



All sparse PCA models are wrong, but some are useful. Part I: Computation of scores, residuals and explained variance

J. Camacho^{a,*}, A.K. Smilde^b, E. Saccenti^c, J.A. Westerhuis^b

^a Department of Signal Theory, Telematics and Communications, School of Computer Science and Telecommunications - CITIC, University of Granada, Granada, Spain

^b Biosystems Data Analysis, University of Amsterdam, Amsterdam, the Netherlands

^c Laboratory of Systems and Synthetic Biology, Wageningen University & Research, Wageningen, the Netherlands

ARTICLE INFO

Keywords:

Sparse principal component analysis
Explained variance
Scores
Residuals
Exploratory data analysis

ABSTRACT

Sparse Principal Component Analysis (sPCA) is a popular matrix factorization approach based on Principal Component Analysis (PCA) that combines variance maximization and sparsity with the ultimate goal of improving data interpretation. When moving from PCA to sPCA, there are a number of implications that the practitioner needs to be aware of. A relevant one is that scores and loadings in sPCA may not be orthogonal. For this reason, the traditional way of computing scores, residuals and variance explained that is used in the classical PCA can lead to unexpected properties and therefore incorrect interpretations in sPCA. This also affects how sPCA components should be visualized. In this paper we illustrate this problem both theoretically and numerically using simulations for several state-of-the-art sPCA algorithms, and provide proper computation of the different elements mentioned. We show that sPCA approaches present disparate and limited performance when modeling noise-free, sparse data. In a follow-up paper, we discuss the theoretical properties that lead to this undesired behavior. We title this series of papers after the famous phrase of George Box “All models are wrong, but some are useful” with the same original meaning: sPCA models are only approximations of reality and have structural limitations that should be taken into account by the practitioner, but properly applied they can be useful tools to understand data.

1. Introduction

In modern data analysis there is an increased interest in sparse (component) methods to obtain models that are simpler and easier to interpret. One of such methods is the popular sparse Principal Component Analysis (sPCA) [1–4]. While the properties of sPCA have already been discussed [3,5,6], there are several aspects that have not been treated in detail and that are relevant for the interpretation of the models.

sPCA is based on Principal Component Analysis (PCA) [7,8], which factorizes a data matrix using the criterion of maximizing variance. The PCA model follows the expression:

$$\mathbf{X} = \hat{\mathbf{T}}\hat{\mathbf{P}}^T + \mathbf{E}, \quad (1)$$

where \mathbf{X} is a $N \times M$ data matrix, $\hat{\mathbf{T}}$ the $N \times A$ score matrix, $\hat{\mathbf{P}}$ the $M \times A$ loading matrix and \mathbf{E} the $N \times M$ matrix of residuals. An interesting property of PCA is that the loadings and the scores are orthogonal and, as a result, they can be computed either using a simultaneous or sequential algorithm.

When moving from PCA to sPCA, there are at least two interrelated consequences that need to be considered: *i*) the appearance of correlation among scores and/or loadings in different components and *ii*) the departure of components from the row-space defined by the original data. The first point makes the traditional way of computing scores, residuals and captured variance used in PCA not applicable in sPCA. Moreover, the captured variance is often used as a criterion for model quality and for selection/comparison of model variants. Therefore, its accurate computation is necessary. The second point affects interpretability, specially for multi-component models. Both effects are especially relevant in those domains in which the interpretation of the scores/residuals is necessary, like in chemometrics.

Zou et al. [2] proposed a method based on the QR decomposition to compute the variance captured by sPCA when scores are correlated; we will show that this method is generally incorrect. Witten et al. [3] related the correlation of the loadings/scores to the departure from the row/column-space of the data and proposed an orthogonalized version of their sPCA method, where scores are made orthogonal; we will show that this orthogonalization reduces the captured variance and dwarfs

* Corresponding author.

E-mail address: josecamacho@ugr.es (J. Camacho).

<https://doi.org/10.1016/j.chemolab.2019.103907>

Received 3 June 2019; Received in revised form 23 November 2019; Accepted 8 December 2019

Available online 19 December 2019

0169-7439/© 2019 Elsevier B.V. All rights reserved.

interpretation, but improves the estimation of sparse loadings. Mackey [5] discussed different deflation methods in the context of sPCA, some of which compute residuals incorrectly, leading to the double-counting of variance, *i.e.*, a portion of variance in the data captured by more than one component. He also contributed a new deflation method which, we will show, still requires the corrections that will be proposed and discussed in this paper.

In this study we present a comparison among several widely used sPCA algorithms and reflect on how scores, residuals and captured variance should be computed for sparse components, something central in chemometrics. Moreover, we show that sPCA approaches present disparate and limited performance when modeling noise-free, sparse data. The theoretical properties that lead to this undesired behavior, along with the departure of components from the row-space is treated in a follow-up paper. The rest of the paper is organized as follows. In Section 2, we review a set of sPCA algorithms considered. In Section 3, we discuss the mathematical properties of sPCA, using PCA as a baseline. In Section 4, we propose two simulated examples that mimic spectral data, and a Monte Carlo experiment, that will illustrate interpretation problems and solutions in the following sections. Section 5 discusses the implications of correlated scores and loadings in sPCA. Section 6 presents the correct computation of scores, residuals and captured variance in sPCA models. Section 7 offers some final considerations and conclusions.

2. Sparse PCA algorithms

Most sPCA algorithms modify the classical PCA by including sparse-inducing constraints or penalties with the L_0 or L_1 norms [9]. Jolliffe et al. [1], in 2003, developed the SCoTLASS algorithm (Simplified Component Technique-LASSO), which incorporates the least absolute shrinkage and selection operator (lasso) [10], with the L_1 norm (absolute values) of the loadings, as a constrain. The SCoTLASS criterion follows:

$$\hat{\mathbf{p}}^{SL} = \arg \max_{\mathbf{p}} \|\mathbf{X}\mathbf{p}\|_F^2 \quad s.t. \quad \|\mathbf{p}\|_1 \leq c, \quad \|\mathbf{p}\|_2^2 = 1 \quad (2)$$

where $\hat{\mathbf{p}}^{SL}$ is the resulting sparse loading, with superscript *SL* referring to SCoTLASS. To obtain successive components, the SCoTLASS optimization constraints the 2nd and further sparse loadings to be orthogonal to the rest. The SCoTLASS criterion is very computational demanding [6], and this makes its use unpractical for data exploration. In addition, the number of nonzero elements in $\hat{\mathbf{p}}^{SL}$ is upper-bounded by the number of observations in the data, which is a structural limitation of the lasso constraint.

2.1. Sparse PCA by Zou, Hastie and Tibshirani, 2006

Zou et al. [2] introduced an alternative formulation to generate sparse components, motivated by the structural limitation of the lasso and the computation cost of SCoTLASS. This new model was also named Sparse PCA (in the following we will use SPCA-Z when referring to this approach, with capital 'S' and 'Z' to differentiate it from the general sPCA technique), although at that time the term was already used to designate the group of sparse variants of PCA. SPCA-Z redefines PCA as a regularized regression problem with the ridge penalty, where the responses are the PCA scores. In this formulation any loading vector of PCA, $\hat{\mathbf{p}}$, equals $\frac{\hat{\mathbf{p}}^R}{\|\hat{\mathbf{p}}^R\|}$, where

$$\hat{\mathbf{p}}^R = \arg \min_{\mathbf{p}} \|\hat{\mathbf{t}} - \mathbf{X}\mathbf{p}\|_F^2 + \lambda_2 \|\mathbf{p}\|_2^2 \quad (3)$$

with superscript *R* referring to *regression*, and $\hat{\mathbf{t}}$ is the corresponding score vector. Therefore, $\hat{\mathbf{p}}^R \propto \hat{\mathbf{p}}$, and $\hat{\mathbf{p}}$ is obtained by normalizing $\hat{\mathbf{p}}^R$ to unit length. This equivalence holds in general for any non-negative λ_2 . Equivalently, the loadings of PCA can be obtained from the following

regression problem followed by a normalization step:

$$\{\hat{\mathbf{P}}^R, \hat{\mathbf{Q}}^R\} = \arg \min_{\mathbf{P}, \mathbf{Q}} \|\mathbf{X} - \mathbf{X}\mathbf{P}\mathbf{Q}^T\|_F^2 + \lambda_2 \sum_{a=1}^A \|\mathbf{p}_a\|_2^2 \quad s.t. \quad \mathbf{Q}^T\mathbf{Q} = \mathbf{I} \quad (4)$$

where \mathbf{p}_a represents the *a*th column vector in \mathbf{P} , with *A* the number of components, and we distinguish between weights $\hat{\mathbf{P}}^R$ and loadings $\hat{\mathbf{Q}}^R$. In similar modeling methods, like Partial Least Squares, weights are used for the derivation of scores from the data and loadings for the identification of the captured variance in \mathbf{X} , *e.g.* for deflation. The advantage of Eq. (4) over Eq. (3) is that the former does not require the pre-computation of the PCA scores. $\hat{\mathbf{P}}^R$ is proportional to the PCA loadings for any non-negative λ_2 . Therefore, we can understand Eq. (4) as a flexible method to compute PCA, where constraints and penalties can be introduced either in the weights or the loadings.

Using this flexible redefinition of PCA the lasso can be implemented as a penalty using a criterion close to the (naive) elastic net [11], which is a combination of both the lasso and the ridge constraint:

$$\{\hat{\mathbf{P}}^{SP}, \hat{\mathbf{Q}}^{SP}\} = \arg \min_{\mathbf{P}, \mathbf{Q}} \|\mathbf{X} - \mathbf{X}\mathbf{P}\mathbf{Q}^T\|_F^2 + \lambda_2 \sum_{a=1}^A \|\mathbf{p}_a\|_2^2 + \lambda_1 \sum_{a=1}^A \|\mathbf{p}_a\|_1 \quad s.t. \quad \mathbf{Q}^T\mathbf{Q} = \mathbf{I} \quad (5)$$

with superscript *SP* referring to SPCA-Z. In the following, we will refer to $\hat{\mathbf{P}}^{SP}$ as sparse loadings and to $\hat{\mathbf{Q}}^{SP}$ as auxiliary loadings to simplify the discussion in connection with other sPCA techniques. The SPCA-Z algorithm is simultaneous, in the sense that all components are identified in the same run. Note the expression in Eq. (5) is not optimal to approximate data, because \mathbf{Q} is constrained to be orthonormal, while at the same time the norms of the vectors in \mathbf{P} are penalized, leaving no flexibility to approximate \mathbf{X} and resulting in very large residuals. Besides, the sparse loadings are not normalized. The authors suggest to normalize them after the optimization, in the last step of their algorithm, but they overlook the need to update $\hat{\mathbf{Q}}$ accordingly. This reflects the fact that authors were only interested in $\hat{\mathbf{P}}^{SP}$ as the output of the algorithm, and the model in Eq. (5) is only a means to arrive at such an output. They also realized that this SPCA-Z algorithm produces highly correlated scores, and concluded that the current method for computing the captured variance as $\text{tr}(\hat{\mathbf{T}}^T\hat{\mathbf{T}})$ used in PCA is too optimistic. For this reason, they suggested to use the QR decomposition of $\hat{\mathbf{T}}$, which is $\hat{\mathbf{T}} = \mathbf{Q}_T\mathbf{R}_T$ for \mathbf{Q}_T orthogonal and \mathbf{R}_T upper triangular, and to compute the captured variance of the model as the sum of squares of the elements in the diagonal of \mathbf{R}_T .

An alternative SPCA algorithm has been published in 2018 (in arXiv since 2012) as part of the SPASM Toolbox [12], which is a sequential variant of SPCA-Z, in which the loadings are computed one at a time. This algorithm is faster than the simultaneous versions, with a small reduction of captured variance.

2.2. The Penalized Matrix Decomposition by Witten, Tibshirani and Hastie, 2009

Witten et al. [3] introduced a new iterative algorithm based on alternating of cardinality-constrained rank-one variance maximization and matrix deflation. For the first task, they proposed the Penalized Matrix Decomposition (PMD), a biconvex problem solved by alternating between the identification of \mathbf{u} (left pseudo-eigenvector) and \mathbf{p} (right pseudo-eigenvector). PMD equals the 1-rank Singular Value Decomposition (SVD) when no constraints are considered. The sPCA formulation in the PMD framework is:

$$\{\hat{\mathbf{p}}_A^P, \hat{\mathbf{u}}_A^P\} = \arg \max_{\mathbf{p}_A, \mathbf{u}_A} \mathbf{u}_A^T \mathbf{X}_{A-1} \mathbf{p}_A \quad s.t. \quad \|\mathbf{p}_A\|_1 \leq c_2, \quad \|\mathbf{p}_A\|_2^2 \leq 1, \quad \|\mathbf{u}_A\|_2^2 \leq 1 \quad (6)$$

where \mathbf{X}_{A-1} denote the residuals after $A - 1$ components have been extracted and $\hat{\mathbf{p}}_A^P$ is obtained using a soft-thresholding operator and $\hat{\mathbf{u}}_A^P$ by a least squares operation, with superscript P referring to PMD. The corresponding pseudo-singular value is obtained as:

$$\hat{d}_A^P = (\hat{\mathbf{u}}_A^P)^T \mathbf{X}_{A-1} \hat{\mathbf{p}}_A^P \quad (7)$$

After each component is obtained, deflation is performed as:

$$\mathbf{X}_A = \mathbf{X}_{A-1} \left(\mathbf{I} - \hat{d}_A^P \hat{\mathbf{u}}_A^P (\hat{\mathbf{p}}_A^P)^T \right) \quad (8)$$

This solution is related to SCoTLASS and sPCA [3], but according to Ref. [6] this method is more efficient, something we have also seen experimentally. Regarding the deflation, they state that while projection deflation works well for true eigenvectors, when using constrained versions of PCA (i.e. sPCA) \mathbf{u} and/or \mathbf{p} are not in the column and row spaces of \mathbf{X} . As an alternative to projection deflation, they propose to compute \mathbf{u} using an orthogonalized factor to all the previous left pseudo-eigenvectors. By incorporating this deflation, orthogonal scores are obtained. However, the authors themselves claim that it is not clear whether orthogonality is a desirable property [3,6]. We will call this the orthogonalized deflation approach.

2.3. Group-wise Principal Component Analysis, 2017

The Group-wise Principal Component Analysis (GPCA) [4] is a sparse approach to PCA proposed by some of the authors. The main difference between GPCA and the sPCA variants presented in the previous sections is that sparsity is not achieved by using a loss function with the lasso constraint but it is defined in terms of groups of correlated variables and it is imposed using a set of nested PCA loops. Every component contains non-zero loadings for a single group of correlated variables, and this component is calibrated using only this subset of variables. GPCA starts with the identification of a set of G (possibly overlapping) groups of correlated variables obtained from a map \mathbf{M} , with elements $m_{ij} \in [-1, 1]$ containing the strength of the relationship between variables i and j . An example of this map is the correlation matrix of \mathbf{X} . In the original formulation of GPCA, the MEDA approach (Missing-data for Exploratory Data analysis) [13] was implemented to define \mathbf{M} , which uses a missing data strategy to estimate the correlation between any two variables, an approach which has been found to be effective in filtering out noise when estimating correlations [13].

Once the G groups have been defined, the GPCA algorithm computes G candidate loading vectors, where the g -th vector has non zero elements associated to the variables in the g -th group. From these, only the loading which explains the largest amount of variability in the data is retained and it is used to deflate data matrix \mathbf{X} . The algorithm iterates until a set of A pseudo-components is extracted.

In GPCA, the deflation by Mackey [5] is used. He showed that the common approach of deflation in PCA is in general not valid for sPCA. This is named projection deflation, and follows:

$$\mathbf{X}_A = \mathbf{X}_{A-1} \left(\mathbf{I} - \hat{\mathbf{p}}_A (\hat{\mathbf{p}}_A^T \hat{\mathbf{p}}_A)^{-1} \hat{\mathbf{p}}_A^T \right) \quad (9)$$

It is easy to show that projection deflation is equivalent to Eq. (8) in PMD when sparsity is only applied in the columns of the data matrix. After normalizing $\hat{\mathbf{p}}_A$ to unit length, Eq. (9) simplifies to:

$$\mathbf{X}_A = \mathbf{X}_{A-1} (\mathbf{I} - \hat{\mathbf{p}}_A \hat{\mathbf{p}}_A^T) \quad (10)$$

Setting $\mathbf{X}\mathbf{X}_{A-1}$ to be the covariance matrix of a data sample \mathbf{X} after $A - 1$ PCs have been extracted, the projection deflation can be written as

$$\mathbf{X}\mathbf{X}_A = (\mathbf{I} - \hat{\mathbf{p}}_A \hat{\mathbf{p}}_A^T) \mathbf{X}\mathbf{X}_{A-1} (\mathbf{I} - \hat{\mathbf{p}}_A \hat{\mathbf{p}}_A^T) \quad (11)$$

For $\hat{\mathbf{p}}_A$ a pseudo-eigenvector of $\mathbf{X}\mathbf{X}_{A-1}$, Mackey [5] shows that projection deflation annihilates the corresponding pseudo-eigenvalue while

maintaining the others in the data. However, when deflating with respect to a series of non-orthogonal pseudo-eigenvectors, projection deflation may lead to re-introduce previously deflated components and to the “double counting” of variance, which can cause misleading results for interpretation. Mackey also shows that this problem can be solved by annihilating the variance captured by the pseudo-eigenvector that is orthogonal to the space spanned by the previous extracted pseudo-eigenvectors. For that, he introduces a combination of projection deflation and orthogonalized deflation, named orthogonalized projection deflation. This, similarly to Ref. [2], adds to the sparse loadings a set of auxiliary orthogonal right pseudo-eigenvectors to perform the deflation.

One way to do this deflation follows a sequential Gram-Schmidt decomposition:

$$\hat{\mathbf{q}}_A = \mathbf{B}_{A-1} \hat{\mathbf{p}}_A \quad (12)$$

$$\mathbf{X}_A = \mathbf{X}_{A-1} (\mathbf{I} - \hat{\mathbf{q}}_A \hat{\mathbf{q}}_A^T) \quad (13)$$

$$\mathbf{B}_A = \mathbf{B}_{A-1} (\mathbf{I} - \hat{\mathbf{q}}_A \hat{\mathbf{q}}_A^T) \quad (14)$$

where \mathbf{B}_0 is initialized to the identity matrix. We will call this approach the Mackey’s generalized deflation and it is the approach used in GPCA.

2.4. Recent developments in sPCA

There has been a wide range of additional developments on sPCA in the literature. A recent review can be found in Ref. [14]. This is however more concerned with machine learning applications of sPCA, rather than data exploration. For instance, a result that is discussed and which is hardly useful in the context of data exploration in highly multivariate data sets is the inconsistency of PCA [15]. A recurrent topic in the literature is the combination of sparsity and robustness [16–19]. An extension for Big Data processing can be found in Ref. [20]. Rasmussen and Bro [21] describe the use of sPCA in exploratory analysis. Bro et al. [22] describe an extension of the lasso penalization to both rows and columns, using a simultaneous approach. Van Deun et al. [23] explore the combination of sPCA with a weighting approach in different exploratory applications.

3. From principal components to sparse principal components

When moving from a classical principal component model to any of its sparse variants there are a number of implications that we think is pivotal to assess, especially in the context of data interpretation:

1. Correlation of scores

The scores resulting from a standard PCA are uncorrelated, while the scores of sPCA model are typically correlated. This correlation complicates proper visualization but also allows for a more flexible modeling. As already discussed, Witten et al. [3] provide a solution to correct for this correlation, but it is not clear whether it should be used. More importantly, care should be taken when computing the captured variance with correlated scores.

2. Correlation of loadings

Correlation of loadings is a relevant problem, in particular when scatter plots of scores are used for interpretation. The visualization in scatter plots assumes orthogonal axes in the original variable space. This holds for the standard PCA components, but it does not necessarily hold for sparse components. Again, care should be taken when computing the captured variance with correlated loadings.

3. Loss of captured variance per principal component.

The PCs are optimal in terms of captured variance. This is a property that makes them especially useful for data interpretation, since a large portion of variance/information is explained by a reduced set of components. This in turn simplifies the visualization of the data, since a minimum number of PCs needs to be visualized to account for a certain level of variance. However, part of the captured variance might not be of interest and could even complicate interpretation. As a relevant example, all PCs include a certain amount of embedded noise [24]. When we move to the sparse context, we simplify the interpretation of loadings, but also need more components to capture the same amount of variance.

4. Moving outside the row-space

Sparse loadings can be outside the data row-space as a result of constraints/penalties applied. As a matter of fact, there is no sparse solution inside the row-space in the presence of noise. If the constraints/penalties are reasonable, we can improve the model quality with this departure. A notable example of this are the non-negative constraints, useful to model non-negative data. While the departure of a sparse component can be advantageous, it also complicates the generation of multi-component models.

We will discuss these implications in detail and quantify them using simulation in the next Sections, except for the departure from the row-space, which will be treated in the follow-up paper.

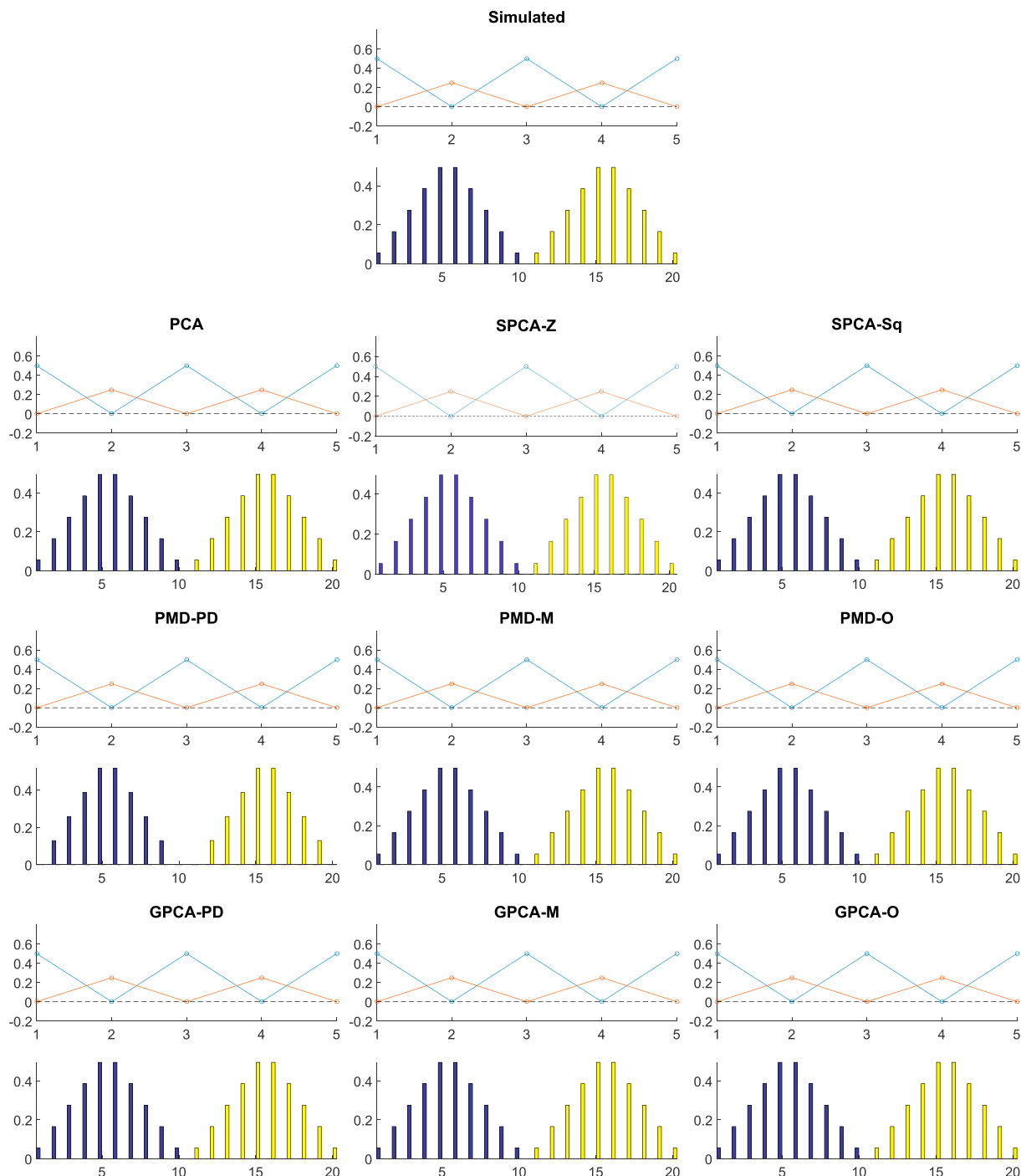


Fig. 1. Orthogonal spectra example: true data and matrix factorization methods.



Fig. 2. Non-orthogonal spectra example: true data and matrix factorization methods.

4. Simulated examples and Monte Carlo study

We will use a number of examples with simulated data to support our discussion, where we will opt not to introduce noise so that we can easily illustrate the behavior of sPCA algorithms. First, we will simulate two different examples for visual illustration, representing two common situations in exploratory data analysis that are of interest for this study: when orthogonality is consistent with the underlying data generation process and when it is not. The motivation is that a subset of sPCA methods and PCA impose orthogonality in some or all the factors, and we want to assess to which extent this is a useful constraint. As a second step we will generalize the results through a Monte Carlo simulation. We will

consider several variants and implementations of sPCA, some of which are evaluated and compared in this paper for the first time:

1. The sPCA algorithm by Zou et al. [2] (SPCA-Z) and the sequential (iterative) implementation in the SPASM toolbox [12] (SPCA-Sq).
2. The PMD algorithm by Ref. [3] with projection (PMD-PD) and orthogonalized deflation (PMD-O) discussed in section 2.2, and a modified version using Mackey's (5) generalized deflation (PMD-M) discussed in section 2.3.
3. The GPCA algorithm by (4) using projection deflation (GPCA-PD), Mackey's (5) generalized deflation (GPCA-M) and orthogonalized deflation by (3) (GPCA-O).

We selected the sPCA by Zou, PMD and GPCA because they span a variety of situations: from simultaneous to sequential calibration of components and using penalties or algorithmic procedures. Furthermore, the first two are probably the most widely used sPCA variants, while GPCA is more consistent with the exploratory methodology in the field of chemometrics.

4.1. Orthogonal spectra

The first example simulates spectra from a mixture of two compounds that result in two orthogonal components. The simulation has the following generating rules:

- The first component has non-zero values for variables {1–10} and the second for variables {11–20}. For each component, loadings follow a shape of spectra: {0.1 0.3 0.5 0.7 0.9 0.9 0.7 0.5 0.3 0.1}. Note loadings are intentionally non-negative, orthogonal and sparse (they contain 0's).
- The scores for five individuals are generated following the pattern:

$$\mathbf{T}(5 \times 2) = \begin{bmatrix} 0.5 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0.25 & 0 & 0.25 & 0 \end{bmatrix}^T$$

Note that the scores are intentionally non-negative and orthogonal.

- The data set \mathbf{X} is generated with dimension 5×20 following $\mathbf{X} = \mathbf{TP}^T$. There is no noise in the data, which means that the data is purely rank 2 and the true profiles are the ones shown above. The low dimension in observations and variables and the absence of noise are not realistic, but are useful to illustrate the performance of the models in a simple scenario.

The data set \mathbf{X} is decomposed using PCA and the different variants of sPCA, and results are shown in Fig. 1. Data is not mean-centered and components are orthogonal. Since all methods perfectly capture the data structure, and are equally interpretable, we shall conclude that they have equal performance when applied to noise-free, orthogonal data, without the requirement to impose orthogonality.

4.2. Non-orthogonal spectra

In the second example, we generate data from a set of sparse components that simulate non-orthogonal spectra with the following generating rules:

- Three components $\mathbf{P}(20 \times 3)$ are simulated: the first has non-zero values for variables {1–10}, the second for variables {6–15} and the third for {11–20}. For each component, the loadings follow a spectral shape like in the example in the previous section. Note loadings are intentionally non-negative and overlapping, and therefore are non-orthogonal, and sparse.
- The scores for five individuals are generated following the pattern:

$$\mathbf{T}(5 \times 3) = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0 \\ 0.25 & 0 & 0.25 & 0 & 0.25 \\ 0 & 0.125 & 0 & 0.125 & 0 \end{bmatrix}^T$$

Note that the scores for the first component are not orthogonal to the other two, but they are all non-negative.

Table 1

Correlation in scores and loadings for the non-orthogonal spectral example. MACS: Mean absolute correlation coefficient of the scores; MACL: Mean absolute correlation coefficient of the loadings.

	Simulated	SPCA-Z	SPCA-Sq	PMD-PD	PMD-O	PMD-M	GPCA-PD	GPCA-M	GPCA-O
MACS	0.43	0.84	0.73	0.66	0	0.58	0.52	0.52	0
MACL	0.17	0.21	0.05	0.09	0.05	0.07	0.03	0.03	0.03

- The data set \mathbf{X} is generated with dimension 5×20 following $\mathbf{X} = \mathbf{TP}^T$. No noise is introduced in the data, which means that the data is purely rank 3.

Fig. 2 shows the result of the simulation and the approximation of data \mathbf{X} by the sparse factorization methods considered along with PCA. Scores were intentionally computed following the standard approach as $\hat{\mathbf{T}} = \mathbf{X}\hat{\mathbf{P}}$, with $\hat{\mathbf{P}}$ the loadings estimated in the algorithms. PCA shows two limitations and fails to reproduce the structure of the data: first, components are not sparse and are constrained to be orthogonal and this induces distortion in the reconstructed spectra; second, scores must also be orthogonal, and to satisfy this condition some score values have to be negative to compensate for the positive ones. Concerning sparse models, they show a disparate performance. The sequential version of SPCA visually outperforms the original SPCA-Z and most other methods in the approximation. GPCA-PD and GPCA-M are also very accurate, but have difficulties to model the second and third loading vector. PMD is also less accurate in determining the loadings. Regarding the scores, orthogonalized methods distort the simulated (non-orthogonal) scores, as expected, and give the same score patterns as PCA.

4.3. Monte Carlo simulation

Based on the previous examples, we designed a Monte Carlo study in which datasets are generated where the components can have a random number of non-zero loadings and the values of the scores and loadings are also randomly selected. With respect to the previous illustrative examples, here the data sets are simulated 1) with more realistic dimensions and structure (50×200 and 5 components) as encountered in practical applications, and 2) are not constrained to be non-negative. Under this simulation scheme, the scores and the loadings are likely correlated, but at a lesser extent than in the case of non-orthogonal spectra in previous section. The generating rules are:

- Five components $\mathbf{P}(200 \times 5)$ are simulated. For each simulated data the loading matrix $\mathbf{P} \sim N(0, 1)$ has size 200×5 , for a total of 1000 loadings. An auxiliary 200×5 matrix $\mathbf{W} \sim N(0, 1)$ is generated, so that for elements of \mathbf{W} below 1 the corresponding elements in \mathbf{P} are set to 0 ($W_{ij} < 1 \Rightarrow P_{ij} = 0$). This procedure generates, on average, 160 non-zero loadings for every 1000 loadings simulated. The resulting loading vectors have a different degree of sparsity, contain both positive and negative values and are usually correlated because the non zero elements can overlap.
- The scores for 50 observations are randomly generated from $N(0, 1)$ and are multiplied by a constant so that their variance follows an exponential decay, similar to previous examples. The scores can be non-orthogonal and contain either positive or negative values.
- The data set \mathbf{X} is generated with dimension 50×200 but no noise is added, like in the previous examples. This means that the data is purely rank 5.

The simulation is repeated 100 times. All sparse algorithms considered require the setting of one or more metaparameters. We optimistically selected the metaparameters for each method to meet on average the known level of sparseness of the data generated. In this way, all methods can be fairly compared.

Results from the non-orthogonal spectra and Monte Carlo simulation

will be used to support the discussion in the following sections.

4.4. Correlation of scores and loadings

As previously discussed, scores and loadings in sPCA can be correlated. To quantify to which extent this happens, we defined and computed the following statistics.

- MACS: Mean absolute correlation coefficient of the scores, measured as the mean of the absolute value of the elements below the diagonal of $\hat{\mathbf{T}}^T \hat{\mathbf{T}} \oslash \|\text{diag}(\hat{\mathbf{T}}^T \hat{\mathbf{T}})\|_2$, where \oslash is the Hadamard (element-wise) division. This parameter ranges between 0 (absence of correlation) and 1 (total correlation).
- MACL: Mean absolute correlation coefficient of the loadings, measured as the mean of the absolute value of the elements below the diagonal of $\hat{\mathbf{P}}^T \hat{\mathbf{P}}$, for normalized loadings. This parameter ranges between 0 (absence of correlation) and 1 (total correlation).

Results are shown in Table 1 for the non-orthogonal spectra and in Fig. 3(a) and (b) for the Monte Carlo simulation. The actual correlation of the simulated data is shown as a baseline. It can be seen that all but the orthogonalized versions provide correlated scores, and that all methods produce correlated loadings. Non-orthogonal methods seem to largely overestimate the true correlation in the scores, while orthogonal methods obviously underestimate it (they yield exactly 0 correlation in all cases). The sequential approaches tend to underestimate the correlation in loadings.

From these results, we can infer that correlation in scores and loadings is commonplace. While this can be beneficial for the interpretation of complex data that can be, in general, non-orthogonal, it also poses several problems and limitations that should be taken into account when analyzing a sparse model. Correlated loadings distort distances in the scatter plots [25,26], while correlated scores and/or loadings may lead to the incorrect estimation of explained and residual variance [5]. More implications resulting from the correlation of loadings and scores will be described in the next sections and in the follow-up paper.

5. Computation of scores, residuals and variance

The non-orthogonality of the loadings has critical implications for the computation of the scores, the residuals and explained variance. While in standard PCA the scores are calculated as

$$\hat{\mathbf{T}} = \mathbf{X}\hat{\mathbf{P}}. \quad (15)$$

in the sparse PCA setting the scores should be computed using the correction formula

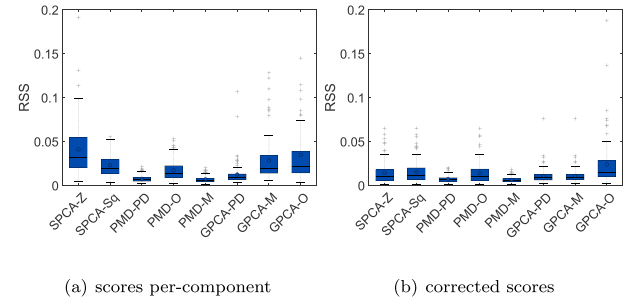


Fig. 4. Box Plots of RSS for the Monte Carlo simulation.

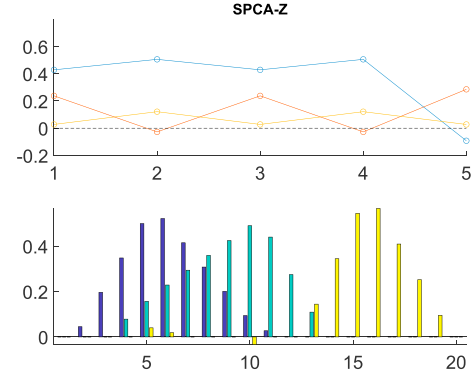


Fig. 5. Non-orthogonal spectra example: corrected SPCA-Z.

$$\hat{\mathbf{T}} = \mathbf{X}\hat{\mathbf{P}}(\hat{\mathbf{P}}^T \hat{\mathbf{P}})^+ \quad (16)$$

since Eq. (15) cannot be applied because loadings are, in general, not orthogonal. The superscript '+' indicates the Moore-Penrose inverse. For orthonormal loadings, Eq. (16) reduces to Eq. (15). A noteworthy characteristic of Eq. (16) is that the scores cannot be accurately computed in a sequential, one-component-at-a-time fashion, since the scores depend on the loadings of the other components.

The level of inaccuracy in the computation of the residual can be assessed using a Monte Carlo simulation. Fig. 3(a) and (c) show a comparison of the correlation observed in the scores computed one-at-a-time ($\hat{\mathbf{T}} = \mathbf{X}\hat{\mathbf{P}}$) and the correlation observed in the scores corrected using Eq. (16). It can be seen that for all methods considered, the correlation between scores is affected by the correction and is reduced in most cases. The correction also induces correlation in orthogonalized methods. The way correction affects the residuals can be quantified using the following RSS statistic:

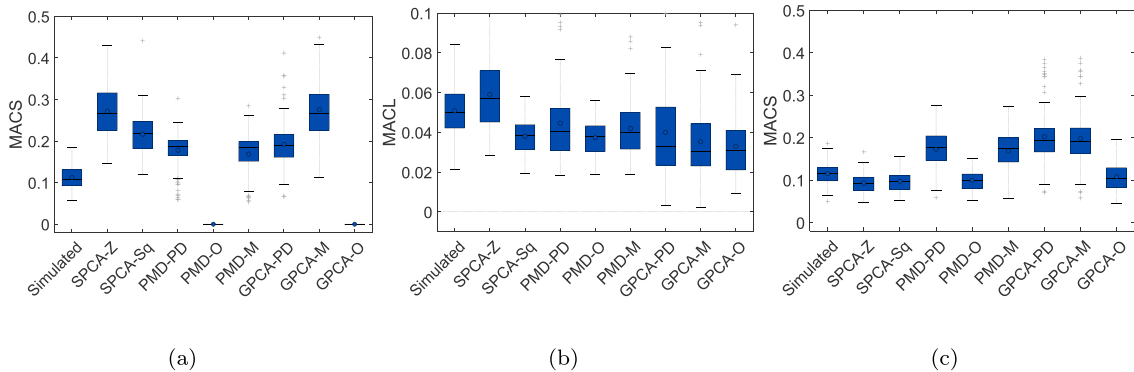


Fig. 3. Box Plots of MACS (a), MACL (b) and MACS with corrected scores (c) for the Monte Carlo simulation. Please, note y-axis are of different scale.

Table 2

Statistics for the calculation of the fraction of variance captured by the different sparse algorithms for the non-orthogonal spectra example. * Indicates that Eq. (16) is used to compute the scores. In the other cases the scores are computed individually per each component.

	SPCA-Z	SPCA-Sq	PMD-PD	PMD-O	PMD-M	GPCA-PD	GPCA-M	GPCA-O
TotQR	1.2730	0.9616	0.9106	1.0000	0.9223	0.9345	0.9345	1.0000
TotT	1.7747	1.0722	1.0000	1.0000	1.0008	1.0000	1.0000	1.0000
TotPT	2.5494	1.1444	1.0689	1.0000	1.0457	1.0435	1.0435	1.0000
TotQR*	0.8241	0.9097	0.8541	1.0000	0.8857	0.8947	0.8947	1.0000
TotT*	0.8606	0.9603	0.9362	1.0000	0.9579	0.9584	0.9584	1.0000
TotPT*	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

- RSS: The residual sum-of-squares normalized by the total sum-of-squares of the data, $\text{tr}(\mathbf{E}^T\mathbf{E})/\text{tr}(\mathbf{X}^T\mathbf{X})$, with $\mathbf{E} = \mathbf{X} - \hat{\mathbf{T}}\hat{\mathbf{P}}^T$. This quantity reflects the quality of the scores capturing the variance in the data, and should be as close to 0 as possible.

Fig. 4 compares residuals before and after correction: the correction reduces the sum-of-squares of the residuals. This reduction comes by definition, given that Eq. (16) is a least squares correction. Note that uncorrected residuals are used within the sPCA deflation-based algorithms to compute new scores, so this computation may be inaccurate.

Fig. 5 shows the results of correcting the scores using Eq. (16) in the non-orthogonal spectra example when simultaneous SPCA-Z is used. Comparing these results with those shown in Fig. 2, it can be seen that the correction is relevant and greatly improves interpretability. However, for sequential approaches, the correction is visually negligible in the non-orthogonal spectra example (not shown).

The correct computation of scores also affects the estimation of the variance explained by each component. Consider a full rank approximation $\mathbf{X} = \hat{\mathbf{T}}\hat{\mathbf{P}}^T$. The variance of \mathbf{X} can be computed as: $\text{tr}(\mathbf{X}^T\mathbf{X}) = \text{tr}(\hat{\mathbf{P}}\hat{\mathbf{T}}^T\hat{\mathbf{T}}\hat{\mathbf{P}}^T)$. For orthogonal loadings, $\text{tr}(\mathbf{X}^T\mathbf{X}) = \text{tr}(\hat{\mathbf{T}}^T\hat{\mathbf{T}}\hat{\mathbf{P}}^T\hat{\mathbf{P}}) = \text{tr}(\hat{\mathbf{T}}^T\hat{\mathbf{T}})$. However, for non-orthogonal loadings the last equality does not hold. In general, the captured/explained variance in a component model with one sparse mode should be measured as $\text{tr}(\hat{\mathbf{P}}\hat{\mathbf{T}}^T\hat{\mathbf{T}}\hat{\mathbf{P}}^T)$. A more

detailed discussion can be found in Appendix A.

To support the previous discussion, we defined and calculated several statistics for (1) the scores computed as $\hat{\mathbf{T}} = \mathbf{X}\hat{\mathbf{P}}$ and (2) for the scores corrected with Eq. (16). These statistics are based on different ways of calculating the captured variance. In all of them, the captured variance is added to the variance in the residuals and normalized by the total sum-of-squares of the data. Therefore, all statistics are expected to be equal to 1. The considered statistics are defined as:

- TotQR: The fraction of total variance computed as: $(\text{QSS} + \text{tr}(\mathbf{E}^T\mathbf{E}))/\text{tr}(\mathbf{X}^T\mathbf{X})$, where QSS is the sum-of-squares of the diagonal of \mathbf{B} from the QR decomposition of $\hat{\mathbf{T}} = \mathbf{A}\mathbf{B}$.
- TotT: The fraction of total variance computed as: $(\text{tr}(\hat{\mathbf{T}}^T\hat{\mathbf{T}}) + \text{tr}(\mathbf{E}^T\mathbf{E}))/\text{tr}(\mathbf{X}^T\mathbf{X})$, where the first term in the numerator is the sum-of-squares of the scores.
- TotPT: The fraction of total variance computed as: $(\text{tr}(\hat{\mathbf{P}}\hat{\mathbf{T}}^T\hat{\mathbf{T}}\hat{\mathbf{P}}^T) + \text{tr}(\mathbf{E}^T\mathbf{E}))/\text{tr}(\mathbf{X}^T\mathbf{X})$, where the first term in the numerator is the sum-of-squares of the reconstructed data.

Numerical results for the non-orthogonal spectra are presented in Table 2. The corrected computations lead to the expected value TotPT = 1. However, this is not the case for non-corrected scores and for other approaches to compute the variance. As discussed previously, the QR

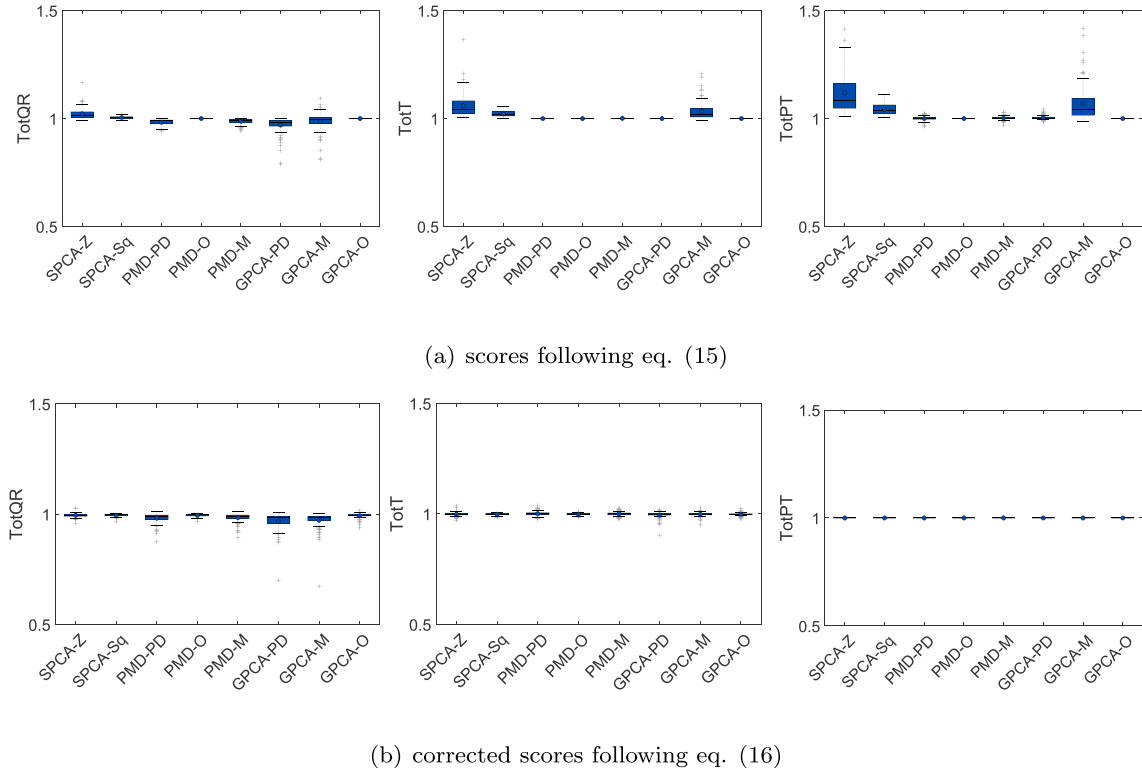


Fig. 6. Statistics for the calculation of the fraction of variance captured by the different sparse algorithms for the Monte Carlo simulation.

decomposition is, in general, an incorrect way to compute variance, since TotQR is different from 1 in most occasions. This also holds for TotT. The amount of variance captured by the orthogonalized methods is the same independently of the statistic used to measure it.

Fig. 6 shows the effect of the correction in the scores in the Monte Carlo simulation. Without the correction, all methods, except for orthogonalized versions, show variance artifacts in TotPT. This especially affects the simultaneous SPCA-Z. After correction, TotPT is systematically equal to 1, while TotQR and TotT are not, indicating that these statistics are not appropriate to describe the variance captured by the sparse model.

6. Conclusion

In this paper we reviewed a number of Sparse Principal Component Analysis (sPCA) algorithms and compared them through simulation. We showed that sPCA approaches present disparate and limited performance when modeling sparse data, even in the optimistic case of noise-free data. In a follow-up paper, we will discuss the theoretical properties that lead to this problem.

In this paper we focused on the inherent properties of sPCA, that make it substantially different to PCA and, for this difference, PCA and sparse PCA models must be handled in different ways. In the light of this,

Appendix B. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.chemolab.2019.103907>.

Appendix A. Calculation of the explained variance

For a standard PCA model it holds

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} = \hat{\mathbf{X}} + \mathbf{E} \quad (\text{A.1})$$

from which it follows that

$$\|\mathbf{X}\|_F^2 = \|\mathbf{T}\mathbf{P}^T + \mathbf{E}\|_F^2. \quad (\text{A.2})$$

Using the general equation $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^T\mathbf{A})$ it follows that

$$\|\mathbf{X}\|^2 = \text{tr}(\mathbf{X}^T\mathbf{X}) = \text{tr}(\mathbf{P}\mathbf{T}^T\mathbf{T}\mathbf{P}^T) + \text{tr}(\mathbf{P}\mathbf{T}^T\mathbf{E}) + \text{tr}(\mathbf{E}^T\mathbf{T}\mathbf{P}^T) + \text{tr}(\mathbf{E}^T\mathbf{E}) \quad (\text{A.3})$$

by simply multiplying out the second part of Equation (A.2). If deflation is performed in the score space then \mathbf{T} are the fixed regressors and a property of the deflation (i.e. regression) is that $\mathbf{T}^T\mathbf{E} = \mathbf{E}^T\mathbf{T} = 0$ and thus

$$\|\mathbf{X}\|^2 = \text{tr}(\mathbf{X}^T\mathbf{X}) = \text{tr}(\mathbf{P}\mathbf{T}^T\mathbf{T}\mathbf{P}^T) + \text{tr}(\mathbf{E}^T\mathbf{E}) = \|\hat{\mathbf{X}}\|^2 + \|\mathbf{E}\|^2 \quad (\text{A.4})$$

which is a simple split-up of explained variation. If the deflation is performed in the loading space then $\mathbf{E}\mathbf{P} = \mathbf{P}^T\mathbf{E}^T = 0$ and Equation (A.4) holds since $\text{tr}(\mathbf{P}\mathbf{T}^T\mathbf{E}) = \text{tr}(\mathbf{E}\mathbf{P}^T\mathbf{T})$.

Remarks regarding calculating the explained variances:

1. The split-up of variation depends on whether Equation (A.1) holds and, additionally, either $\mathbf{E}\mathbf{P} = 0$ or $\mathbf{T}^T\mathbf{E} = 0$.
2. There is no requirement regarding the orthogonality of \mathbf{T} and/or \mathbf{P} .
3. There is also no requirement of the columns of \mathbf{T} being in the column space of \mathbf{X} nor the columns of \mathbf{P} in the row space of \mathbf{X} .

These equations become simpler if there is orthogonality. If $\mathbf{T}^T\mathbf{T} = \mathbf{I}$ then

$$\|\mathbf{X}\|^2 = \text{tr}(\mathbf{X}^T\mathbf{X}) = \text{tr}(\mathbf{P}\mathbf{T}^T\mathbf{T}\mathbf{P}^T) + \text{tr}(\mathbf{E}^T\mathbf{E}) = \text{tr}(\mathbf{P}\mathbf{P}^T) + \text{tr}(\mathbf{E}^T\mathbf{E}) = \text{tr}(\mathbf{P}^T\mathbf{P}) + \text{tr}(\mathbf{E}^T\mathbf{E}). \quad (\text{A.5})$$

Similarly, if $\mathbf{P}^T\mathbf{P} = \mathbf{I}$ then

$$\|\mathbf{X}\|^2 = \text{tr}(\mathbf{X}^T\mathbf{X}) = \text{tr}(\mathbf{P}\mathbf{T}^T\mathbf{T}\mathbf{P}^T) + \text{tr}(\mathbf{E}^T\mathbf{E}) = \text{tr}(\mathbf{P}^T\mathbf{P}\mathbf{T}^T\mathbf{T}) + \text{tr}(\mathbf{E}^T\mathbf{E}) = \text{tr}(\mathbf{T}^T\mathbf{T}) + \text{tr}(\mathbf{E}^T\mathbf{E}) \quad (\text{A.6})$$

we provided corrected equations for quantities such as captured variances, scores and residuals. These corrections are relevant, taking into account that scores and residuals are central in chemometrics and related disciplines for data visualization and interpretation. Furthermore, explained variance is useful for comparison among model variants, and also to compare sPCA with other modeling approaches.

Statement

All authors were equally involved in the research and preparation of the paper.

Declaration of competing interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgement

This work is partly supported by the Spanish Ministry of Economy and Competitiveness and FEDER funds through project TIN2017-83494-R and the ‘‘Plan Propio de la Universidad de Granada’’.

which explains why sometimes calculating explained variances using only scores or loadings may work.

References

- [1] I.T. Jolliffe, N.T. Trendafilov, M. Uddin, A modified principal component technique based on the LASSO, *J. Comput. Graph. Stat.* 12 (3) (2003) 531–547.
- [2] Hui Zou, Hastie Trevor, Tibshirani Robert, Sparse principal component analysis, *J. Comput. Graph. Stat.* 15 (2006) 265–286.
- [3] M. Witten Daniela, Tibshirani Robert, Hastie Trevor, A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, *Biostatistics* 10 (2009) 515–534.
- [4] Camacho José, A. Rodríguez-Gómez Rafael, Saccenti Edoardo, Group-wise principal component analysis for exploratory data analysis, *J. Comput. Graph. Stat.* 26 (2017) 501–512.
- [5] Mackey Lester, Deflation methods for sparse PCA, *NIPS (News Physiol. Sci.)* 1–8 (2008).
- [6] Hastie Trevor, Tibshirani Robert, Wainwright Martin, *Statistical Learning with Sparsity: the Lasso and Generalizations*, Chapman & Hall/CRC, 2015.
- [7] I.T. Jolliffe, *Principal Component Analysis*, EEUU: Springer Verlag Inc, 2002.
- [8] J.E. Jackson, *A User's Guide to Principal Components*, Wiley-Interscience, England, 2003.
- [9] Richtárik Peter, Takác Martin, Selin Damla Ahipasaoglu, Alternating Maximization: Unifying Framework for 8 Sparse PCA Formulations and Efficient Parallel Codes *CoRR*, 2012 abs/1212.4137.
- [10] Tibshirani Robert, Regression selection and shrinkage via the lasso, *J. R. Stat. Soc. Ser. B* 58 (1994) 267–288.
- [11] H. Zou, T. Hastie, Regularization and variable selection via the elastic-net, *J. R. Stat. Soc.* 67 (2005) 301–320.
- [12] Sjöstrand Karl, Clemmensen Line, Larsen Rasmus, Einarsson Gudmundur, Ersbøll Bjarne, SpaSM: a matlab toolbox for sparse statistical modeling, *Journal of Statistical Software, Articles* 84 (2018) 1–37.
- [13] J. Camacho, Missing-data theory in the context of exploratory data analysis, *Chemometr. Intell. Lab. Syst.* 103 (2010) 8–18.
- [14] Hui Zou, Lingzhou Xue, A selective overview of sparse principal component analysis, *Proc. IEEE* 106 (2018) 1311–1320.
- [15] M. Johnstone Iain, Arthur Yu Lu, On consistency and sparsity for principal components analysis in high dimensions, *J. Am. Stat. Assoc.* 104 (2009) 682–693. PMID: 20617121.
- [16] Qian Zhao, DeYu Meng, ZongBen Xu, Robust sparse principal component analysis, *Sci. China Inf. Sci.* 57 (2014) 1–14.
- [17] Shuangyan Yi, Zhihui Lai, Zhenyu He, Cheung Yiu-ming, Yang Liu, Joint sparse principal component analysis, *Pattern Recognit.* 61 (2017) 524–536.
- [18] Ling Wang, Jihao Yin, The robust sparse PCA for data reconstructive via weighted elastic, *Net Lecture Notes in Electrical Engineering* 202 (2012) 225–235.
- [19] Mia Hubert, Reynkens Tom, Eric Schmitt, Tim Verdonck, Sparse PCA for high-dimensional data with outliers, *Technometrics* 58 (2016) 424–434.
- [20] Weifeng Liu, H. Zhang, Dapeng Tao, Y. Wang, K. Lu, Large-scale paralleled sparse principal component analysis, *Multimed. Tools Appl.* 75 (2016) 1481–1493.
- [21] Morten Arendt Rasmussen, Bro Rasmus, A tutorial on the Lasso approach to sparse modeling, *Chemometr. Intell. Lab. Syst.* 119 (2012) 21–31.
- [22] Bro Rasmus, Papalexakis Evangelos E., Acar Evrim, Sidiropoulos Nicholas D., Cocustering—a useful tool for chemometrics *J. Chemom.* 26:256–263.
- [23] Deun Katrijn Van, Thorrez Lieven, Coccia Margherita, et al., Weighted sparse principal component analysis, *Chemometr. Intell. Lab. Syst.* (2019) 103875.
- [24] E.R. Malinowski, *Factor Analysis in Chemistry*, Wiley, 2002.
- [25] A.L. Kiers Henk, Some procedures for displaying results from three-way methods, *J. Chemom.* 14 (2000) 151–170.
- [26] Geladi Paul, Manley Marena, Lestander Torbjörn, Scatter plotting in multivariate data analysis, *J. Chemom.* 17 (2003) 503–511.