Jack of all trades, Master of None: The Trade-offs in Sparse PCA Methods for Diverse Purposes

Guerra Urzola, Rosember

Publication date:
2023

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

Jack of all Trades, Master of None: The Trade-offs in Sparse PCA Methods for Diverse Purposes

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 aan gewezen commissie in de Aula van de Universiteit op **woensdag 1 november 2023 om 14.00 uur**

door

Rosember Isidoro Guerra Urzola,

geboren te El Bagre, Colombia

Promotor:	prof. dr. K. Sijtsma	(Tilburg University)
Copromotores:	dr. K. Van Deun	(Tilburg University)
	dr. J.C. Vera Lizcano	(Tilburg University)
Ieden Promotiecommissie:	prof. dr. J.K. Vermunt	(Tilburg University)
	dr. M. Balvert	(Tilburg University)
	prof. dr. M.J. de Rooij	(Leiden University)
	prof. dr. P.J.F. Groenen	(Erasmus University Rotterdam)

Tilburg University financially supported printing.

Printed and cover designed by: Proefschrift-aio

To a dreamer.
Thanks for never giving up.

Contents

Chapter 1

Introduction

This dissertation focuses on Sparse Principal Component Analysis. In this chapter, we introduce the background of Principal Component Analysis. Then, we present the sparse Principal Component Analysis problem and state the aims and limitations of the dissertation. Finally, we outline the dissertation by describing the content per chapter.

1.1 Background

1.1.1 Principal Component Analysis

With the rise of digitalization, massive amounts of information are at the disposal of many researchers and practitioners. Analyzing this information is challenging due to the associated computational challenges and often the highly involved interpretation of the results. Principal Component Analysis (PCA) is one of the most widely used multivariate techniques to summarize variables in a much lower dimension with minimal loss of information (Hotelling, 1933; I. Jolliffe, 2002; Wold, Esbensen, & Geladi, 1987). There are many examples of PCA applications in many fields. In neuroscience, a variant of PCA is used to characterize the response property of a neuron in response to a time-varying stimulus (Brenner, Bialek, & De Ruyter Van Steveninck, 2000). In quantitative finance, PCA can be applied to the risk management of interest rate derivative portfolios (Alexander, 2008), and to perform a market analysis or diversify the risk by finding the best direction to invest (Pasini, 2017). PCA has been incorporated with Artificial Intelligence (AI) techniques to improve performance in applications such as anomaly detection, classification, image processing, and pattern recognition (Mohammed, Khalid, Osman, & Helali, 2016). A recent study applied PCA to the COVID-19 genome sequence to provide potential virus mutations for further studies (B. Wang & Jiang, 2021). The purpose of using PCA varies per application. This dissertation considers two purposes for PCA: Exploratory Data Analysis (EDA) and Dimensionality Reduction (DR).

PCA for Exploratory Data Analysis

Without loss of generality, let $\mathbf{X} \in \mathbb{R}^{I \times J}$ be a data matrix centered and scaled to unit variance. When the purpose is EDA, PCA can give insight into the data structure by finding the correlation between the new reduced set of variables and $\mathbf{X}$. This purpose is associated with the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{P}} \quad & \left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{P}^\top\mathbf{P} = \mathbf{I}, \end{aligned} \tag{1.1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathbf{W} \in \mathbb{R}^{J \times K}$ is a matrix used to project $\mathbf{X}$ to build the new set of variables $\mathbf{T} = \mathbf{X}\mathbf{W}$, and $\mathbf{P} \in \mathbb{R}^{J \times K}$ is the loadings matrix that expresses the strength of association between new set variables $\mathbf{T}$ and the original variables $\mathbf{X}$. The solution of problem (1.1) is given by the k-first right singular vectors of $\mathbf{X}$.

PCA for Dimension Reduction

When the purpose is DR, PCA is a model-free technique that focuses only on finding a set of orthonormal vectors $\mathbf{W}$ to form the new variables such that the variance is maximized. This is done classically by solving the following optimization problem:

$$\begin{aligned} \max_{\mathbf{W}} \quad & \text{Tr}(\mathbf{W}\mathbf{X}^\top\mathbf{X}\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^\top\mathbf{W} = \mathbf{I}, \end{aligned} \tag{1.2}$$

where $\text{Tr}(\cdot)$ denotes the trace matrix operator. $\mathbf{W} \in \mathbb{R}^{J \times K}$ is called weights matrix, and it weighs the variables $\mathbf{X}$ to construct the new variables $\mathbf{T}$. The solution of problem (1.2) is given by the k-first eigenvectors of the matrix $\mathbf{X}^\top\mathbf{X}$. Therefore, due to the mathematical relationship between the Singular Value Decomposition (SVD) of $\mathbf{X}$ and the Eigenvector Decomposition (EVD) of $\mathbf{X}^\top\mathbf{X}$, the problems (1.1) and (1.2) yield the same solutions for $\mathbf{P}$ and $\mathbf{W}$.

Besides its easy implementation and availability in various software packages, PCA methods generally do not provide interpretable solutions for the new set of variables. It can be seen that the new set of variables $\mathbf{T}$ is formed as a linear combination of all the original variables, where the coefficients of these linear combinations are given by $\mathbf{W}$. This makes the new variables challenging to interpret. Furthermore, in a high-dimensional setting (many more variables than observations, $I \ll J$), the solution for $\mathbf{P}$ and $\mathbf{W}$ presents inconsistency (Johnstone & Lu, 2009). Solutions with many zero entries for $\mathbf{P}$ and $\mathbf{W}$ have been proposed to tackle the interpretability and consistency problems. These solutions are discussed in the next section.

1.1.2 Sparse PCA

A sparse array (matrix or vector) is an array populated mainly with zero as elements (Yan, Wu, Liu, & Gao, 2018). Sparse solutions to the PCA problem have been proposed in the literature to improve interpretability and consistency. For example, in medical applications such as cancer research, when applying PCA to reduce the dimension, extracting the relevant features is a critical challenge (Hsu, Huang, & Chen, 2014). In finance, where trading strategies are based on PCA, fewer assets represent fewer transaction costs. PCA methods that attain sparse solutions are known as sparse PCA methods. A classical way to formulate a sparse PCA problem is to limit the number of nonzero elements on the loadings and weights by a constraint. This is called cardinality constraint. We refer to PCA models with this type of constraint as *cardinality-constrained PCA*. However, due to their combinatorial nature, the computational complexity of cardinality-constrained PCA problems is NP-hard (Natarajan, 1995). Different classes of methods have been proposed in the statistical and optimization literature to address relaxed versions of the cardinality-constrained PCA problem via Mixed-Integer semidefinite optimization (d'Aspremont, Bach, & Ghaoui, 2007; d'Aspremont, El Ghaoui, Jordan, & Lanckriet, 2004), branch-and-bound methods (Berk & Bertsimas, 2019; Moghaddam, Weiss, & Avidan, 2005), a cutting-plane method to certifiable optimality (Bertsimas, Cory-Wright, & Pauphilet, 2022), or penalized methods that add a sparsity-inducing penalty to the objective function (Journée, Nesterov, Richtárik, & Sepulchre, 2010; Zou, Hastie, & Tibshirani, 2006). A sparsity-inducing penalty is a function that forces the solution associated with the penalized optimization problem to be sparse. Examples of sparsity-inducing penalties include, but are not limited to, the l_1-norm (or LASSO) (Tibshirani, 1996), the l_0-norm (Journée et al., 2010), and the Elastic Net penalty (Zou et al., 2006). We refer to sparse PCA formulations with these types of penalty as *Penalized PCA*. Table 1.1 summarizes some sparse PCA formulations by combining PCA formulations with sparsity-inducing principles. It can be observed that some of the formulations seek only one weight vector. It is a common practice to find the subsequent weights by replacing the matrix $\mathbf{X}$ with a deflated version (MacKey, 2009).

Table 1.1: Sparse PCA formulations

Sparsity inducing	Feasible Region	Objective
Cardinality-Constrained	$\{\mathbf{w} \in \mathbb{R}^J : \mathbf{w}^\top \mathbf{w} \leq 1, \|\mathbf{w}\|_0 \leq \rho\}$	$\max_{\mathbf{w}}\{\mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w}\}$
$l_1 - norm$ Penalized	$\{\mathbf{w} \in \mathbb{R}^J : \mathbf{w}^\top \mathbf{w} \leq 1\}$	$\max_{\mathbf{w}}\{\mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w} - \lambda\|\mathbf{w}\|_1\}$
$l_0 - norm$ Penalized	$\{\mathbf{w} \in \mathbb{R}^J : \mathbf{w}^\top \mathbf{w} \leq 1\}$	$\max_{\mathbf{w}}\{\mathbf{w}^\top \mathbf{X}^\top \mathbf{X}\mathbf{w} - \lambda\|\mathbf{w}\|_0\}$
Cardinality-Constrained	$\{\mathbf{W}, \mathbf{P} \in \mathbb{R}^{J\times K} : \mathbf{P}^\top \mathbf{P} = \mathbf{I}, \mathbf{w}_j^\top \mathbf{w}_j \leq 1, \|\mathbf{W}\|_0 \leq \rho\}$	$\min\{\left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\right\|_F^2\}$
$l_1 - norm$ Penalized	$\{\mathbf{W}, \mathbf{P} \in \mathbb{R}^{J\times K} : \mathbf{P}^\top \mathbf{P} = \mathbf{I}, \mathbf{w}_j^\top \mathbf{w}_j \leq 1\}$	$\min\{\left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\right\|_F^2 + \sum_{k=1}^K \lambda_k^l \left\|\mathbf{w}_k\right\|_1\}$

$\|\cdot\|_0$ denotes the number of nonzero elements, $\|\cdot\|_1$ is the sum of the absolute values, and K is the number of components.

Given all the sparse PCA formulations and methods to attain sparse solutions,

a natural question is: Which is the best sparse PCA formulation? This dissertation aims to answer this question based on two perspectives: Data Analysis and Optimization. This is elaborated on in the following section.

1.2 Research objectives and limitations

Most of the sparse PCA formulations are based on PCA formulations. This encourages the misconception that some properties of PCA could be extended to the sparse PCA version. For example, PCA problems (1.1) and (1.2) present equivalent solutions for loadings or weights due to the relationship between the SVD and the EVD. However, when sparsity is imposed, this equivalence is lost because neither SVD nor EVD is a solution to the problem. Additionally, if sparsity is imposed on the loadings (weights), it does not imply sparsity on the weights (loadings). Therefore, each sparse PCA model presents a different objective and sparse structure. One of this dissertation's aims is to guide the use and implementation of sparse PCA methods by studying which combination of PCA formulation and sparse method best serves data analysis purposes from the perspective of several existing approaches and statistical measures.

Proposing methods that find optimal solutions remains one of the main topics in the optimization literature of sparse PCA. In terms of computational complexity, the cardinality-constrained PCA problem is an NP-hard problem. Recently, a few new methods have been proposed such that they provided near optimal certifiable solutions with proven better performance than other sparse PCA methods regarding objective value (Berk & Bertsimas, 2019; Bertsimas et al., 2022; Li & Xie, 2020). However, these methods can not handle data sets with many variables, which are common in many scientific applications nowadays (e.g., genetics). The use of penalized PCA methods is motivated by the need to obtain sparse solutions rapidly, their computational tractability, scaling, and their statistical properties of shrinkage. Nevertheless, the benefits of penalized methods have been assessed only via numerical experiments. There are no theoretical guarantees of optimality; that is, existing methods rely on heuristic solutions. This dissertation also aims to study, from a theoretical point of view, the optimality properties of two well-known penalized PCA methods.

This dissertation focuses on solving different optimization problems associated with sparse PCA. Focusing on these problems, we do not engage in some practical implementation steps regarding the estimation or selection of hyperparameters, such as the dimension of the new set of variables, the cardinality per loadings or weights, or the penalization parameter estimation. All these values are unrealistically assumed to be known when solving each optimization problem, although we suggest some ways to estimate or select these values.

1.3 Outline of the dissertation

This section summarizes the content of each chapter. Since some chapters were written independently as journal articles, there may be some overlaps and inconsistencies in terminology and notation between chapters.

Chapter 2 When implementing PCA methods for EDA or DR, weights and loadings converge to the same solution due to the mathematical relationship between the SVD and the EVD (I. T. Jolliffe & Cadima, 2016). However, the weights and loadings are no longer the same when cardinality constraints or sparsity-inducing penalties are added to obtain sparse solutions. Chapter 2 clarifies this misconception and offers guidelines for choosing among different sparse PCA methods. We thoroughly discuss several sparse PCA formulations and methods regarding whether sparsity is imposed on the loadings or the weights. Through an extensive numerical experiment with synthetic data sets, we assess the performance of these methods on measures such as the squared relative error, the misidentification rate, and the percentage of explained variance. Chapter 2 ends with two empirical implementations, one using item scores on a questionnaire measuring the Big Five personality traits (Dolan, Oort, Stoel, & Wicherts, 2009) and the other using gene expression profiles of lymphoblastoid cells (Nishimura et al., 2007). The former is aimed at EDA, and the latter is aimed at DR purposes.

Chapter 3 In the regression literature, obtaining the sparse optimal solution through the cardinality constraint is an NP-hard problem (Natarajan, 1995). Penalized regression methods have been put forward in the literature to obtain a sparse solution due to their computational tractability and statistical shrinkage properties that avoid inflation of the coefficients for a better trade-off between bias and variance. However, significant progress has been made in solving the sparse regression problem via cardinality constraint for a large number of variables to global optima (Bertsimas, King, & Mazumder, 2016; Bertsimas & Van Parys, 2020). Chapter 3 focuses on the least squares approach to PCA (Zou et al., 2006) and compares two methods to achieve sparsity in the weights, using a sparsity-inducing penalty and its cardinality-constrained counterpart. The performance of these two methods is compared using different statistical measures, such as the recovery rate of the sparse structure, mean absolute bias, mean variance, and mean squared error. Finally, both methods are illustrated with an empirical application in a high-dimensional data set that contains gene expression profiles.

Chapter 4 Penalized PCA has been used in the literature to obtain a sparse solution to the PCA problem, as such methods are computationally tractable and have good scaling and statistical properties (see Chapter 3). However, penalized PCA methods rely on heuristic methods without guaranteeing optimality (Bertsimas et al., 2022). Chapter 4 studies the necessary optimality conditions for a penalized

PCA problem. Using an alternating scheme, we characterize the sparsity-inducing penalties that lead to a solution given by a continuous thresholding rule. In addition, a mathematical analysis is conducted, showing that the solution provided by this method satisfies the necessary optimality condition when the minimum eigenvalue of the covariance matrix is greater than one.

Chapter 5 As discussed in Chapter 4, penalized PCA methods are favorable concerning computational tractability, scalability, and statistical properties. Despite their practical advantage, penalized PCA methods rely on heuristic solutions without guaranteeing optimality. In Chapter 5, we study a penalized PCA problem using cardinality as a sparsity-inducing penalty. To solve this problem, we propose a minorization-maximization method and show that it achieves local optimality of the original problem when the minimum eigenvalue of the covariance matrix is greater than one.

Chapter 2

A guide for sparse PCA: model comparison and applications

Abstract

PCA is a popular tool for exploring and summarizing multivariate data, especially those consisting of many variables. PCA, however, is often not simple to interpret, as the components are a linear combination of the variables. To address this issue, numerous methods have been proposed to sparsify the nonzero coefficients in the components, including rotation-thresholding methods and, more recently, PCA methods subject to sparsity-inducing penalties or constraints. Here, we offer guidelines on how to choose among the different sparse PCA methods. Current literature misses clear guidance on the properties and performance of the different sparse PCA methods, often relying on the misconception that the equivalence of the formulations for ordinary PCA also holds for sparse PCA. To guide potential users of sparse PCA methods, we first discuss several popular sparse PCA methods in terms of where the sparseness is imposed on the loadings or on the weights, assumed model, and optimization criterion used to impose sparseness. Second, using an extensive simulation study, we assess each of these methods by means of performance measures such as squared relative error, misidentification rate, and percentage of explained variance for several data generating models and conditions for the population model. Finally, two examples using empirical data are considered.

Keywords: dimension reduction, exploratory data analysis, high dimension-low sample size, regularization, sparse principal component analysis

Guerra-Urzola, R., Van Deun, K., Vera, J.C., & Sijtsma, K. A Guide for Sparse PCA: Model Comparison and Applications. *Psychometrika* **86**, 893–919 (2021). https://doi.org/10.1007/s11336-021-09773-2

2.1 Introduction

Principal component analysis (PCA) is one of the oldest and most popular multivariate analysis techniques used to summarize a (large) set of variables in low dimension with minimum loss of information (I. T. Jolliffe & Cadima, 2016; Wold et al., 1987). In particular, PCA is one of the most popular techniques used to analyze (ultra-) high-dimensional data consisting of many more variables than observations, and its use has become more widespread over recent years. PCA is mainly used to summarize the individual variables' scores by a few derived components based on a linear combination of the individual variables. These new variables are known as component scores and are often used as a data pre-processing step to deal with a large number of variables, e.g., to reduce the number of predictor variables to account for collinearity issues in regression analysis. The coefficients of the linear combination, used to derive the component scores, are known as component weights (Adachi & Trendafilov, 2016). Additionally, PCA can give insight into the data structure via the correlation between component scores and variables. These correlations are known as component loadings.

In PCA, there is a long-standing tradition to look for sparse representations where the variables are associated with only one or a few components (Kaiser, 1958). The sparse structure facilitates interpretation, and the need for such a representation is especially warranted in the case of an extensive collection of variables. Moreover, sparse representations have been employed not only for interpretational issues but also to deal with the inconsistency of the estimated component loadings or weights in the high-dimensional setting (Johnstone & Lu, 2009).

There is a substantial volume of work in sparse PCA based on different formulations of PCA and using different approaches to achieve sparsity. We categorize sparse PCA methods by their estimation aim: sparse loadings or sparse weights. To obtain sparse loadings, Kaiser (1958), I. T. Jolliffe (1995), Cadima and Jolliffe (1995), and Kiers (1994) used a rotation of the PCA solution to obtain a simple structure, and H. Shen and Huang (2008), and Papailiopoulos, Dimakis, and Korokythakis (2013) introduced a least-squares low-rank approximation with sparsity inducing penalties such as the lasso (Tibshirani, 1996). For sparse weights, I. T. Jolliffe, Trendafilov, and Uddin (2003) modified the original PCA problem to satisfy the lasso penalty (SCoTLASS) while Zou et al. (2006) used a lasso penalized least-squares approach to obtain sparsity. d'Aspremont et al. (2004) and d'Aspremont et al. (2007) established a sparse PCA method subject to a cardinality constraint based on semidefinite programming (SDP), while Journée et al. (2010) and X. T. Yuan and Zhang (2013) introduced variations of the well-known power method to achieve sparse PCA solutions using sparsity inducing penalties.

Most of the formulations for sparse PCA are based on different formulations of PCA; thus, the corresponding optimization problems solved are different and—unlike ordinary PCA—do not yield equivalent solutions. Importantly, the different methods result in sparse estimates for different model structures. Hence, the selected method should depend on the objective of the analysis and the assumed

model structure for which sparsity is desired. These differences in sparse PCA formulations have remained mostly unnoticed in the literature, which highlights the need for a thorough comparison of the methods under different data generating models—imposing sparsity on different model structures—and concerning different performance measures. The objective of our research is to provide a guide for using sparse PCA, emphasizing the differences in purposes, objectives, and performance among several sparse PCA approaches. We present a review of the most relevant sparse PCA methods used for sparse loadings and sparse weights estimation. We assess these methods by conducting an extensive simulation study using three types of sparse data structures and performance measures such as squared relative error, misidentification rate, and percentage of explained variance. Finally, we use two empirical data sets to illustrate how to use these methods in practice. The data sets consist of item scores on a questionnaire measuring the Big Five personality (Dolan et al., 2009) and gene expression profiles of lymphoblastoid cells used to distinguish different forms of autism (Nishimura et al., 2007). The former example relies on questionnaire data for which researchers wish to understand the correlation patterns in the data (e.g., knowing which items are highly correlating and hinting at an underlying component or construct). In contrast, the latter example relies on high-dimensional data collected in a classification setting where a reduction of the large set of variables is performed as a pre-processing step[1]. Results from the simulation study and empirical applications suggest that sparse loadings methods are more suitable for exploratory data analysis, while sparse weights methods are more suitable for summarization.

The paper is organized as follows. Sect. 2.2 describes different approaches and drawbacks of PCA. In Sect. 2.3, the leading methods for sparse PCA are briefly discussed. Simulation studies are presented in Sect. 2.4 and two examples using empirical data sets are presented in Sect. 2.5. Concluding remarks are made in section 2.6. Next, we collect our notation for our readers' convenience.

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 will use capital letters (of the letter used to run an index) to denote cardinality (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 l_0-norm $\|\mathbf{x}\|_0$ is the number of nonzero elements of $\mathbf{x}$, the l_1-norm is defined by $\|\mathbf{x}\|_1 = \sum_{j=1}^J |x_j|$, and the Euclidean distance by $\|\mathbf{x}\| = (\sum_{j=1}^J x_j^2)^{1/2}$. Given a matrix $\mathbf{X} \in \mathbb{R}^{I\times J}$, its i-th row and j-th column entry is denoted by $x_{i,j}$, $\|\mathbf{X}\|_F^2 = \sum_{i=1}^I \sum_{j=1}^J |x_{i,j}|^2$ denotes the squared Frobenius norm, and $\mathrm{Tr}(\mathbf{X}) = \sum_{i=1}^I x_{i,i}$ denotes the trace operator when $\mathbf{X}$ is square matrix ($I = J$). We use the notation $\mathbf{X}_K \in \mathbb{R}^{I\times K}$, with $K < J$, for the matrix whose columns are the first K columns of $\mathbf{X}$. Given a scalar $\delta \in \mathbb{R}$, $[\delta]_+ = \max(0, \delta)$. The soft-thresholding operator is defined as $S(x, \lambda) = \mathrm{sign}(x)[|x| - \lambda]_+$, where sign denotes the sign of x. Finally, when formulating an optimization problem, s.t. means "subject to".

[1]The MATLAB and R codes used to perform the simulation study and applications are available from `https://github.com/RosemberGuerra/sparsePCA`

2.2 Principal Component Analysis Overview

This section aims to review different formulations for PCA and their relation to the Singular Value Decomposition (SVD) and the Eigenvalue Decomposition (EVD). PCA formulations are presented in Sect. 2.2.1. Sect. 2.2.2 discusses the lack of consistency in the estimation of the component loadings/weights and the difficulties to interpret the component scores—the main drawbacks of PCA found in the literature. Let us define $\mathbf{X} \in \mathbb{R}^{I \times J}$ as the data matrix (i.e., I observations and J variables) and $K < J$ as the number of desired components. Without loss of generality, we follow the common practice of assuming that all the data are centered and scaled to unit variance, that is $\mathbf{X}^\top \mathbf{1}_I = \mathbf{0}_J$ and $\widehat{\mathbf{\Omega}} = \frac{1}{I-1}\mathbf{X}^\top\mathbf{X}$ denotes the sample correlation matrix (I. T. Jolliffe & Cadima, 2016).

2.2.1 PCA Formulations

Several disciplines rely on the following structure for the data set (Whittle, 1952),

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E}, \tag{2.1}$$

where $\mathbf{T} \in \mathbb{R}^{I \times K}$, $\mathbf{P} \in \mathbb{R}^{J \times K}$, $\mathbf{P}^\top\mathbf{P} = \mathbf{I} \in \mathbb{R}^{K \times K}$, $\mathbf{I}$ denotes de identity matrix, and $\mathbf{E} \in \mathbb{R}^{I \times J}$ is the error matrix uncorrelated to $\mathbf{T}\mathbf{P}^\top$. $\mathbf{P}$ is called the component loadings matrix, and $p_{j,k}$ are the component loadings, which express the strength of the connection between the variables and the component scores $\mathbf{T}$. In this model, the component scores are linear combinations of the original variables; therefore, they can be expressed as $\mathbf{T} = \mathbf{X}\mathbf{W}$, where the elements $w_{j,k}$ express the weights used in this combination. The elements of the matrix $\mathbf{W} \in \mathbb{R}^{J \times K}$ are named component weights. For this approach, the goal of PCA is to minimize the squared Frobenius norm of the error matrix $\mathbf{E}$ (also known as the least-squares approach). The problem is formulated as:

$$\begin{aligned} (\widehat{\mathbf{T}}, \widehat{\mathbf{P}}) = \underset{\mathbf{T},\mathbf{P}}{\operatorname{argmin}} \quad & \left\|\mathbf{X} - \mathbf{T}\mathbf{P}^\top\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{P}^\top\mathbf{P} = \mathbf{I}. \end{aligned} \tag{2.2}$$

A solution of problem (2.2) can be obtained from the truncated SVD of $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$, with $\mathbf{U} \in \mathbb{R}^{I \times K}$ and $\mathbf{V} \in \mathbb{R}^{J \times K}$ semi-orthogonal matrices such that $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I} \in \mathbb{R}^{K \times K}$ and $\mathbf{D} \in \mathbb{R}^{K \times K}$ a diagonal matrix (Eckart & Young, 1936). Thus, $\widehat{\mathbf{T}} = \mathbf{U}\mathbf{D}$ and $\widehat{\mathbf{P}} = \mathbf{V}$ provide the solution of problem (2.2).

In psychometrics, it is common to find PCA formulations where problem (2.2) is modified as follows (ten Berge, 1986),

$$\begin{aligned} (\widehat{\mathbf{T}}, \widehat{\mathbf{P}}) = \underset{\mathbf{T},\mathbf{P}}{\operatorname{argmin}} \quad & \left\|\mathbf{X} - \mathbf{T}\mathbf{P}^\top\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{T}^\top\mathbf{T} = (I-1)\mathbf{I}. \end{aligned} \tag{2.3}$$

The solution of problem (2.3) can be obtained using the SVD of $\mathbf{X}$ by taking $\widehat{\mathbf{T}} = (I-1)^{1/2}\mathbf{U}$ and $\widehat{\mathbf{P}} = (I-1)^{-1/2}\mathbf{V}\mathbf{D}^2$. Hence,

$$\begin{aligned}\widehat{\mathbf{T}} &= (\mathbf{X}-\mathbf{E})\mathbf{P}(\mathbf{P}^\top\mathbf{P})^{-1}\\ &= (I-1)^{1/2}\mathbf{X}\mathbf{V}\mathbf{D}^{-1}.\end{aligned}$$

Therefore, the component weights matrix for problem (2.3) is $\widehat{\mathbf{W}} = (I-1)^{1/2}\mathbf{V}\mathbf{D}^{-1}$. Additionally, problem (2.3) is commonly formulated as an explicit combination of the original variables (ten Berge, 2005), considering $\mathbf{T} = \mathbf{X}\mathbf{W}$ that is,

$$\begin{aligned}(\widehat{\mathbf{W}}, \widehat{\mathbf{P}}) = \underset{\mathbf{W},\mathbf{P}}{\operatorname{argmin}} \quad & \left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\right\|_F^2\\ \text{s.t.} \quad & \mathbf{T}^\top\mathbf{T} = (I-1)\mathbf{I}.\end{aligned}$$

The classical way to define PCA is to find the component weight matrix $\mathbf{W} \in \mathbb{R}^{J\times K}$, having orthogonal vectors that maximize the variance of the components. Formally, consider the following formulation:

$$\begin{aligned}\widehat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}} \quad & \operatorname{Tr}\left(\mathbf{W}^\top\widehat{\mathbf{\Omega}}\mathbf{W}\right)\\ \text{s.t.} \quad & \mathbf{W}^\top\mathbf{W} = \mathbf{I}.\end{aligned} \tag{2.4}$$

A solution for problem (2.4) can be obtained from the EVD (Hotelling, 1933) of the covariance matrix $\widehat{\mathbf{\Omega}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, taking $\widehat{\mathbf{W}} = \mathbf{V}$ as the matrix formed by eigenvectors corresponding the K largest eigenvalues.

The orthogonality constraints in PCA formulations (2.2) and (2.4) and principal axes orientation imply their equivalence. More precisely, component loadings and component weights are both equal to $\mathbf{V}$. To see this, notice that using the SVD of $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, the EVD for $\mathbf{\Omega} = \mathbf{X}^\top\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^\top$ is obtained (I. T. Jolliffe & Cadima, 2016). Thus, $\mathbf{D}^2$ is the diagonal matrix containing the eigenvalues of $\mathbf{\Omega}$ (the square of the singular values of $\mathbf{X}$) in decreasing order: $d_{11}^2 \geq d_{22}^2 \geq \ldots \geq d_{JJ}^2$. Then, the matrix of component weights $\widehat{\mathbf{W}} = \mathbf{V}$ coincides with the matrix $\widehat{\mathbf{P}}$ of component loadings defined by PCA formulation (2.2). However, this equivalence does not hold exactly for PCA formulation (2.3) because the orthogonality constraint is imposed on the component scores. Instead, under formulation (2.3), $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{P}}$ are proportional to $\mathbf{V}$.

[2]It can be shown that the element $p_{j,k}$ is the correlation between variable $\mathbf{x}_j$ and component score $\mathbf{t}_k$

2.2.2 PCA Drawbacks

Interpretation and Non-uniqueness

Principal component scores are a linear combination of the original variables. That makes them difficult to interpret. For instance, when using data containing measures with different units, the linear combination does not have a definite meaning. A common practice to tackle this problem is to use the correlation matrix instead of the covariance matrix (I. T. Jolliffe & Cadima, 2016). That is to standardize the variables so all of them are on the same scale.

Rotation techniques are commonly used to help practitioners interpret the component loadings. The rotation is done to obtain component loadings values close to either 0 or 1, such that only the most relevant variables are considered for interpretation purposes (see Sect. 2.3.1 for further discussion). The rotation can be implemented using an orthogonal rotation matrix $\mathbf{Q}$, which does not modify the amount of variance accounted by all components together but rather redistributes the variance across the variables by choosing a different system of orthogonal axes. However, because of the several possible choices for the rotation matrix $\mathbf{Q}$, non-unique solutions in problems (2.2) and (2.4) are achieved (Hastie et al., 2000).

Inconsistency in the High-Dimensional Setting

As mentioned above, the solution of the model-free PCA formulation (2.4) is the leading eigenvector of the covariance matrix. The inconsistency of this leading eigenvector has been studied by analyzing the angle between its population and estimated value under different asymptotical conditions for the dimensionality of the data set. For instance, Johnstone and Lu (2009) show that

$$P\left(\lim_{I\to\infty} R^2(\hat{\mathbf{v}}_1, \mathbf{v}_1) = R^2_\infty(\omega, c)\right) = 1,$$

where $\mathbf{v}_1$ is the leading population eigenvector, $\hat{\mathbf{v}}_1$ its estimate, and $R^2(\hat{v}_1, v_1)$ the cosine of the angle between $\hat{\mathbf{v}}_1$ and $\mathbf{v}_1$. $\omega > 0$ stands for the limiting signal-to-noise ratio, $c = \lim_{I\to\infty} J/I$, and $R^2_\infty = (\omega^2 - c)_+/(\omega^2 + c\omega)$. This result implies that $\hat{\mathbf{v}}_1$ is a consistent estimate of $\mathbf{v}_1$ if and only if $c = 0$. Therefore, in the high-dimensional setting ($J \gg I$), the estimator of the component weights in the PCA formulation (2.4) is inconsistent. Similarly, the estimation of the leading eigenvalue is shown to be inconsistent under random matrix theory (e.g., when I and J tend to infinity and the ratio I/J converges to a constant) (Baik & Silverstein, 2006; Johnstone & Lu, 2009; Nadler, 2008; Paul, 2007) and in the high-dimensional low sample (HDLS) (e.g., J tends to infinity, and I is fixed) (Jung & Marron, 2009; D. Shen, Shen, & Marron, 2016). On the other hand, Jung and Marron (2009) shows that when I is fixed, the angle between $\hat{\mathbf{v}}_1$ and $\mathbf{v}_1$ goes to 0 with probability 1 if the leading eigenvalues are extremely large in comparison with the number of variables J, yet

the components scores are shown to be inconsistent (D. Shen, Shen, Zhu, & Marron, 2016).

2.3 Sparse Principal Component Analysis Overview

Sparse PCA has been proposed as a solution to the difficulties encountered in interpreting the component scores of ordinary PCA, non-uniqueness, and the inconsistency of the component loadings/weights (c.f. Sect. 2.2.2). Research efforts have focused on reformulations for PCA, where component loadings or component weights have as many zero elements as possible. In this section, we present six sparse PCA methods that are well established in the literature and for which implementations are available. Our selection of methods was also chosen to reflect the different PCA formulations (2.2), (2.3), and (2.4). This section aims to show the differences in the purposes and objectives of sparse PCA methods. The emphasis is on the fact that while the ordinary PCA formulations (2.2) and (2.4) are equivalent (see Sect. 2.2.1), for sparse PCA the corresponding formulations are not equivalent, so that the obtained results heavily depend on the chosen methodology. Sparse PCA methods for estimating the loadings are presented in Section 2.3.1 while sparse PCA methods for estimating the weights are presented in 2.3.2.[3]

2.3.1 Sparse Loadings

Principal component analysis, when used to explore structure and patterns in data, relies on the model structure presented in Eq. (2.1). Interpreting the components is based on inspecting the loadings because these reveal how strongly the variables contribute to the components. More precisely, in problem (2.2), the component loadings $\mathbf{P}$ represent the regression coefficients in the multiple regression of $\mathbf{x}_j$ on the k component scores $\mathbf{t}_k$.[4] Note that with orthogonal component scores, this is a regression problem with independent predictors, and with proper normalization constraints, the loading is equal to the correlation. Then, having sparse component loadings gives a clearer interpretation in the sense that variables are explained only by one or a few components. In this section, we present two frequently used methodologies for this purpose.

Sparse PCA Via Rotation and Thresholding: Varimax and Simplimax

The first attempts to achieve a component structure with variables being explained by one component only while having zero loadings for the other components are simple structure rotations followed by thresholding. Simple structure rotation,

[3] Ning-min and Jing (2015), Trendafilov (2014), Zou and Xue (2018) give a wide list of more methods for both purposes

[4] Observe that from (2.1) it follows that $\mathbf{x}_j = \sum_k \mathbf{t}_k p_{j,k} + \mathbf{e}_j$ which is the linear regression equation with dependent variable $\mathbf{x}_j$ and predictor variables $\mathbf{t}_k$.

which was adopted from factor analysis, (I. Jolliffe, 2002; I. T. Jolliffe, 1989, 1995, Chap. 11), relies on the rotational freedom of Eq. (2.1):

$$\begin{aligned}\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E} = \mathbf{T}(\mathbf{Q}^{-\mathbf{1}})^\top(\mathbf{P}\mathbf{Q})^\top + \mathbf{E} \\ \mathbf{X} = \mathbf{T}_{rotated}\mathbf{P}_{rotated}^\top + \mathbf{E}\end{aligned} \tag{2.5}$$

with $\mathbf{Q}$ a non-singular transformation matrix usually orthogonal (hence $\mathbf{Q}$ is a rotation matrix) or oblique[5] (Jennrich, 2004, 2006).

This approach is applied in two steps. First, the component scores and component loadings are obtained from solving problem (2.2). Second, a rotation matrix $\mathbf{Q}$ is found by optimizing a criterion that leads to a simple structure of $\mathbf{PQ}$. In this study, we consider two well-known methods: Varimax (Kaiser, 1958) that maximizes the variance of the squared component loadings, hence encouraging loadings to be as close to either 0 or 1 as possible, and Simplimax (Kiers, 1994) that finds an oblique matrix such that the rotated loading matrix comes closest (in the least square sense) to a matrix with (at least) a given number of zero values. Oblique rotation matrices are often used when the component scores are expected to be correlated. The rotated loadings will—in general—not be precisely zero, but in practice, small loadings are neglected (including not printing the value of small loadings in leading software packages such as SPSS), which boils down to treating them as having a zero value (I. Jolliffe, 2002, p.269). This practice is called thresholding and is considered *ad hoc*. Importantly, as discussed by Cadima and Jolliffe (1995), the thresholding approach is misleading in the sense that another subset of variables may better approximate the data as in Eq. (2.5).

Sparse PCA Via Regularized SVD: sPCA-rSVD

Taking the close connection between the SVD and PCA as a point of departure, H. Shen and Huang (2008) proposed a sparse PCA method based on adding a regularization penalty to the least-squares PCA criterion in problem (2.3). Their so-called sparse PCA via regularized SVD (sPCA-rSVD) method solves the following problem:

$$\begin{aligned}(\hat{\mathbf{t}}, \hat{\mathbf{p}}) = \underset{\mathbf{t},\mathbf{p}}{\operatorname{argmin}} \left\|\mathbf{X} - \mathbf{t}\mathbf{p}^\top\right\|_F^2 + \mathcal{P}_\lambda(\mathbf{p}) \\ \text{s.t.} \quad \|\mathbf{t}\|_2^2 = 1,\end{aligned} \tag{2.6}$$

where $\hat{\mathbf{t}}\hat{\mathbf{p}}^\top$ is the best rank-one approximation of the data matrix $\mathbf{X}$ (Eckart & Young, 1936), $\mathbf{t}$ is the first component score vector and $\mathbf{p}$ the corresponding loading vector and $\mathcal{P}_\lambda$ a particular penalty term that imposes sparsity over the component loadings. Three different sparsity inducing penalties are considered in H. Shen and

[5] A non-singular matrix $\mathbf{Q} \in \mathbb{R}^{K\times K}$ is called oblique if $\mathbf{Q}^\top\mathbf{Q}$ is a correlation matrix (Trendafilov, 2014).

Huang (2008), including the l_1-norm of the loadings, also known as the lasso. Problem (2.6) is used to find the first component score and component loading vectors, the subsequent pairs $(\hat{\mathbf{t}}_\mathbf{k}, \hat{\mathbf{p}}_\mathbf{k})$ with $k > 1$ are obtained by solving problem (2.6) for the residual matrix (i.e., $\mathbf{X} - \hat{\mathbf{t}}\hat{\mathbf{p}}^\top$). H. Shen and Huang (2008) solved the problem by alternating between the optimization of $\mathbf{t}$ given $\hat{\mathbf{p}}$ and $\mathbf{p}$ given $\hat{\mathbf{t}}$; they also discuss that the conditional optimization problem of the loadings is separable in the variables. Such separability has two major advantages. First, all loadings can be optimized simultaneously using simple expressions (e.g., soft-thresholding of the inner product of the observed variable and component scores), which implies very efficient computation even in the high-dimensional setting; Second, it means that the problem can be solved for a fixed number of zero coefficients. Trendafilov and Adachi (2015) used this advantages to solve the least-squares PCA problem (2.3) with orthogonal $\mathbf{T}$ for $k > 1$ subject to a cardinality constraint.

2.3.2 Sparse Weights

In this section, we present different methodologies to estimate the sparse component weights matrix $\mathbf{W}$. Given that the role of $\mathbf{W}$ is to weight the original variables to form $\mathbf{T} = \mathbf{XW}$, sparsity is desired on $\mathbf{W}$. In this way, the component scores $\mathbf{T}$ would be summarized by a weighted linear combination of those variables in $\mathbf{X}$ with nonzero elements in $\mathbf{W}$.

Sparse PCA Via Elastic Net Regularization: SPCA

One of the most popular methods for PCA with sparse component weights was proposed by Zou et al. (2006). They showed that the component weights[6] are proportional to the solution of the ridge regression, and sparsity can be attained by adding a lasso penalty. Zou et al. (2006) proposed to solve the following problem

$$(\widehat{\mathbf{W}}, \widehat{\mathbf{P}}) = \underset{\mathbf{W},\mathbf{P}}{\operatorname{argmin}} \left\|\mathbf{X} - \mathbf{XWP}^\top\right\|_F^2 + \sum_{k=1}^{K} \lambda \left\|\mathbf{w}_k\right\|^2 + \sum_{k=1}^{K} \lambda_{1,k} \left\|\mathbf{w}_k\right\|_1 \quad \text{s.t. } \mathbf{P}^\top\mathbf{P} = \mathbf{I}. \tag{2.7}$$

The terms $\sum_{k=1}^{K} \lambda \left\|\mathbf{w}_k\right\|^2$ and $\sum_{k=1}^{K} \lambda_{1,k} \left\|\mathbf{w}_k\right\|_1$ are the ridge and lasso penalties, respectively. To solve the problem (2.7) for given values of λ and $\lambda_{1,k}$, Zou et al. (2006) proposed an alternating minimization algorithm while updates $\mathbf{W}$ and $\mathbf{P}$ alternately with the other variable is fixed to its current estimate until some stopping criterion is reached. The update of $\mathbf{P}$ conditional upon fixed $\mathbf{W}$ is the orthogonal Procrustes rotation problem with known optimal solution (S., Golub, & Loan, 1991). The conditional update of the weights $\mathbf{W}$ can be written as an elastic net regression problem that regresses the component scores $\mathbf{t}_\mathbf{k}$ on the J variables $\mathbf{x}_\mathbf{j}$ (Zou & Hastie,

[6] Referred as loadings in Zou et al. (2006).

2005). Note that in the high-dimensional setting, this becomes a high-dimensional regression problem with known numerical issues (Hastie, Tibshirani, & Friedman, 2009). Then, as the lasso penalty yields at most I nonzero coefficients, in the high-dimensional setting, the ridge penalty is included. Efficient procedures have been proposed for the elastic net regression problem, such as the LARS-EN algorithm (Efron et al., 2004), cyclic coordinate descent (Friedman, Hastie, Höfling, & Tibshirani, 2007), and proximal gradient techniques (Beck & Teboulle, 2009). However, these algorithms remain subject to computational issues in the high-dimensional setting (G. X. Yuan, Ho, & Lin, 2011). Furthermore, a major challenge when using the elastic net method is proper tuning of the penalties. In this respect, the LARS-EN algorithm has the benefit that it allows defining the number of nonzero values a priori.

Sparse PCA Via Cardinality Penalty: pathSPCA

d'Aspremont et al. (2007) focused on the problem of maximizing the variance of the components with a cardinality penalty,

$$\hat{\mathbf{w}} = \underset{\|\mathbf{w}\|\leq 1}{\operatorname{argmax}} \|\mathbf{X}\mathbf{w}\|^2 - \rho \|\mathbf{w}\|_0 , \tag{2.8}$$

with ρ a parameter controlling the sparsity. d'Aspremont et al. (2007) proposed a greedy algorithm that provides candidate indexes I_r for r nonzero elements. Then the sparse component weights vector is the solution of the problem (2.8) given I_r, which is:

$$\hat{\mathbf{w}} = \underset{\left\{\mathbf{w}_{I_r^c}=0,\, \|\mathbf{w}\|=1\right\}}{\operatorname{argmax}} \|\mathbf{X}\mathbf{w}\|^2 - \rho r,$$

where I_r^c is the complement set of I_r, e.g., the position with zero element in $\mathbf{w}$. This algorithm is called *pathSPCA*.

Sparse PCA Via Lasso Penalty: GPower

Journée et al. (2010) showed that the sparse PCA formulation based on maximizing the (scaled) standard deviation of the component scores using a lasso penalty,

$$\hat{\mathbf{w}} = \underset{\|\mathbf{w}\|=1}{\operatorname{argmax}} \|\mathbf{X}\mathbf{w}\| - \lambda \|\mathbf{w}\|_1 , \tag{2.9}$$

is equivalent to solving initially:

$$\hat{\mathbf{z}} = \underset{\|\mathbf{z}\|\leq 1}{\operatorname{argmax}} \left\|S(\mathbf{X}^\top \mathbf{z}, \lambda)\right\|^2 , \tag{2.10}$$

where the soft-thresholding function $S(\mathbf{X}^\top \mathbf{z}, \lambda)$ is applied component wise. Once $\hat{\mathbf{z}}$ is obtained, $\hat{\mathbf{w}} = S(\mathbf{X}^\top \hat{\mathbf{z}}, \lambda) / \left\|S(\mathbf{X}^\top \hat{\mathbf{z}}, \lambda)\right\|$, which gives the sparsity pattern $S(\mathbf{X}^\top \hat{\mathbf{z}}, \lambda)$

for $\mathbf{w}$. Then, the component weights are obtained via the ordinary PCA (problem (2.4)) by removing the corresponding zero variables from the original data set $\mathbf{X}$. Note that the problem of solving for the J-dimensional vector $\hat{\mathbf{w}}$ is reformulated in terms of solving for a I-dimensional vector $\mathbf{z}$. In the high-dimensional setting, this avoids to search in a large space. A gradient scheme is used to solve problem (2.10). Additionally to problem (2.9), Journée et al. (2010) also considered the problem of maximizing the variance subject to a cardinality penalty.

2.3.3 Sparse PCA: Summary

PCA can be formulated as optimization problems with the solutions happening to be equivalent (see Sect. 2.2.1). However, when having sparsity constraints in the formulation, neither the SVD of the data set nor the EVD of the covariance matrix is the solution to the sparse PCA problem. Given the lack of awareness of the different formulations and goals of PCA, it is not clear whatsoever when to use which method. In this section, we have discussed several methods for sparse PCA that all share the principle of Ockham's razor to represent the data in a reliable though simple way. Table 2.1 summarizes the described methods: each of them imposes sparsity either on the component loadings or on the component weights. The last column of Table 2.1, "Algorithm", indicates whether components are extracted one by one (deflation approach) or all together (block approach).

To impose sparsity, PCA methods rely on one of three popular techniques: rotation, the addition of a penalty, or a constraint (usually l_0 or l_1[7]). Many of the sparse PCA formulations are complex to solve, and a considerable amount of work is of an algorithmic nature; proposed algorithms are often subject to local optima and without guaranteed convergence. Moreover, some of the procedures also fail in terms of memory or are very slow to compute. Such algorithmic issues are not the focus here, yet they may affect the numerical performance of the methods.

Table 2.1: Summary of methods for sparse PCA

Method	Estimated	Objective	Sparsity	Algorithm
VARIMAX	**P**	Rotation	Threshold	Block
SIMPLIMAX	**P**	Rotation	Threshold	Block
sPCA-rSVD	**P**	low-rank	l_1	Deflating
SPCA	**W**	Max. variance	l_1 and l_2	Block
pathSPCA	**W**	Max. variance	l_0	Deflating
GPower	**W**	Max. variance	l_1	Deflating

[7]Note that for l_1 it is possible to find a dual representation though this is not always the case for the l_0 pseudo-norm; see, e.g., Bertsimas et al. (2016).

2.4 Simulation Study

A crucial question that we want to address using simulated data is when to use which sparse PCA method. As discussed throughout the paper, choosing the proper approach depends on the assumed model (sparse component loadings, sparse component weights, or both) and the performance of the method concerning various criteria. Here, we will use four measures to assess the performance of the six sparse PCA methods discussed in Sect. 3.2.2.

2.4.1 Design

An essential factor in any simulation is the assumed data-generating model. Most of the reported simulation studies for sparse PCA are based on the spiked covariance model for which data follow a multivariate distribution with zero mean, variance $\mathbf{\Omega} = \mathbf{V}\mathbf{D}\mathbf{V}^\top$, with sparse leading eigenvectors $\mathbf{V}_K$, and the K largest eigenvalues much larger than the remaining ones. Papers using this model include Zou et al. (2006), H. Shen and Huang (2008), Johnstone and Lu (2009). Another model that has been considered is the sparse standard factor model that relies on Eq. (2.1), that is, $\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E}$ with $\mathbf{P}$ sparse, and noise $\mathbf{E}$ independent of the components scores $\mathbf{T}$; see Adachi and Trendafilov (2016) for an example of a simulation study using this model. Also, more relaxed versions have been considered under the same name.[8] Here, we will rely on three versions of the 'factor model' set up such that they correspond to the data model structure assumed by the sparse PCA methods considered in this study. First, consider

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E} \tag{2.11}$$

with $\mathbf{P}$ sparse and $\mathbf{T}^\top\mathbf{T} = \mathbf{I}$; note that the model in Eq. (2.11) corresponds to the structure imposed by Adachi and Trendafilov (2016). Second, considering the component scores explicitly as a function of the weights,

$$\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{P}^\top + \mathbf{E} \tag{2.12}$$

with $\mathbf{W}$ sparse and, third, the same model in Eq. (2.12) but, with $\mathbf{P}$ and $\mathbf{W}$ being sparse simultaneously.

For generating the synthetic data sets, besides the data-generating model, we also considered the following factors and levels: sample size with levels $I = 100, 500$, number of variables with levels $J = 10, 100, 1000$, number of components with levels $K = 2, 3$, percentage of variance accounted for the data set with levels VAF $= 80\%, 95\%, 100\%$, and proportion of sparsity with levels PS $= 0.0, 0.5, 0.8$ or PS $= 0.7, 0.8, 0.9$ when data are generated with component loadings and component weights being equal, sparse, and orthogonal. These higher levels of sparsity

[8] Note that outside psychology, the least-squares model with component scores and loadings is often wrongly named factor model.

allow for avoiding overlap of the nonzero values, making it possible to have sparse structures that are orthogonal. For each of the three types of models, a fully crossed design was used, resulting in $2 \times 3 \times 2 \times 3 \times 3 = 108$ conditions. For each condition, 100 data sets were generated, ending up with a total of $10,800$ data sets in each of the three data generating regimes. The data generation design is summarized in Table 2.2.

TABLE 2.2: Simulation design factors and their levels

Model	sparse	I	J	K	VAF	PS	Repetions
$\mathbf{X} = \mathbf{TP}^\top + \mathbf{E}$	$\mathbf{P}$	100, 500	10, 100, 1000	2, 3	80%, 95%, 100%	0.0, 0.5, 0.8	100
$\mathbf{X} = \mathbf{XWP}^\top + \mathbf{E}$	$\mathbf{W}$	100, 500	10, 100, 1000	2, 3	80%, 95%, 100%	0.0, 0.5, 0.8	100
$\mathbf{X} = \mathbf{XWP}^\top + \mathbf{E}$	$\mathbf{P}$ and $\mathbf{W}$	100, 500	10, 100, 1000	2, 3	80%, 95%, 100%	0.7, 0.8, 0.9	100

I sample size, J N. of variables, K N. of components, VAF variance accounted, PS proportion of sparsity.

Data were generated using one of three algorithms: Algorithm 1 is used for generating data with a sparse component loadings structure, Algorithm 2 generates data with a sparse component weights structure, and Algorithm 3 generates data with orthogonal and equal sparse component loadings and weights. Every algorithm begins with a rank-K decomposition obtained from the compact SVD decomposition of data generated from a multivariate normal distribution. Algorithm 1 then imposes sparsity on the component loadings $\mathbf{P} = \mathbf{VD}$ and has orthogonal component scores $\mathbf{T} = \mathbf{U}$; Algorithm 2 imposes sparsity on the component weights $\mathbf{W} = \mathbf{V}$. For Algorithm 3, there are two scenarios: (1) For the model that assumes $\mathbf{P}$ sparse, $\mathbf{W} = \mathbf{VD}^{-1}$, and (2) for the models that assume $\mathbf{W}$ sparse, $\mathbf{P} = \mathbf{V}$. Additionally, every algorithm considers additive noise $\mathbf{E}$ distributed according to a multivariate normal distribution with mean $\mathbf{0}$ and variance proportional to the identity matrix, such that the final data set has the desired VAF. This error structure has also been considered in leading sparse PCA papers (e.g., Johnstone & Lu, 2009; H. Shen & Huang, 2008; Zou et al., 2006) while Van Deun et al. (2019) considers generalizations of sparse PCA to data with non-additive noise. Each data set was analyzed using the six sparse PCA methods previously discussed: PCA with simple thresholding of the rotated loadings using either Varimax or Simplimax rotation, sPCA-rSVD, SPCA, pathSPCA, and GPower. Also, the performance of each method on each data set was assessed using the following performance measures: the squared relative error (SRE) of the model parameters, the misidentification rate (MR) of zero versus the nonzero status of the sparse coefficients, the percentage of explained variance (PEV), and the cosine similarity (also known as Tucker's coefficient of congruence). The performance measures are defined as follows.

- The SRE is used to assess how well each method estimates the model component scores, component loadings, and/or component weights. For a matrix $\mathbf{A}$, the SRE is defined by

$$\text{SRE}(\mathbf{A}) = \frac{\left\| \widehat{\mathbf{A}} - \mathbf{A} \right\|_F^2}{\|\mathbf{A}\|_F^2},$$

Algorithm 1: Data generation: Sparse Component loadings.

Input : I, J, K, PS, and VAF
Output: $\mathbf{X} \in \mathbb{R}^{I \times J}$

Generate $\mathbf{X}_{initial}$ by sampling I vectors from $\mathcal{N}(\mathbf{0}_J, \mathbf{I}_J)$;
Obtain $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ via the truncated SVD: $\mathbf{X}_{initial} = \mathbf{UDV}^\top$;
Replace by zero the PS proportion of elements of $\mathbf{V}$ having the smallest absolute value;
Normalize each column of $\mathbf{V}$ to a unit vector;
$\mathbf{P} \leftarrow \mathbf{VD}$;
$\mathbf{T} \leftarrow \mathbf{U}$;
$\mathbf{X} \leftarrow \mathbf{TP}^\top + f\mathbf{E}$ with $\mathbf{E}$ having I vectors drawn from $\mathcal{N}(\mathbf{0}_J, \mathbf{I}_J)$ and f such that VAF $= \|\mathbf{TP}^\top\|^2/(\|\mathbf{TP}^\top\|^2 + f^2\|\mathbf{E}\|^2)$.

Algorithm 2: Data generation: Sparse Component Weights.

Input : I, J, K, PS, and VAF
Output: $\mathbf{X} \in \mathbb{R}^{I \times J}$

Generate $\mathbf{X}_{initial}$ by sampling I vectors from $\mathcal{N}(\mathbf{0}_J, \mathbf{I}_J)$;
Obtain $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ via the truncated SVD: $\mathbf{X}_{initial} = \mathbf{UDV}^\top$;
Replace the elements of $\mathbf{V}$ with the smallest absolute value by 0, according to the level of sparsity;
Normalize each column of $\mathbf{V}$ to a unit vector;
$\mathbf{T} = \mathbf{X}_{initial}\mathbf{V}$;
$\mathbf{P}$ is the solution of $\mathbf{X}_{initial} = \mathbf{TP}^\top$;
$\mathbf{X} \leftarrow \mathbf{X}_{initial}\mathbf{VP}^\top + f\mathbf{E}$ with $\mathbf{E}$ having I vectors drawn from $\mathcal{N}(\mathbf{0}_J, \mathbf{I}_J)$ and f such that VAF $= \|\mathbf{TP}^\top\|^2/(\|\mathbf{TP}^\top\|^2 + f^2\|\mathbf{E}\|^2)$.

Algorithm 3: Data generation: Sparse Component Weights and loadings.

Input : I, J, K, PS, and VAF
Output: $\mathbf{X} \in \mathbb{R}^{I \times J}$

Generate $\mathbf{X}_{initial}$ by sampling I vectors from $\mathcal{N}(\mathbf{0}_J, \mathbf{I}_J)$;
Obtain $\mathbf{U}$, $\mathbf{D}$, and $\mathbf{V}$ via the truncated SVD: $\mathbf{X}_{initial} = \mathbf{UDV}^\top$;
Replace by zero the PS proportion of elements of $\mathbf{V}$ having the smallest absolute value;
Normalize and orthogonalize $\mathbf{V}$, preserving the zero elements;
if *model relies on* $\mathbf{X} = \mathbf{TP}^\top + \mathbf{E}$, **then**
    $\mathbf{T} \leftarrow \mathbf{U}$;
    $\mathbf{W} \leftarrow \mathbf{VD}^{-1}$;
    $\mathbf{P} \leftarrow \mathbf{VD}$;
end
if *model relies on maximization of the variance,* **then**
    $\mathbf{W} \leftarrow \mathbf{V}$;
    $\mathbf{P} = \mathbf{W}$;
    $\mathbf{T} = \mathbf{X}_{initial}\mathbf{W}$;
    $\mathbf{P}$ is the solution of $\mathbf{X}_{initial} = \mathbf{TP}^\top$;
end
$\mathbf{X} \leftarrow \mathbf{TP}^\top + f\mathbf{E}$ with $\mathbf{E}$ having I vectors drawn from $\mathcal{N}(\mathbf{0}_J, \mathbf{I}_J)$ and f such that VAF $= \|\mathbf{TP}^\top\|^2 / (\|\mathbf{TP}^\top\|^2 + f^2\|\mathbf{E}\|^2)$.

with $\widehat{\mathbf{A}}$ representing the estimated matrix. Values close to zero indicate good recovery of the original model matrix by the method, while values close to or higher than one indicate bad recovery. The SRE is calculated for the component scores $\mathbf{T}$, component loadings $\mathbf{P}$, and component weights $\mathbf{W}$. The cosine similarity (or Tucker congruence) between matrices $\mathbf{A}$ and $\mathbf{B}$ with dimension $I \times K$ is defined as

$$CosSim(\mathbf{A}, \mathbf{B}) = \frac{1}{K} \sum_{k=1}^{K} \frac{\mathbf{a}_k^\top \mathbf{b}_k}{\|\mathbf{a}_k\| \, \|\mathbf{b}_k\|} \tag{2.13}$$

with $\mathbf{a}_k$ and $\mathbf{b}_k$ the k-th column of matrix $\mathbf{A}$ and $\mathbf{B}$, respectively. This value is calculated between the estimated component loadings and the population component weights $CosSim(\widehat{\mathbf{P}}, \mathbf{W})$, the estimated component weights and the population component loadings $CosSim(\widehat{\mathbf{W}}, \mathbf{P})$, and the estimated and population component scores $CosSim(\widehat{\mathbf{T}}, \mathbf{T})$. The $CosSim$ is only calculated for the simulation settings representing a mismatch between the sparse constraints imposed by the data generating model and those imposed by the method.

- The misidentification rate assesses how badly each model captures the sparse structure of the data set. MR is defined as the percentage of zero values that are not recovered; that is,

$$\text{MR} = 1 - \frac{\#\text{ of correctly classified zero elements}}{\#\text{ of zero-elements}}.$$

  MR is a value in the interval $[0, 1]$. When MR $= 0$, all zeros in the generated model structure have been estimated as a zero by the sparse PCA method, while MR $= 1$ means that none of the zeros in the model structure has been estimated as a zero by the method. Hence, methods set up to identify the underlying sparse structure should have MR values close to zero. Note that in simulation conditions with the proportion of sparsity set to zero, the MR is not calculated.

- The percentage of explained variance was implemented to assess how well the sparse component solution explains the variance in the generated data. PEV is defined as

$$\text{PEV} = 1 - \frac{\left\| \widehat{\mathbf{X}} - \mathbf{X} \right\|_F^2}{\|\mathbf{X}\|_F^2}.$$

  where $\widehat{\mathbf{X}}$ represents the recovered data set and it is defined as $\widehat{\mathbf{X}} = \widehat{\mathbf{T}}\widehat{\mathbf{P}}^\top$. PEV is a value in the interval $[0, 1]$ and is desired to be close to the variance accounted by the generated data (VAF); a PEV value greater than VAF means that the model extracts some of the residual variation (i.e. the noise), which is a sign of overfitting.

Note that—except for PEV—all performance measures are sensitive to order permutations and changing of the sign of the component scores, loadings, or weights. However, the methods considered here have sign invariance, and some of them also have permutational invariance. Therefore, to make our measurement robust, we considered all possible permutations of the component loadings/weights—including changes in their sign—and calculated all measurements with the combination that produces the minimum SRE (or *CosSim* when used).

2.4.2 Results

Overview

We present the results for three different types of conditions. In condition type I, the sparse structure of the generated data matches the sparse structure of the methods. In condition type II, the data have been generated with more constraints than those set by the methods. Finally, in condition type III, we assume a mismatch between generated and estimated sparse structures (that is, analyzing data generated with sparse loadings using a method that yields sparse weights and vice versa, see Table 2.3). In Figs. 2.1, 2.2, and 2.3, we report results for the settings that include two components, a PS equal to 50% and 80% for condition types I and III, and VAF equal to 80%. Each panel contains a boxplot of a performance measure. Within each panel, a dashed line divides the boxplots for sparse loadings methods (at the left side of the dashed line) from those for sparse weights methods. For condition type-II, the settings with two components scores and VAF equal to 80% were included.[9] All analyses were performed using the actual values of the number of components and the sparsity level available in the simulation setting. Therefore, differences in performance are not the result of an improper tuning of the meta-parameters by the methods.

Table 2.3: Simulation description summary.

Condition	Sparse structure	Algorithm	Measurements		
Type I	**P**	Alg-I	SRE	MR	PEV
	W	Alg-II	SRE	MR	PEV
Type II	**P** and **W**	Alg-III	SRE	MR	PEV
	P and **W**	Alg-III	SRE	MR	PEV
Type III	**W**	Alg-II	CosSim	MR	PEV
	P	Alg-I	CosSim	MR	PEV

[9] Settings with three components and with the PS equal to 0% are available as Online Resource 1.

Condition Type I: Matching Sparsity

The first type of conditions that we discuss are those with data generated using the same model structures as the corresponding methods. Therefore, data generated by Algorithm 1 were analyzed with thresholding of rotated loadings and sPCA-rSVD, while data generated by Algorithm 2 were analyzed with SPCA, pahSPCA, and GPower. Figure 2.1 shows the results of the different performance measures for the simulation setting with two components and VAF equal to 80%. It can be observed that among the methods with sparse loadings, both thresholded Varimax and sPCA-rSVD perform reasonably well on all performance measures and in all settings. Thresholded Simplimax, on the other hand, only performs well with respect to explaining the variance. Comparing Varimax with sPCA-rSVD, we found that sPCA-rSVD has the lowest MR in all conditions and has a better recovery of the loadings and component scores in situations with many variables ($J > 10$). We found a strong effect of the level of sparsity on the MR. MR is lower when the PS is higher: This is mainly an artefact as the maximal MR is $1 - .6/.8 = 0.25$ when the sparsity is 80% and 1 when it is 50%. For Varimax and sPCA-rSVD (and in some conditions also for Simplimax), some effect of the number of variables can be observed: Better results were obtained when the number of variables increased. This is contrary to expectations, given reported issues for high-dimensional data (see Sect. 2.2.2). However, as explained previously in Sect. 3.2.2, the estimation of the loadings with the sPCA-rSVD method boils down to univariate regressions.

Among the methods imposing sparsity on the weights, GPower shows the best performance in general. For the SRE on the component weights and component scores (first and second row), it always had the lowest values when the proportion of sparsity was 80%. For different parameter settings, GPower and SPCA presented similar results. Related to the PEV and MR, GPower and SPCA showed favorable performance, although GPower obtained the best performance on the latter. Both for SPCA and GPower, it holds that their SRE performance decreased with an increasing number of variables; the estimation problem, with sparse component weights, suffers from the high dimensionality as the estimation of the weights streamlines to a high-dimensional regression problem. Finally, pathSPCA had the worst performance on every measure. For the MR, pathSPCA obtained values close to the maximum possible, and the SRE was always close to or greater than 1.

Condition Type II: Double Sparsity

In condition type II, the data were generated with the component loadings and component weights simultaneously sparse, relying on Algorithm 3. Figure 2.2 shows the results for the performance measures in the conditions with two components and VAF equal to 80%. We found that sPCA-rSVD and GPower maintained good performance and showed the best performance for sparse loadings and sparse weights methods, respectively. Both rotation techniques and sPCA-rSVD performed better in general, with a reduction of the SRE of the component loadings and scores, a

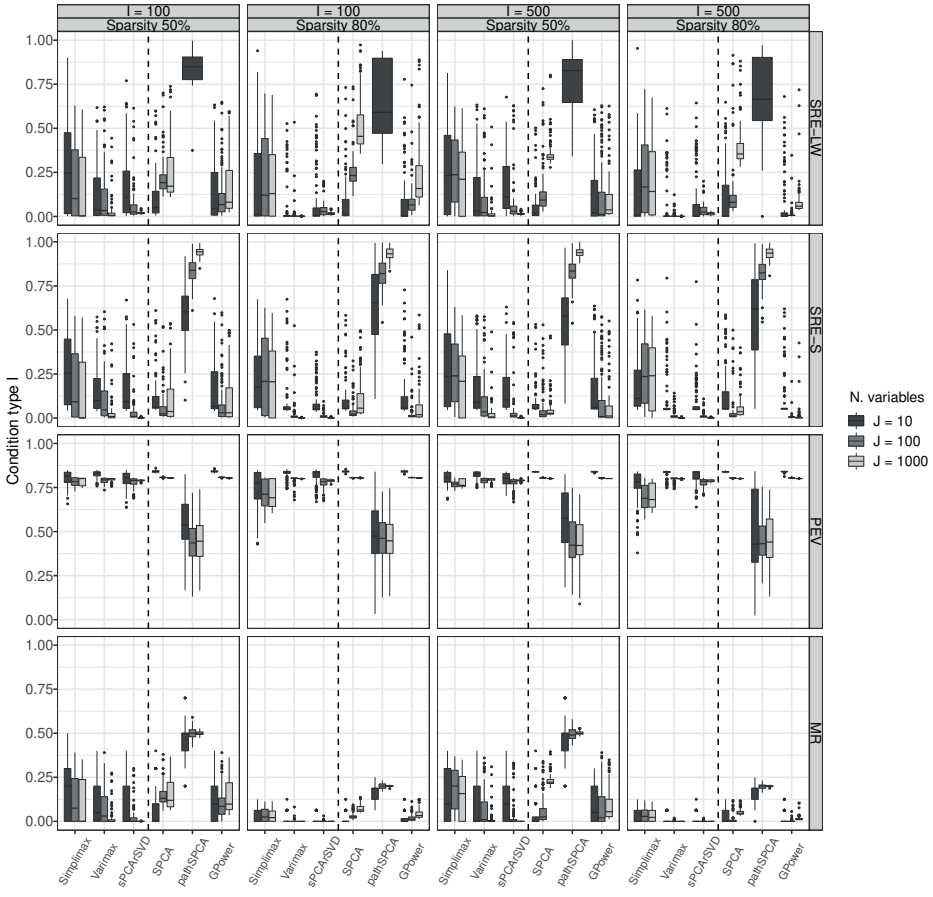


FIGURE 2.1: Matching sparsity: Boxplots of the performance measures in conditions with 80% of variance accounted by the model in the data and two components. Within each panel, a dashed line divides the boxplots for sparse loadings methods (at the left side of the dashed line) from those for sparse weights methods. The top row summarizes the squared relative error (SRE-LW) for the loadings (at the left of the dashed line) and weights (at the right of the dashed line), the second row the SRE-S for the component scores, the third row (PEV) the proportion of variance in the data explained by the estimated model, and the bottom row the misidentification rate (MR).

reduction of the MR, and a slight increment of the PEV. The performance of SPCA is much worse in the settings with 100 and 1,000 variables for all measures but the PEV, which remains around 80%. PathSPCA still performs badly, especially with respect to MR, where it almost attains the maximum possible value.

Besides comparisons within each purpose (**P** and **W**), comparisons between the two purposes can also be made (**P** vs **W**). In condition types I and II, sPCA-rSVD outperformed GPower on all measures but PEV, where they showed similar performance. This indicates that methods for sparse component loadings recover better the sparse component loading structure than those methods for sparse component weights recover the sparse component weight structure. The comparison also indicates that sparse component weights methods have higher PEV.

Condition Type III: Mismatching Sparsity

In condition type III, the sparse structures were mismatched between generated and estimated structures; that is, data generated with sparse component weights were analyzed with sparse loadings methods, while data with sparse component loadings were analyzed with methods for sparse weights. This implies that sparse loadings methods were assessed using data generated with Algorithm 2, and sparse weights methods were assessed using data generated with Algorithm 1. Additionally, the similarity measure described in Eq. (2.13) was used to assess the recovery of the component loadings/weights and scores instead of SRE.

Figure 2.3 summarizes the results for the setting with two components and VAF equal to 80%. Note that for the sparse loadings methods, the recovery of the component weights is calculated (and thus not of the component loadings), while for sparse weights methods, the recovery of the component loadings is calculated. All methods for sparse loadings—thus imposing sparse component loadings—recover the *component weights* and component scores well; Simplimax even obtains better results than Varimax in the conditions with 50% of sparsity and in some conditions also than sPCA-rSVD. Compared to condition types I and II, when 80% sparsity is imposed and $J > I$ the PEV drops. This can be understood by the fact that data were generated with sparse component weights while they were estimated with sparse component loadings, the latter having a more direct impact on the recovered data $\hat{x}_{ij}$ than the former.

Methods for sparse weights show the same pattern of results as in condition type I and notably maintain the same PEV as in condition types I and II. GPower outperformed SPCA in most of the settings and measures, although the latter still shows reasonably good results except with respect to MR in the high-dimensional settings. Compared to condition type I, GPower also outperformed SPCA on the MR in conditions with 50% of sparsity; its performance improved in this condition with mismatched sparsity. PathSPCA performed badly on every measure. Additionally, GPower outperformed sPCA-rSVD on all measures and in almost all conditions except for those with $J = 10$. Taken together, these results suggest that an underlying

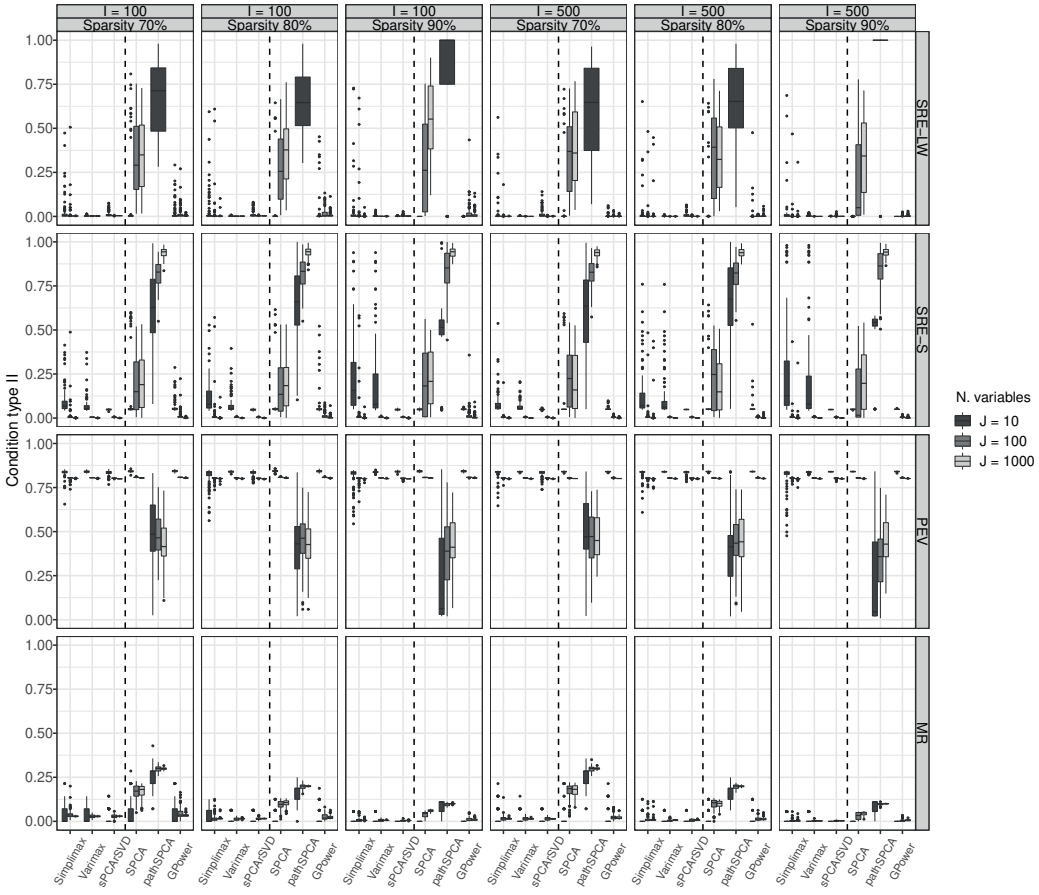


FIGURE 2.2: Double sparsity: Boxplots of the performance measures in conditions with 80% of variance accounted by the model in the data and two components. Within each panel, a dashed line divides the boxplots for sparse loadings methods (at the left side of the dashed line) from those for sparse weights methods. The top row summarizes the squared relative error (SRE-LW) for the loadings (at the left of the dashed line) and weights (at the right of the dashed line), the second row the SRE-S for the component scores, the third row (PEV) the proportion of variance in the data explained by the estimated model, and the bottom row the misidentification rate (MR).

sparse component loading structure can be recovered better by a sparse component weight method and with higher PEV than vice versa.

We used Figs. 2.4 and 2.5 to summarize the MR and PEV of the three condition types. First we discuss MR. The robustness of the methods in capturing the sparse structure under varying data generation schemes can be observed in Fig. 2.4. We can see, for example, that Simplimax showed its best MR in the conditions where sparseness is imposed on the component weights (condition types II and III). On the other hand, Varimax and sPCA-rSVD showed their best results in condition type-I. SPCA presented only good results when the $I = 10$ in the three condition types and poor performance in the other ones. GPower, although being a method that imposes sparseness on the weights, has a better recovery of the sparse structure when data are generated with sparse loadings (condition types II and III). Second, regarding the PEV (see Fig. 2.5), GPower and SPCA showed the best PEV under each condition type, and methods for sparse loadings only had a comparable PEV when data were generated with sparseness both on loadings and weights (condition type II). On both measures, MR and PEV, pathSPCA consistently showed poor performance across every condition type. Additionally, comparing the MR of GPower (sPCA-rSVD) in condition type I with sPCA-rSVD (GPower) performance in condition type III, we see that the sparse loading structure of that sPCA-rSVD does a better job in finding back the sparse component weight structure for data generated with a sparse component weight structure. GPower, however, is not better in finding the underlying sparse loading structure than sPCA-rSVD.

The different results in condition types I and II that we observe in Fig. 2.4 further support the hypothesis that sparse component loadings and sparse component weights should be treated differently. If sparse component loadings and sparse component weights were the same, we would have observed the same results in conditions type I and III, which is not the case. In condition type II, it is assumed that both component loadings and weights have the same spare structure, and methods for sparse loadings showed a better performance recovering the sparse structure in the data sets.

2.4.3 Summary

Here we focus on two essential aims of a sparse PCA analysis, namely recovering the sparseness structure (which variables are associated with the components and which ones are not) and explaining maximal variance in a parsimonious way. (This is using components that are a linear combination of a few variables only.) When recovery of the sparseness structure is the aim, a sparse loading approach (preferably sPCA-rSVD) should be used unless the data have an underlying sparse weight structure (in the latter case, the GPower approach with sparse weights should be used). When summarizing the variables with a few derived variables that explain maximal variance and are based on a linear combination of a few variables only is the goal, a sparse weight approach should be used, preferably GPower.

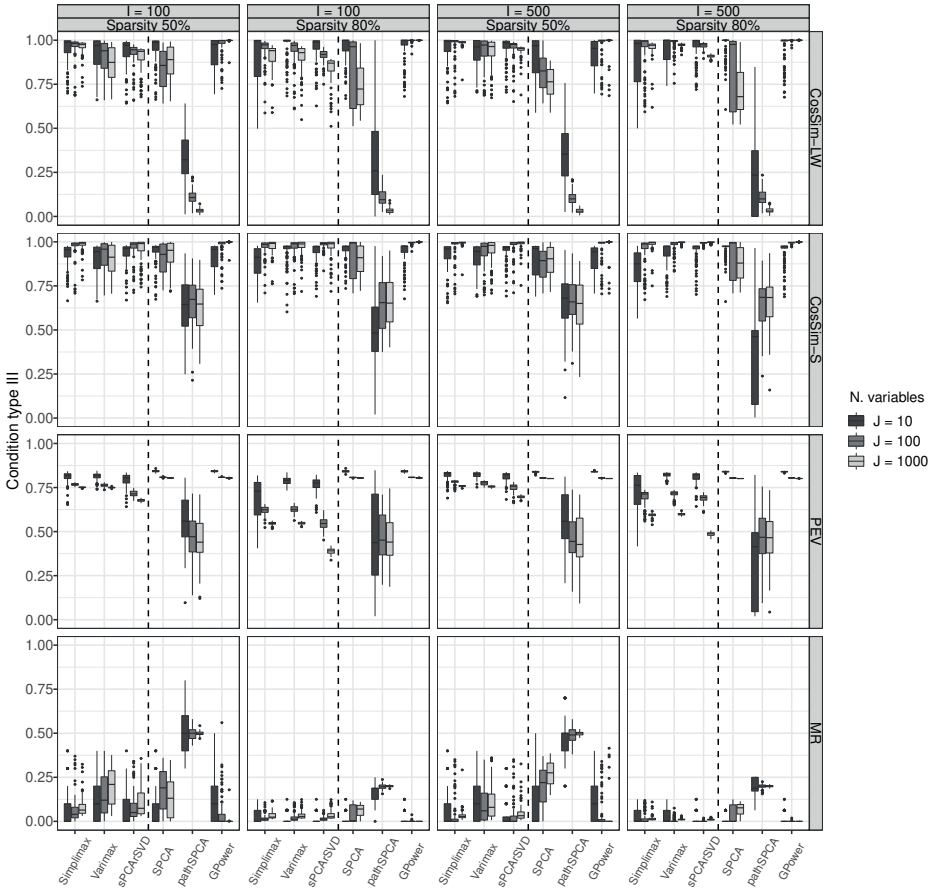


FIGURE 2.3: Mismatching sparsity: Boxplots of the performance measures in conditions with 80% of variance accounted by the model in the data and two components. Within each panel, a dashed line divides the boxplots for sparse loadings methods (at the left side of the dashed line) from those for sparse weights methods. The top row summarizes the squared relative error (SRE-LW) for the loadings (at the left of the dashed line) and weights (at the right of the dashed line), the second row the SRE-S for the component scores, the third row (PEV) the proportion of variance in the data explained by the estimated model, and the bottom row the misidentification rate (MR).

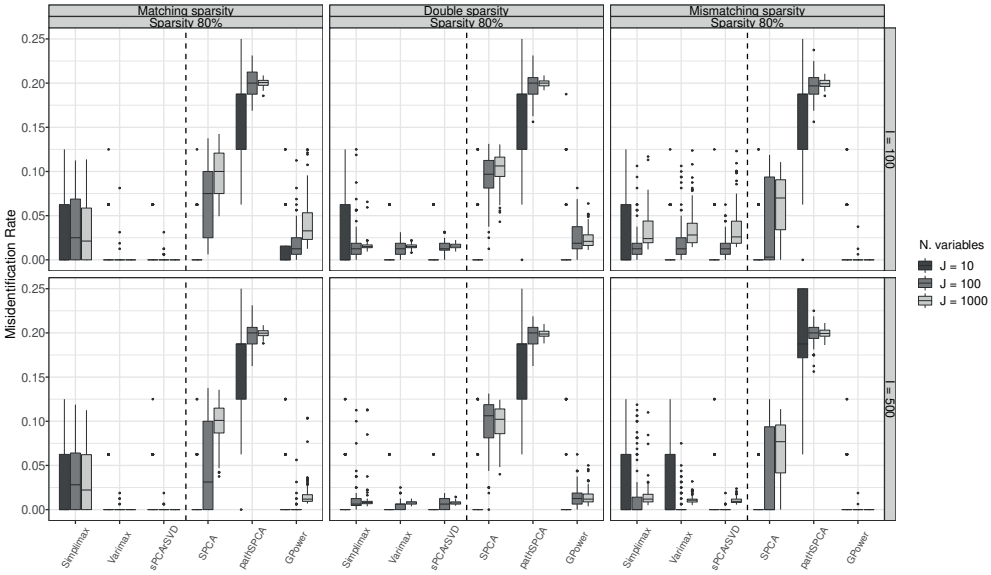


FIGURE 2.4: Misidentification rate (MR): boxplots of the MR in conditions with 80% of variance accounted by the model in the data, a proportion of sparsity of 0.8, and two components. Within each panel, a dashed line is used to divide the boxplots for sparse loadings methods (at the left side of the dashed line) from those for sparse weights methods.

Although the present results convincingly favor sPCA-rSVD and GPower, we should acknowledge that we unrealistically used knowledge about the number of components and the level of sparseness to implement the methodologies. These factors' actual values are only available in simulation studies and not when using empirical data sets. Then, parameters such as the proportion of sparsity and the number of components require additional techniques to select them. Those techniques are out of the scope of this study. The following section illustrates the implementation of sparse PCA methodologies using empirical data sets.

2.5 Empirical Applications

In this section, we use two empirical data sets to illustrate the application of sparse PCA in practice. We used a highly structured data set with variables designed to measure one of five underlying psychological constructs. Here, the aim of the sparse PCA analysis is to reveal the sparse structure that underlies the data: each variable is expected to be associated with one component only. A second data set

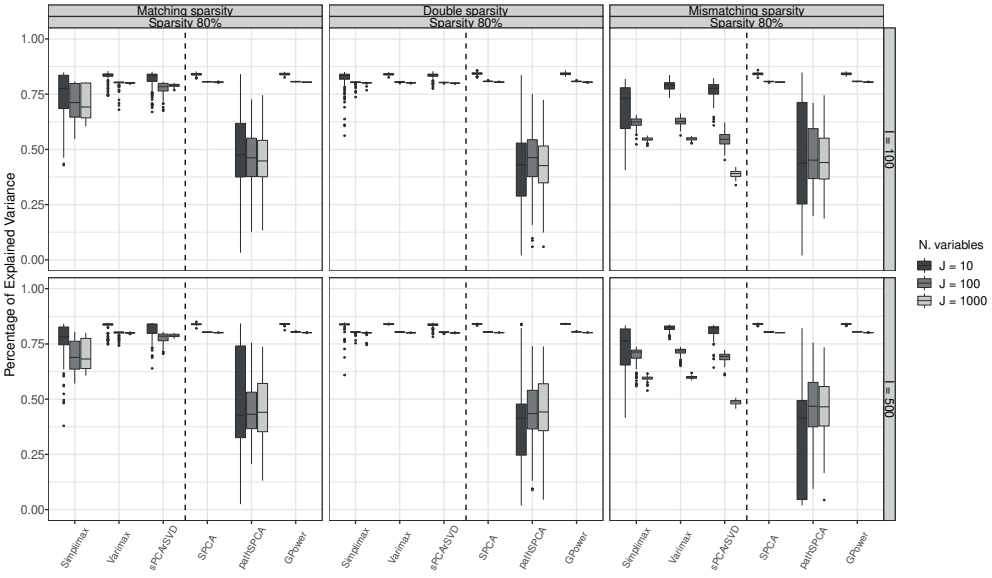


FIGURE 2.5: Percentage of explained variance (PEV): boxplots of the PEV in conditions with 80% of variance accounted by the model in the data, a proportion of sparsity of 0.8, and two components. Within each panel, a dashed line is used to divide the boxplots for sparse loadings methods (at the left side of the dashed line) from those for sparse weights methods.

was selected to show the use of sparse PCA as a summarization tool in the high-dimensional setting. For this purpose, we analyze an ultra-high dimensional genetic data set with the aim of finding a limited set of genes that allows us to classify subjects into one of three groups (two autism related groups and a control group).

An important issue that needs to be addressed for these empirical applications, and that was not addressed in the simulation study, is the choice of the number of components and the level of sparsity. For the number of components, we rely on the literature and substantive arguments made therein. For the proportion of sparsity, we rely on a data driven method, namely the *Index of Sparseness (IS)* introduced by Trendafilov (2014), that was shown to outperform other methods such as cross-validation and the BIC in estimating the true proportion of sparsity (Gu & Van Deun, 2019). The *IS* is defined as

$$IS = PEV_{sparse} \times PEV_{pca} \times PS$$

with PEV_{sparse}, PEV_{pca}, and PS denoting the PEV using a sparse method, PEV using ordinary PCA, and the proportion of sparsity (loadings or weights), respectively.

The IS value increases with the goodness-of-fit PEV_{sparse}, the higher adjusted variance PEV_{pca}, and the sparseness: the level of sparsity is determined by maximizing IS.

2.5.1 Big Five Data

We used data on the Big Five personality dimensions publicly available from the R-package *qgraph* (Epskamp, Cramer, Waldorp, Schmittmann, & Borsboom, 2012), henceforth called Big Five data. The data set contains the scores of 500 individuals on the NEO-PI-R questionnaire (McCrae & John, 1992) consisting of five sets of 48 items (i.e., 240 items in total), each set measuring one of the Big Five personality traits (Neuroticism, Extraversion, Openness to Experience, Agreeableness, and Conscientiousness) (Dolan et al., 2009). For this kind of data, interest is usually in the correlation patterns in the data (component loadings); therefore, each variable was mean-centered and scaled to unit variance. Following the design of the questionnaire, we chose $K = 5$ five components. Ordinary PCA explained 24% of the total variance; this is the maximal amount of variance that can be explained with 5 components. We will analyze these data with six sparse PCA methods. Yet, before doing so, we first need to tune the level of sparseness. As sPCA-rSVD showed the best performance in the simulation study, we use this method in combination with IS to determine the level of sparseness. Figure 2.6 shows the values for the IS and PEV as a function of the proportion of sparsity for sPCA-rSVD, calculated as the proportion of the 5×240 loadings that are zero. The maximum IS for sPCA-rSVD is attained at a sparsity proportion of 0.73 having 18% explained variance. This proportion of sparsity corresponds to a sparse model having only 64 non-zero out of 240 loadings for each component; this is reasonably close to the 48 non-zero loadings that may be expected on the basis of the design of the questionnaire.

The biplot representation of the first two components after running PCA and SPCA-rSVD are shown in Figure 2.7. Each variable is represented by an oriented vector and each subject by a dot. Figure 2.7a depicts the first two PCA components. Each item loads on both components and the solution is hard to interpret; sparseness has been introduced to improve interpretability. The biplot representation of the two first sPCA-rSVD components is shown in Figure 2.7b. Most of the items load just on one component; this makes interpretation of the components easy.

Table 2.4 presents a summary of the number of items in each set that have a non-zero loading for the five components. Using sPCA-rSVD, except for the fourth component, most non-zero loadings belong to one particular item set. For instance, from the 64 items that load on component 1, 34 belong to Neuroticism and 17 to Extraversion; on the other hand, items having a non-zero loading on component 2, mainly belong to Agreeableness (29 items), and Extraversion (19 items). Hence, the components are strongly associated with one specific trait; this is especially true for the third component (mainly Conscientiousness items) and the fifth component (mostly Openness items). On the fourth component, relatively many items from both Extraversion and Agreeableness load. The prior expectation may be that the

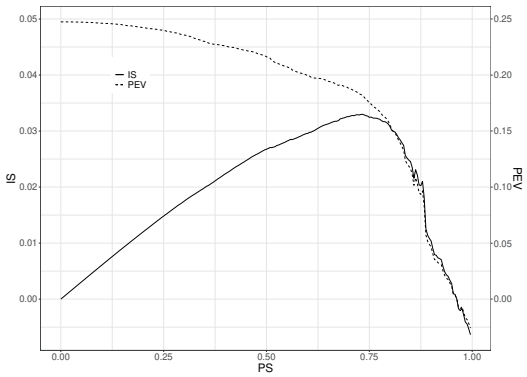


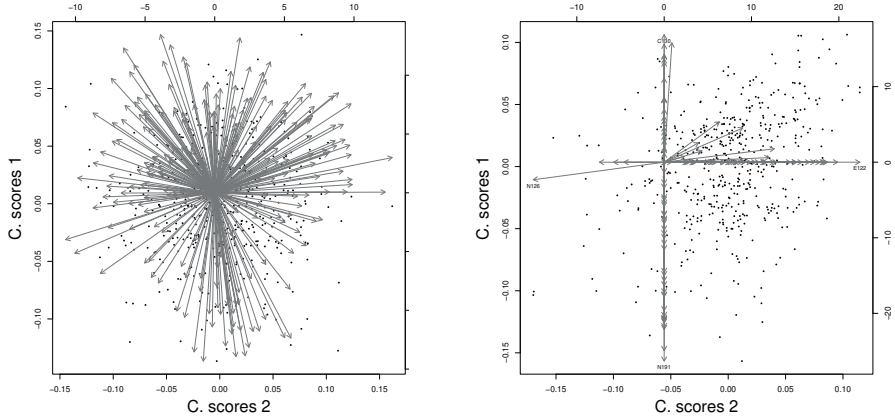
FIGURE 2.6: *Index of Sparseness* (IS) and Percentage of explained variance (PEV) against the proportion of sparsity (PS).

items of one set load only on one particular component, and thus, it invalidates the sPCA-rSVD method. Yet, many studies have shown the type of pattern found here, for example, high cross-loadings for Extraversion and Agreeableness after Procrustes rotation to the predefined structure (McCrae, Costa, & Martin, 2005).

To illustrate the comparative performance on the same empirical data, we implemented the other methods using the Big Five data set with the total number of non-zero coefficients fixed to the one found for sPCA-rSVD. As can be seen from Table 2.4, the Varimax results largely reflect the design underlying the questionnaire with items designed to measure a particular trait loading only on one particular component. Simplimax, on the other hand, does not recover the underlying structure; it has no component that is clearly dominated by the extraversion items, and the conscientiousness trait does not show up as a single component but rather as two (components 2 and 3). Using methods with sparse weights, the zero/non-zero pattern of the SPCA weights is very similar to the pattern of the Simplimax loadings. However, SPCA explains only 13% of the variance. PathSPCA showed no particular structure, each component is a weighted combination of variables related to all traits, and these components explain only 9% of the variance. Finally, by using GPower, 22% of the variance can be explained. However, the summary representations by the GPower components do not include the variables related to Neuroticism; this trait practically disappeared. Only two and one variable of the Neuroticism set of items have a non-zero weight for components 1 and 2, respectively. Additionally, items designed to measure the Openness trait underlie three of the five components (namely, components 2, 3, and 5).

Overall, the results presented in Table 2.4 highlight the importance of taking the purpose of analysis into account when choosing the sparse PCA method. We observe that methods imposing sparseness on the loadings are more suitable for the purpose of exploratory data analysis than methods imposing sparseness on the component weights. The sparsity pattern of the sPCA-rSVD and Varimax loadings reflected the

questionnaire design underlying the data best, even though the latter showed poor performance on every performance measure in the simulation study. On the other hand, GPower explained the most variance but could not recover the personality traits from the data. Finally, in line with the simulation study, pathSPCA failed to explain a reasonable amount of variance and to recover the underlying traits.



(A) Ordinary PCA (B) sPCA-rSVD

FIGURE 2.7: Biplot: the dots in each subplot represent the component scores, the arrows the component loadings.

TABLE 2.4: Sparse loading and weights composition by trait (OCEAN).

	sPCArSVD					Varimax					Simplimax				
	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
Openness	0	9	1	4	41	1	0	8	5	42	0	17	9	4	30
Concientiousness	9	3	11	43	2	7	7	3	44	4	15	0	23	31	7
Extraversion	17	19	21	6	9	16	15	30	5	7	15	10	6	7	11
Agreeableness	4	29	23	2	5	3	33	16	4	4	6	33	13	14	5
Neuroticism	34	4	8	9	7	37	9	7	6	7	28	4	13	8	11
Total non-zero	64	64	64	64	64	64	64	64	64	64	64	64	64	64	64
	SPCA					pathSPCA					Gpower				
	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$	$\mathbf{w}_5$	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$	$\mathbf{w}_5$	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$	$\mathbf{w}_5$
Openness	0	17	4	13	25	16	12	14	12	10	27	4	12	41	33
Concientiousness	15	0	26	24	8	15	15	11	10	13	11	3	42	11	15
Extraversion	15	10	15	6	16	16	10	14	14	10	3	34	5	10	12
Agreeableness	6	27	13	10	3	15	9	11	17	12	39	4	1	5	5
Neuroticism	28	10	6	11	12	17	10	12	9	16	1	2	0	0	0
Total non-zero	64	64	64	64	64	79	56	62	62	61	81	47	60	67	65

Note. Each column represent the number of items in each loading/weight that have a non-zero value in each trait. The components were ordered such that the number of non-zero loading/weights on the diagonal is maximized.

2.5.2 Gene Expression Data

To illustrate sparse PCA used as a summarization tool, we rely on publicly available gene expression data comparing 14 male control subjects to 13 male autistic subjects[10]. The autism subjects were further subdivided into two groups: a group of six with autism caused by a fragile X mutation (*FMR1-FM*) and a group of seven with autism caused by a 15q11–q13 duplication (*dup15q*). For each subject, the transcription rates of $43,893$ probes, corresponding to $18,498$ unique genes, were obtained; hence, the number of variables is much larger than the number of observations, with known numerical issues for generalized linear models (Hastie et al., 2009). Often the approach followed to account for such high-dimensionality is to first reduce the large set of variables to a few components. Because it showed the best performance in the simulation study, we will use the GPower method to select the relevant genes that summarize the component scores.

Prior to analyzing the data, we centered and scaled them to unit variance; in this way, we focused on the correlation between the expression values. Following the original publication, we select $K = 3$ three components (Nishimura et al., 2007). Figure 2.8 shows the *IS* and PEV as a function of the proportion of sparsity. The maximal PEV with three components, obtained with ordinary PCA, accounts for 32% of the total variance. The maximum value of *IS* is reached at a proportion of sparsity of 0.97 with a PEV of 31%. This corresponds to 3% or $4,323$ non-zero component weights, spread over $4,323$ different variables, each having exactly one non-zero weight. Therefore, we found an efficient reduction of the high-dimensional data to just three derived variables (the component score vectors) using approximately 10% of the original variables while losing only 1% of the variance accounted compared to when all variables are used in constructing the components via ordinary PCA.

When using the other sparse PCA methods, only sPCA-rSVD can handle the dimension of the data set computationally. However, if sPCA-rSVD had been used as a summarization tool with the same optimal proportion of sparseness found for GPower (PS = 0.97), virtually 0% of the variance would have been explained, evidencing that methods imposing sparsity in the weights are more suitable for summarization purpose.

Figure 2.9 shows the scatter plot of the three component scores. From Figure 2.9a, we observe that the first component separates the individuals with autism from the control group; this could be expected as the largest source of variation in the data is the distinction between control and autistic subjects. One may notice that Nishimura et al. (2007) constructed component scores using a subset of 293 probes with significant differences in expression between the three groups in an analysis

[10]The data can be accessed from the NCBI GEO database (Nishimura et al., 2007), using accession number GSE7329. After personally contacting the corresponding author, we were informed that the data for the individuals GSM176586 (autism with FMR1FM, AU046707), GSM176589 (autism with FMR1FM, AU046708), and GSM176615 (control, AU1165305) were not correctly stored in the database. Therefore, the data for these individuals were not used in our analyses.

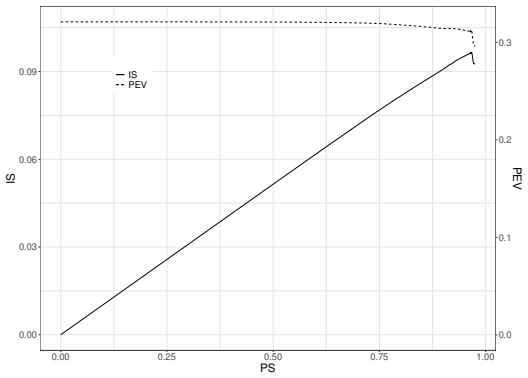
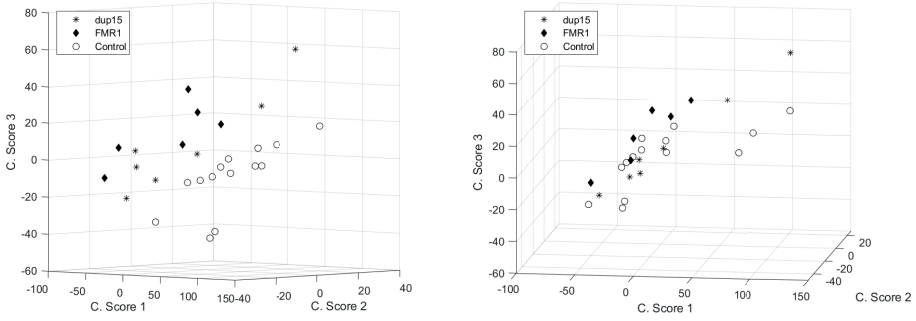


FIGURE 2.8: *Index of sparseness* and Percentage of explained variance against the proportion of sparsity when applying GPower to the gene expression data set.

of variance (ANOVA). In other words, the authors used an informed approach to select the relevant genes while sparse PCA methods (here GPower) do not rely on such external information; still, a separation between the two large groups can be observed from Figure 2.9b.



(A) GPower view 1 (B) GPower view 2

FIGURE 2.9: Scatter plot of component scores.

2.6 Concluding Remarks

As explained in this study, different PCA formulations give the same estimated scores and lead to estimates of the model coefficients that are the same or only differ up to scaling or rotation. Not surprisingly, little attention has been given to existing differences between the PCA methods, which is exemplified by the different

meanings given to the term 'loadings' in the literature. Based on these different formulations of PCA, different methods for sparse PCA have been proposed, where most of the attention has been given to the different ways of imposing sparsity and the numerical procedures used to solve the optimization problems. But, the sparse PCA methods are different on a more fundamental level, and this is seldomly discussed; the (implicitly) assumed data-generating model is often overlooked while sparsity is imposed on different model structures (either the component weights or the component loadings). Also, sparse PCA may serve different purposes in which some methods may be better than others. For instance, for exploratory data analysis, finding structure in the data and attaching meaning to the components is of primary importance. Then, a good recovery of the relevant variables and the structure therein is required. For summarization, the primary focus is to find component scores that maximally account for the variance in the data. Here, the focus is on the proportion of explained variance and, sometimes, on recovering the component scores.

To offer users of sparse PCA guidance on which method to use and under what circumstances, in a simulation study, we compared six popular methods under three data-generating schemes and four performance measures. Assuming matching sparsity (e.g., generating data with a sparse loading model and estimating them back with a method for sparse loadings), sPCA-rSVD was the preferred method based on every performance criterion for sparse loadings methods, and GPower was the best method among the sparse weights methods. In psychology, a common practice is to threshold the loadings obtained after rotation to a simple structure. In our simulation study, thresholding sometimes gave good results but sometimes also produced much worse results than the sPCA-rSVD approach. Considering that the data generating model may be unknown and that there may be a mismatch in sparsity, sPCA-rSVD is overall the best method for recovering the relevant variables, and GPower performs best in terms of explained variance.

Finally, from a practical point of view, the availability of software is of utmost importance for the use of data analysis methods. Unfortunately, sPCA-rSVD and GPower have not been (yet) implemented in major software packages such as SPSS. GPower, to our knowledge, is currently only available in Matlab. sPCA-rSVD with a cardinality constraint is available in the *ClusterSSCA* R-package (S. Yuan, De Roover, Dufner, Denissen, & Van Deun, 2021) while a penalized approach is part of the *RegularizedSCA* R-package (Gu, de Schipper, & Van Deun, 2019).

Acknowledgements

We wish to thank the referees and Associate Editor for their thoughtful work and their important recommendations that led to a substantial improvement of this study.

Chapter 3

Sparsifying the least-squares approach to PCA: comparison of lasso and cardinality constraint

Abstract

Sparse PCA methods are used to overcome the difficulty of interpreting the solution obtained from PCA. However, constraining PCA to obtain sparse solutions is an intractable problem, especially in a high-dimensional setting. Penalized methods are used to obtain sparse solutions due to their computational tractability. Nevertheless, recent developments permit efficiently obtaining good solutions of cardinality-constrained PCA problems allowing comparison between these approaches. Here, we conduct a comparison between a penalized PCA method with its cardinality-constrained counterpart for the least-squares formulation of PCA imposing sparseness on the component weights. We compare the penalized and cardinality-constrained methods through a simulation study that estimates the sparse structure's recovery, mean absolute bias, mean variance, and mean squared error. Additionally, we use a high-dimensional data set to illustrate the methods in practice. Results suggest that using cardinality-constrained methods leads to better recovery of the sparse structure.

Keywords: Cardinality constraint, Sparse PCA, Penalized linear regression

Guerra-Urzola, R., de Schipper, N.C., Tonne, A., Sijtsma, K., Vera, J.C., & Van Deun, K. Sparsifying the least-squares approach to PCA: comparison of lasso and cardinality constraint. *Advances in Data and Classification* (2022).
https://doi.org/10.1007/s11634-022-00499-2

3.1 Introduction

Principal component analysis (PCA) is a widely used analysis technique for dimension reduction and exploratory data analysis. PCA can be formulated as a variance maximization problem or a residual sum of squares minimization problem with both formulations yielding the same solution (see, e.g., Adachi & Trendafilov, 2016; I. T. Jolliffe, 1986). The component scores resulting from PCA are linear combinations of all variables, making their interpretation difficult, especially in the high-dimensional setting. Therefore, obtaining component scores that are based on a linear combination of a few variables only while still retaining most of the information in the original data is attractive. Such methods are categorized as sparse PCA.

Sparse PCA problems are usually formulated either as an extension of the PCA formulations by adding a cardinality constraint or as a convex relaxation of the constrained PCA formulation by adding penalties. Sparsity is attained either on the weights or loadigns, and unlike in PCA, their solution is not longer equivalent (see, e.g., Guerra-Urzola, Van Deun, Vera, & Sijtsma, 2021). In the context of the variance maximization PCA formulation, d'Aspremont et al. (2004), Yang, Ma, and Buja (2014), and Berk and Bertsimas (2019) added cardinality constraints on the weights, while d'Aspremont et al. (2007), and Journée et al. (2010) formulated the problem as a convex relaxation thereof adding different penalties. Additionally, Richtárik, Jahani, Ahipaşaoğlu, and Takáč (2021) presented eight different formulations based on either cardinality constraints or sparseness-inducing penalties. For the least-squares formulation of PCA, most sparse PCA methods rely on the use of different penalties (Gu & Van Deun, 2016; H. Shen & Huang, 2008; Van Deun, Smilde, van der Werf, Kiers, & Van Mechelen, 2009; Zou et al., 2006). Adachi and Trendafilov (2016) considered the sparse PCA problem in this least-squares context by imposing a cardinality constraint on the loadings. In this paper, we consider the sparse version of the least-squares formulation of PCA, where the sparsity is imposed on the weights.

Penalized methods for sparse PCA rely on alternating optimization procedures where the update of the sparse structure (weights or loadings) usually boils down to a penalized regression problem. Penalized regressions, such as LASSO (Tibshirani, 1996), have been put forward in the literature to obtain sparse solutions due to their computational tractability (Tibshirani, 2011) and their statistical nature of shrinking the nonzero coefficients. This shrinkage avoids inflation of the coefficients resulting in a better bias-variance trade-off of the estimators. However, penalized methods are heuristics that, although they provide feasible solutions, are not able to find the best subset of coefficients unless stringent conditions on the data hold (Tibshirani, 2011, p. 277). Finding the optimal subset of coefficients is an NP-hard problem (Natarajan, 1995). Nevertheless, significant progress has been recently made in solving the sparse linear regression problem via cardinality constraints for a large number of variables to optimality (Bertsimas et al., 2016; Bertsimas & Van Parys, 2020). This work is the departing point for our study. First, it shows that due to the advances in optimization, it is natural to reconsider the solvability/quality relation

between cardinality based and convex penalized relaxations of sparse formulations. Second, it opens the venue to use procedures for solving the cardinality-constrained linear regression as a subroutine to solve cardinality-constrained versions of sparse PCA.

In this paper, we compare the well-known sparse PCA method proposed by Zou et al. (2006) to its cardinality-constrained counterpart (problem (3.2)). Both methods rely on sparsifying the weights in the least-squares formulation of PCA. To our knowledge, the cardinality-constrained approach for this formulation has not yet been proposed in the literature; Therefore, we introduce it in Section 3.2.2. Both methods use an alternating scheme where sparsity is achieved via a penalized or cardinality-constrained linear regression step. We compare the performance of the methods in a simulation study using different measures such as the recovery rate of the sparse structure, mean absolute bias, mean variance, and mean squared error. Additionally, we illustrate the use of the methods in practice with an empirical data set containing gene expression profiles of lymphoblastoid cells used to distinguish different forms of autism (Nishimura et al., 2007). The results from the simulation study suggest that cardinality-constrained PCA has a better recovery of the sparse structure yet a similar bias-variance trade-off as the penalized counterpart.

The remainder of this paper is structured as follows: First, Sect. 3.2 introduces PCA, sparse PCA, and penalized PCA. In Sect. 3.3, we present the simulation study comparing the performance of the methods on different measures. Sect. 3.4 presents an example using a real high-dimensional data set. Finally, in Sect. 3.5, conclusions are presented.

3.2 Methods

We first present the notation used in the remainder of the paper. 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 l_1-norm is defined by $\|\mathbf{x}\|_1 = \sum_{j=1}^J |x_j|$, and the Euclidean distance by $\|\mathbf{x}\|_2 = (\sum_{j=1}^J x_j^2)^{1/2}$. Given a matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$, its i-th row and j-th column entry is denoted 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.

In this section, we introduce the PCA formulation on which the paper focuses. Sparse PCA variants of the formulation are then obtained by either adding a cardinality constraint or a convex penalty to the PCA objective.

3.2.1 PCA

Given a data matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ that contains I observations on J variables, in PCA, it is assumed that the data can be decomposed as,

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

where $\mathbf{W} \in \mathbb{R}^{J \times K}$ is the weights matrix, $\mathbf{P} \in \mathbb{R}^{J \times K}$ is the loadings matrix, $\mathbf{E} \in \mathbb{R}^{I \times J}$ is the residual matrix, and $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$. Ordinary PCA can be formulated as the following least squares optimization problem:

$$\begin{aligned} \widehat{\mathbf{W}}, \widehat{\mathbf{P}} = \underset{\mathbf{W},\mathbf{P}}{\operatorname{argmin}} \quad & \left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{P}^\top\mathbf{P} = \mathbf{I}. \end{aligned} \tag{3.1}$$

The solution to the PCA formulation in (3.1) can be obtained using the truncated Singular Value Decomposition (SVD) of the data matrix $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ (I. Jolliffe, 2002), with $\mathbf{U} \in \mathbb{R}^{I \times K}$ and $\mathbf{V} \in \mathbb{R}^{J \times K}$ semi-orthogonal matrices, and $\widehat{\mathbf{W}} = \widehat{\mathbf{P}} = \mathbf{V}$. The linear combinations $\mathbf{T} = \mathbf{X}\mathbf{W}$ represent the component scores. In general, the estimated weights matrix resulting from the truncated SVD contains all nonzero elements making the interpretation of the component scores difficult when J is large.

3.2.2 Cardinality-Constrained PCA

Starting from PCA formulation (3.1), the sparse PCA problem can be formulated as a best subset selection problem for a subset of size ρ (where ρ between 0 and $J \cdot K$ is given) as follows,

$$\begin{aligned} \widehat{\mathbf{W}}, \widehat{\mathbf{P}} = \underset{\mathbf{W},\mathbf{P}}{\operatorname{argmin}} \quad & \left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{P}^\top\mathbf{P} = \mathbf{I}, \\ & \|\mathbf{W}\|_0 \leq \rho, \end{aligned} \tag{3.2}$$

with $\|\mathbf{W}\|_0$ denoting the number of nonzero coefficients in $\mathbf{W}$. Sparse PCA methods based on the least squares criterion in (3.1) have been only considered by adding penalties (see Sect. 3.2.3 for details). A solution to the cardinality-constrained problem (3.2) has not been proposed yet. Here, we propose an alternating optimization procedure to obtain feasible solutions of good quality. That is, fix $\mathbf{W}$ and obtain $\widehat{\mathbf{P}}$ by the well-known reduced rank Procrustes rotation (ten Berge, 2005; Zou et al., 2006),

$$\begin{aligned} \widehat{\mathbf{P}} = \underset{\mathbf{P}}{\operatorname{argmin}} \quad & \left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{P}^\top\mathbf{P} = \mathbf{I}. \end{aligned} \tag{3.3}$$

Also, fix $\mathbf{P}$ and obtain $\widehat{\mathbf{W}}$ via the cardinality-constrained linear regression problem,

$$\begin{aligned} \widehat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmin}} \quad & \|\operatorname{vec}(\mathbf{X}) - (\mathbf{P} \otimes \mathbf{X})\operatorname{vec}(\mathbf{W})\|_2^2 \\ \text{s.t.} \quad & \|\operatorname{vec}(\mathbf{W})\|_0 = \rho, \end{aligned} \tag{3.4}$$

where $\otimes$ denotes the Kronecker product, and $\operatorname{vec}(\cdot)$ the vectorization of a matrix which converts the matrix into a column vector. A numerical procedure that solves problem (3.4) was proposed by Adachi and Kiers (2017) as a special case of a majorize-minimize (Hunter & Lange, 2004) or iterative majorization (Kiers, 2002) procedure. The update for $\widehat{\mathbf{W}}$ in each iteration is given by:

$$\operatorname{vec}(\mathbf{W}_{new}) = T_\rho\left(\operatorname{vec}(\mathbf{W}_{old} - \alpha^{-1}\mathbf{X}^\top\mathbf{X}[\mathbf{W}_{old} - \mathbf{P}])\right), \tag{3.5}$$

where α is the maximum eigenvalue of $\mathbf{X}^\top\mathbf{X}$ and, for a vector $\mathbf{x} \in \mathbb{R}^n$, the thresholding operator $T_\rho(\mathbf{x}) \in \mathbb{R}^n$ denotes the vector obtained from $\mathbf{x}$ keeping the values of the ρ elements of $\mathbf{x}$ having the largest absolute value, and setting the remaining ones equal to zero. Notice that the updating step for $\mathbf{W}$ in Eq. (3.5) is equal to the update of a projected gradient scheme with fixed step size α^{-1}. The use of majorization ensures that the resulting sequence of loss values is non-increasing. At each iteration, to obtain an approximate solution to (3.4), our algorithm relies on a procedure where the main complexity is to sort a matrix of dimension $J \times K$, and thus, this procedure can be applied even for large values of J. We call the full alternating procedure cardinality-constrained PCA (CCPCA). In Appendix A.1, the CCPCA algorithm is presented in detail.

It is important to mention that the proposed algorithm does not guarantee finding a global optimum of problem (3.4). Instead, with each conditional update of either the component weights or loadings, the loss function is monotonically decreasing. For alternating algorithms of the type considered here, obtaining a stationary point is guaranteed under some compactness assumptions on the feasible set of the subproblems (Huang, Sidiropoulos, & Liavas, 2016; Tseng, 2001). Such compact structure can be obtained by adding the constraint $\|\mathbf{w}_k\|_2 \leq 1$ for $k = 1, \ldots, K$. However, this type of regularization constraint does not appear in the least square error formulation of PCA (see problem (3.1)) and therefore has not been added to the cardinality-constrained version of the sparse formulation either.

Defining sparse PCA as a best subset problem has not been the method of choice in the statistical literature, given that it belongs to the class of NP-hard problems. Another reason to find sparse solutions by adding convex penalties, such as the LASSO, is the belief that these have a better bias-variance tradeoff resulting in better predictive accuracy in the context of regression. Recently, given the algorithmic and computational-power progress in the last few decades, it has been shown that the cardinality-constrained regression problem can be solved for a large number of variables (in the 100,000s) (Bertsimas & Van Parys, 2020). For instance, Bertsimas et al. (2016) found the cardinality-constrained regression approach to be superior to

LASSO regression not only in terms of recovering the correct subset of variables but also in terms of predictive performance, which is contrary to expectations based on the bias-variance trade-off. However, Hastie, Tibshirani, and Tibshirani (2017) have extended the simulations of (Bertsimas et al., 2016) focusing on prediction accuracy and found that the cardinality-constrained regression approach outperformed the LASSO regression only when there was a high signal-to-noise ratio.

3.2.3 Penalized PCA

A well-known sparse PCA method based on penalizing (3.1) was proposed by Zou et al. (2006). The method, named SPCA, is based on the following formulation,

$$\begin{aligned} \widehat{\mathbf{W}}, \widehat{\mathbf{P}} = \underset{\mathbf{W},\mathbf{P}}{\operatorname{argmin}} \quad & \left\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\right\|_F^2 + \lambda \sum_{k=1}^{K} \|\mathbf{w}_k\|_2^2 + \sum_{k=1}^{K} \lambda_k^l \|\mathbf{w}_k\|_1 \\ \text{s.t.} \quad & \mathbf{P}^\top\mathbf{P} = \mathbf{I}, \end{aligned}$$

with $\sum_{k=1}^{K} \|\mathbf{w}_k\|_1$ the LASSO penalty (tuned using $\lambda_k^l \geq 0$) and $\sum_{k=1}^{K} \|\mathbf{w}_k\|_2^2$ the ridge penalty (tuned using $\lambda \geq 0$). For fixed values of λ_k^l and λ, SPCA is an alternating minimization algorithm that updates $\mathbf{W}$ given $\mathbf{P}$ and vice-versa. Obtaining $\widehat{\mathbf{P}}$ given a fixed value for $\mathbf{W}$ is also done via the reduced rank Procrustes Rotation problem (ten Berge, 2005). And obtaining $\widehat{\mathbf{W}}$ given a fixed value for $\mathbf{P}$ is achieved using the elastic net penalized regression problem(Zou & Hastie, 2005) that is defined by adding the LASSO and ridge penalties to the ordinary regression problem. The LASSO penalty sets some of the coefficients to exactly zero, while the ridge penalty shrinks the coefficients and regularizes the problem in the high-dimensional setting ($J > I$); i.e., it allows for more nonzero coefficients than the number of observations, see also Zou et al. (2006).

Using a penalized regression as one of the alternating steps presents some advantages and disadvantages. On the one hand, shrinkage of all coefficients reduces the variance of the estimated coefficients; hence the coefficients estimated under a penalized regime may be more accurate than those obtained via cardinality constraints (Hastie et al., 2017). On the other hand, although penalized regressions find sparse feasible solutions, the correct subset of nonzero variables is only recovered under stringent conditions (see Bertsimas et al. (2016) and references therein). In the next section, we assess and compare the performance of CCPCA and SPCA in a simulation study. We focus on sparse structure recovery (zero and nonzero weights) and the accuracy of the estimated weights.

3.3 Simulation Study

To compare the statistical properties of the penalized and the cardinality-constrained sparse PCA methods described in Section 3.2 above, we conducted a simulation

study where two types of measures are of interest: the recovery of the weights support matrix (correctly identifying the set of non-zero weights) and the accuracy of the estimates in terms of bias and variance. To measure the former, we use the total sparse structure recovery rate (TSS%), and for the latter, we use the mean absolute bias (MAB), mean variance (MVAR), and mean squared error (MSE)[1].

3.3.1 Design

We set the number of observations to $I = 100$, the number of variables to $J = 50, 100, 500$, and the number of components to $K = 3$. We have also set the level of sparsity to 20% and 80% (i.e., when $J \cdot K = 300$, we have 60 and 240 weights that are equal to zero, respectively), and the noise level to 5%, 20%, and 80%. The design results in $3 \cdot 2 \cdot 3 = 18$ different design conditions. For each condition, $R = 100$ data sets were generated. The data generation procedure is detailed in Appendix A.2. The resulting data sets were analyzed using the CCPCA algorithm programmed in the `R` software for statistical computing (*R: The R Project for Statistical Computing*, n.d.), and SPCA with LARS using the elastic net `R`-package (*R: Elastic-Net for Sparse Estimation and Sparse PCA*, n.d.). Both algorithms were run with one initial value based on the SVD decomposition of the data. We supplied the analysis with the actual number of components. The tuning parameter of the ridge penalty for SPCA was left at the default value of 10^{-6}; this is a small value such that the focus remains on comparing the cardinality constraint to the LASSO penalty as a means to sparsify the PCA problem in (3.1).

The analysis is divided into two cases depending on whether the cardinality of $\mathbf{W}$ is known or not. When the cardinality is known, we supply the analysis with the true cardinality. When the cardinality is unknown, we rely on a data-driven method, namely the Index-of-sparseness (IS) introduced by Trendafilov (2014). The IS has been shown to outperform other methods, such as cross-validation and the BIC, in estimating the actual proportion of sparsity (Gu et al., 2019). The IS is defined as

$$IS = \text{PEV}_{sparse} \cdot \text{PEV}_{pca} \cdot (1 - \tfrac{\rho}{J \cdot K})$$

with PEV_{sparse} and PEV_{pca} denoting the proportion of explained variance using a sparse method and ordinary PCA, respectively. The IS value increases with the goodness-of-fit PEV_{sparse}, the higher adjusted variance PEV_{pca}, and the sparseness. The cardinality of the weights is determined by maximizing the IS.

To assess the recovery of the weights matrix, we calculate the total sparse structure recovery rate, defined as:

$$\text{TSS\%} = \frac{\sum_{j,k} \text{supp}(\mathbf{W}, \widehat{\mathbf{W}})_{j,k}}{J \cdot K} \tag{3.6}$$

[1]Computational diagnostic are outside the scope of this study. These diagnostics mainly depend on the selected method for estimating, and more computationally efficient methods have been proposed for SPCA (Erichson et al., 2020)

where

$$\text{supp}(\mathbf{W}, \widehat{\mathbf{W}})_{j,k} = \begin{cases} 1, & w_{j,k} = 0 \quad \text{and} \quad \widehat{w}_{j,k} = 0 \\ 1, & w_{j,k} \neq 0 \quad \text{and} \quad \widehat{w}_{j,k} \neq 0 \\ 0, & \text{Otherwise.} \end{cases}$$

Therefore, TSS% takes into account both correct identification of the zero and nonzero values. To assess the accuracy of the actual value of the estimates, we calculate the MAB, MVAR, and MSE. These measures are defined as,

$$\text{MAB} = \frac{1}{J \cdot K \cdot R} \sum_j \sum_k \sum_r |\overline{\widehat{w}}_{j,k} - w_{j,k}^{(r)}|,$$

$$\text{MVAR} = \frac{1}{J \cdot K \cdot R} \sum_j \sum_k \sum_r (\overline{\widehat{w}}_{j,k} - \widehat{w}_{j,k}^{(r)})^2,$$

$$\text{MSE} = \frac{1}{J \cdot K \cdot R} \sum_j \sum_k \sum_r (w_{j,k} - \widehat{w}_{j,k}^{(r)})^2,$$

where $\overline{\widehat{w}}_{j,k} = \frac{1}{R} \sum_r \widehat{w}_{j,k}^{(r)}$ and $r = 1 \ldots R$ a running index for the generated data sets. As CCPCA and SPCA solutions are indeterminate with respect to the sign and order of the component weight vectors $\mathbf{w}_k$, we matched $\widehat{\mathbf{w}}_k$ to the true $\mathbf{w}_k$ based on the highest proportion of total recovery in Eq. (3.6).

3.3.2 Results

Figures 3.1 and 3.2 show the total recovery rate with cardinality either set to the cardinality used to generate the sparse weights or to the value that maximizes the IS, respectively. From Figure 3.1, it can be observed that in almost all conditions, CCPCA has a higher proportion of correctly identified weights than SPCA. Only when the noise level is 80% and the proportion of sparsity 20%, both methods present similar results on average. When the cardinality of the component weights is treated as unknown and tuned using the IS, we observe in Figure 3.2 that the recovery rate mainly depends on the proportion of sparsity. When $PS = 20\%$, SPCA has a higher recovery rate, and when $PS = 80\%$, CCPCA has a higher recovery rate. This result may be explained by the fact that CCPCA can achieve more variance with less variables than SPCA (see Figure A.1). Therefore, the cardinality of CCPCA is always lower than SPCA's cardinality and the cardinality used to generate the data sets (see Figure A.2) The MAB, MVAR, and MSE of the estimators from CCPCA and SPCA are reported in Tables 3.1 and 3.2 when the cardinality is set equal to the cardinality used to generate the weights and as tuned with the IS, respectively. It can be observed in Table 3.1 that the MAB of CCPCA is higher than the SPCA's MAB, although by a small margin only. The MVAR of the CCPCA weights is approximately equal to the MVAR of SPCA weights when there is little noise in the data (5%). In the case of a higher noise level (20% and 80%), the MVAR of SPCA is lower than that of CCPCA; this can be attributed to the shrinkage effect of the

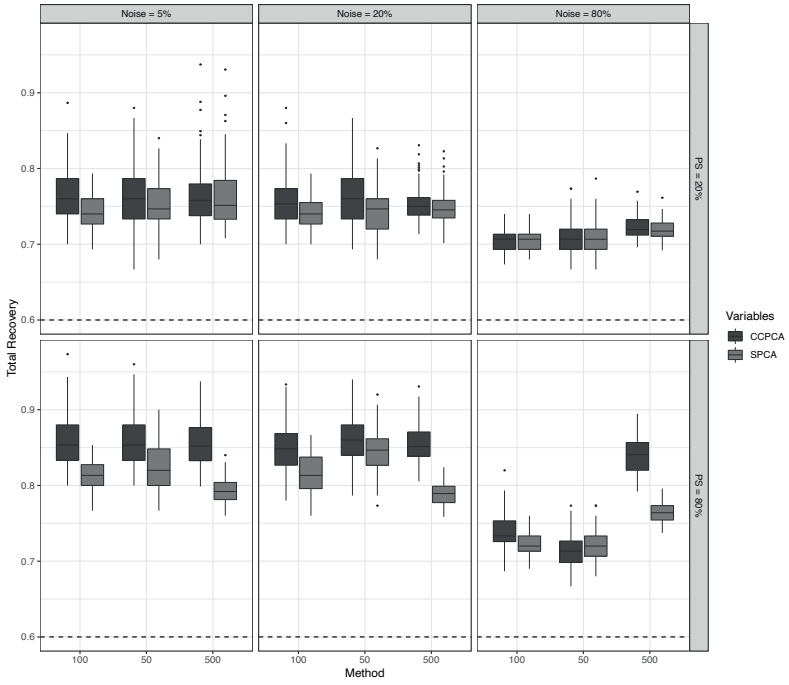


FIGURE 3.1: Proportion of correctly identified weights with the same cardinality as used to generate the data. The dashed line at 0.6 indicates the minimum recovery rate that can be obtained given 20% or 80% of sparsity.

penalties in SPCA. The MSE in case of little noise in the data and 20% of sparsity is slightly lower for CCPCA than SPCA, while the MSE is lower for SPCA in case of 20% noise and 80% sparsity. If we turn to the case where the cardinality was tuned using the IS (Table 3.2), in all conditions, the MAB is smaller for CCPCA than for SPCA while the MVAR and MSE are higher for CCPCA than for SPCA: here, we clearly see the beneficial effect of the shrinkage penalties that introduce a higher bias though resulting in a much lower variance.

Overall, these results suggest that CCPCA recovers better the sparse structure, especially under high levels of sparsity and noise. When the cardinality of the component weights is known, the recovery rates are, in general, satisfactory to good. When the cardinality is not known, and the IS is used to tune the cardinality, SPCA performs reasonably well on data with low levels of sparsity while CCPCA performs reasonably well on data with high levels of sparsity. Additionally, as mentioned in the statistical literature, the penalized method (SPCA) has higher bias though lower variance, while the cardinality-constrained method (CCPA) has less bias but higher variance.

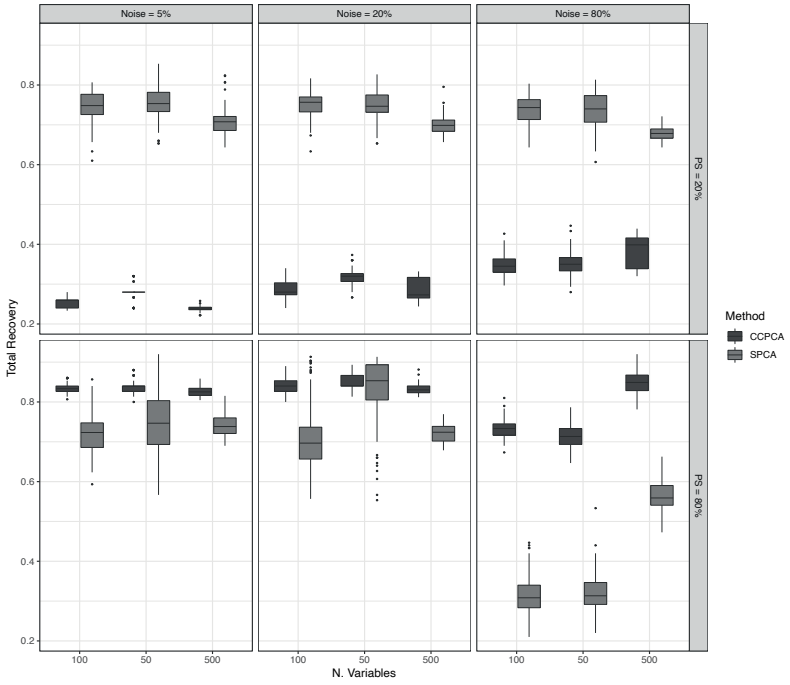


FIGURE 3.2: Proportion of correctly identified weights with cardinality tuned using the index of sparseness.

3.4 Empirical Application

In this section, we use an empirical data set to illustrate how the methods described in this study can be used in practice as a pre-processing step to reduce a large set of variables in the context of classification. We use a publicly available gene expression data set comparing 14 male control subjects to 13 male autistic subjects.[2] The autism subjects were further subdivided into two groups: a group of six with autism caused by a fragile X mutation (FMR1-FM) and seven with autism caused by a 15q11–q13 duplication (dup15q). The transcription rates of 43,893 probes, corresponding to 18,498 unique genes, were obtained for each subject.

[2]The data set can be accessed from the NCBI GEO database (Nishimura et al., 2007), using accession number GSE7329. After personally contacting the corresponding author, we were informed that the data for the individuals GSM176586 (autism with FMR1FM, AU046707), GSM176589 (autism with FMR1FM, AU046708), and GSM176615 (control, AU1165305) were not correctly stored in the database. Therefore, those observations were not included in our analyses. In the supporting code, code is included that allows to download and pre-processing of the data set automatically.

TABLE 3.1: MAB, MVAR, and MSE of the estimators from CCPCA and SPCA

		Noise Level = 5%					
		Sparsity = 20%			Sparsity = 80%		
		J=50	J=100	J=500	J=50	J=100	J=500
MAB	CCPCA	0.1082	0.0760	0.0339	0.0669	0.0474	0.0211
	SPCA	0.0999	0.0701	0.0314	0.0571	0.0401	0.0171
MVAR	CCPCA	0.0198	0.0099	0.0020	0.0214	0.0105	0.0021
	SPCA	0.0198	0.0099	0.0020	0.0198	0.0099	0.0020
MSE	CCPCA	0.0410	0.0200	0.0037	0.0408	0.0212	0.0041
	SPCA	0.0427	0.0196	0.0040	0.0386	0.0205	0.0041
		Noise Level = 20%					
MAB	CCPCA	0.1084	0.0762	0.0341	0.0663	0.0480	0.0212
	SPCA	0.1005	0.0702	0.0315	0.0539	0.0399	0.0171
MVAR	CCPCA	0.0198	0.0099	0.0020	0.0211	0.0106	0.0021
	SPCA	0.0198	0.0099	0.0020	0.0198	0.0099	0.0020
MSE	CCPCA	0.0422	0.0206	0.0041	0.0417	0.0214	0.0043
	SPCA	0.0411	0.0206	0.0042	0.0392	0.0208	0.0041
		Noise Level = 80%					
MAB	CCPCA	0.1108	0.0783	0.0345	0.0828	0.0573	0.0220
	SPCA	0.1018	0.0717	0.0315	0.0569	0.0396	0.0172
MVAR	CCPCA	0.0198	0.0099	0.0020	0.0279	0.0134	0.0022
	SPCA	0.0198	0.0099	0.0020	0.0198	0.0099	0.0020
MSE	CCPCA	0.0403	0.0204	0.0038	0.0489	0.0236	0.0041
	SPCA	0.0403	0.0202	0.0039	0.0400	0.0200	0.0039

Note. The estimates are based on 100 replicated data sets.

Prior to analyzing the data, we centered and scaled each column to unit variance and followed Nishimura et al. (2007) to choose the number of components $K = 3$. Therefore, the total cardinality of the component weights is $131,679$. To select the cardinality, we rely on the IS. Figure 3.3 shows the IS and PEV as a function of the cardinality of the weights using CCPCA and SPCA.[3] The maximal PEV with three components, obtained with ordinary PCA, accounts for 32% of the variance. The maximum value of IS for CCPCA is reached at a cardinality of $23,499$ with a PEV of 30% while the maximal IS for SPCA is reached at a cardinality of $42,283$ with a PEV of 22%. This is also in accordance with our earlier observation in the

[3]To handle these ultra-high dimensional data, we use the SPCA model with $\lambda = \infty$, see Zou et al. (2006) for further details.

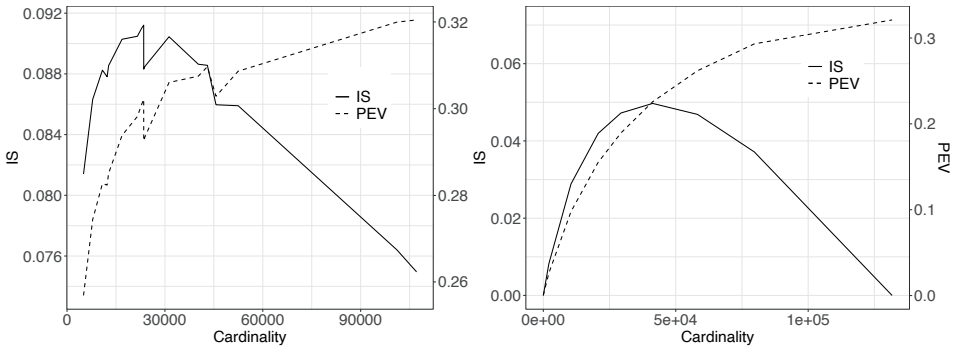
TABLE 3.2: MAB, MVAR, and MSE of the estimators from CCPCA and SPCA with cardinality tuned using the index-of-sparseness.

		Noise Level = 5%					
		Sparsity = 20%			Sparsity = 80%		
		J=50	J=100	J=500	J=50	J=100	J=500
MAB	CCPCA	0.0764	0.0533	0.0234	0.0526	0.0365	0.0160
	SPCA	0.1016	0.0710	0.0295	0.0681	0.0495	0.0221
MVAR	CCPCA	0.0536	0.0321	0.0077	0.0345	0.0176	0.0047
	SPCA	0.0198	0.0099	0.0020	0.0198	0.0099	0.0020
MSE	CCPCA	0.0468	0.0263	0.0064	0.0173	0.0090	0.0030
	SPCA	0.0104	0.0050	0.0008	0.0091	0.0045	0.0008
		Noise level = 20%					
MAB	CCPCA	0.0769	0.0532	0.0242	0.0529	0.0375	0.0162
	SPCA	0.1019	0.0715	0.0297	0.0560	0.0483	0.0218
MVAR	CCPCA	0.0377	0.0220	0.0045	0.0275	0.0151	0.0035
	SPCA	0.0198	0.0099	0.0020	0.0198	0.0099	0.0020
MSE	CCPCA	0.0267	0.0167	0.0033	0.0107	0.0070	0.0018
	SPCA	0.0118	0.0053	0.0009	0.0050	0.0048	0.0008
		Noise level = 80%					
MAB	CCPCA	0.0820	0.0577	0.0260	0.0824	0.0574	0.0210
	SPCA	0.1053	0.0747	0.0300	0.1053	0.0729	0.0242
MVAR	CCPCA	0.0279	0.0141	0.0028	0.0282	0.0135	0.0023
	SPCA	0.0198	0.0099	0.0020	0.0198	0.0099	0.0020
MSE	CCPCA	0.0439	0.0189	0.0018	0.0435	0.0167	0.0009
	SPCA	0.0369	0.0152	0.0011	0.0368	0.0152	0.0011

Note. The estimates are based on 100 replicated data sets.

simulation study, which showed that CCPCA can explain more variance with less variables than SPCA.

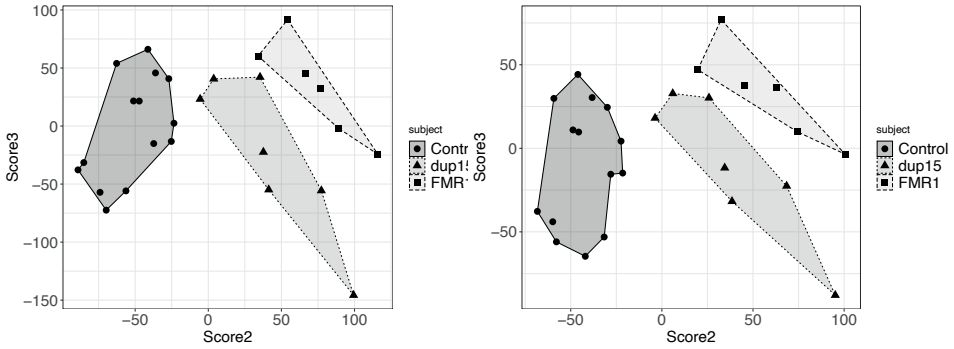
When plotting the second component score against the third one (Figures 3.4a and 3.4b), we observe a separation of the individuals with autism from the control group and between the individuals with autism caused by the fragile X mutation and by the 15q11-q13 duplication. The former could be expected as the largest source of variation in the data is the distinction between control and autistic subjects. One may notice that in Nishimura et al. (2007), this classification of the three groups is observed as well. However, Nishimura et al. (2007) constructed component scores using a subset of 293 probes with a significant difference in expression between the three groups in an analysis of variance (ANOVA). This means that an informed approach was used to select the relevant genes while CCPCA and SPCA



(A) CCPCA (B) SPCA

FIGURE 3.3: Index of sparseness (IS) and proportion of explained variance (PEV) against Cardinality

do not construct component scores with the aim of discrimination; still, a separation between the three groups can be observed from Figures 3.4a and 3.4b.



(A) CCPCA (B) SPCA

FIGURE 3.4: Scatter plot of the component scores of component 2 against component 3.

3.5 Conclusion

We introduce a cardinality-constraint based method (CCPCA) and compare its performance with the performance of a penalty based method (SPCA). Both methods are designed to attain sparse weights in PCA. Both follow an alternating optimization procedure where sparsity is achieved via either a penalized or a cardinality-constrained linear regression problem. Penalized regressions have been propounded

in the statistical literature for reasons of computational and statistical efficiency. Recently, significant progress has been made in solving cardinality-constrained regression problems finding feasible solutions in the case of many variables.

We compared the CCPCA and SPCA methods through a simulation study assessing the recovery of the sparse structure (zero and nonzero) and the accuracy of the estimates. Regarding the recovery rate, CCPCA showed better results than SPCA in almost all conditions when both methods were supplied with actual cardinality. When the cardinality needed to be estimated, CCPCA presented a better solution when the cardinality was set to a small number of variables. For the accuracy of the estimates, both presented similar performance with known cardinality, while SPCA shows more bias and less variance with unknown cardinality. Additionally, we used real, high-dimensional data to evaluate these methods in practice. CCPCA and SPCA efficiently reduced the dimension without losing much of the explained variance using only a fraction of the original variables in the data. From the simulation and the real example, CCPCA can explain more variance with fewer variables than SPCA.

CCPCA and SPCA are freely available to be used in `R`-software. When using them, it is essential to consider that both methods are subject to local minima. It is a common practice to implement a multi-start procedure and select the solution with the smallest objective function, but the obtained solutions will still be subject to local optima. For future work, it would be interesting to analyse the conditions for optimality for sparse PCA methods.

Acknowledgments. We wish to thank the referees and Associate Editor for their thoughtful work and their recommendations. In our opinion, these led to a substantial improvement of the paper.

Authors contribution statement. All authors contributed to the study's conception and design. All authors read and approved the final manuscript.

Chapter 4

Penalized PCA framework: thresholding operators and optimality conditions

Abstract

Penalized PCA methods are widely used to find sparse solutions to the PCA problem due to their computational tractability and scalability. These benefits have been assessed via numerical experiments without theoretical justification. This paper presents a theoretical analysis of a penalized PCA method used to find sparse solutions to the PCA problem. The paper derives a necessary optimality condition for penalized PCA problems and characterizes penalties that lead to a thresholding operator as the solutions to the optimization problem. An alternating thresholding method is proposed to solve the penalized PCA problem, and it is shown that the method converges to a solution that satisfies the necessary optimality condition when the minimum eigenvalue of the covariance matrix is greater than one. Additionally, the paper applies the framework to a family of l_1-norm penalties and proposes two new penalized PCA formulations.

Keywords: Sparse PCA, penalties, optimality conditions, thresholding operators

4.1 Introduction

Principal Component Analysis (PCA) is one of the oldest and most used data analysis tools to summarize a data set with a few new variables or scores. PCA has applications in many fields, such as medicine, biology, Artificial Intelligence (AI), and finance (Mohammed et al., 2016; Pasini, 2017). Given that the new scores are a weighted combination of all variables in the data, scores lack interpretability most of the time. To improve interpretability, having scores formed by only a few of the most representative variables is a desired property when applying PCA to gain interpretation. For example, a critical challenge in cancer research is to reduce dimension and extract the relevant features when analyzing high-dimensional data sets (Hsu et al., 2014). Besides interpretation, analyzing fewer variables helps to reduce transactional and operations costs in financial applications and speeds up computation in AI, among other benefits. PCA solutions that consider only a few variables are known as sparse PCA.

A direct approach to formulating the sparse PCA problem is to add a cardinality constraint. However, this results in an NP-hard problem (Natarajan, 1995). As an alternative, sparsity-inducing penalties are broadly used to obtain sparse solutions to the PCA problem. We refer to these types of formulations as Penalized PCA. Penalized formulations are motivated by their computational tractability, scalability, and statistical property of shrinkage (Guerra-Urzola et al., 2022). Despite their practical benefits, methods proposed in the literature for solving penalized PCA problems have no guarantee of optimality (locally or globally) as they rely on heuristic solutions (Berk & Bertsimas, 2019; Bertsimas & Van Parys, 2020). In addition, there are no verifiable necessary and/or sufficient local or global optimality conditions for the penalized PCA problems studied in the literature. Consequently, reported numerical experiments that compare the performance of different penalized methods miss a theoretical underpinning. We focus on establishing a necessary optimality condition for the following penalized PCA formulation:

$$\max_{\mathbf{w}\in\mathcal{B}} \quad \|\mathbf{X}\mathbf{w}\| - \delta(|\mathbf{w}|), \tag{4.1}$$

where $\mathbf{X} \in \mathbb{R}^{I\times J}$ a data set, $\delta(\cdot)$ is a sparsity-inducing penalty, and $\mathcal{B} = \{\mathbf{x} \in \mathbb{R}^J \mid \|\mathbf{x}\| \leq 1\}$ is the unit Euclidean ball in $\mathbb{R}^J$.

Several sparsity-inducing penalties $\delta(\cdot)$ have been proposed in the PCA context. The Elastic Net penalty is the first sparsity-inducing penalty used to formulate a penalized PCA problem (Zou et al., 2006). The Elastic Net penalty combines the well-known Lasso (l_1-norm) and Ridge (l_0-norm) penalties allowing for a greater number of nonzero coefficients than using only the Lasso penalty. H. Shen and Huang (2008) proposed a formulation with three sparsity-inducing penalties: l_1-norm, l_0-norm, and SCAD penalty. The SCAD penalty leads to unbiased estimates when the coefficients are large enough (Fan & Li, 2001). Leng and Wang (2009) considered the adaptive l_1-norm to take into account the estimated relative importance of each

parameter. The above mentioned methods used an alternating optimization scheme to solve each proposed problem.

Problem (4.1) was first introduced by Journée et al. (2010) with the l_1 and l_0 norms as penalties $\delta(\cdot)$. Journée et al. (2010) solved the problem by introducing a gradient method to maximize a convex function known as the generalized power method (GPower). The GPower method has proven advantageous performance regarding recovering the sparse structure, explained variance, and scalability (Guerra-Urzola et al., 2021). Sriperumbudur, Torres, and Lanckriet (2011) also used the l_0-norm as a sparsity-inducing penalty and solved the problem using a majorization-maximization algorithm. Finally, Richtárik et al. (2021) used the l_1 and l_0 norms as sparsity-inducing penalties and proposed an alternating optimization method with proven equivalence to the GPower method[1].

In this paper, we use an alternating optimization scheme similar to Richtárik et al. (2021) to solve problem (4.1). The main contributions of this paper are twofold. First, we consider a necessary optimality condition stating that there are no feasible directions for the function value of problem (4.1) to be improved at a given point. We conduct a numerical analysis of the alternating scheme showing that the solution satisfies this necessary optimality condition under a suitable condition for $\mathbf{X}$. Necessary optimality conditions have been studied in the cardinality-constrained PCA context (Beck & Vaisbourd, 2016) but not for the penalized PCA problem (4.1). Second, the alternating optimization scheme allows us to characterize the sparsity-inducing penalties δ that lead to a continuous thresholding operator as the solution to the optimization problem. We prove that penalties with a singularity at the origin lead to a solution given by a continuous thresholding rule. Then, we propose two new penalized PCA formulations using the SCAD penalty and adaptive l_1-norm in model (4.1) as sparsity-inducing penalties.

The remainder of the paper proceeds as follows. Section 4.2 presents the algorithm, convergence analysis, and the necessary optimality conditions of problem (4.1). Section 4.3 will present examples of sparsity-inducing penalties and introduce model (4.1) with the SCAD penalty and adaptive l_1-norm. Finally, Section 4.4 presents some concluding remarks.

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 l_0-norm is denoted by $\|\mathbf{x}\|_0$ and represents the number of nonzero elements in $\mathbf{x}$. The l_1-norm is defined by $\|\mathbf{x}\|_1 = \sum_{j=1}^{J} |x_j|$, and the Euclidean distance by $\|\mathbf{x}\| = (\sum_{j=1}^{J} x_j^2)^{1/2}$. By $\mathcal{B} = \{\mathbf{x} \in \mathbb{R}^J \mid \|\mathbf{x}\| \leq 1\}$ we refer to the unit Euclidean ball in $\mathbb{R}^J$. We also define the support of $\mathbf{w}$ as the set of indexes with a non-zero element, and we denote it by $\text{supp}(\mathbf{w}) \equiv \{j | w_j \neq 0\}$.

[1]There are several more sparse PCA methods proposed in the literature. Here, we mention the most relevant methods to solve a penalized PCA problem. See Chapter 2 for a comprehensive overview of sparse PCA methods.

4.2 Theoretical Framework

In this section, we introduce a numerical method to solve problem (4.1) (Sect. 4.2.1). Then we study the provided solution (Sect. 4.2.2) and show under which condition it satisfies the necessary optimality condition (Sect. 4.2.3).

4.2.1 Alternating Thresholding Method

Because $\|\mathbf{a}\| = \max_{\mathbf{b}\in\mathcal{B}} \mathbf{b}^\top\mathbf{a}$, problem (4.1) can be equivalently reformulated as:

$$(\mathbf{w}^*, \mathbf{z}^*) = \underset{\mathbf{w},\mathbf{z}\in\mathcal{B}}{\operatorname{argmax}}\ \mathbf{z}^\top\mathbf{X}\mathbf{w} - \delta(|\mathbf{w}|). \tag{4.2}$$

where $\delta(\cdot)$ is applied element-wise to $\mathbf{w}$.

We solve problem (4.2) via an alternating optimization scheme. First, $\mathbf{z}$ is obtained as $\mathbf{z}^* = \mathbf{X}\mathbf{w}/\|\mathbf{X}\mathbf{w}\|$ for a given value $\mathbf{w}$. Second, for a given value of $\mathbf{z}$, $\mathbf{w}$ is obtained as $\mathbf{w}^* = T_\delta(\mathbf{X}^\top\mathbf{z})$ where

$$T_\delta(\mathbf{h}) := \underset{\mathbf{w}\in\mathcal{B}}{\operatorname{argmax}}\ \mathbf{h}^\top\mathbf{w} - \delta(|\mathbf{w}|), \tag{4.3}$$

for any $\mathbf{h}$. Then, we use Algorithm 4 to solve problem (4.2).

Algorithm 4: Alternating Thresholding Method

Input : $\mathbf{X}, \mathbf{w}^0$
Output: $\mathbf{w}^*$

repeat
  From $t = 0$, in iteration t,
  $\mathbf{z}^t = \frac{\mathbf{X}\mathbf{w}^t}{\|\mathbf{X}\mathbf{w}^t\|}$
  $\mathbf{w}^{t+1} = T_\delta(\mathbf{X}^\top\mathbf{z}^t)$
until *a stopping criterion is satisfied*;

The next result states the condition for $\delta(\cdot)$ such that the solution to problem (4.3) is a continuous threshold rule.

Proposition 4.2.1. *Consider problem* (4.3). *Suppose the penalty* $\delta(\cdot)$ *is separable in each component* j, *and it has a singularity at the origin. In that case, the optimal solution to the problem* (4.3) *is given by a continuous threshold rule denoted by* $T_\delta(\mathbf{h})$, *i.e.,* $\delta(\cdot)$ *is a sparsity-inducing penalty.*

Proof. The Lagrangian of problem (4.3) is $\mathcal{L} = \mathbf{h}^\top \mathbf{w} - \delta(|\mathbf{w}|) - \mu(\|\mathbf{w}\|^2 - 1)$. Then, the first-order conditions of problem (4.3) are:

$$\begin{aligned} h_j - \text{sign}(w_j)(\delta'(|w_j|) + 2\mu|w_j|) &= 0, \quad \text{for } j \in [J] \\ \|\mathbf{w}\|^2 &\leq 1 \\ \mu &\geq 0 \\ \mu(\|\mathbf{w}\|^2 - 1) &= 0. \end{aligned} \tag{4.4}$$

Taking into account that $0 \leq \delta'(|w_j|) + 2\mu|w_j|$, we analyze two cases: $|h_j| < \min_{w_j \neq 0}\{\delta'(|w_j|) + 2\lambda|w_j|\}$ and $|h_j| > \min_{w_j \neq 0}\{\delta'(|w_j|) + 2\lambda|w_j|\}$, for $j \in [J]$. When $|h_j| < \min_{w_j \neq 0}\{\delta'(|w_j|) + 2\lambda|w_j|\}$ the partial derivative of the Lagrangian is negative for all positive w_j's and positive for all negative w_j's. Therefore, there must be a maximum at $w_j = 0$ for $|h_j| < \min_{w_j \neq 0}\{\delta'(|w_j|) + 2\lambda|w_j|\}$ leading to a threshold rule that depends on the value of h_j. In the other case, when $|h_j| > \min_{w_j \neq 0}\{\delta'(|w_j|) + 2\lambda|w_j|\}$, and if the minimum of $\{\delta'(|w_j|) + 2\lambda|w_j|\}$ is reached at 0, there exists w_j that satisfies the equation (4.4) so that the solution is continuous in h_j. Figure 4.1 provides further insight into these two cases. We conclude that a sparsity-inducing penalty that is sparse and continuous must be singular at the origin. □

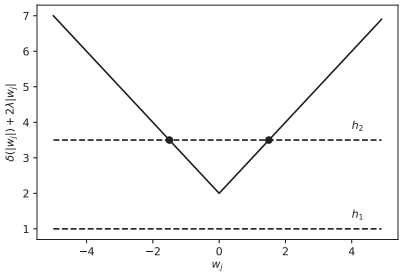


FIGURE 4.1: Sufficient condition of the penalties for having sparsity and continuity of the thresholding: $|h_1| < \min_{w_j \neq 0}\{\delta'(|w_j|) + 2\lambda|w_j|\}$ and $|h_2| > \min_{w_j \neq 0}\{\delta'(|w_j|) + 2\lambda|w_j|\}$, for some $j \in [J]$.

From now on, we assume that the penalty $\delta(\cdot)$ has a singularity at the origin. Fan and Li (2001) also studied this assumption in the context of a penalized linear regression problem.

4.2.2 Convergence Analysis

For clarity, let us define the objective of problem (4.1) as $Q(\mathbf{w}) = \|\mathbf{X}\mathbf{w}\| - \delta(|\mathbf{w}|)$, σ_{min} and σ_{max} denote the minimum and maximum eigenvalues of the covariance

matrix $\mathbf{X}^\top\mathbf{X}$, respectively. We start the convergence analysis by showing that the sequence generated by Algorithm 4 converges in value.

Proposition 4.2.2. *Let $\mathbf{w}^0 \in \mathcal{B}$. Let $\{\mathbf{w}^t\}_{t\geq 1}$ be the sequence generated using Algorithm 4 starting at $\mathbf{w}^0$. Then, $\{Q(\mathbf{w}^t)\}_{t\geq 1}$ is increasing and $\lim_{t\to\infty} Q(\mathbf{w}^t)$ exists.*

Proof. In iterate $t+1$, Algorithm 4 is equivalent to

$$\mathbf{w}^{t+1} = \operatorname{argmax}\left\{ \frac{\mathbf{w}^{t\top}\mathbf{X}^\top\mathbf{X}\mathbf{w}}{\|\mathbf{X}\mathbf{w}^t\|} - \delta(|\mathbf{w}|) : \mathbf{w} \in \mathcal{B} \right\}. \tag{4.5}$$

By the definition of $Q(\cdot)$,

$$\begin{aligned} Q(\mathbf{w}^{t+1}) - Q(\mathbf{w}^t) &= \left\|\mathbf{X}\mathbf{w}^{t+1}\right\| - \delta(|\mathbf{w}^{t+1}|) - \left\|\mathbf{X}\mathbf{w}^t\right\| + \delta(|\mathbf{w}_t|) \\ &\geq \left\|\mathbf{X}\mathbf{w}^{t+1}\right\| - \frac{\mathbf{w}^{t\top}\mathbf{X}^\top\mathbf{X}\mathbf{w}^{t+1}}{\left\|\mathbf{X}\mathbf{w}^t\right\|} \\ &\geq 0. \end{aligned}$$

The first inequality comes from Eq. (4.5), and the last inequality follows by applying the Cauchy-Schwarz inequality. Furthermore, $Q(\cdot)$ is bounded above by σ_{max} in the feasible set $\mathcal{B}$. Therefore, the sequence $\{Q(\mathbf{w}^t)\}_{t\geq 1}$ increases and is bounded above, which implies that $\lim_{t\to\infty} Q(\mathbf{w}^t)$ exists. □

Given that the feasible set $\mathcal{B}$ is a compact set, we can conclude that the sequence generated using Algorithm 4 possesses accumulation points. In the Lemma 4.2.1, we show that the sequence generated by Algorithm 4 converges under the assumption that $\sigma_{min} > 1$. This implies that $\operatorname{supp}(\mathbf{w}^t)$ stabilizes; that is, the support is the same for all $t > N$ for some N.

Lemma 4.2.1. *Let $\mathbf{w}^0 \in \mathcal{B}$. Let $\{\mathbf{w}^t\}_{t\geq 1}$ be the sequence generated using Algorithm 4 starting at $\mathbf{w}^0$. For all $\epsilon > 0$, if the minimum eigenvalue of the matrix $\mathbf{X}^\top\mathbf{X}$ is greater than 1, there exists $N \in \mathbb{N}$ such that $\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 < \epsilon$ for all $t > N$.*

Proof. Let us denote $Q^* = \lim_{t\to\infty} Q(\mathbf{w}^t)$. We show that $\sum_{t=0}^{\infty} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2$ converges, which implies the desired result. This is done by showing that the sequence

of partial sums $\sum_{t=0}^{N} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2$ is bounded. It follows from Proposition 4.2.2:

$$\begin{aligned} Q(\mathbf{w}^{t+1}) - Q(\mathbf{w}^t) &\geq \frac{\|\mathbf{X}\mathbf{w}^{t+1}\|\|\mathbf{X}\mathbf{w}^t\| - 1 + (1 - \mathbf{w}^{t\top}\mathbf{\Sigma}\mathbf{w}^{t+1})}{\|\mathbf{X}\mathbf{w}^t\|} \\ &\geq \frac{\|\mathbf{X}\mathbf{w}^{t+1}\|\|\mathbf{X}\mathbf{w}^t\| - 1 + \|\mathbf{X}(\mathbf{w}^{t+1} - \mathbf{w}^t)\|^2}{\sigma_{max}^{1/2}} \\ &\geq \frac{\|\mathbf{X}\mathbf{w}^{t+1}\|\|\mathbf{X}\mathbf{w}^t\| - 1}{\sigma_{max}^{1/2}} + \frac{\sigma_{min}}{\sigma_{max}^{1/2}}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \\ &\geq \frac{\sigma_{min} - 1}{\sigma_{max}^{1/2}} + \frac{\sigma_{min}}{\sigma_{max}^{1/2}}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \\ Q(\mathbf{w}^{t+1}) - Q(\mathbf{w}^t) + \frac{1 - \sigma_{min}}{\sigma_{max}^{1/2}} &\geq \frac{\sigma_{min}}{\sigma_{max}^{1/2}}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2. \end{aligned}$$

When $\sigma_{min} > 1$, the quantity $1 - \sigma_{min} < 0$, which implies that

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

Therefore, summing up both sides of the last inequality,

$$\sum_{t=1}^{\infty} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \leq \frac{\sigma_{max}^{1/2}}{\sigma_{min}}[Q^* - Q(\mathbf{w}_0)].$$

□

Proposition 4.2.3. *Let $\mathbf{w}^0 \in \mathcal{B}$. Let $\{\mathbf{w}^t\}_{t\leq 1}$ be the sequence generated using Algorithm 4 starting at $\mathbf{w}^0$. Assuming that the minimum eigenvalue of the matrix $\mathbf{X}^\top\mathbf{X}$ is greater than 1, there exists $N \in \mathbb{N}$ such that for all $t > N$, $\text{supp}(\mathbf{w}^{t+1}) = \text{supp}(\mathbf{w}^t)$.*

Proof. Without loss of generality, let us define λ_δ as the threshold rule in operator T_δ (see proposition 4.2.1). Taking any $0 < \epsilon < \lambda_\delta$, if $w_j^t > \lambda_\delta$ and $w_j^{t+1} = 0$, then $\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 > \epsilon$, which is impossible for $t > N$ for some N due to Lemma 4.2.1. □

Lemma 4.2.1 and Proposition 4.2.3 give an upper bound on the number of iterations t^* it takes for Algorithm 4 to produce a small step size and therefore the stability of the support. In fact,

$$t^* \geq \frac{\sigma_{max}^{1/2}[Q^* - Q(\mathbf{w}_0)]}{\sigma_{min}\epsilon^2} - 1 \Rightarrow \min_{0\leq t\leq t^*} \|\mathbf{w}^{t+1} - \mathbf{w}^t\| < \epsilon.$$

Lemma 4.2.2. *Let $\mathbf{w}^0 \in \mathcal{B}$. Let $\{\mathbf{w}^t\}_{t\leq 1}$ be the sequence generated by Algorithm 4 starting at $\mathbf{w}^0$. Let $\mathbf{w}^*$ be an accumulation point of $\{\mathbf{w}^t\}_{t\leq 1}$. Then $\mathbf{w}^* = T_\delta\left(\frac{\mathbf{X}^\top\mathbf{X}\mathbf{w}^*}{\|\mathbf{X}\mathbf{w}^*\|}\right)$.*

Proof. Consider a convergent subsequence $\{\mathbf{w}^{t_s}\}_{t_s \geq 1}$ of the sequence $\{\mathbf{w}^t\}_{t \leq 1}$ such that $\mathbf{w}^* = \lim_{s \to \infty} \mathbf{w}^{t_s}$. Given that $\delta(\cdot)$ has a singularity at the origin, T_δ a continuous operator (see Proposition 4.2.1). Thus,

$$T_\delta\left(\frac{\mathbf{X}^\top \mathbf{X}\mathbf{w}^*}{\|\mathbf{X}\mathbf{w}^*\|}\right) = T_\delta\left(\lim_{s\to\infty} \frac{\mathbf{X}^\top \mathbf{X}\mathbf{w}^{t_s}}{\|\mathbf{X}\mathbf{w}^{t_s}\|}\right) = \lim_{s\to\infty} \mathbf{w}^{t_s+1} = \lim_{s\to\infty} \mathbf{w}^{t_s} = \mathbf{w}^*.$$

The last two equalities hold by Lemma 4.2.1. □

4.2.3 Necessary Optimality Conditions

We show below that the solution from Algorithm 4 converges to a point at which there is no *feasible ascent direction* for problem (4.1). Before we state this result, we define a feasible ascent direction.

Definition 1 (Feasible Ascent Direction, Beck (2014))**.** *Consider the problem*

$$\max_{\mathbf{x}\in\Omega} \; G(\mathbf{x}). \tag{P}$$

A vector $\mathbf{d}$ *is called a feasible ascent direction in* ω *at* $\mathbf{x}$ *for* (P) *if* $\nabla G(\mathbf{x})^\top \mathbf{d} > 0$*, and there exists* $\epsilon > 0$ *such that* $\mathbf{x} + \mu\mathbf{d} \in \omega$ *for all* $\mu \in [0, \epsilon]$*.*

Proposition 4.2.4 (Necessary Optimality Condition For Local Optimizer)**.** *Let* $\mathbf{w}^*$ *be a local optimizer of the problem* (4.1)*, then there exits no feasible ascent direction at* $\mathbf{w}^*$*.*

Proof. Following Lemma 11.2 in Beck (2014) with $-G(\mathbf{x})$ and feasible descent direction, we obtain the desired result. □

We consider the absence of feasible ascent directions as a necessary optimality condition for the problem (4.1). Now, we move forward to show that the accumulation point $\mathbf{w}^*$ satisfies the necessary optimality conditions stated in Proposition 4.2.4.

Proposition 4.2.5 (Necessary Optimality Conditions)**.** *Let* $\mathbf{w}^0 \in \mathcal{B}$*. Let* $\{\mathbf{w}^t\}_{t\leq 1}$ *be the sequence generated using Algorithm 4 starting at* $\mathbf{w}^0$*. Let* $\mathbf{w}^*$ *be an accumulation point of* $\{\mathbf{w}^t\}_{t\leq 1}$*. If the minimum eigenvalue of the matrix* $\mathbf{X}^\top\mathbf{X}$ *is greater than* 1 *and* $\delta(\cdot)$ *has a singularity at the origin, then there is no feasible ascent direction at* $\mathbf{w}^*$ *for problem* (4.1)*.*

Proof. Let $Q_2(\mathbf{w}) = \frac{\mathbf{w}^{*\top}\mathbf{X}^\top\mathbf{X}\mathbf{w}}{\|\mathbf{X}\mathbf{w}^{*\top}\|} - \delta(|\mathbf{w}|)$. From Lemma 4.2.2, it holds that $\mathbf{w}^* = T_\delta\left(\frac{\mathbf{X}^\top\mathbf{X}\mathbf{w}^*}{\|\mathbf{X}\mathbf{w}^*\|}\right) := \operatorname{argmax}\{Q_2(\mathbf{w}) : \|\mathbf{w}\| \leq 1\}$. Thus, there is no feasible ascent direction in $\mathcal{B}$ at $\mathbf{w}^*$ for Q_2. But, $\nabla Q(\mathbf{w}^*) = \nabla Q_2(\mathbf{w}^*)$. Then, there is no feasible direction in $\mathcal{B}$ at $\mathbf{w}^*$ for problem (4.1) either. □

In this section, we showed that the sequence generated using Algorithm 4 converges in value, that $\text{supp}(\mathbf{w}^t)$ stabilizes for $t > N$, for some N. We also proved that an accumulation point of the sequence generated by Algorithm 4 satisfies the necessary optimality condition for the problem (4.1). In Sect. 4.3, we introduce some sparsity-inducing penalties that fit the theoretical framework presented in this section.

4.3 Sparsity-Inducing Penalties and Operators

In Sect. 4.2, we showed that if the minimum eigenvalue of the covariance matrix is greater than 1 and the sparsity-inducing penalty has a singularity at the origin, then Algorithm 4 converges to a solution that satisfies the necessary optimality condition for problem (4.1). This section introduces some sparsity-inducing penalties that satisfy these conditions. Particularly in Sect. 4.3.1, we consider problem (4.1) with the l_1-norm as penalty. Also, we introduce the SCAD and adaptive l_1 penalties. Additionally, in Sect. 4.3.2, we consider the special case of problem (4.1) with the l_0-norm penalty. This one does not entirely satisfy the theoretical framework presented in Sect. 4.2, but it can be applied in some cases. We consider it relevant to include the l_0-norm penalty in this study due to its use in finding sparse PCA solutions (Journée et al., 2010; Sriperumbudur et al., 2011).

4.3.1 l_1-norm Penalties

From proposition 4.2.1, when the sparsity-inducing penalty $\delta(\cdot)$ has a singularity at the origin, it leads to a solution defined by a continuous threshold rule. This desired property is satisfied when considering l_1-norm as a sparsity-inducing penalty or combinations of it (Fan & Li, 2001). This section introduces the l_1-norm and SCAD penalties. As can be observed in Figure 4.2, these penalties have a singularity at the origin.

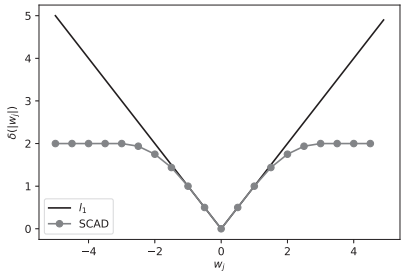


FIGURE 4.2: Norm l_1 and SCAD sparsity-inducing penalties.

l_1-norm

The l_1-norm is introduced as a sparsity-inducing penalty in the linear regression context for variable selection (Tibshirani, 1996) and extended to the PCA problem (Journée et al., 2010; H. Shen & Huang, 2008). The l_1-norm is defined as $\|\mathbf{w}\|_1 = \sum_{j=1}^{J} |w_j|$. Under this penalty problem (4.2) becomes

$$(\mathbf{w}^*, \mathbf{z}^*) = \underset{\mathbf{w},\mathbf{z}\in\mathcal{B}}{\operatorname{argmax}} \ \mathbf{z}^\top \mathbf{X}\mathbf{w} - \lambda \sum_{j=1}^{J} |w_j|. \tag{4.6}$$

Then, for a given value of $\mathbf{z}$, the Lagrangian of problem (4.6) is given by:

$$\mathcal{L}(\mathbf{w}, \lambda_2) = (\mathbf{X}^\top \mathbf{z})\mathbf{w} - \lambda \sum_{j\in J} |w_j| - \lambda_2(\|\mathbf{w}\|^2 - 1),$$

and the KKT-conditions are:

$$\begin{aligned} \mathbf{X}_j^\top \mathbf{z} - \lambda \operatorname{sign}(w_j) - 2\lambda_2 w_j = &\quad 0 \quad \forall j \in [J] \\ \mathbf{w}^\top \mathbf{w} \leq &\quad 1 \\ \lambda_2 \geq &\quad 0 \\ \lambda_2(\mathbf{w}^\top \mathbf{w} - 1) = &\quad 1. \end{aligned}$$

Solving the KKT equations, we have that:

$$w_j = \frac{\operatorname{sign}(\mathbf{X}_j^\top \mathbf{z})[|\mathbf{X}_j^\top \mathbf{z}| - \lambda]_+}{\|[|\mathbf{X}^\top \mathbf{z}| - \lambda]_+\|}.$$

Then, at iteration $t+1$, the updated of $\mathbf{w}$ is given by,

$$\mathbf{w}^{t+1} = \frac{S_\lambda(\mathbf{X}^\top \mathbf{z}^t)}{\|S_\lambda(\mathbf{X}^\top \mathbf{z}^t)\|},$$

where S_λ is the *soft-thresholding* defined component-wise as $S_\lambda(u) = \operatorname{sign}(u)[|u| - \lambda]_+$. In Figure 4.3a, we observe that the soft-thresholding operator is a continuous operator that shrinks small values to zero and large values towards the threshold (bias).

SCAD Penalty

The Smoothly Clipped Absolute Deviation (SCAD) was introduced by Fan and Li (2001) to mitigate the shrinkage effect when using the l_1-norm as a penalty. SCAD penalty is a continuous piecewise function that presents a smooth transition between the soft-thresholding and the identity operator. Then, it eliminates the shrinkage effect of the soft thresholding on large values. The SCAD penalty, denoted by $\mathcal{P}(\cdot)$,

is defined as

$$\mathcal{P}(|w_i|) = \begin{cases} \lambda|w_i|, & \text{if } |w_i| \leq \lambda \\ \frac{2a\lambda|w_i| - w_i^2 - \lambda^2}{2(a-1)}, & \text{if } \lambda < |w_i| \leq a\lambda \\ \frac{(a+1)\lambda^2}{2}, & \text{otherwise} \end{cases} \tag{4.7}$$

for some $a > 2$ and $\lambda > 0$. When using SCAD as penalty, problem (4.2) becomes,

$$(\mathbf{w}^*, \mathbf{z}^*) = \underset{\mathbf{w},\mathbf{z}\in\mathcal{B}}{\operatorname{argmax}} \ \mathbf{z}^\top \mathbf{X}\mathbf{w} - \sum_{j=1}^{J} \mathcal{P}(|w_j|).$$

Then, at iteration $t+1$, the updated of $\mathbf{w}$ is given by,

$$\mathbf{w}^{t+1} = \frac{T_{\text{SCAD}}(\mathbf{X}^\top \mathbf{z}^t)}{\|T_{\text{SCAD}}(\mathbf{X}^\top \mathbf{z}^t)\|},$$

where T_{SCAD} is called the SCAD operator, and it is defined component-wise by,

$$T_{\text{SCAD}}(u) = \begin{cases} S_\lambda(u), & \text{if } |u| \leq 2\lambda \\ \frac{\text{sign}(u)[(a-1)|u| - a\lambda]}{a-2}, & \text{if } 2\lambda < |u| \leq a\lambda \\ u, & \text{if } |u| > a\lambda \end{cases}. \tag{4.8}$$

In Figure 4.3b, we observe that the SCAD operator is a continuous operator that leaves unpenalized values larger than $a\lambda$.

Adaptive l_1-norm

The adaptive l_1-norm penalty was introduced by Zou (2006) to penalize each coefficient independently in the linear regression context. The adaptive l_1-norm is defined as $\sum_{j=1}^{J} \lambda_j |w_j|$, with $\lambda_j >$ for all $j \in [J]$. When using adaptive l_1-norm as penalty, problem (4.2) becomes,

$$(\mathbf{w}^*, \mathbf{z}^*) = \underset{\mathbf{w},\mathbf{z}\in\mathcal{B}}{\operatorname{argmax}} \ \mathbf{z}^\top \mathbf{X}\mathbf{w} - \sum_{j=1}^{J} \lambda_j |w_j|.$$

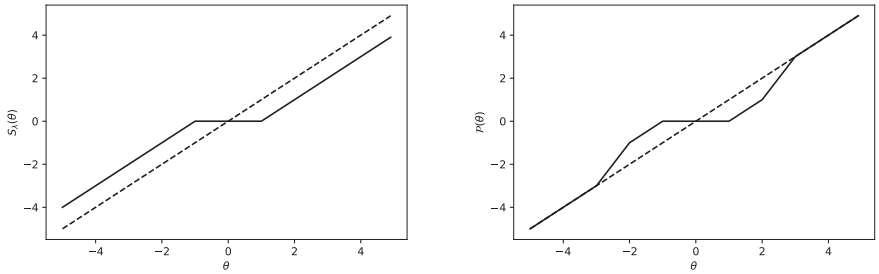
Then, in iteration $t+1$, Algorithm 4 updates $\mathbf{w}$ is given by,

$$\mathbf{w}^{t+1} = \frac{S_{\boldsymbol{\lambda}}(\mathbf{X}^\top \mathbf{z}^t)}{\|S_{\boldsymbol{\lambda}}(\mathbf{X}^\top \mathbf{z}^t)\|},$$

with $S_{\boldsymbol{\lambda}}$ defined component-wise by $S_{\boldsymbol{\lambda}}(\mathbf{u})_j = S_{\lambda_j}(u_j)$.

4.3.2 l_0-norm

The l_0-norm of $\mathbf{u} \in \mathbb{R}^J$ is defined as the number of nonzero elements in $\mathbf{u}$. Often, it is used to relax the cardinality constraint. The l_0-norm can be defined as $\sum_{j\in J} \mathbb{1}_{w_j \neq 0}$,



(A) Soft-Thresholding, $\lambda = 2$ (B) SCAD, $\lambda = 2$ and $a = 3$

FIGURE 4.3: Soft-Thresholding and SCAD Operators

with $\mathbb{1}_{w_j \neq 0}$ is an indicator operator that assigns 1 if $w_j \neq 0$ and 0 otherwise. When using the l_0-norm as penalty, problem (4.3) becomes,

$$(\mathbf{w}^*, \mathbf{z}^*) = \underset{\mathbf{w},\mathbf{z} \in \mathcal{B}}{\operatorname{argmax}} \ \mathbf{z}^\top \mathbf{X}\mathbf{w} - \lambda \sum_{j \in J} \mathbb{1}_{w_j \neq 0}. \tag{4.9}$$

Then the Lagrangian of problem (4.9) for a given value of $\mathbf{z}$ is:

$$\mathcal{L}(\mathbf{w}, \lambda_2) = (\mathbf{X}^\top \mathbf{z})\mathbf{w} - \lambda \sum_{j \in J} 1_{w_j \neq 0} - \lambda_2(\|\mathbf{w}\|^2 - 1),$$

and the KKT conditions by:

$$\begin{aligned} \mathbf{X}_j^\top \mathbf{z} - 2\lambda_2 w_j = &\ 0 \quad \forall j \in [J] \\ \mathbf{w}^\top \mathbf{w} \leq &\ 1 \\ \lambda_2 \geq &\ 0 \\ \lambda_2(\mathbf{w}^\top \mathbf{w} - 1) = &\ 1 \end{aligned}$$

Solving the KKT system, in iteration $t+1$, Algorithm 4 updates $\mathbf{w}$ is given by,

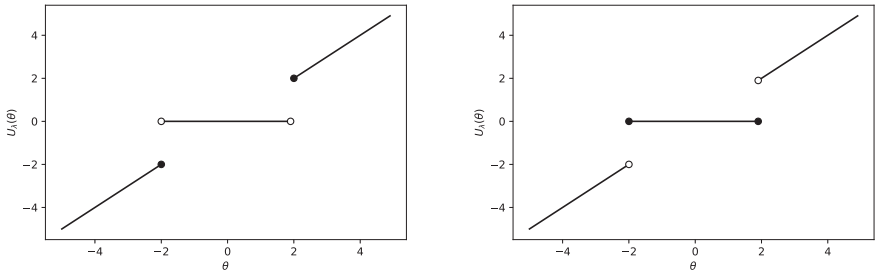
$$\mathbf{w}^{t+1} = \frac{U_\lambda(\mathbf{X}^\top \mathbf{z}^t)}{\|U_\lambda(\mathbf{X}^\top \mathbf{z}^t)\|}.$$

Where $U_\lambda(\cdot)$ is the *hard-thresholding* operator defined component-wise by

$$U_\lambda(\mathbf{u})_j = \begin{cases} 0, & \text{if } \frac{u_j^2}{\|\mathbf{u}\|} \leq \lambda \\ u_j & \text{if } \frac{u_j^2}{\|\mathbf{u}\|} \geq \lambda. \end{cases} \tag{4.10}$$

For the results in Sect. 4.2 to hold, the solution should be given by a continuous thresholding rule. The hard-thresholding operator presents a discontinuity only

at the threshold (see Figure 4.4). Notice that if the w_j^t does not converge to the threshold, then Algorithm 4 does not see this discontinuity, and the theoretical framework in Sect. 4.2 can be applied.



(A) Threshold rule $\geq \lambda$ (B) Threshold rule $> \lambda$

FIGURE 4.4: Hard-Thresholding Operator

4.4 Conclusion

The benefits and performance of the penalized PCA problem (4.1) have been studied in the literature based on numerical experiments without a theoretical explanation of their advantages. We analyzed an alternating thresholding method and showed that the solution satisfies the necessary conditions for optimality when the minimum eigenvalue of the covariance matrix is greater than one. Additionally, we characterized penalties that lead to a solution given by a continuous thresholding rule. We considered the l_1-norm as a sparsity-inducing penalty and proposed two new formulations of the penalized PCA problem (4.1) under the SCAD and adaptive l_1-norm penalties.

Having the minimum eigenvalue of the covariance matrix greater than one is a restrictive assumption to work with in practice. For instance, when dealing with high-dimensional data sets, the covariance matrix is known to be positive semidefinite with some eigenvalues equal to zero. For future work, it would be interesting to explore ways to relax this assumption so that the alternating thresholding method in Algorithm 4 can be shown to work for a broader range of empirical applications.

Chapter 5

Optimal penalized PCA using cardinality as sparsity-inducing penalty

Abstract

Methods for solving penalized principal component analysis (PCA) are widely used to find sparse solutions due to their computational tractability and scalability. One of the main criticisms of penalized 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 data set, including those in high dimensionality. Numerical experiments involving a synthetic data set and an empirical data set are conducted to demonstrate the implication of this condition in practice.

Keywords: Sparsity-inducing penalty, optimality conditions, PCA

5.1 Introduction

Sparse PCA methods have been proposed in the literature to gain interpretability and consistency in PCA solutions. Adding a cardinality constraint to the PCA problem seems to be a natural choice for achieving a sparse solution. However, it results in an NP-hard problem (Natarajan, 1995). To address this impracticality, relaxations that consider sparsity-inducing penalties have been used to achieve sparsity in the PCA solution. We call this kind of method Penalized PCA. Penalized methods are favorable regarding computational tractability, scalability, and statistical properties (Guerra-Urzola et al., 2022). Despite their practical advantage, penalized PCA methods in the literature rely on heuristic solutions without a theoretical guarantee of optimality.

Several penalties have been proposed to induce specific sparse structures in the solution. The most common of these penalties are the l_0 and l_1 norms. Representative work to solve the penalized PCA problem, using the norms l_0 and l_1, includes the well-known iterative GPower algorithm (Journée et al., 2010) and the alternating optimization scheme presented by Richtárik et al. (2021). On the other hand, Sriperumbudur et al. (2011) 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

$$\max_{\mathbf{w}\in\mathcal{B}} \ \mathbf{w}^\top\boldsymbol{\Sigma}\mathbf{w} - \alpha\|\mathbf{w}\|_0, \tag{5.1}$$

with $\alpha > 0$ denoting the penalty parameter, $\boldsymbol{\Sigma} = \mathbf{X}^\top\mathbf{X}$ 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 5.1, and show that it achieves a locally optimal solution to problem (5.1). Local optimality is attained by our method under the condition that the smallest eigenvalue of $\boldsymbol{\Sigma}$ is greater than 1. We show that this condition can be met for any data set by employing a simple procedure to transform the $\boldsymbol{\Sigma}$ matrix. The procedure can also ensure the condition to be met for high-dimensional data sets where $\boldsymbol{\Sigma}$ is positive semidefinite. To our knowledge, there are a few methods that studied the necessary optimality conditions (Sriperumbudur et al. (2011) and Chapter 4), but none have been able to prove optimality.

The remainder of the paper is as follows. Section 5.2 presents the minorization-maximization method and convergence analysis. In Section 5.3, we illustrate our method in a numerical setting using synthetic and real data sets. Finally, Section 5.4 provides a conclusion. Next, we collect our notation for the convenience of our readers.

[1]For a comprehensive review of penalized PCA method see Chapters 2, 3, and 4.

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 Euclidean distance 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.

5.2 Theoretical Framework

We use a minorization-maximization (MM) scheme in Sect. 5.2.1, the solution of which is given by an iterative thresholding algorithm in Sect. 5.2.2. We present some convergence analysis in Sect. 5.2.3 and show that our method converges to a local optimum solution of the problem (5.1).

5.2.1 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:

$$\mathbf{x}^{t+1} \in \operatorname*{argmax}_{\mathbf{x}} \; G(\mathbf{x}, \mathbf{x}^t). \tag{5.2}$$

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.

$$F(\mathbf{x}^{t+1}) \geq G(\mathbf{x}^{t+1}, \mathbf{x}^t) \geq G(\mathbf{x}^t, \mathbf{x}^t) = F(\mathbf{x}^t), \tag{5.3}$$

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. (2011) used an MM

algorithm for the penalized PCA problem, and the classical power method to solve the largest eigenvalue of a positive semidefinite matrix can also be derived from the MM perspective (Lange, 2016).

5.2.2 MM implementation for Problem (5.1)

For clarity, let us define the objective of problem (5.1) as $C(\mathbf{w}) = \mathbf{w}^\top \boldsymbol{\Sigma} \mathbf{w} - \alpha \|\mathbf{w}\|_0$. We propose the following minorizing function S over $\mathcal{B} \times \mathcal{B}$ as

$$S(\mathbf{w}, \mathbf{z}) = \mathbf{w}^\top \boldsymbol{\Sigma} \mathbf{w} - \alpha \|\mathbf{w}\|_0 - (\mathbf{w} - \mathbf{z})^\top (\boldsymbol{\Sigma} - \mathbf{I})(\mathbf{w} - \mathbf{z}) \tag{5.4}$$

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

$$\mathbf{w}^{t+1} \in \underset{\mathbf{w} \in \mathcal{B}}{\operatorname{argmax}} \; S(\mathbf{w}, \mathbf{w}^t), \tag{5.5}$$

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

Iterative Hard Thresholding

We now show that the update presented in Eq. (5.5) is equivalent to an iterative hard-thresholding rule. Let us consider the lagrangian of problem (5.5) as

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

Then, the KKT-conditions conditions are given by:

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

its a solution

$$\hat{\mathbf{w}} = \frac{(\boldsymbol{\Sigma} - \mathbf{I})\mathbf{w}^t}{\|(\boldsymbol{\Sigma} - \mathbf{I})\mathbf{w}^t\|}. \tag{5.6}$$

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_λ is defined component-wise as

$$U_\alpha(\mathbf{y})_j = \begin{cases} 0 & \text{if } \frac{y_j^2}{\|\mathbf{y}\|} < \alpha \\ y_j & \text{if } \frac{y_j^2}{\|\mathbf{y}\|} \geq \alpha \end{cases}. \tag{5.7}$$

We propose the algorithm 5 to find an optimal solution to problem (5.1).

Algorithm 5: Iterative threshold rule

Input : $\mathbf{\Sigma}, \mathbf{w}_0$
Output: $\mathbf{w}^*$
begin
  repeat
    In iteratation $t+1$,
    $\mathbf{w}^{t+1} = \frac{U_\alpha([\mathbf{\Sigma}-\mathbf{I}]\mathbf{w}^t)}{\|U_\alpha([\mathbf{\Sigma}-\mathbf{I}]\mathbf{w}^t)\|}$
  until $\mathbf{w}^{t+1} = \mathbf{w}^t$;
end

5.2.3 Convergence Analysis

We now conduct a convergence analysis of the solution achieved using the MM scheme in Eq. (5.5). We begin by showing in Lemma 5.2.1 that the sequence generated by Algorithm 5 increases and converges in value.

Lemma 5.2.1. *Let* $\mathbf{w}^0 \in \mathcal{B}$. *Let* $\{\mathbf{w}^t\}_{t\geq 1}$ *be the sequence generated using the MM scheme in Eq.* (5.5) *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 Eq. (5.5), 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 5.2.2, we show the sufficient condition to

guarantee that the use of the MM scheme in Eq. (5.5) converges and meets this stopping criterion $\mathbf{w}^{t+1} = \mathbf{w}^t$.

Lemma 5.2.2. *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 Eq.* (5.5) *starting at* $\mathbf{w}^0$. *Then,* $\lim_{t\to\infty} \|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 = 0$.

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

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

The last inequality comes from the inequalities in the proof of Lemma 5.2.1. 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 5.2.2 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 5 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 (5.1) using $\hat{\mathbf{\Sigma}}$ is equivalent to use $\mathbf{\Sigma}$ as follows.

$$\begin{aligned}
\mathbf{w}^* \in &\operatorname*{argmax}_{\mathbf{w}\in\mathcal{B}} \ \mathbf{w}^\top\mathbf{\Sigma}\mathbf{w} - \alpha\|\mathbf{w}\|_0 + \tau 1 \\
\Leftrightarrow &\operatorname*{argmax}_{\mathbf{w}\in\mathcal{B}} \ \mathbf{w}^\top\mathbf{\Sigma}\mathbf{w} - \alpha\|\mathbf{w}\|_0 + \tau\mathbf{w}^\top\mathbf{w} \\
\Leftrightarrow &\operatorname*{argmax}_{\mathbf{w}\in\mathcal{B}} \ \mathbf{w}^\top(\mathbf{\Sigma} + \tau\mathbf{I})\mathbf{w} - \alpha\|\mathbf{w}\|_0. \qquad (5.8)
\end{aligned}$$

This '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 et al., 2011).

Let the support $\text{supp}(\mathbf{w}) \equiv \{j|w_j \neq 0\}$ be the set of indexes with a nonzero element in $\mathbf{w}$. Lemma 5.2.2 implies that the support of the sequence generated using Algorithm 5 stabilizes, that is, it is the same after some N. This is stated in Corollary 5.2.1.

Corollary 5.2.1. *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 Eq.* (5.5)

starting at $\mathbf{w}^0$. *Then there exists* $N \in \mathbb{N}$ *such that, for all* $t > N$, $\text{supp}(\mathbf{w}^{t+1}) = \text{supp}(\mathbf{w}^t)$.

Proof. Let $\sigma_{max} > 1$ be the maximum eigenvalues of the matrix $\mathbf{\Sigma}$. If $w_j^t \neq 0$, we have from Eq. (5.7) that

$$\begin{aligned} w_j^{t\,2} = \frac{(U_\alpha([\mathbf{\Sigma}-\mathbf{I}]_j^\top \mathbf{w}^{t-1}))^2}{\|U_\alpha([\mathbf{\Sigma}-\mathbf{I}]\mathbf{w}^{t-1})\|^2} = \frac{([\mathbf{\Sigma}-\mathbf{I}]^\top \mathbf{w}^{t-1})_j^2}{\|U_\alpha([\mathbf{\Sigma}-\mathbf{I}]\mathbf{w}^{t-1})\|^2} \geq & \frac{([\mathbf{\Sigma}-\mathbf{I}]^\top \mathbf{w}^{t-1})_j^2}{\|[\mathbf{\Sigma}-\mathbf{I}]\mathbf{w}^{t-1}\|^2} \\ w_j^{t\,2}\|[\mathbf{\Sigma}-\mathbf{I}]\mathbf{w}^{t-1}\| \geq & \frac{([\mathbf{\Sigma}-\mathbf{I}]^\top \mathbf{w}^{t-1})_j^2}{\|[\mathbf{\Sigma}-\mathbf{I}]\mathbf{w}^{t-1}\|} \geq \alpha \\ w_j^{t\,2}(\sigma_{max}-1) \geq & \alpha \\ w_j^{t\,2} \geq & \frac{\alpha}{\sigma_{max}-1} \end{aligned} \tag{5.9}$$

Now, let consider any ϵ such that $0 < \epsilon < \alpha/(\sigma_{max}-1)$. From Lemma 5.2.2, it exists $N \in \mathbb{N}$ such that for any $t > N$, $\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 < \epsilon$. If $\text{supp}(\mathbf{w}^{t+1}) \neq \text{supp}(\mathbf{w}^t)$, there exists $j \in \text{supp}(\mathbf{w}^t \setminus \text{supp}(\mathbf{w}^{t+1}))$, which implies that $\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \geq \frac{\alpha}{\sigma_{max}-1}$ from Eq. (5.9). This is a contradiction.

From Corollary 5.2.1 and Eq. (5.6), it can be observed that when the support stabilizes, algorithm 5 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 5 is a local optimum of problem (5.1). This is stated in Theorem 5.2.1.

Local Optimizer

To finalize this section, we show that any solution obtained from Algorithm 5 is a local optimum of problem (5.1).

Proposition 5.2.1. *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 Eq.* (5.5) *starting at* $\mathbf{w}^0$ *and ending at* $\mathbf{w}^*$. *Let* $\mathbf{d} \in \mathcal{B}$ *be a feasible direction of problem* (5.1). *Then* $\text{supp}(\mathbf{w}^*) \subseteq \text{supp}(\mathbf{w}^* + \delta\mathbf{d})$ *for any* $0 < \delta < \sqrt{\alpha/(\sigma_{max}-1)}$ *with* σ_{max} *the maximum eigenvalue of* $\mathbf{\Sigma}$.

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

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

The second equality comes from the assumption that $j \in supp(\mathbf{w}^*) \setminus \text{supp}(\mathbf{w}^* + \delta\mathbf{d})$ and the last inequality from Eq. (5.9). Therefore, there is no $j \in supp(\mathbf{w}^*) \setminus \text{supp}(\mathbf{w}^* + \delta\mathbf{d})$, which implies the desired result. □

Theorem 5.2.1. *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 Eq.* (5.5) *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 consider any δ such that $0 < \delta < \sqrt{\alpha/(\sigma_{max} - 1)}$. From Proposition 5.2.1, it holds that $\text{supp}(\mathbf{w}^*) \subseteq \text{supp}(\mathbf{w}^* + \delta\mathbf{d})$. If $\text{supp}(\mathbf{w}^*) = \text{supp}(\mathbf{w}^* + \delta\mathbf{d})$, $\mathbf{w}^*$ is the solution of the Power method when the support stabilizes (see Corollary 5.2.1). 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

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

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

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

In both cases, we show that δ exists. □

5.3 Numerical Examples

In Lemma 5.2.2, we showed that the step size in Algorithm 5 convergences when the minimum eigenvalue (σ_{min}) of the matrix $\mathbf{\Sigma}$ is larger than 1. Here we illustrate this finding by administering our method on simulated and empirical data sets for which the condition is not satisfied. We illustrate that Algorithm 5 does not converge for these specific data sets and how this can be overcome by the aforementioned transformation of data in Eq. (5.8).

5.3.1 Synthetic Data Set

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

$$\underbrace{\begin{bmatrix} -0.302 \\ 0 \\ 0 \\ 0.302 \\ -0.905 \end{bmatrix}}_{\mathbf{v}_1} \underbrace{[5]}_{\sigma_1} \underbrace{\begin{bmatrix} -0.302 \\ 0 \\ 0 \\ 0.302 \\ -0.905 \end{bmatrix}^\top}_{\mathbf{v}_1^\top} = \underbrace{\begin{bmatrix} 0.455 & 0 & 0 & -0.455 & 1.364 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -0.455 & 0 & 0 & 0.455 & -1.364 \\ 1.364 & 0 & 0 & -1.364 & 4.091 \end{bmatrix}}_{\mathbf{\Sigma}} \tag{5.10}$$

By defining only the first eigenvalue ($\sigma_1 = 5$), the remaining four eigenvalues are defined as zero. Therefore, the smallest eigenvalue of the matrix $\mathbf{\Sigma}$ would be $\sigma_{min} = 0$. Then, implementing Algorithm 5, with penalty parameter $\alpha = 0.7$, in this particular setting, the sequence diverges; see Figure 5.3. Now, we illustrate that with a transformed matrix $\hat{\mathbf{\Sigma}} = \mathbf{\Sigma} + \tau\mathbf{I}$, with $\tau > 1$, Algorithm 5 converges (Figure 5.4) 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 5.5).

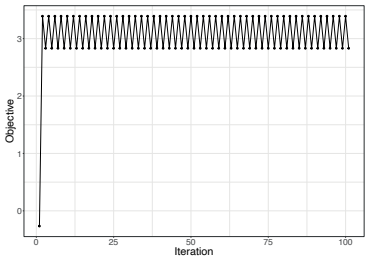


FIGURE 5.1: $C(\mathbf{w})$

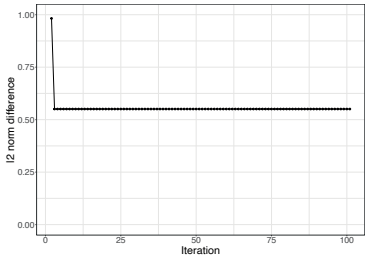
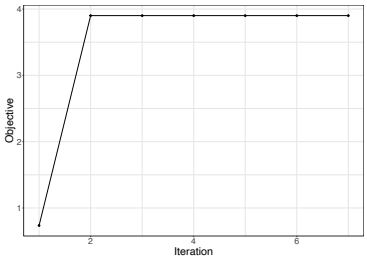


FIGURE 5.2: $\left\|\mathbf{w}^t - \mathbf{w}^{t+1}\right\|$

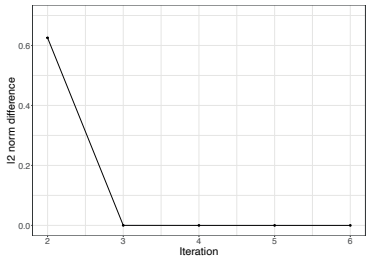
FIGURE 5.3: Divergence for the simulated $\mathbf{\Sigma}$. The objective function is displayed in (a), while (b) shows the l_2 norm of the difference between the iterates.

5.3.2 Empirical Data Set

We imported the '16S data', which relates 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 data set is characterized by 1674 measurements from 162 observation units. We imported the data set from the R-package 'mixOmics' (Rohart, Gautier, Singh, & Lê Cao, 2017).



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



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

FIGURE 5.4: Convergence with the transformation on the simulated data

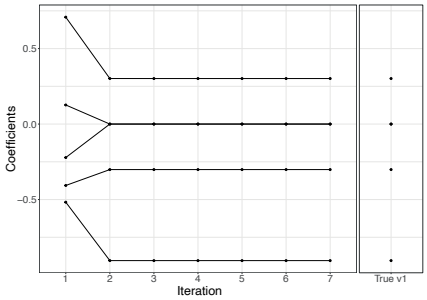
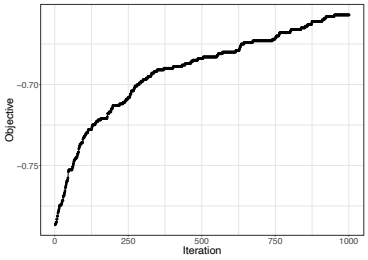
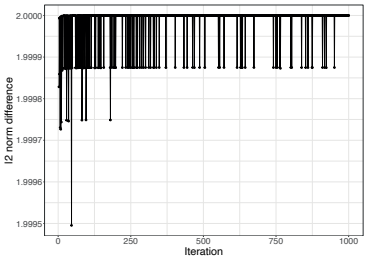


FIGURE 5.5: Series of $\mathbf{w}^t$ compared to the true eigenvector $\mathbf{v}_1$. Each line in the left panel represents each element of $\mathbf{w}^t$.

We perform the eigenvalue decomposition on the $\mathbf{\Sigma}$ matrix, which results in $\sigma_{max} = 0.603 < 0$. This implies $\sigma_{min} < 1$. With the penalty parameter $\alpha = 0.001$, we found that Algorithm 5 did not converge in 100000 iterations, see Figure 5.6c.



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

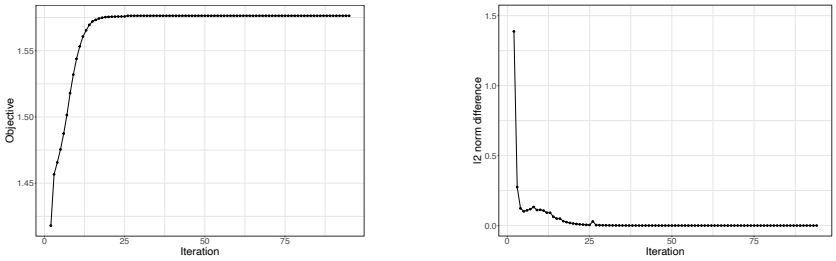


(B) $\left\|\mathbf{w}^t - \mathbf{w}^{t+1}\right\|$

(C) Divergence for the 16S data

We observe in Figure 5.6c that the objective $C(\mathbf{w})$ continues to increase and that the norm l_2 between two consecutive points does not decrease over iterations.

However, the sequence converges successfully when administered to the transformed matrix $\hat{\boldsymbol{\Sigma}}$, with the same initial vector and penalty parameter. Figure 5.7 shows that convergence is achieved in 96 iterations.



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

FIGURE 5.7: Convergence with the transformation on the 16S data

5.4 Conclusion

This paper considers the penalized PCA based on variance maximization and the l_0-norm as a sparsity-inducing penalty. We 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 (Sriperumbudur et al. (2011) and Chapter 4), this is the first account to prove optimality in the context of penalized PCA. Based on the MM principle, we derived an iterative method with 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 using a synthetic and an empirical data set.

For future work, it would be worth studying the optimality conditions using other types of penalties. It would be along the lines of the work in Chapter 4 that provided the optimal conditions necessary for specific penalties within penalized PCA. Additionally, the condition regarding the covariance matrix's minimum eigenvalue would be an interesting research topic. The same condition was also found for an alternating method in Chapter 4, which differs from our approach. It appears that the condition may apply to penalized PCA problems in general.

Chapter 6

Epilogue

This dissertation was conducted as a joint project between the Methodology and Statistics Department and the Operations Research Department at Tilburg University. We focus on the sparse Principal Component Analysis problem, addressing questions of interest in both disciplines. This dissertation aims to guide the use and implementation of sparse PCA methods that serve the data analysis purpose and study the optimality properties of some penalized PCA methods. For data analysis purposes, Chapter 2 provided a guide on using and implementing sparse PCA methods compared to several popular sparse PCA methods. Chapter 3 compared two ways of achieving sparse solutions. Chapter 4 moved into the theoretical aspect by developing a framework for the necessary optimality conditions of a penalized PCA method. Chapter 5 proposed a method with local optima solutions.

We clarified the misconception that PCA properties could be extended to sparse PCA problems, mainly that loadings and weights are the same. One of the main conclusions is that sparse PCA methods differ more fundamentally in their parameters and objectives and may better serve different purposes, such as Exploratory Data Analysis and Dimension Reduction. It was shown through extensive numerical experiments and real-life data sets that sPCA-rSVD offers the best results for Exploratory Data Analysis, finding structure in the data set, and assigning meaning to the scores. GPower is the best method for Dimension Reduction, finding new variables that maximally account for the variance in the data set. Regarding the sparsity-inducing scheme, this study indicates that cardinality-constrained methods can achieve more accounted variance with fewer variables than penalized methods. Then, using cardinality constraints results in more variance and less bias than using a penalty counterpart.

Penalized PCA methods have been mainly introduced and studied in the statistical literature. From the optimization point of view, one of the main criticisms is that the benefits of using a penalized PCA method have been shown only through numerical experiments without theoretical results on their optimality. We theoretically proved that an equivalent method to the well-known GPower satisfies necessary optimality conditions. Considering the l_0-norm as the sparsity-inducing penalty and minorization-maximization method, we show that the solution is a local optimum.

In conclusion, while it may be difficult to declare a single Sparse PCA method as the best, it is essential to recognize that different methods are better suited to

serve diverse purposes and objectives. Researchers should carefully consider their specific objectives when selecting a method for their analysis.

6.1 A Note on Statistics and Optimization

Which is the best sparse PCA method? We started this joint project thinking about solving this question of common interest in the statistics and optimization fields. Although statistics and optimization fields are interested in sparse PCA methods that enjoy computational tractability and scalability, they differ in other fundamental aspects. In statistics, there is great interest in sparse PCA methods with advantageous data analysis properties such as support recovery, estimation, and interpretation. Optimization concerns are directed to methods to find locally or globally optimal solutions efficiently.

6.1.1 Statistics

Formulations of most statistical models for sparse PCA, such as penalized PCA, are based on the least square or likelihood principle, which involves an optimization problem. Nevertheless, methods to solve penalized PCA models are not assessed based on the optimality of their solutions but using numerical experiments to measure their statistical properties. This raises some fundamental questions about the role of optimality in these models and whether they should be formulated and estimated differently based on statistical principles. An example that can be used as a departure point for such characterization is the Bayes Matrix factorization proposed by W. Wang and Stephens (2021), which assumes that the data can be factorized as:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E}.$$

W. Wang and Stephens (2021) used Bayesian estimation with prior distributions for $\mathbf{T}$ and $\mathbf{P}$ that induce sparsity on the final posterior distribution. Additionally, the number of columns of $\mathbf{P}$ is automatically selected by the number of components with probability mass 0 in the prior distribution.

6.1.2 Optimization

The optimization literature of sparse PCA has mainly focused on solving the cardinality-constrained PCA problem. Mix Integer Optimization (MIO) techniques have been recently used to solve this problem to certifiable optimality. This is partly due to the speedup factor of MIO solvers in the past decades by incorporating theoretical and practical advances with the exponentially increasing speed of supercomputers (Bixby, 2012). However, MIO methods for cardinality-constrained PCA are still not highly used by practitioners due to their scalability limitations and their computational tractability. So far, MIO can solve to certifiable optimality the cardinality-constrained PCA problem up to 1000's variables. Yet, those methods should be

designed for greater use by practitioners and researchers in fields different than optimization.

6.2 Future Directions

6.2.1 Bridge between statistics and optimization

Statistics and optimization fields have focused on formulating and solving the sparse PCA models that serve a particular interest of each discipline. For example, more methods with desirable data analysis properties have been proposed in statistics, and in optimization, more methods that attain specific optimal criteria have been proposed. However, the potential benefits of cross-disciplinary collaborations have not been fully explored yet. A promising direction would be to investigate techniques to assess the optimality properties of any sparse solution obtained from a penalized PCA method. Such a method would provide a valuable tool for researchers in statistics and optimization to analyze and interpret high-dimensional data sets efficiently.

6.2.2 How important is optimality in statistics?

Penalized PCA methods were introduced in the statistical literature about two decades ago. Since then, many methods have been proposed to solve the sparse PCA problem considering different sparse structures. The main criticism of penalized methods from the optimization point of view is that they rely on solutions without guaranteeing optimality, that is, heuristic solutions. Future research should investigate the implications of not having optimal solutions in penalized PCA methods. Specifically, the study should focus on understanding the trade-off between the quality of the heuristic solutions obtained and the optimality of the solutions. This could involve developing new theoretical frameworks for evaluating the performance of penalized PCA methods, including bounds on the optimality gap and characterizing the conditions under which heuristic solutions are likely to be close to optimal solutions. Such research could provide valuable insights into the limitations and strengths of penalized PCA methods and guide the development of more effective and reliable sparse PCA algorithms.

6.2.3 Self-Contained method

The current sparse PCA literature assumes that essential parameters such as the number of components, the proportion of sparsity, or the penalty parameter are known a priori. However, these parameters are unknown in many real-world applications and must be estimated. A promising direction for future research is to develop a self-contained data-driven method that can estimate these parameters and incorporate them into the optimization problem. Specifically, the research should

focus on developing new optimization algorithms that can estimate optimal values of these hyper-parameters simultaneously with the sparse loadings or weights. This could involve using techniques such as machine learning algorithms to learn the optimal values of the parameters from data. Such research could provide valuable insights into the optimal design of sparse PCA algorithms and facilitate their widespread use in various applications and fields.

Appendix A

A.1 Algorithm: CCPCA

We present a detailed description of the algorithm used to solve formulation (3.2). We have implemented an alternating procedure fixing one variable at a time. To estimate $\mathbf{P}$ with $\mathbf{W}$ fixed (see problem (3.3)), Procruste rotation has been used (ten Berge, 2005; Zou et al., 2006). That is, the solution to (3.3) is $\widehat{\mathbf{P}} = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors of $\mathbf{X}^\top\mathbf{X}\mathbf{W}$, respectively. To estimate $\mathbf{W}$ with $\mathbf{P}$ fixed (see problem (3.4)), the cardinality-constrained regression algorithm in Adachi and Kiers (2017) is implemented, which uses a majorization minimization approach (for a short overview see Hunter and Lange (2004)).

Following Kiers (2002), we can majorize (3.4) as follows,

$$\|\text{vec}(\mathbf{X}) - (\mathbf{P} \otimes \mathbf{X})\text{vec}(\mathbf{W})\|_2^2 \leq c + \alpha\|\mathbf{b} - \text{vec}(\mathbf{W})\|_2^2, \tag{A.1}$$

where c is a constant with respect to $\mathbf{W}$, α is the maximum eigenvalue of $\mathbf{X}^\top\mathbf{X}$ and $\mathbf{b}$ is given by $\mathbf{b} = \text{vec}(\mathbf{W}) - \alpha^{-1}\text{vec}\left(\mathbf{X}^\top\mathbf{X}(\mathbf{W} - \mathbf{P})\right)$. The right hand side in (A.1) is minimized when $\mathbf{b} = T_\rho\left(\text{vec}(\mathbf{W})\right)$. In fact, the same updating formula of the weights appears in the proximal gradient algorithm presented in Bertsimas et al. (2016) (Eq 3.8 in Algorithm 1).

In some instances, it can be more useful to specify the cardinality constraint per column of $\mathbf{W}$. This leads to more control over the sparsity level in the weights pertaining to specific components. This can be done by adding a cardinality constraints per $\mathbf{w}_k$ as follows,

$$\begin{aligned} \widehat{\mathbf{W}} = \underset{\mathbf{W}}{\text{argmin}} \quad & \|\text{vec}(\mathbf{X}) - (\mathbf{P} \otimes \mathbf{X})\text{vec}(\mathbf{W})\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{w}_k\|_0 = \rho_k \qquad\qquad \forall k = 1, 2, 3, \ldots, K, \end{aligned}$$

where ρ_k denotes the number of nonzero per component weight. In this case, the updating formula in each iteration becomes,

$$\mathbf{w}_{k,new} := T_{\rho_k}\left(\mathbf{w}_{k,old} - \alpha^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{w}_{k,old} - \mathbf{p}_k)\right).$$

An implementation of Algorithm (6) is freely available in R-software (*R: The R Project for Statistical Computing*, n.d.) and downloadable from the corresponding author's github page. As discussed in Section 3.2, This algorithm does not guarantee

Algorithm 6: CCPCA algorithm for sparse PCA

Input : $\mathbf{X}, K, \boldsymbol{\rho}, \mathbf{W}_0$
Output: $\widehat{\mathbf{W}}$

while Δ *lossfunction value* $> \epsilon$ **do**
  in iteration i
  $\widehat{\mathbf{P}} \leftarrow$ Procruste rotation$(\mathbf{X}, \widehat{\mathbf{W}})$
  for $k \leftarrow 1$ **to** K **do**
    $\mathbf{w}_{k,i} := T_{\rho_k}\left(\mathbf{w}_{k,i-1} - \alpha^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{w}_{k,i-1} - \mathbf{p}_k)\right)$
  end
end
return $\widehat{\mathbf{W}}$;

finding a global optimum of problem (3.4). A way of improving the solution is by initializing $\mathbf{W}$ with the estimates $\widehat{\mathbf{W}}$ from PCA. This "warm" start will nudge the algorithm in the right direction, minimizing the risk that the algorithm will end up in a local minimum of large value, far from optimal. Another way to improve the solution is by using multiple starts. The procedure can be started multiple times with different values for $\mathbf{W}_0$, and the result with the smallest loss function value is retained. Multiple starts are more costly, which especially adds up when K and ρ still need to be determined using model selection.

A.2 Data Generation

The data for the simulation study was generated from the following decomposition,

$$\mathbf{X} = \mathbf{X}\mathbf{W}\mathbf{P}^\top,$$

where $\mathbf{W} \in \mathbb{R}^{J\times K}$, $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$ and $\mathbf{P} = \mathbf{W}$. The orthogonality condition on $\mathbf{W}$ links to the PCA identification constraints and it has been considered before in simulation studies by Camacho, Smilde, Saccenti, and Westerhuis (2020); Camacho, Smilde, Saccenti, Westerhuis, and Bro (2021); Guerra-Urzola et al. (2021). The matrix $\mathbf{W}$ is constructed such that it contains a given level of sparsity. To achieve this, we used the following iterative procedure. First, a random matrix $\mathbf{W}$ is generated with zero weights in the desired places. Then, the orthogonality of the columns is attempted by applying the Gram-Schmidt orthogonalization procedure only in the intersection of the nonzero weights between two columns of $\mathbf{W}$. When $\mathbf{W}$ only has sets of columns that contain non-overlapping sparsity patterns, this immediately results into orthogonal columns, but when the columns in $\mathbf{W}$ have overlapping sparsity patterns the procedure will not always lead to $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$ on the first pass. In such cases, multiple passes are needed in order to achieve orthogonality (additional coefficients might need to be set equal to zero). Some sparsity patterns are impossible, for example, an initialization where $\mathbf{W}$ does not have full column

rank or an initial set that degenerates to a linearly dependent set after multiple passes. In those cases, the algorithm fails to converge.

After a suitable $\mathbf{W}$ has been obtained, $\mathbf{\Sigma}$ is constructed by taking $\mathbf{\Sigma} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^\top$. Here, $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues of the J components underlying the full decomposition. We specify these eigenvalues such that the first K components account for a set amount of structural variance and the remaining eigenvalues for a set amount of noise variance. The data matrices $\mathbf{X}$ having a desired underlying sparse structure and noise level can then be obtained by sampling from the multivariate normal distribution with zero mean vector and variance-covariance $\mathbf{\Sigma}$. Note that the generation scheme used here is very restrictive and may not be applicable to empirical data. Additionally, in our experiments, generating non-orthogonal weights did not change the results in any noticeable way. This code is publicly available at the author's github page.

A.3 Additional Plots

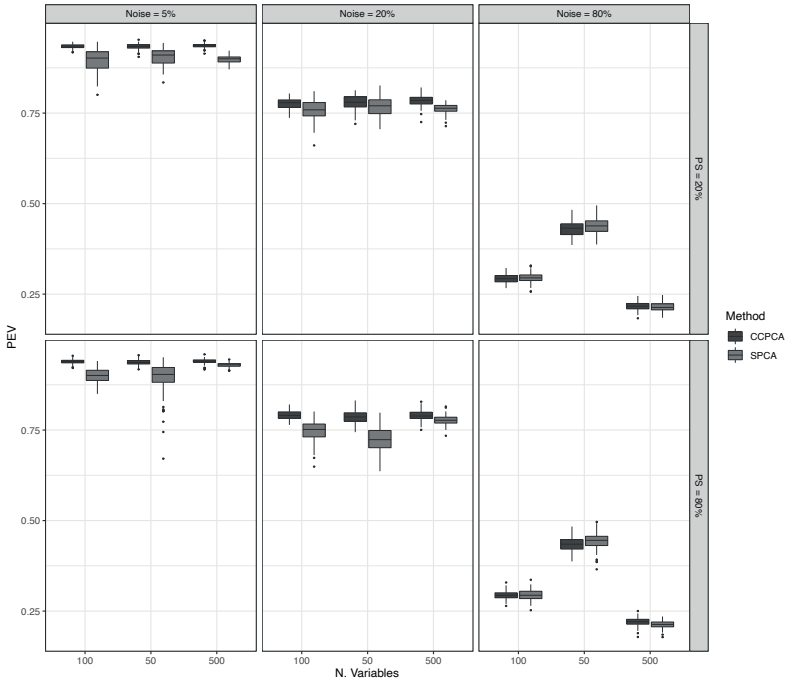


FIGURE A.1: Proportion of explained variance (PEV) with cardinality tuned using the index of sparseness.

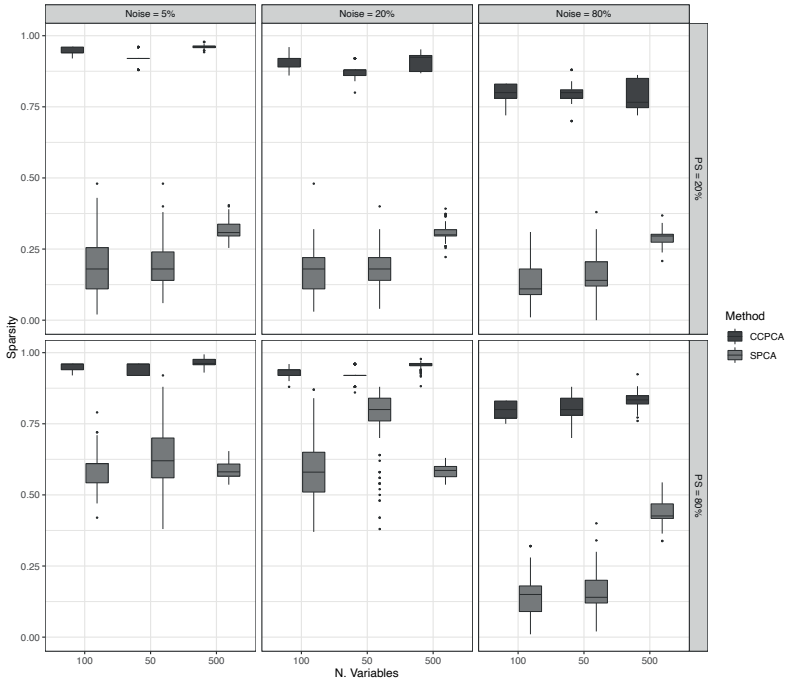


FIGURE A.2: Estimated proportion sparsity with cardinality tuned using the index of sparseness.

Acknowledgements

I started fantasizing about doing a Ph.D. at 16 years old. Since then, I have directed all my energy and thoughts to pursue this dream. However, I could not have done it alone. Throughout all these years, there were people whose contributions have been invaluable in reaching this goal. To the people who taught me, fed me, cared for me, and put a roof over my head, I want to thank you!

Mi corazón rebosa de gratitud por mi amada familia, especialmente por mis queridos padres, cuya devoción inquebrantable, cariño y aliento constante han sido mi fortaleza en las buenas y en las malas. A mi madre, **Janynce Urzola**, quien es una verdadera heroína. Tu fuerza inquebrantable, valentía y amor han sido el pegamento que mantiene unida a nuestra familia. A mi padre, **Isidoro Guerra**, quien me inculcó el valor de la educación, que ha transformado mi vida más allá de lo que algún día pude imaginar. No tengo palabras para expresar cuánto los valoro y aprecio por estar siempre a mi lado, por nunca perder la fe en mí, y por darme esperanza y alegría cada día. Soy bendecido por tenerlos en mi vida y los amo con todo mi corazón.

A mis hermanos, **Mario Guerra** y **Francheska Guerra**, les expreso toda mi gratitud por el apoyo y la admiración que siempre me han mostrado. Perseguir este sueño ha tenido uno de los costos más altos de mi vida, al estar separado de ambos. No ha habido un solo día desde que dejé nuestro hogar en el que no los haya extrañado. Su amor y aliento han iluminado los momentos más difíciles de mi vida. Espero que este trabajo los inspire a seguir sus sueños con el mismo entusiasmo y determinación con los que yo lo hice.

A mis abuelos, gracias por enseñarme la felicidad en las cosas simples de la vida. Sin su apoyo durante toda mi vida, no sería la persona que soy hoy. A mi abuelo **Víctor Urzola**, me siento bendecido por haber compartido parte de mi vida contigo. Gracias por tus chistes, el delicioso café y el amor incondicional. A mi abuela **Martha Barrera**, gracias por ser una segunda madre. Desde que nací, me has dado todo lo que he necesitado, querido y más.

To **Andrés Ramírez Hassan**, you saw potential in me and gave me the opportunity to be part of your research team in one of the most challenging moments of my life. It changed my life for good. I learned from your hard work, dedication, and commitment to excel. I will always be grateful for that. To **Paula Almonacid**, you were part of that change, too. Thank you for your encouragement, trust, and advice. To **Manuel Salvador**, whose friendship, sense of humor, and always happy mood helped me endure the challenging moments during my master's.

To my Ph.D. supervisors, I would like to express my sincere gratitude for your guidance, patience, and mentorship throughout this journey. **Juan Vera**, I already miss our weekly meetings where we explored entertaining mathematical proofs and had some beers together, of course. I truly enjoyed the challenges and appreciate your support. **Katrijn Van Deun**, thank you for being open to my ideas and always guiding me. Your support and understanding during the entire process, especially when I was unwell, meant a lot to me. Lastly, **Klaas Sijtsma**, your pieces of advice were always sharp and helped me with publications and the timely completion of my Ph.D. I couldn't have asked for a better supervising team.

During my Ph.D., I have made friends and colleagues who have significantly influenced my well-being and happiness all these years.

All my gratitude to my Tex-Mex carnal, **Aarón Villareal**, whose camaraderie and brotherhood have brought unwavering support, encouragement, endless laughter, and cherished memories into my life. ¡gracias wey! I want to show my gratitude to **Soogeun Park**, whose friendship and brotherhood have made a living abroad feel like home. No words can express what your friendship means to me, ¡gracias rey! To **Fábio Generoso**, I hope we can continue our yearly trip tradition. Perhaps one day, we will return to the USA to revive enigmas such as "What Happens to Mohamed" or sit on the unforgettable line J singing at 3 a.m. Obrigado doutor! To **Lina Palomino**, thank you for your support and friendship throughout the years. Those Havana nights always provided a rejuvenating break from my Ph.D. To **Shuai Yuan**, with whom I shared office for three years, thank you for welcoming me to the Chinese community. I enjoyed our dinner on the 6th floor from the "illegal Chinese kitchen," as I liked to call it. To **Esther Massen**, thank you for being so caring and understanding. I admire your discretion and highly value your opinion, Dankjewel. To **Edoardo Constantini** and **Damiano D'Urso**, thank you for teaching me how to flirt in Italian. It always came in handy. Tue tranquille, faccio io. Grazie mille.

I'm grateful to all of you for your pivotal roles in my Ph.D. journey. Your support, encouragement, and kindness have greatly impacted me, and I want to express my heartfelt thank you to each of you. While I may not have been able to mention everyone individually in my acknowledgments, please know that your contributions are deeply appreciated and have been instrumental in my success. Thank you for being a part of this incredible journey with me.

Tilburg, September 1, 2023 *Rosember Isidoro Guerra Urzola*

References

Adachi, K., & Kiers, H. A. (2017). Sparse Regression Without Using a Penalty Function.. Retrieved from http://www.jfssa.jp/taikai/2017/table/program_detail/pdf/1-50/10009.pdf

Adachi, K., & Trendafilov, N. T. (2016, dec). Sparse principal component analysis subject to prespecified cardinality of loadings. *Computational Statistics*, *31*(4), 1403–1427. Retrieved from http://link.springer.com/10.1007/s00180-015-0608-4 doi: 10.1007/s00180-015-0608-4

Alexander, C. E. (2008). Market risk analysis. Volume II, Practical financial econometrics.

Baik, J., & Silverstein, J. W. (2006, jul). Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, *97*(6), 1382–1408. doi: 10.1016/j.jmva.2005.08.003

Beck, A. (2014). *Introduction to Nonlinear Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics. Retrieved from https://doi.org/10.1137/1.9781611973655 doi: 10.1137/1.9781611973655

Beck, A., & Teboulle, M. (2009, jan). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, *2*(1), 183–202. Retrieved from http://epubs.siam.org/doi/10.1137/080716542 doi: 10.1137/080716542

Beck, A., & Vaisbourd, Y. (2016). The Sparse Principal Component Analysis Problem: Optimality Conditions and Algorithms. *Journal of Optimization Theory and Applications*, *170*(1), 119–143. doi: 10.1007/s10957-016-0934-x

Berk, L., & Bertsimas, D. (2019, sep). Certifiably optimal sparse principal component analysis. *Mathematical Programming Computation*, *11*(3), 381–420. Retrieved from https://link.springer.com/article/10.1007/s12532-018-0153-6 doi: 10.1007/s12532-018-0153-6

Bertsimas, D., Cory-Wright, R., & Pauphilet, J. (2022). Solving Large-Scale Sparse PCA to Certifiable (Near) Optimality. *Journal of Machine Learning Research*, *23*.

Bertsimas, D., King, A., & Mazumder, R. (2016, apr). Best subset selection via a modern optimization lens. *Annals of Statistics*, *44*(2), 813–852. doi: 10.1214/15-AOS1388

Bertsimas, D., & Van Parys, B. (2020, feb). Sparse high-dimensional regression: Exact scalable algorithms and phase transitions. *The Annals of Statistics*, *48*(1), 300–323. doi: 10.1214/18-AOS1804

Bixby, R. E. (2012). A Brief History of Linear and Mixed-Integer Programming

Computation. *Documenta Mathematica · Extra*, *ISMP ISMP*, 107–121.

Brenner, N., Bialek, W., & De Ruyter Van Steveninck, R. (2000, jun). Adaptive rescaling maximizes information transmission. *Neuron*, *26*(3), 695–702. doi: 10.1016/S0896-6273(00)81205-2

Cadima, J., & Jolliffe, I. T. (1995, jan). Loadings and correlations in the interpretation of principal components. *Journal of Applied Statistics*, *22*(2), 203–214. doi: 10.1080/757584614

Camacho, J., Smilde, A. K., Saccenti, E., & Westerhuis, J. A. (2020, jan). All sparse PCA models are wrong, but some are useful. Part I: Computation of scores, residuals and explained variance. *Chemometrics and Intelligent Laboratory Systems*, *196*, 103907. doi: 10.1016/j.chemolab.2019.103907

Camacho, J., Smilde, A. K., Saccenti, E., Westerhuis, J. A., & Bro, R. (2021, jan). All sparse PCA models are wrong, but some are useful. Part II: Limitations and problems of deflation. *Chemometrics and Intelligent Laboratory Systems*, *208*, 104212. doi: 10.1016/j.chemolab.2020.104212

d'Aspremont, A., Bach, F., & Ghaoui, L. E. (2007, jul). Optimal Solutions for Sparse Principal Component Analysis. *Journal of Machine Learnig Research*, *99*, 1269–1294. Retrieved from `https://www.di.ens.fr/~fbach/sparse_PCA_Aspremont_Bach_ElGhaoui_JMLR.pdf`

d'Aspremont, A., El Ghaoui, L., Jordan, M. I., & Lanckriet, G. R. (2004). A Direct Formulation for Sparse PCA Using Semidefinite Programming. *SSRN Electronic Journal*, *49*(3), 434–448. Retrieved from `http://www.ssrn.com/abstract=563524` doi: 10.2139/ssrn.563524

Dolan, C. V., Oort, F. J., Stoel, R. D., & Wicherts, J. M. (2009, apr). Testing measurement invariance in the target rotated multigroup exploratory factor model. *Structural Equation Modeling*, *16*(2), 295–314. doi: 10.1080/10705510902751416

Eckart, C., & Young, G. (1936, sep). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211–218. Retrieved from `http://link.springer.com/10.1007/BF02288367` doi: 10.1007/BF02288367

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

Epskamp, S., Cramer, A. O., Waldorp, L. J., Schmittmann, V. D., & Borsboom, D. (2012). Qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software*, *48*(4), 69. doi: 10.18637/jss.v048.i04

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. Retrieved from `https://www.jstor.org/stable/3085904` doi: 10.1198/016214501753382273

Friedman, J., Hastie, T., Höfling, H., & Tibshirani, R. (2007, dec). Pathwise coordinate optimization. *The Annals of Applied Statistics*, *1*(2), 302–332. doi: 10.1214/07-aoas131

Gu, Z., de Schipper, N. C., & Van Deun, K. (2019, dec). Variable Selection in the Regularized Simultaneous Component Analysis Method for Multi-Source Data Integration. *Scientific Reports*, *9*(1), 18608. Retrieved from `http://www.nature.com/articles/s41598-019-54673-2` doi: 10.1038/s41598-019-54673-2

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, oct). RegularizedSCA: Regularized simultaneous component analysis of multiblock data in R. *Behavior Research Methods*, *51*(5), 2268–2289. doi: 10.3758/s13428-018-1163-z

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/s11634-022-00499-2` doi: 10.1007/s11634-022-00499-2

Guerra-Urzola, R., Van Deun, K., Vera, J. C., & Sijtsma, K. (2021, dec). A Guide for Sparse PCA: Model Comparison and Applications. *Psychometrika*, *86*(4), 893–919. Retrieved from `https://link.springer.com/article/10.1007/s11336-021-09773-2` doi: 10.1007/s11336-021-09773-2

Hastie, T., Tibshirani, R., Eisen, M. B., Alizadeh, A., Levy, R., Staudt, L., ... Brown, P. (2000, aug). 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome biology*, *1*(2), 1–21. Retrieved from `https://genomebiology.biomedcentral.com/articles/10.1186/gb-2000-1-2-research0003` doi: 10.1186/GB-2000-1-2-RESEARCH0003

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Springer Series in Statistics* (Vol. 27) (No. 2). Retrieved from `https://doi.org/10.1007/978-0-387-84858-7` doi: 10.1007/b94608

Hastie, T., Tibshirani, R., & Tibshirani, R. J. (2017, jul). Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso. Retrieved from `http://arxiv.org/abs/1707.08692`

Hotelling, H. (1933, sep). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417–441. doi: 10.1037/h0071325

Hsu, Y. L., Huang, P. Y., & Chen, D. T. (2014, jun). *Sparse principal component analysis in cancer research* (Vol. 3) (No. 3). NIH Public Access. Retrieved from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4692276/` doi: 10.3978/j.issn.2218-676X.2014.05.06

Huang, K., Sidiropoulos, N. D., & Liavas, A. P. (2016, oct). A Flexible and Efficient

Algorithmic Framework for Constrained Matrix and Tensor Factorization. In *Ieee transactions on signal processing* (Vol. 64, pp. 5052–5065). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/TSP.2016.2576427

Hunter, D. R., & Lange, K. (2004). A Tutorial on MM Algorithms. *American Statistician*, *58*(1), 30–37. doi: 10.1198/0003130042836

Jennrich, R. I. (2004). Rotation to simple loadings using component loss functions: The orthogonal case. In *Psychometrika* (Vol. 69, pp. 257–273). Psychometric Society. doi: 10.1007/BF02295943

Jennrich, R. I. (2006, mar). Rotation to simple loadings using component loss functions: The oblique case. *Psychometrika*, *71*(1), 173–191. doi: 10.1007/s11336-003-1136-B

Johnstone, I. M., & Lu, A. Y. (2009, jun). On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, *104*(486), 682–693. doi: 10.1198/jasa.2009.0121

Jolliffe, I. (2002). *Principal Component Analysis*. New York: Springer-Verlag. Retrieved from `http://link.springer.com/10.1007/b98835` doi: 10.1007/b98835

Jolliffe, I. T. (1986). *Principal Component Analysis*. New York, NY: Springer New York. Retrieved from `http://link.springer.com/10.1007/978-1-4757-1904-8` doi: 10.1007/978-1-4757-1904-8

Jolliffe, I. T. (1989). Rotation of Ill-Defined Principal Components. *Applied Statistics*, *38*(1), 139. Retrieved from `https://www.jstor.org/stable/10.2307/2347688?origin=crossref` doi: 10.2307/2347688

Jolliffe, I. T. (1995, jan). Rotation of principal components: Choice of normalization constraints. *Journal of Applied Statistics*, *22*(1), 29–35. doi: 10.1080/757584395

Jolliffe, I. T., & Cadima, J. (2016, apr). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202. Retrieved from `https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202` doi: 10.1098/rsta.2015.0202

Jolliffe, I. T., Trendafilov, N. T., & Uddin, M. (2003, sep). A Modified Principal Component Technique Based on the LASSO. *Journal of Computational and Graphical Statistics*, *12*(3), 531–547. doi: 10.1198/1061860032148

Journée, M., Nesterov, Y., Richtárik, P., & Sepulchre, R. (2010). *Generalized Power Method for Sparse Principal Component Analysis Rodolphe Sepulchre* (Vol. 11; Tech. Rep.).

Jung, S., & Marron, J. S. (2009, dec). PCA consistency in High Dimension, Low Sample Size context. *Annals of Statistics*, *37*(6 B), 4104–4130. doi: 10.1214/09-AOS709

Kaiser, H. F. (1958, sep). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, *23*(3), 187–200. doi: 10.1007/BF02289233

Kiers, H. A. (1994, dec). Simplimax: Oblique rotation to an optimal target with simple structure. *Psychometrika*, *59*(4), 567–579. doi: 10.1007/BF02294392

Kiers, H. A. (2002, nov). Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems. *Computational Statistics and Data Analysis*, *41*(1), 157–170. doi: 10.1016/S0167-9473(02)00142-1

Lange, K. (2016). *MM Optimization Algorithms*. Society for Industrial and Applied Mathematics. doi: 10.1137/1.9781611974409

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

Leng, C., & Wang, H. (2009). On general adaptive sparse principal component analysis. *Journal of Computational and Graphical Statistics*, *18*(1), 201–215. Retrieved from `https://www.jstor.org/stable/25703561` doi: 10.1198/jcgs.2009.0012

Li, Y., & Xie, W. (2020, aug). Exact and Approximation Algorithms for Sparse PCA. Retrieved from `https://arxiv.org/abs/2008.12438v1` doi: 10.48550/arxiv.2008.12438

MacKey, L. (2009). Deflation methods for sparse PCA. In *Advances in neural information processing systems 21 - proceedings of the 2008 conference* (pp. 1017–1024).

McCrae, R. R., Costa, P. T., & Martin, T. A. (2005). The NEO-PI-3: A more readable Revised NEO Personality Inventory. *Journal of Personality Assessment*, *84*(3), 261–270. doi: 10.1207/s15327752jpa8403_05

McCrae, R. R., & John, O. P. (1992, jun). An Introduction to the Five-Factor Model and Its Applications. *Journal of Personality*, *60*(2), 175–215. Retrieved from `https://onlinelibrary.wiley.com/doi/10.1111/j.1467-6494.1992.tb00970.x` doi: 10.1111/j.1467-6494.1992.tb00970.x

Moghaddam, B., Weiss, Y., & Avidan, S. (2005). Spectral bounds for sparse PCA: Exact and greedy algorithms. In *Advances in neural information processing systems* (pp. 915–922).

Mohammed, S., Khalid, A., Osman, S., & Helali, R. G. (2016, dec). Usage of Principal Component Analysis (PCA) in AI Applications. *International Journal of Engineering Research and Technology (IJERT)*, *5*(12), 372–375. doi: 10.17577/IJERTV5IS120291

Nadler, B. (2008, dec). Finite sample approximation results for principal component analysis: A matrix perturbation approach. *Annals of Statistics*, *36*(6), 2791–2817. doi: 10.1214/08-AOS618

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

Nguyen, H. D. (2017, mar). An introduction to Majorization-Minimization algorithms for machine learning and statistical estimation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(2), e1198. Retrieved from

https://onlinelibrary.wiley.com/doi/full/10.1002/widm.1198 doi: 10.1002/widm.1198

Ning-min, S., & Jing, L. (2015, dec). A Literature Survey on High-Dimensional Sparse Principal Component Analysis. *International Journal of Database Theory and Application*, *8*(6), 57–74. doi: 10.14257/ijdta.2015.8.6.06

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

Papailiopoulos, D. S., Dimakis, A. G., & Korokythakis, S. (2013, may). Sparse PCA through low-rank approximations. In *30th international conference on machine learning, icml 2013* (pp. 1784–1792). PMLR. Retrieved from https://proceedings.mlr.press/v28/papailiopoulos13.html

Pasini, G. (2017). Principal component analysis for stock portfolio management. *International Journal of Pure and Apllied Mathematics*, *115*(1), 153–167. doi: 10.12732/ijpam.v115i1.12

Paul, D. (2007). Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, *17*(4), 1617–1642.

R: Elastic-Net for Sparse Estimation and Sparse PCA. (n.d.). Retrieved from https://search.r-project.org/CRAN/refmans/elasticnet/html/00Index.html

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., & Lê Cao, K. A. (2017, nov). mixOmics: An R package for 'omics feature selection and multiple data integration. *PLoS Computational Biology*, *13*(11), e1005752. Retrieved from https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005752 doi: 10.1371/journal.pcbi.1005752

R: The R Project for Statistical Computing. (n.d.). Retrieved from https://www.r-project.org/

S., G. W., Golub, G. H., & Loan, C. F. V. (1991). Matrix Computations. *Mathematics of Computation*, *56*(193), 380. doi: 10.2307/2008552

Shen, D., Shen, H., & Marron, J. S. (2016). A general framework for consistency of principal component analysis. *Journal of Machine Learning Research*, *17*, 1–34.

Shen, D., Shen, H., Zhu, H., & Marron, J. S. (2016, oct). The statistics and mathematics of high dimension low sample size asymptotics. In *Statistica sinica* (Vol. 26, pp. 1747–1770). Institute of Statistical Science. doi: 10.5705/ss.202015.0088

Shen, H., & Huang, J. Z. (2008, jul). Sparse principal component analysis via regularized low rank matrix approximation. , *99*(6), 1015–1034. doi: 10.1016/

j.jmva.2007.06.007

Sriperumbudur, B. K., Torres, D. A., & Lanckriet, G. R. (2011). A majorization-minimization approach to the sparse generalized eigenvalue problem. *Machine Learning*, *85*(1-2), 3–39. doi: 10.1007/s10994-010-5226-3

ten Berge, J. M. (1986). Some Relationships Between Descriptive Comparisons of Components from Different Studies. *Multivariate Behavioral Research*, *21*(1), 29–40. doi: 10.1207/s15327906mbr2101_2

ten Berge, J. M. (2005). Least Squares Optimization in Multivariate Analysis. *Leiden University*, *34*, 96.

Tibshirani, R. (1996, jan). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. Retrieved from https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1996.tb02080.x doi: 10.1111/j.2517-6161.1996.tb02080.x

Tibshirani, R. (2011, jun). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(3), 273–282. Retrieved from https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2011.00771.x doi: 10.1111/j.1467-9868.2011.00771.x

Trendafilov, N. T. (2014). From simple structure to sparse components: A review. *Computational Statistics*, *29*(3–4), 431–454. doi: 10.1007/s00180-013-0434-5

Trendafilov, N. T., & Adachi, K. (2015, sep). Sparse Versus Simple Structure Loadings. *Psychometrika*, *80*(3), 776–790. doi: 10.1007/s11336-014-9416-y

Tseng, P. (2001, jun). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, *109*(3), 475–494. doi: 10.1023/A:1017501703105

Van Deun, K., Smilde, A. K., van der Werf, M. J., Kiers, H. A., & Van Mechelen, I. (2009, dec). A structured overview of simultaneous component based data integration. *BMC Bioinformatics*, *10*(1), 246. Retrieved from https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-246 doi: 10.1186/1471-2105-10-246

Van Deun, K., Thorrez, L., Coccia, M., Hasdemir, D., Westerhuis, J. A., Smilde, A. K., & Van Mechelen, I. (2019, dec). Weighted sparse principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *195*. doi: 10.1016/j.chemolab.2019.103875

Wang, B., & Jiang, L. (2021, jan). Principal Component Analysis Applications in COVID-19 Genome Sequence Studies. *Cognitive Computation*, *1*, 1–12. Retrieved from https://link.springer.com/article/10.1007/s12559-020-09790-w doi: 10.1007/s12559-020-09790-w

Wang, W., & Stephens, M. (2021). Empirical bayes matrix factorization. *Journal of Machine Learning Research*, *22*(120), 1–40. Retrieved from http://jmlr.org/papers/v22/20-589.html

Whittle, P. (1952). On principal components and least square methods of factor analysis. *Scandinavian Actuarial Journal*, *1952*(3-4), 223–239. doi: 10.1080/03461238.1955.10430696

Wold, S., Esbensen, K., & Geladi, P. (1987, aug). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *2*(1-3), 37–52. doi: 10.1016/0169-7439(87)80084-9

Yan, D., Wu, T., Liu, Y., & Gao, Y. (2018, may). An efficient sparse-dense matrix multiplication on a multicore system. In *International conference on communication technology proceedings, icct* (Vol. 2017-Octob, pp. 1880–1883). Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICCT.2017.8359956

Yang, D., Ma, Z., & Buja, A. (2014, oct). A Sparse Singular Value Decomposition Method for High-Dimensional Data. *Journal of Computational and Graphical Statistics*, *23*(4), 923–942. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/10618600.2013.858632 doi: 10.1080/10618600.2013.858632

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, S., De Roover, K., Dufner, M., Denissen, J. J., & Van Deun, K. (2021, oct). Revealing Subgroups That Differ in Common and Distinctive Variation in Multi-Block Data: Clusterwise Sparse Simultaneous Component Analysis. *Social Science Computer Review*, *39*(5), 802–820. doi: 10.1177/0894439319888449

Yuan, X. T., & Zhang, T. (2013). Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research*, *14*(1), 899–925.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, *101*(476), 1418–1429. doi: 10.1198/016214506000000735

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. doi: 10.1111/j.1467-9868.2005.00503.x

Zou, H., Hastie, T., & Tibshirani, R. (2006, jun). Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, *15*(2), 265–286. Retrieved from http://www.tandfonline.com/doi/abs/10.1198/106186006X113430 doi: 10.1198/106186006X113430

Zou, H., & Xue, L. (2018, aug). A Selective Overview of Sparse Principal Component Analysis. *Proceedings of the IEEE*, *106*(8), 1311–1320. doi: 10.1109/JPROC.2018.2846588