210U processor (base clock speed of 2.11 GHz) and 8GB of RAM was used to fit the SMPCovR model presented in Chapter 4 to two datasets that appeared in the previous chapters. On the Pittsburgh Cold Study (PCS) data with 187 predictors, 16 outcomes from 46 observations, the SMPCovR model reported in Section 4.4.1 was fitted a hundred times, resulting in 0.729 seconds per run on average. However, the weights matrix in this model was very sparse (about 94% sparsity) which tends to be less burdening. By decreasing the lasso parameter and keeping all other parameters constant, a SMPCovR model with about 50% sparsity in weights took 1.587 seconds on average. The Autism dataset employed in Chapter 5 consists of 1107 variables and 27 observations without a distinction of predictors and outcomes. We took the first 30 variables as outcomes and the remaining 1077 as predictors, and fitted the SMPCovR model with parameters such that the weights matrix would be about 94% sparse. From the 100 runs, the average time taken was 29.111 seconds. Lastly, SMPCovR model with about 50% sparsity in weights took 49.192 seconds on average. In Chapter 4, the model selection strategy for the PCS data evaluated 3600 different models with 5-fold cross-validation; the method was therefore administered 18000 times. Even if we assume that each run would take 0.729 seconds for PCS data and 29.111 seconds for the Autism data, such a model selection procedure would take a little more than 3.5 hours and 6 days, respectively. In this time-consuming setting, users would be encouraged to consider narrower ranges of tuning parameters which complicates finding the optimal model.

Although the implementation has been done in Rcpp which significantly speeds up the estimation process compared to only relying on R, there is certainly room for improvement. Setting aside the computational strategies for speed-up such as parallel computing, one straightforward direction that the PCovR methodology for multiblock data can take is to impose the sparsity on the loadings instead of the weights. While no sparse PCovR methods have yet been posed with sparse loadings, benefits brought about by the sparse loadings approach can be inferred from sparse PCA literature that compared it against the sparse weights approach (see for example Chapter 5 or Guerra-Urzola et al. (2021)). Conditional estimation of the sparse loadings becomes a univariate (or low-dimensional) regression problem which involves substantially less computational strain than sparse weights estimation which is often a high-dimensional regression problem. Although the sparse loadings approach has been found to fall short in the amount of variance explained compared to the sparse weights approach, it would be a worthwhile direction towards sparse multiblock PCovR with better computational feasibility.

7.4.2 Novel formulations for sparse PCovR

In the same vein, future directions in PCovR could capitalize on the active research carried out within sparse PCA to propose novel extensions. The sparse PCA problem has been a longstanding topic which has been reformulated in a variety of ways. While some formulations take the approach of a minimization problem (e.g. Zou et al., 2006), others have been proposed as a maximization problem (e.g. d'Aspremont et al., 2004; Jolliffe et al., 2003; Journée et al., 2010). Existing sparse PCA methods also differ in the way of estimating multiple components: the *block* approach that solves for multiple components at once (e.g. Adachi & Trendafilov, 2016) and the *deflation* approach that extracts one component at a time (e.g. H. Shen & Huang, 2008). Lastly, there have been several ways for the methods to induce sparsity on the coefficients. Some penalized the coefficients (e.g. d'Aspremont, Bach, & El Ghaoui, 2008), whereas others imposed a constraint on the number of non-zero coefficients (e.g. X. T. Yuan & Zhang, 2013). Following this categorization, the PCovR methods in Chapters 2, 3 and 4 rely on a minimization problem which adopts the block approach to find sparse weights by penalization. Studies by Zou and Xue (2018) and Guerra-Urzola et al. (2021) can be referred to as overviews for the various sparse PCA methods. Although there has not been a comprehensive investigation which looks into all of these sparse PCA methods regarding their performance on computational feasibility and the quality of estimated coefficients, generalized power method tackling a maximization problem (Journée et al., 2010) has been found effective in the selective comparison conducted by Guerra-Urzola et al. (2021). Therefore, composing and tackling a maximization problem for sparse PCovR may be an interesting topic of future investigation; pursuit of developments in sparse PCA appears to be a promising strategy to bring improvements to the PCovR methods.

References

Abdi, H., & Valentin, D. (2007). Multiple correspondence analysis. *Encyclopedia of measurement and statistics*, *2*(4), 651–657.

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), 433–459.

Adachi, K., & Trendafilov, N. T. (2016). Sparse principal component analysis subject to prespecified cardinality of loadings. *Computational Statistics*, *31*(4), 1403–1427.

Agresti, A. (2003). *Categorical data analysis*. John Wiley & Sons.

An, B., & Zhang, B. (2017). Simultaneous selection of predictors and responses for high dimensional multivariate linear regression. *Statistics & Probability Letters*, *127*, 173–177.

Babor, T. F., Higgins-Biddle, J., Saunders, J., & Monteiro, M. (2001). The alcohol use disorders identification test: Guidelines for use in. *World Health Organization. Recuperado de https://apps. who. int/iris/handle/10665/67205*.

Babyak, M. A. (2004). What you see may not be what you get: a brief, nontechnical introduction to overfitting in regression-type models. *Psychosomatic medicine*, *66*(3), 411–421.

Barker, M., & Rayens, W. (2003). Partial least squares for discrimination. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *17*(3), 166–173.

Barnes, D., Covinsky, K., Whitmer, R., Kuller, L., Lopez, O., & Yaffe, K. (2009). Predicting risk of dementia in older adults: The late-life dementia risk index. *Neurology*, *73*(3), 173–179.

Boqué, R., & Smilde, A. K. (1999). Monitoring and diagnosing batch processes with multiway covariates regression models. *AIChE Journal*, *45*(7), 1504–1520.

Botella, J., Huang, H., & Suero, M. (2015). Meta-analysis of the accuracy of tools used for binary classification when the primary studies employ different references. *Psychological methods*, *20*(3), 331.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, *37*(4), 373–384.

Cadima, J., & Jolliffe, I. T. (1995). Loading and correlations in the interpretation of principle compenents. *Journal of Applied Statistics*, *22*(2), 203–214.

Chavent, M., Kuentz-Simonet, V., Labenne, A., & Saracco, J. (2014). Multivariate analysis of mixed data: The r package pcamixdata. *arXiv preprint* *arXiv:1411.4911*.

Chen, D.-W., Miao, R., Deng, Z.-Y., Lu, Y.-Y., Liang, Y., & Huang, L. (2020). Sparse logistic regression with l1/2 penalty for emotion recognition in electroencephalography classification. *Frontiers in neuroinformatics*, *14*, 29.

Chen, K., Chan, K.-S., & Stenseth, N. C. (2012). Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *74*(2), 203–221.

Chen, L., & Huang, J. Z. (2012). Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, *107*(500), 1533–1545.

Chung, D., & Keles, S. (2010). Sparse partial least squares classification for high dimensional data. *Statistical applications in genetics and molecular biology*, *9*(1).

Cohen, S., Kamarck, T., Mermelstein, R., et al. (1994). Perceived stress scale. *Measuring stress: A guide for health and social scientists*, *10*(2), 1–2.

Cook, R. D., Helland, I., & Su, Z. (2013). Envelopes and partial least squares regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *75*(5), 851–877.

Cook, R. D., Li, B., & Chiaromonte, F. (2010). Envelope models for parsimonious and efficient multivariate linear regression. *Statistica Sinica*, 927–960.

d'Aspremont, A., Bach, F., & El Ghaoui, L. (2008). Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, *9*(7).

d'Aspremont, A., Ghaoui, L., Jordan, M., & Lanckriet, G. (2004). A direct formulation for sparse pca using semidefinite programming. *Advances in neural information processing systems*, *17*.

De Jong, S., & Kiers, H. A. (1992). Principal covariates regression: part i. theory. *Chemometrics and Intelligent Laboratory Systems*, *14*(1-3), 155–164.

de Schipper, N. C., & Van Deun, K. (2018). Revealing the joint mechanisms in traditional data linked with big data. *Zeitschrift für Psychologie*, *226*(4), 212.

de Schipper, N. C., & Van Deun, K. (2021). Model selection techniques for sparse weight-based principal component analysis. *Journal of Chemometrics*, *35*(2), e3289.

Ding, B., & Gentleman, R. (2005). Classification using generalized partial least squares. *Journal of Computational and Graphical Statistics*, *14*(2), 280–298.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., Ishwaran, H., Knight, K., ... Tibshirani, R. (2004, apr). Least angle regression. *Annals of Statistics*, *32*(2), 407–499. doi: 10.1214/009053604000000067

Erichson, N. B., Zheng, P., Manohar, K., Brunton, S. L., Kutz, J. N., & Aravkin,

A. Y. (2020). Spare principal compenent analysis via variable projection. *SIAM Journal on Applied Mathematics*, *80*(2), 977–1002. doi: 10.1137/18M1211350

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, *96*(456), 1348–1360.

Fan, J., Liao, Y., & Mincheva, M. (2013). Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *75*(4), 603–680.

Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, *78*(383), 553–569.

Fraley, R. C., Waller, N. G., & Brennan, K. A. (2000). An item response theory analysis of self-report measures of adult attachment. *Journal of personality and social psychology*, *78*(2), 350.

Friedman, J., Hastie, T., & Tibshirani, R. (2010a). A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*.

Friedman, J., Hastie, T., & Tibshirani, R. (2010b). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, *33*(1), 1.

Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., Simon, N., & Qian, J. (2021). Package 'glmnet'. *CRAN R Repositary*.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2001). *The elements of statistical learning* (Vol. 1) (No. 10). Springer series in statistics New York.

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, *58*(3), 453–467.

Gabriel, K. R. (1978). Least squares approximation of matrices by additive and multiplicative models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 186–196.

Gajjar, S., Kulahci, M., & Palazoglu, A. (2017). Selection of non-zero loadings in sparse principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *162*, 160–171.

Gallagher, J., Hudgens, E., Williams, A., Inmon, J., Rhoney, S., Andrews, G., … others (2011). Mechanistic indicators of childhood asthma (mica) study: piloting an integrative design for evaluating environmental health. *BMC Public Health*, *11*(1), 344.

Gizer, I. R., Ficks, C., & Waldman, I. D. (2009). Candidate gene studies of adhd: a meta-analytic review. *Human genetics*, *126*(1), 51–90.

Goldberg, L. R., et al. (1999). A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. *Personality psychology in Europe*, *7*(1), 7–28.

Grizenko, N., Fortier, M.-E., Zadorozny, C., Thakur, G., Schmitz, N., Duval, R., & Joober, R. (2012). Maternal stress during pregnancy, adhd symptomatology in children and genotype: gene-environment interaction. *Journal of the Canadian Academy of Child and Adolescent Psychiatry*, *21*(1), 9.

Gross, J. J., & John, O. P. (2003). Individual differences in two emotion regulation processes: implications for affect, relationships, and well-being. *Journal of personality and social psychology*, *85*(2), 348.

Gu, Z., Schipper, N. C. d., & Van Deun, K. (2019). Variable selection in the regularized simultaneous component analysis method for multi-source data integration. *Scientific reports*, *9*(1), 1–21.

Gu, Z., & Van Deun, K. (2016, nov). A variable selection method for simultaneous component based data integration. *Chemometrics and Intelligent Laboratory Systems*, *158*, 187–199. Retrieved from `https://linkinghub.elsevier.com/retrieve/pii/S0169743916301848` doi: 10.1016/j.chemolab.2016.07.013

Gu, Z., & Van Deun, K. (2019). Regularizedsca: Regularized simultaneous component analysis of multiblock data in r. *Behavior research methods*, *51*(5), 2268–2289.

Guerra-Urzola, R., de Schipper, N. C., Tonne, A., Sijtsma, K., Vera, J. C., & Van Deun, K. (2022, apr). Sparsifying the least-squares approach to PCA: comparison of lasso and cardinality constraint. *Advances in Data Analysis and Classification*, 1–18. Retrieved from `https://link.springer.com/article/10.1007/s11634-022-00499-2https://link.springer.com/10.1007/s11634-022-00499-2` doi: 10.1007/s11634-022-00499-2

Guerra-Urzola, R., Van Deun, K., Vera, J. C., & Sijtsma, K. (2023). Penalized PCA Framework: Thresholding Operators and Optimality Conditions. *Working Paper*.

Guerra-Urzola, R., Van Deun, K., Vera, J. C., & Sijtsma, K. (2021). A guide for sparse pca: Model comparison and applications. *Psychometrika*, 1–27.

Guo, C., Kang, J., & Johnson, T. D. (2020). A spatial bayesian latent factor model for image-on-image regression. *Biometrics*.

Gvaladze, S., Vervloet, M., Van Deun, K., Kiers, H. A., & Ceulemans, E. (2021). Pcovr2: A flexible principal covariates regression approach to parsimoniously handle multiple criterion variables. *Behavior Research Methods*, 1–21.

Hair, J. F., Ringle, C. M., & Sarstedt, M. (2011). Pls-sem: Indeed a silver bullet. *Journal of Marketing theory and Practice*, *19*(2), 139–152.

Hasin, Y., Seldin, M., & Lusis, A. (2017). Multi-omics approaches to disease. *Genome biology*, *18*(1), 83.

Hauser, R. M., & Warren, J. R. (1997). Socioeconomic indexes for occupations: A review, update, and critique. *Sociological methodology*, *27*(1), 177–298.

Heij, C., Groenen, P. J., & van Dijk, D. (2007). Forecast comparison of principal component regression and principal covariate regression. *Computational statistics & data analysis*, *51*(7), 3612–3625.

Helfrecht, B. A., Cersonsky, R. K., Fraux, G., & Ceriotti, M. (2020). Structure-property maps with kernel principal covariates regression. *Machine Learning: Science and Technology*, *1*(4), 045021.

Hill, L. S., Reid, F., Morgan, J. F., & Lacey, J. H. (2010). Scoff, the development of an eating disorder screening questionnaire. *International journal of eating disorders*, *43*(4), 344–351.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, *24*(6), 417.

Hu, J., Huang, J., Liu, X., & Liu, X. (2022). Response best-subset selector for multivariate regression with high-dimensional response variables. *Biometrika*.

Hu, J., Liu, X., Liu, X., & Xia, N. (2022). Some aspects of response variable selection and estimation in multivariate linear regression. *Journal of Multivariate Analysis*, *188*, 104821.

Hwang, H., Cho, G., Jung, K., Falk, C. F., Flake, J. K., Jin, M. J., & Lee, S. H. (2021). An approach to structural equation modeling with both factors and components: Integrated generalized structured component analysis. *Psychological Methods*, *26*(3), 273.

IBM Corp. (2020). *Ibm spss statistics for windows.* Armonk, NY: IBM Corp.

Izenman, A. J. (1975). Reduced-rank regression for the multivariate linear model. *Journal of multivariate analysis*, *5*(2), 248–264.

Jackson, G. G., DOWLING, H. F., SPIESMAN, I. G., & BOAND, A. V. (1958). Transmission of the common cold to volunteers under controlled conditions: I. the common cold as a clinical entity. *AMA archives of internal medicine*, *101*(2), 267–278.

Jia, J., & Yu, B. (2010). On model selection consistency of the elastic net when p » n. *Statistica Sinica*, 595–611.

Johnson, J. A. (2014). Measuring thirty facets of the five factor model with a 120-item public domain inventory: Development of the ipip-neo-120. *Journal of Research in Personality*, *51*, 78–89.

Johnson, J. A. (2018). *Data from johnson, j. a. (2014). measuring thirty facets of the five factor model with a 120-item public domain inventory: Development of the ipip-neo-120.* Retrieved from osf.io/wxvth.

Johnstone, I. M. (2001). On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, 295–327.

Johnstone, I. M., & Lu, A. Y. (2009). On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, *104*(486), 682–693.

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.

Jolliffe, I. T. (1986). *Principal component analysis*. Springer New York. Retrieved from http://dx.doi.org/10.1007/978-1-4757-1904-8 doi: 10.1007/978-1-4757-1904-8

Jolliffe, I. T. (2002). *Principal component analysis*. Springer.

Jolliffe, I. T., Trendafilov, N. T., & Uddin, M. (2003). A modified principal component technique based on the lasso. *Journal of computational and Graphical Statistics*, *12*(3), 531–547.

Journée, M., Nesterov, Y., Richtárik, P., & Sepulchre, R. (2010). Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, *11*(Feb), 517–553.

Kahn, R. S., Khoury, J., Nichols, W. C., & Lanphear, B. P. (2003). Role of dopamine transporter genotype and maternal prenatal smoking in childhood hyperactive-impulsive, inattentive, and oppositional behaviors. *The Journal of pediatrics*, *143*(1), 104–110.

Kang, O.-S., Chang, D.-S., Jahng, G.-H., Kim, S.-Y., Kim, H., Kim, J.-W., … others (2012). Individual differences in smoking-related cue reactivity in smokers: an eye-tracking and fmri study. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, *38*(2), 285–293.

Kawano, S. (2021). Sparse principal component regression via singular value decomposition approach. *Advances in Data Analysis and Classification*, 1–29.

Kawano, S., Fujisawa, H., Takada, T., & Shiroishi, T. (2015). Sparse principal component regression with adaptive loading. *Computational Statistics & Data Analysis*, *89*, 192–203.

Kawano, S., Fujisawa, H., Takada, T., & Shiroishi, T. (2018). Sparse principal component regression for generalized linear models. *Computational Statistics & Data Analysis*, *124*, 180–196.

Kiers, H. A., & Smilde, A. K. (2007). A comparison of various methods for multivariate regression with highly collinear variables. *Statistical Methods and Applications*, *16*(2), 193–228.

Kiers, H. A., & ten Berge, J. M. (1989). Alternating least squares algorithms for simultaneous components analysis with equal component weight matrices in two or more populations. *Psychometrika*, *54*(3), 467–473.

Kim, J., Zhang, Y., & Pan, W. (2016). Powerful and adaptive testing for multi-trait and multi-snp associations with gwas and sequencing data. *Genetics*, *203*(2), 715–731.

Kreutzmann, S., Svensson, V. T., Thybo, A. K., Bro, R., & Petersen, M. A. (2008). Prediction of sensory quality in raw carrots (daucus carota l.) using multi-block ls-parpls. *Food Quality and Preference*, *19*(7), 609–617.

Lange, K. (2016). *Mm optimization algorithms*. SIAM.

Lange, K., Hunter, D. R., & Yang, I. (2000, mar). Optimization Transfer Using Surrogate Objective Functions. *Journal of Computational and Graphical Statistics*, *9*(1), 1–20. Retrieved from `http://www.tandfonline.com/doi/abs/10.1080/10618600.2000.10474858` doi: 10.1080/10618600.2000.10474858

Lê Cao, K.-A., Boitard, S., & Besse, P. (2011). Sparse pls discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems. *BMC bioinformatics*, *12*(1), 253.

Lê Cao, K.-A., Rossouw, D., Robert-Granié, C., & Besse, P. (2008). A sparse pls for variable selection when integrating omics data. *Statistical applications in genetics and molecular biology*, *7*(1).

Lock, E. F., Hoadley, K. A., Marron, J. S., & Nobel, A. B. (2013). Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The annals of applied statistics*, *7*(1), 523.

Lorenzo-Seva, U., & Ten Berge, J. M. (2006). Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology*, *2*(2), 57–64.

Luo, S. (2020). Variable selection in high-dimensional sparse multiresponse linear regression models. *Statistical Papers*, *61*(3), 1245–1267.

Måge, I., Menichelli, E., & Næs, T. (2012). Preference mapping by po-pls: Separating common and unique information in several data blocks. *Food quality and preference*, *24*(1), 8–16.

Måge, I., Smilde, A. K., & van der Kloet, F. M. (2019). Performance of methods that separate common and distinct variation in multiple data blocks. *Journal of Chemometrics*, *33*(1), e3085.

Mayer, J., Rahman, R., Ghosh, S., & Pal, R. (2018). Sequential feature selection and inference using multi-variate random forests. *Bioinformatics*, *34*(8), 1336–1344.

McCrae, R. R., & Costa Jr, P. T. (2008). Empirical and theoretical status of the five-factor model of personality traits.

McFadden, D., et al. (1973). Conditional logit analysis of qualitative choice behavior.

McNeish, D. M. (2015). Using lasso for predictor selection and to assuage overfitting: A method long overlooked in behavioral sciences. *Multivariate Behavioral Research*, *50*(5), 471–484.

Meinshausen, N., & Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(4), 417–473.

Mioshi, E., Dawson, K., Mitchell, J., Arnold, R., & Hodges, J. R. (2006). The addenbrooke's cognitive examination revised (ace-r): a brief cognitive test battery for dementia screening. *International Journal of Geriatric Psychiatry: A journal of the psychiatry of late life and allied sciences*, *21*(11), 1078–1085.

Monto, A. S., Gravenstein, S., Elliott, M., Colopy, M., & Schweinle, J. (2000). Clinical signs and symptoms predicting influenza infection. *Archives of internal medicine*, *160*(21), 3243–3247.

Moos, R. H. (1990). Conceptual and empirical approaches to developing family-based assessment procedures: Resolving the case of the family environment scale. *Family process*, *29*(2), 199–208.

Nakaya, H. I., Wrammert, J., Lee, E. K., Racioppi, L., Marie-Kunze, S., Haining, W. N., … others (2011). Systems biology of vaccination for seasonal influenza in humans. *Nature immunology*, *12*(8), 786.

Natarajan, B. K. (1995, apr). Sparse Approximate Solutions to Linear Systems. *SIAM Journal on Computing*, *24*(2), 227–234. Retrieved from `http://epubs.siam.org/doi/10.1137/S0097539792240406` doi: 10.1137/S0097539792240406

Nelemans, S. A., Van Assche, E., Bijttebier, P., Colpin, H., Van Leeuwen, K., Verschueren, K., … Goossens, L. (2019). Parenting interacts with oxytocin polymorphisms to predict adolescent social anxiety symptom development: A novel polygenic approach. *Journal of abnormal child psychology*, *47*(7), 1107–1120.

Nguyen, H. D. (2017). An introduction to majorization-minimization algorithms for machine learning and statistical estimation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(2), e1198.

Nishimura, Y., Martin, C. L., Vazquez-Lopez, A., Spence, S. J., Alvarez-Retuerto, A. I., Sigman, M., … Geschwind, D. H. (2007, jul). Genome-wide expression profiling of lymphoblastoid cell lines distinguishes different forms of autism and reveals shared pathways. , *16*(14), 1682–1698. doi: 10.1093/hmg/ddm116

Obozinski, G., Taskar, B., & Jordan, M. (2006). Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep*, *2*(2.2), 2.

Oladzad, A., Porch, T., Rosas, J. C., Moghaddam, S. M., Beaver, J., Beebe, S. E., … others (2019). Single and multi-trait gwas identify genetic factors associated with production traits in common bean under abiotic stress environments.

G3: Genes, Genomes, Genetics, *9*(6), 1881–1892.

Park, M. Y., & Hastie, T. (2008). Penalized logistic regression for detecting gene interactions. *Biostatistics*, *9*(1), 30–50.

Park, S., Ceulemans, E., & Van Deun, K. (2020). Sparse common and distinctive covariates regression. *Journal of Chemometrics*, e3270.

Peng, J., Zhu, J., Bergamaschi, A., Han, W., Noh, D.-Y., Pollack, J. R., & Wang, P. (2010). Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *The annals of applied statistics*, *4*(1), 53.

Raiche, G., Magis, D., & Raiche, M. G. (2020). Package 'nfactors'. *Repository CRAN*, 1–58.

Raîche, G., Walls, T. A., Magis, D., Riopel, M., & Blais, J.-G. (2013). Non-graphical solutions for cattell's scree test. *Methodology*.

Rasmussen, M. A., & Bro, R. (2012). A tutorial on the lasso approach to sparse modeling. *Chemometrics and Intelligent Laboratory Systems*, *119*, 21–31.

Richtárik, P., Jahani, M., Ahipaşaoğlu, S. D., & Takáč, M. (2021, sep). Alternating maximization: unifying framework for 8 sparse PCA formulations and efficient parallel codes. *Optimization and Engineering*, *22*(3), 1493–1519. Retrieved from `https://link.springer.com/10.1007/s11081-020-09562-3` doi: 10.1007/s11081-020-09562-3

Rohart, F., Gautier, B., Singh, A., & Cao, K. (2017). mixomics: an r package for 'omics feature selection and multiple data integration. *biorxiv.org*. Retrieved from `http://biorxiv.org/content/early/2017/05/05/108597`

Schneider, B., & Waite, L. J. (2008). The 500 family study [1998-2000: United states]. *Ann Arbor, MI: Inter-university Consortium for Political and Social Research [distributor]. https://doi. org/10.3886/ICPSR04549. v1*.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, *31*(1), 1–10.

Schouteden, M., Van Deun, K., Pattyn, S., & Van Mechelen, I. (2013). Sca with rotation to distinguish common and distinctive information in linked data. *Behavior research methods*, *45*(3), 822–833.

Shen, D., Shen, H., & Marron, J. (2016). A general framework for consistency of principal component analysis. *The Journal of Machine Learning Research*, *17*(1), 5218–5251.

Shen, D., Shen, H., & Marron, J. S. (2013). Consistency of sparse pca in high dimension, low sample size contexts. *Journal of Multivariate Analysis*, *115*, 317–333.

Shen, H., & Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis*, *99*(6), 1015–1034.

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics*, *22*(2), 231–245.

Singh, A., Gautier, B., Shannon, C. P., Vacher, M., Rohart, F., Tebbutt, S. J., & Le Cao, K.-A. (2016). Diablo–an integrative, multi-omics, multivariate method for multi-group classification. *BioRxiv*, 067611.

Smilde, A. K., Måge, I., Naes, T., Hankemeier, T., Lips, M. A., Kiers, H. A., ... Bro, R. (2017). Common and distinct components in data fusion. *Journal of Chemometrics*, *31*(7), e2900.

Smilde, A. K., Westerhuis, J. A., & Boque, R. (2000). Multiway multiblock component and covariates regression models. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *14*(3), 301–331.

Sriperumbudur, B. K., Torres, D. A., & Lanckriet, G. R. (2011a). A majorization-minimization approach to the sparse generalized eigenvalue problem. *Machine Learning*, *85*(1-2), 3–39. doi: 10.1007/s10994-010-5226-3

Sriperumbudur, B. K., Torres, D. A., & Lanckriet, G. R. (2011b). A majorization-minimization approach to the sparse generalized eigenvalue problem. *Machine learning*, *85*(1), 3–39.

Steiger, H., Labonté, B., Groleau, P., Turecki, G., & Israel, M. (2013). Methylation of the glucocorticoid receptor gene promoter in bulimic women: associations with borderline personality disorder, suicidality, and exposure to childhood abuse. *International Journal of Eating Disorders*, *46*(3), 246–255.

Stein, J. L., Hua, X., Lee, S., Ho, A. J., Leow, A. D., Toga, A. W., ... others (2010). Voxelwise genome-wide association study (vgwas). *neuroimage*, *53*(3), 1160–1174.

Steinley, D., & Brusco, M. J. (2008). Selection of variables in cluster analysis: An empirical comparison of eight procedures. *Psychometrika*, *73*(1), 125–144.

Su, Z., Zhu, G., Chen, X., & Yang, Y. (2016). Sparse envelope model: efficient estimation and response variable selection in multivariate linear regression. *Biometrika*, *103*(3), 579–593.

Taylor, M. K., Sullivan, D. K., Ellerbeck, E. F., Gajewski, B. J., & Gibbs, H. D. (2019). Nutrition literacy predicts adherence to healthy/unhealthy diet patterns in adults with a nutrition-related chronic condition. *Public health nutrition*, *22*(12), 2157–2169.

ten Berge, J. M. (1993). *Least squares optimization in multivariate analysis*. DSWO Press, Leiden University Leiden.

Tenenhaus, A., Philippe, C., Guillemot, V., Le Cao, K.-A., Grill, J., & Frouin, V. (2014). Variable selection for generalized canonical correlation analysis. *Biostatistics*, *15*(3), 569–583.

Tenenhaus, A., & Tenenhaus, M. (2011). Regularized generalized canonical correlation analysis. *Psychometrika*, *76*(2), 257–284.

Tenenhaus, M., Tenenhaus, A., & Groenen, P. J. (2017). Regularized generalized canonical correlation analysis: a framework for sequential multiblock component methods. *Psychometrika*, *82*(3), 737–777.

Thomson, S. (2018). *Achievement at school and socioeconomic background—an educational perspective* (Vol. 3) (No. 1). Nature Publishing Group.

Thybo, A. K., Bechmann, I. E., Martens, M., & Engelsen, S. B. (2000). Prediction of sensory texture of cooked potatoes using uniaxial compression, near infrared spectroscopy and low field1h nmr spectroscopy. *LWT-Food Science and Technology*, *33*(2), 103–111.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Trendafilov, N. T. (2014). From simple structure to sparse components: a review. *Computational Statistics*, *29*(3), 431–454.

Trendafilov, N. T., & Adachi, K. (2015). Sparse versus simple structure loadings. *psychometrika*, *80*(3), 776–790.

Tu, Y., & Lee, T.-H. (2019). Forecasting using supervised factor models. *Journal of Management Science and Engineering*, *4*(1), 12–27.

Tucker, J. S. (2002). Health-related social control within older adults' relationships. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, *57*(5), P387–P395.

Tucker, L. R. (1951). *A method for synthesis of factor analysis studies* (Tech. Rep.). Educational Testing Service Princeton Nj.

Tutun, S., Ahmed, A. A., Irgil, S., Yesilkaya, I., Analytics, D., & Khasawneh, M. T. (2019). Detecting psychological symptom patterns using regularized multinomial logistic regression. In *2019 institute of industrial and systems engineers annual conference and expo, iise 2019* (p. 967087).

Van Deun, K., Crompvoets, E. A., & Ceulemans, E. (2018). Obtaining insights from high-dimensional data: sparse principal covariates regression. *BMC bioinformatics*, *19*(1), 104.

Van Deun, K., Smilde, A. K., Van Der Werf, M. J., Kiers, H. A., & Van Mechelen, I. (2009). A structured overview of simultaneous component based data integration. *Bmc Bioinformatics*, *10*(1), 1–15.

Van Deun, K., Thorrez, L., Coccia, M., Hasdemir, D., Westerhuis, J. A., Smilde, A. K., & Van Mechelen, I. (2019). Weighted sparse principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *195*, 103875.

Van Deun, K., Wilderjans, T. F., Van den Berg, R. A., Antoniadis, A., & Van Mechelen, I. (2011). A flexible framework for sparse simultaneous component based data integration. *BMC bioinformatics*, *12*(1), 448.

Van Mechelen, I., & Smilde, A. K. (2010). A generic linked-mode decomposition model for data fusion. *Chemometrics and Intelligent Laboratory Systems*, *104*(1), 83–94.

Vervloet, M., Kiers, H. A., Van den Noortgate, W., & Ceulemans, E. (2015). Pcovr: An r package for principal covariates regression. *Journal of Statistical Software*, *65*(8), 1–14.

Vervloet, M., Van Deun, K., Van den Noortgate, W., & Ceulemans, E. (2013). On the selection of the weighting parameter value in principal covariates regression. *Chemometrics and Intelligent Laboratory Systems*, *123*, 36–43.

Vervloet, M., Van Deun, K., Van den Noortgate, W., & Ceulemans, E. (2016). Model selection in principal covariates regression. *Chemometrics and Intelligent Laboratory Systems*, *151*, 26–33.

Wang, W., & Fan, J. (2017). Asymptotics of empirical eigenstructure for high dimensional spiked covariance. *Annals of statistics*, *45*(3), 1342.

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: the panas scales. *Journal of personality and social psychology*, *54*(6), 1063.

Westerhuis, J. A., & Coenegracht, P. M. (1997). Multivariate modelling of the pharmaceutical two-step process of wet granulation and tableting with multiblock partial least squares. *Journal of Chemometrics: A Journal of the Chemometrics Society*, *11*(5), 379–392.

Whittle, P. (1952). On principal components and least square methods of factor analysis. *Scandinavian Actuarial Journal*, *1952*(3-4), 223–239.

Wilderjans, T. F., Ceulemans, E., & Meers, K. (2013). Chull: A generic convex-hull-based model selection method. *Behavior research methods*, *45*(1), 1–15.

Witten, D. M., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, *10*(3), 515–534.

Wold, H. (1982). Soft modeling: the basic design and some extensions. *Systems under indirect observation*, *2*, 343.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37–52.

Wold, S., Martens, H., & Wold, H. (1983). The multivariate calibration problem in chemistry solved by the pls method. In *Matrix pencils* (pp. 286–293). Springer.

Wold, S., Ruhe, A., Wold, H., & Dunn, W., Iii. (1984). The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, *5*(3), 735–

743.

Wold, S., Sjöström, M., & Eriksson, L. (2001). Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, *58*(2), 109–130.

Yuan, G. X., Ho, C. H., & Lin, C. J. (2011). An improved glmnet for L1-regularized logistic regression. In *Proceedings of the acm sigkdd international conference on knowledge discovery and data mining* (pp. 33–41). doi: 10.1145/2020408.2020421

Yuan, M., & 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.

Yuan, X. T., & Zhang, T. (2013). Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, *14*(1), 899–925.

Zamdborg, L., & Ma, P. (2009). Discovery of protein–dna interactions by penalized multivariate regression. *Nucleic acids research*, *37*(16), 5246–5254.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, *101*(476), 1418–1429.

Zou, H., & 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.

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, *15*(2), 265–286.

Zou, H., & Xue, L. (2018). A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, *106*(8), 1311–1320.

Summary

Many research fields of today enjoy the unprecedented availability of large data collected from different sources concerning the same observation units. For example, a block of questionnaire data regarding health practices and another block of gene expression data are jointly analyzed to study the onset of lung cancer. In studying the predictive mechanisms giving rise to an outcome, such joint data (known as multiblock data) offer a distinct opportunity to identify mechanisms characterized by a combination of variables from different data sources. It is only possible to pinpoint the interaction between smoking and a certain genetic susceptibility as a determinant of lung cancer by employing a multiblock data setup. Multiblock data analysis hence helps obtaining a comprehensive understanding of an outcome by finding these mechanisms of multi-source nature.

However, identifying such mechanisms in relation to multiple blocks is far from easy. They are known to take a subtler manifestation in the data compared to other mechanisms that are solely rooted in individual data bocks. This problem is especially pronounced when the data blocks are large and heterogeneous from each other. Moreover, there are issues pertaining to multiblock data that complicate the construction of predictive models. Multiblock datasets often contain a large number of predictor variables that are highly correlated with each other, or unimportant to the research question. Presence of these predictors disallows stable estimation of model coefficients and renders the derived models to consist of an excessive number of coefficients which are impractical to inspect. Additionally, there are challenges concerning the outcome variable; some data problems involve a continuous outcome, while others a categorical outcome. There may also be data settings comprised with multiple outcome variables.

In proposing novel methods that address these challenges, we employed principal covariate regression (PCovR) as a basis. PCovR is a method that summarizes the predictor variables into 'principal covariates' that account for the prediction of the outcome variables. We put forward extensions of PCovR by making the following adaptations. First, the problems of unimportant and highly correlated predictors were tackled by introducing regularization penalties when deriving the covariates. Second, the covariates were modified such that they are distinguished into two types: those that are uniquely in relation with single data blocks (distinctive covariates) and others that associate with multiple blocks jointly (common covariates). This distinction enables capturing of the mechanisms chracterized by

a mixture of variables from multiple blocks. Third, we devised three variants of the PCovR extension to target the aforementioned outcome variable settings. For example, our method was combined with logistic regression to address a categorical outcome.

Chapter 2 presents the extension of PCovR for a multiblock setting. While this method targeted a regression problem, it was directly adapted into a classification problem in Chapter 3. The two methods showed competitive performance in outcome prediction and retrieving the true predictive mechanisms compared to methods with the same set of goals. Whereas Chapters 2 and 3 only addressed a single outcome variable, Chapter 4 tackled a setting with multiple outcome variables. A method that performs variable selection for both predictor and outcome variables simultaneously was devised therein. In looking out for ways in which PCovR can be improved to better suit large and multiblock data, topics within sparse principal component analysis (PCA) were visited in Chapters 5 and 6. This is because sparse PCA serves as the basis for the PCovR extensions in this dissertation. An algorithm for sparse PCA that guarantees local optimality of the solutions has been proposed in Chapter 5, hinting at the future step for extending PCovR. Lastly, the overlooked consequences of introducing sparsity to PCA were studied in Chapter 6. It was found that imposing the sparsity on the weights leads to more unstable results than when the loadings are made sparse, which is a result that has a direct implication in extending PCovR. Altogether, this dissertation puts forward novel PCovR methods that allow unlocking the potential within multiblock data and points out where future opportunities may lie in the next line of research.