Sparse PCA for high-dimensional data with outliers

Mia Hubert Tom Reynkens Eric Schmitt

Tim Verdonck

Department of Mathematics, KU Leuven

Leuven, Belgium

June 25, 2015

Abstract

A new sparse PCA algorithm is presented which is robust against outliers. The approach is based on the ROBPCA algorithm which generates robust but non-sparse loadings. The construction of the new ROSPCA method is detailed, as well as a selection criterion for the sparsity parameter. An extensive simulation study and a real data example are performed, showing that it is capable of accurately finding the sparse structure of datasets, even when challenging outliers are present. In comparison with a projection pursuit based algorithm, ROSPCA demonstrates superior robustness properties and comparable sparsity estimation capability, as well as significantly faster computation time.

Keywords: dimension reduction, outlier detection, robustness

1 INTRODUCTION

Principal Component Analysis (PCA) is a popular technique used for dimension reduction. The idea is to find a number of uncorrelated linear combinations of the original variables that capture most of the covariance structure of the original data. These combinations are called the Principal Components (PCs). Those directions are chosen such that they are orthogonal and sequentially maximize the variance of the projected data. Typically one does not use all the PCs, but only the first k explaining a sufficient portion of the total variance (i.e. information) of the original data. Despite its advantages, Classical Principal Component Analysis (CPCA) also has several drawbacks; two of which we will focus on.

First, CPCA often results in PCs that are difficult to interpret because most of the loadings are neither very small nor very large in absolute value. To increase interpretability, sparse PCA methods were developed to estimate PCs with many zero loadings. This is useful when the data is high-dimensional, since only a subset of the original variables may need to be analyzed or measured. Two popular methods for performing sparse PCA are SCoTLASS (Jolliffe et al. 2003) and SPCA (Zou et al. 2006).

Second, it is well known that outliers present in the data can heavily effect the CPCA estimates. Several robust alternatives for CPCA have been proposed including a Projection Pursuit PCA approach (PP-PCA) (Li and Chen 1985; Hubert et al. 2002; Croux and Ruiz-Gazen 2005), spherical PCA (Locantore et al. 1999), and ROBPCA (Hubert et al. 2005).

In this paper, we propose a new method, RObust Sparse PCA (ROSPCA), combining the advantageous properties of sparse and robust PCA. Previous work on this problem has been done by Croux et al. (2013), who developed a sparse version of the robust PP-PCA method by integrating sparsity principles into the formulation of PP-PCA. Since we believe that the detection of outliers may be the more difficult, and crucial, challenge, we approach the problem from a different direction, and develop a sparse modification of the robust ROBPCA method. The main difference is that we partially separate the outlier detection step from the sparsification step. As we detail in this paper, doing so results in greater robustness and more accurate sparse estimates.

Note that our model assumptions are different from those studied in Candès et al.

(2011) and Zhou et al. (2010). Whereas we are searching for a subspace spanned by sparse vectors, in the latter papers not the subspace but the errors are supposed to be sparse. This allows to recover the subspace exactly with a convex optimization program.

In Section 2 we first give a summary of existing methods for sparse and/or robust PCA, and then we detail our new method together with a new criterion to select the sparsity parameter. Section 3 contains the results of a simulation study, whereas Section 4 illustrates ROSPCA on a real dataset. Finally, Section 5 contains conclusions and directions for further research.

2 METHODS

2.1 Classical PCA

To fix notation, we begin by defining PCA for a data matrix, $\boldsymbol{X} = \boldsymbol{X}_{n,p} \in \mathbb{R}^{n \times p}$. In general, the subscripts denote the dimensions of the matrix and will only be added when appropriate. The p-dimensional observations in $\boldsymbol{X}$ are denoted by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. The loadings of the PCs, i.e. the components of the linear combinations, are in the columns of the orthogonal *loadings matrix* $\boldsymbol{P}$. Given estimated loadings $\boldsymbol{P}$ and center $\hat{\boldsymbol{\mu}}$, projecting the centered $\boldsymbol{X}$ on the new directions yields the *scores matrix* $\boldsymbol{T} = (\boldsymbol{X} - \boldsymbol{1}_n \hat{\boldsymbol{\mu}}')\boldsymbol{P}$, with $\boldsymbol{1}_n$ a column vector consisting of n ones.

Classical PCA can be described as searching for a $\hat{\boldsymbol{\mu}}$ and $\boldsymbol{P}$ such that the scores have maximal variance, and are uncorrelated. The PCA directions then correspond to the eigenvectors of the classical covariance matrix $\boldsymbol{S}$ of $\boldsymbol{X}$, whereas the variance of the data projected on an eigenvector is equal to the corresponding eigenvalue of $\boldsymbol{S}$. Note that when the variances of the original variables differ greatly, the data should first be standardized. If one uses the componentwise standard deviation, this comes down to computing the eigenvectors of the correlation matrix of $\boldsymbol{X}$.

Typically, $k \ll p$ dimensions are needed to express the information in the data. Various approaches exist to select the number of components to retain, k. One of the simplest and most popular is the *scree plot.* It plots the sorted, decreasing eigenvalues versus their index. The number of components corresponding to the point at which an elbow in the plot occurs is then selected. Following the selection of the number of components, only the first k columns of $\boldsymbol{P}$ are used and denoted as $\boldsymbol{P}_{p,k} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k]$.

2.2 Sparse PCA

Sparse PCA has the advantage of making the interpretation of the PCs easier. A simple way to accomplish this is to set all loadings with absolute value smaller than a certain threshold to zero. This method is called *simple thresholding.* Cadima and Jolliffe (1995) noticed that this method can be potentially misleading. For example, one should also look at the standard deviations of variables to determine the contribution of a variable to a certain PC.

To overcome the issues of that early method, a number of methods have been developed. One of these is SCoTLASS, which was proposed by Jolliffe et al. (2003). It integrates an L_1 constraint with PCA, yielding sparse loadings. The resulting objective function seeks the orthogonal loadings $\boldsymbol{p}_j$ maximizing the variance explained by the fitted model, subject to the constraint $\|\boldsymbol{p}_j\|_1 \leqslant \eta_j$, a sparsity constraint, where $\|\boldsymbol{p}_j\|_1$ is the L_1 norm of $\boldsymbol{p}_j$. We will work with the dual of this problem:

$$\boldsymbol{p}_j = \underset{\|\boldsymbol{p}\|=1, \boldsymbol{p}\perp\boldsymbol{p}_1, \ldots, \boldsymbol{p}\perp\boldsymbol{p}_{j-1}}{\operatorname{argmax}} \boldsymbol{p}'\boldsymbol{S}\boldsymbol{p} - \lambda_j \|\boldsymbol{p}\|_1, \tag{1}$$

where $\boldsymbol{p}_j$ is the jth PCA direction. Under this formulation, λ_j is the sparsity parameter for SCoTLASS, in place of η_j. A higher value of λ_j corresponds to greater sparsity, and a value of zero corresponds to no sparsity.

2.3 Robust PCA

The loadings matrix estimated by CPCA and sparse PCA is very sensitive to outliers. Robust principal component analysis addresses this issue. Two well known robust PCA methods are robust Projection Pursuit PCA (PP-PCA) and ROBPCA. PP-PCA maximizes a robust measure of spread to obtain consecutive directions on which the data is projected. Croux and Ruiz-Gazen (2005) proposed a version that serves as the basis for one variant of sparse, robust PCA. The ROBPCA method (Hubert et al. 2005) combines ideas from projection pursuit and robust covariance estimation. These approaches will be discussed in greater detail below, when we encounter sparse versions.

To detect PCA outliers, two notions of distance are used: robust score distances and orthogonal distances. The *robust score distance* (SD) measures the robust statistical distance from a PC score to the center of the scores. For an observation $\boldsymbol{x}_i$, the robust score distance is defined as

$$\mathrm{SD}_i = \sqrt{\sum_{j=1}^{k} \frac{(\boldsymbol{t}_i)_j^2}{l_j}} = \sqrt{\boldsymbol{t}_i' \boldsymbol{L}^{-1} \boldsymbol{t}_i}, \tag{2}$$

with k the number of PCs, $(\boldsymbol{t}_i)_j$ the jth component of the ith score $\boldsymbol{t}_i$ and $\boldsymbol{L}$ the diagonal matrix containing the robust eigenvalues corresponding to the robust PCs. We set the cut-off for observations with high SD values at $c_{\mathrm{SD}} = \sqrt{\chi^2_{k,0.975}}$, the square root of the 97.5% quantile of a chi-squared distribution with k degrees of freedom. This is justified when the scores are approximately normally distributed.

The *orthogonal distance* (OD) of an observation $\boldsymbol{x}_i$ to the PCA subspace is given by

$$\mathrm{OD}_i = \|\boldsymbol{x}_i - \hat{\boldsymbol{\mu}} - \boldsymbol{P}_{p,k}\boldsymbol{t}_i\|. \tag{3}$$

Note that $\hat{\boldsymbol{\mu}} + \boldsymbol{P}_{p,k}\boldsymbol{t}_i$ is the projection of $\boldsymbol{x}_i$ on the PCA subspace determined by $\boldsymbol{P}_{p,k}$ and $\hat{\boldsymbol{\mu}}$. To obtain a cut-off for the orthogonal distances, we follow the approach taken in Hubert et al. (2005). This makes use of the Wilson-Hilferty approximation for a chi-squared distribution, which implies that the orthogonal distances to the power 2/3 are approximately normally distributed. To obtain estimates of the center and scale of this distribution we use the univariate MCD (Rousseeuw 1984), a robust estimator that searches for the subset of size $\frac{n}{2} < h \leqslant n$ that has the smallest variance and bases location

($\hat{\mu}_{MCD}$) and scale ($\hat{\sigma}_{MCD}$) estimates on it. Given these parameters, the cut-off is defined as $c_{\text{OD}} = (\hat{\mu}_{\text{MCD}} + \hat{\sigma}_{\text{MCD}} z_{0.975})^{3/2}$, with $z_{0.975}$ the 97.5% quantile of the standard normal distribution.

2.4 SRPCA

Croux et al. (2013) proposed a robust, sparse method that combines ideas from the PP approach and sparse PCA. It will be used as a benchmark in our simulations and a real data example. Their approach consists of adding the L_1 penalty into the PP equations. The method thus looks for directions that maximize the scale of the data projected on them under the constraint that the loadings of these directions should not be too large. The jth sparse PCA direction is given by

$$\tilde{\boldsymbol{p}}_j = \begin{cases} \underset{\|\boldsymbol{p}\|=1}{\operatorname{argmax}}\ S(\boldsymbol{p}'\boldsymbol{x}_1, \ldots, \boldsymbol{p}'\boldsymbol{x}_n) - \lambda_1 \|\boldsymbol{p}\|_1 & \text{if } j = 1 \\ \underset{\|\boldsymbol{p}\|=1,\, \boldsymbol{p}\perp\tilde{\boldsymbol{p}}_1, \ldots, \boldsymbol{p}\perp\tilde{\boldsymbol{p}}_{j-1}}{\operatorname{argmax}}\ S(\boldsymbol{p}'\boldsymbol{x}_1, \ldots, \boldsymbol{p}'\boldsymbol{x}_n) - \lambda_j \|\boldsymbol{p}\|_1 & \text{if } 1 < j \leqslant p, \end{cases} \tag{4}$$

where S is a measure of scale. If one uses the sample standard deviation for S, this method is nothing more than SCoTLASS. To obtain robust principal components, Croux et al. (2013) suggest to use the robust Q_n estimator of scale (Rousseeuw and Croux 1993). The Q_n is the first quartile of the pairwise distances between the elements of a vector. The data are typically centered using a robust estimator for the center (e.g. using the L_1-median). Then, one applies the PP steps on the $\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}$ (for $1 \leqslant i \leqslant n$), with $\hat{\boldsymbol{\mu}}$ the robust estimate for the center.

The sparsity parameter λ_j can vary across the different PCs. Croux et al. (2013) make the relative importance of the L_1 penalty comparable across the different PCs. This means that there is a similar degree of sparsity across the PCs. They take $\lambda_j = \lambda v_j$ where v_j can be defined as follows. Suppose we have found the $j-1$ first PC directions and denote by $X_j^{\perp}$ the data projected on the space orthogonal to the space spanned by the $j-1$ first PC directions. The number v_j is then the average of the variance measure S^2 applied to the columns of $X_j^{\perp}$. Note that v_1 is the average of the variance measure S^2

applied to the columns of X. This definition is used in the `R` packages *pcaPP* (Filzmoser et al. 2014) and *rrcovHD* (Todorov 2014) and differs slightly from the definition in Croux et al. (2013). Hence, there is only one tuning parameter to select: the sparsity parameter λ. We denote this method by SRPCA as in Todorov and Filzmoser (2013).

To find the sparse PCA directions in (4), the expressions need to be maximized over a p-dimensional space. This optimization problem is non-convex. The Grid algorithm of Croux et al. (2007) is an accurate algorithm that is used to obtain the PCA directions in the PP approach. In Croux et al. (2013), the authors extend it for sparse PCA and provide a detailed description of the algorithm. Since SRPCA is a generalization of the PP approach, Croux et al. (2013) proposed to extend the Grid algorithm to compute the sparse directions. Henceforth, we will use this algorithm to compute the sparse loadings of SCoTLASS and SRPCA. By default, the maximum number of iterations is equal to 10, but we noticed that the algorithm does not yet converge then. We use a maximum of 75 iterations instead which provides stable results.

2.5 ROSPCA

Hubert et al. (2005) proposed a robust PCA algorithm combining ideas from projection pursuit and the MCD estimator, which they called ROBPCA. Many steps in ROBPCA anticipate those of ROSPCA as the robustness properties of the latter derive almost directly from the former. Intuitively, they can be compared as follows. ROBPCA finds an outlier-free subset which determines a robust subspace. Then, it projects the data onto this subspace to estimate the eigenvectors and eigenvalues robustly. The ROSPCA method (RObust Sparse PCA) integrates sparse PCA into ROBPCA. In doing so, ROSPCA finds a subset that determines a robust, *sparse* subspace, and then estimates the eigenvectors and eigenvalues while preserving sparsity.

Not surprisingly the method contains two hyperparameters: α which determines the degree of robustness and λ which regulates the sparsity. The value of α must satisfy $0.5 \leqslant \alpha < 1$ and needs to be chosen in advance. It constitutes a lower bound on the number of regular observations, so at most $(1-\alpha)100\%$ of the n data points are allowed to be outlying. If no a priori information about the amount of outliers is available, we recommend to set $\alpha = 0.5$, yielding maximal robustness. The choice of the sparsity parameter λ will be discussed in Section 2.6.

The ROSPCA algorithm consists of an outlier detection part (step 1), and a sparsification part (steps 2 and 3):

1. The first part is similar to ROBPCA, so we describe it only shortly. When a standardization is appropriate, the variables are first robustly standardized by means of the componentwise median and the Q_n. Then using the SVD of the resulting data matrix, the p-dimensional data space is reduced to the affine subspace spanned by the n observations. We denote the resulting data matrix (of rank at most $n-1$) by $\tilde{\boldsymbol{X}}$. Next, for each $\tilde{\boldsymbol{x}}_i$ the Stahel-Donoho outlyingness is computed as

   $$\text{outl}(\tilde{\boldsymbol{x}}_i) = \max_{\boldsymbol{v} \in B} \frac{|\tilde{\boldsymbol{x}}_i'\boldsymbol{v} - \hat{\mu}_{\text{MCD}}(\tilde{\boldsymbol{x}}_j'\boldsymbol{v})|}{\hat{\sigma}_{\text{MCD}}(\tilde{\boldsymbol{x}}_j'\boldsymbol{v})} \tag{5}$$

   where $\hat{\mu}_{\text{MCD}}$ and $\hat{\sigma}_{\text{MCD}}$ are the univariate MCD estimators of location and scale. The set $\boldsymbol{B}$ consists of all directions $\boldsymbol{v}$ passing through two data points (or a random subset of these directions if n is very large).

   Thereafter, the $h_0 = \lceil \alpha n \rceil + 1$ observations with smallest outlyingness are considered, they are mean-centered and SVD is applied to them to find the k-dimensional subspace most closely to them (in L_2-norm). Here, the scree plot can be used to find an appropriate value for k, or the cumulative percent variation (CPV). For example, one could select k such that $\text{CPV} = \sum_{j=1}^k s_j^2 / \sum_{j=1}^p s_j^2 \geqslant 80\%$ with s_j the singular values of the SVD decomposition. Next, following Engelen et al. (2005), given the orthogonal distances to the preliminary subspace, we consider all observations with ODs smaller than the corresponding cut-off (as explained in Section 2.3). This yields an outlier-free index set H_1 of size h_1, which typically will be larger than h_0, in particular when α is chosen much smaller than the real proportion of regular observations.

2. Whereas ROBPCA applies CPCA on the observations from H_1, ROSPCA now uses sparse PCA. More precisely, we first standardize the data points of $\boldsymbol{X}$ with indices in H_1 using the componentwise median and the Q_n. Performing sparse PCA on them, by means of the Grid-based implementation of SCoTLASS with sparsity parameter λ, yields the sparse loadings matrix $\boldsymbol{P}_1 \in \mathbb{R}^{p \times k}$.

   We then perform an additional reweighting step that incorporates information about the sparse structure of the data, forming a bridge between the sparse and robust components of the algorithm and increasing efficiency. We discard variables with zero loadings on all k PCs and we then compute the orthogonal distances to the estimated sparse PCA subspace. This yields an index set H_2 of observations with orthogonal distance smaller than the cut-off corresponding to these new orthogonal distances. We now standardize the subset of $\boldsymbol{X}$ with indices in H_2 using the componentwise median and the Q_n of the observations in H_1 (we use the same standardization as in the first time sparse PCA is applied). Then, sparse PCA is applied onto them, again by means of the Grid-based implementation of SCoTLASS with sparsity parameter λ. To get a full loadings matrix $\boldsymbol{P}_2$, we also need to add zero rows for all discarded variables to the estimated loadings matrix. The k-dimensional scores after reweighting are then given by $\boldsymbol{T} = (\boldsymbol{X} - \mathbf{1}_n \hat{\boldsymbol{\mu}}_1')\boldsymbol{P}_2$, with $\hat{\boldsymbol{\mu}}_1'$ the median of the observations in H_1. Intuitively, the goal of this reweighting is to recapture information from observations that are only outlying due to their behavior on variables that are found to be unimportant in our model, and use this information to obtain better estimates of the loadings corresponding to the important variables. Such observations will still have high OD values since the variables on which they are outlying will be compared to zero loadings in $\boldsymbol{P}_2$.

3. Finally, the eigenvalues are estimated robustly by applying the Q_n^2 estimator on the scores of the observations with indices in H_2. We need to use a robust measure of scale because observations with low OD and high SD that are included can influence the eigenvalue estimation. In order to robustly estimate the center, we compute the score distances and look at all observations of H_2 with a score distance smaller than the

corresponding cutoff, this is the set H_3. We then estimate the center by the mean of these observations which gives the final center $\hat{\boldsymbol{\mu}}$ and the final scores $\boldsymbol{T} = (\boldsymbol{X} - \mathbf{1}_n\hat{\boldsymbol{\mu}}')\boldsymbol{P}_2$. We finally recompute the estimates of the eigenvalues by computing the sample variance of the (new) scores of the observations with indices in H_3 (the observations with low OD and high SD are not included anymore). The eigenvalues are sorted in descending order, so the order of the PCs may change. The columns of the loadings and scores matrices are changed accordingly.

Note that when it is not necessary to standardize the data, we only center the data as in the scheme above, but do not scale them.

2.6 Selection of sparsity parameters

SRPCA, SCoTLASS and ROSPCA use a scalar sparsity parameter λ in the Grid algorithm. Croux et al. (2013) select λ using a BIC (Bayesian Information Criterion) type criterion. It looks at the ratio of residual variances and the degree of sparsity of the loadings matrix. These residual variances are computed by applying the Q_n^2 estimator to the sums of the squared OD statistics of the sparse and unconstrained PCA models. However, in our simulations and real data examples, this BIC approach selects λ values that are noticeably too sparse for ROSPCA, so we only use it for SRPCA. We choose λ by minimizing a BIC (Bayesian Information Criterion) type criterion based on the conventional formulation derived to use the Residual Sum of Squares (RSS). Our BIC-type criterion is:

$$\mathrm{BIC}(\lambda) = \ln\left(\frac{1}{h_1 p}\sum_{i=1}^{h_1}\mathrm{OD}^2_{(i)}(\lambda)\right) + \mathrm{df}(\lambda)\frac{\ln(h_1 p)}{h_1 p}, \tag{6}$$

where h_1 is the size of H_1, and $\mathrm{OD}_{(i)}(\lambda)$ is the ith smallest orthogonal distance for the model when using λ as the sparsity parameter. This criterion is similar to the BIC in regression, with the PCA orthogonal distances in place of the regression residuals. In ordinary regression, the residuals are univariate. Because the ODs are norms of p-dimensional vectors, we have to include p in (6). Moreover we use h_1 instead of n as this denotes the size of an outlier-free subset which does not depend on λ. After reweighting, if contamination is not high, h_1 is often close to n. Similar to Croux et al. (2013), $\mathrm{df}(\lambda)$ is taken as the number of non-zero loadings when λ is used as the sparsity parameter.

The first part of the criterion measures the quality of the fit whereas the second term penalizes for model complexity, reflecting a trade-off between accuracy and sparsity. In practice, we select λ by minimizing the BIC over the interval $[0, \lambda_{\max}]$ where $\lambda_{\max}$ gives full sparseness (exactly one non-zero loading per PC). We do this by looking at a grid of (usually equidistant) λ values over this interval.

Note that the computation of the index set H_1 in ROSPCA (step 1) does not depend on the choice of the sparsity parameter. It is therefore not necessary to run the full method each time we compute the BIC for a certain λ value. We perform the parts that are independent of λ only once and we then use this, for each value of λ we look at, as input for the parts that depend on the sparsity parameter (steps 2 and 3). This approach reduces the computation time and can lead to a considerable speed-up if many λ values need to be evaluated. This computational improvement cannot be applied to the SRPCA and SCoTLASS methods because in that case the Grid algorithm fully depends on the value of λ.

The computation time of ROSPCA is the result of its initial outlier detection part (step 1) and the remaining steps 2 and 3 to obtain sparsity. Figure 1 displays the computation times in seconds of ROSPCA (left) and SRPCA (right) for a range of values of n and p, and for $k = 2$ and $\lambda = 0$ using R 3.1.1 (R Core Team 2014) on Windows 7 (64-bit) OS with an Intel Core i7-3770 CPU @ 3.40GHz. The ROSPCA plot contains a further breakdown of computation time between the sparse and total computation times. The difference is the computation time attributable to the outlier detection step, which becomes more time consuming as n increases. Both ROSPCA and SRPCA show an increase in computation time as a function of n and p. The effect is noticeably stronger though for SRPCA, which shows much higher computation times as a function of both parameters (note the difference in the y-axis). This is primarily due to the way that the methods achieve robustness. ROSPCA performs a single outlier detection step, and then in the following steps it calculates the computationally inexpensive standard deviation for each direction in the Grid algorithm. In contrast, SRPCA relies on the comparatively slower Q_n statistic because robustness is achieved at the same time as sparsity is imposed. Note that the computation time of SRPCA is independent of the sparsity parameter λ. For ROSPCA, the computation time will decrease with λ since for higher values of λ, more variables can be excluded in the additional reweighting step which decreases the computation time of the second execution of SCoTLASS. We used $\lambda = 0$ to construct Figure 1, so computation times are lower when more sparsity is imposed using a higher value of λ.

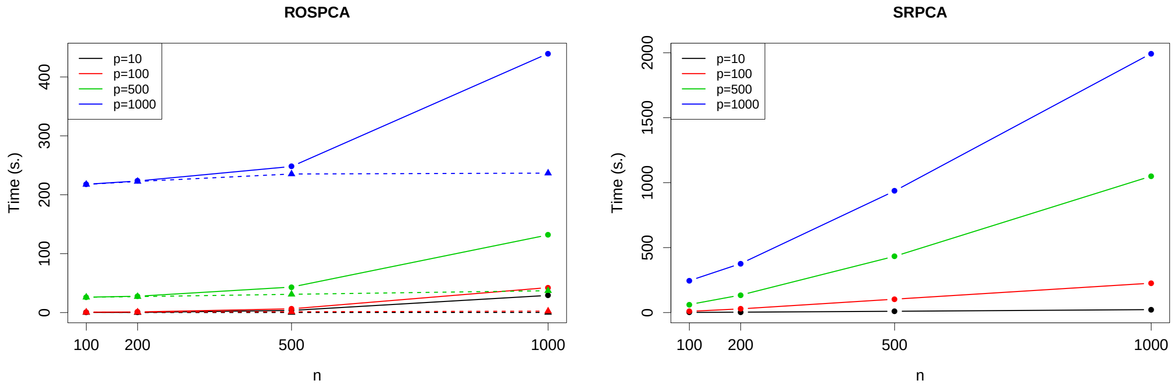


Figure 1: Computational performance of ROSPCA (left) and SRPCA (right) for varying values of n and p. The ROSPCA plot displays both the sparse (dashed line) and total (solid line) computation times.

3 SIMULATIONS

3.1 Layout of the simulation study

To evaluate the robustness, accuracy and sparsity of ROSPCA, we compare its performance with that of SRPCA, SCoTLASS, CPCA and ROBPCA on outlier-free and contaminated data. In specifying our simulations, we generate data from a multivariate normal distribution with a covariance matrix that has sparse eigenvectors. A varying proportion of the observations are replaced with outliers in order to test the robustness of the methods. We first standardize the data so that performing CPCA results in computing the eigenvectors (and -values) of the correlation matrix. Therefore, we need to generate a correlation matrix with sparse eigenvectors. First, we give a detailed description of the setup. Next, we evaluate the accuracy of the different PCA methods on the simulated data using performance measures based on the estimated loadings.

Let $\mathbb{R}^p$, with $p \geqslant 8$, be our original data space, and let $k = 2$ be the number of important components. We generate a correlation matrix such that it has sparse eigenvectors. We design the correlation matrix to have 3 groups of variables with no correlation between variables from different groups. The first two groups consist of b variables each, where b is an integer that we choose to be at least 4. The correlation between the different variables of the group is equal to $a_1 \in [-1, 1]$ for group 1 and $a_2 \in [-1, 1]$ for group 2. The third group contains the remaining $p - 2b$ variables, which we specify to be uncorrelated. Our correlation matrix $\boldsymbol{R}$ is thus equal to

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}(a_1) & \mathbf{0}_{b\times b} & \mathbf{0}_{b\times(p-2b)} \\ \mathbf{0}_{b\times b} & \boldsymbol{R}(a_2) & \mathbf{0}_{b\times(p-2b)} \\ \mathbf{0}_{(p-2b)\times b} & \mathbf{0}_{(p-2b)\times b} & \boldsymbol{I}_{p-2b} \end{pmatrix}$$

with $\boldsymbol{R}(x)$ the $b\times b$-matrix with ones on the diagonal and off-diagonal elements $x \in [-1, 1]$, and $\boldsymbol{I}_{p-2b}$ the $(p - 2b)$-dimensional identity matrix. When $a_1 > a_2$, the first two sparse eigenvectors are given by $\boldsymbol{p}_1 = -\frac{1}{\sqrt{b}}\boldsymbol{q}_1$ and $\boldsymbol{p}_2 = -\frac{1}{\sqrt{b}}\boldsymbol{q}_2$ with $\boldsymbol{q}_1 \in \mathbb{R}^p$ a vector with the first b elements equal to one and zero elsewhere, and $\boldsymbol{q}_2 \in \mathbb{R}^p$ a vector with the second b elements equal to one and zero elsewhere. The first b variables should therefore have zero loadings for the second PC, and similarly for the next b variables and the first PC. It is also clear that the variables from the last group should have zero loadings for both PCs. The order of the first two eigenvectors is changed when a_1 is smaller than a_2. The statements about the zero loadings can be adapted accordingly. Note that the eigenvectors are, neglecting their order, independent of the choice of a_1 and a_2.

Next, the correlation matrix $\boldsymbol{R}$ is transformed into the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{V}^{\frac{1}{2}}\boldsymbol{R}\boldsymbol{V}^{\frac{1}{2}}$, where $\boldsymbol{V}$ is the diagonal matrix containing the variances of the variables to be detailed later. The n observations are generated from a p-variate normal distribution with mean

$\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}$. Standard normally distributed noise terms are also added to each of the p variables to make the sparse structure of the data harder to detect. This gives a dataset $\boldsymbol{X} = \boldsymbol{X}_u + \boldsymbol{X}_{noise}$ with $\boldsymbol{X}_u \sim N_p(\mathbf{0}, \mathbf{\Sigma})$ and $\boldsymbol{X}_{noise} \sim N_p(\mathbf{0}, \boldsymbol{I}_p)$. Finally, $100\varepsilon\%$ of the data points are randomly replaced by outliers. We consider different proportions of outliers, namely $\varepsilon = 0, 0.1, 0.2, 0.3, 0.4$. These outliers are generated from a p-variate normal distribution $N_p(\boldsymbol{\mu}_{out}, \sigma^2_{out}\boldsymbol{I}_p)$ with $\boldsymbol{\mu}_{out} = 25(0, -4, 4, 2, 0, 4, -4, 2, 3, -3, \ldots, 3, -3)'$ and $\sigma^2_{out} = 20$, as in Croux et al. (2013). Importantly, these outliers do not follow the correlation structure determined by $\boldsymbol{R}$. They will therefore bias non-robust sparse methods trying to estimate the sparse structure. We also denote the dataset with the outliers by $\boldsymbol{X}$.

First, we consider a low-dimensional setting with $p = 10$ dimensions and $b = 4$ in our simulations, so we have two blocks of four useful variables and the last two variables are noise. We take $a_1 = 0.9$ and $a_2 = 0.5 < a_1$ which gives eigenvalues $3.7, 2.5, 1, 1, 0.5, 0.5, 0.5, 0.1, 0.1, 0.1$ and the first two eigenvectors of $\boldsymbol{R}$ are given by $\boldsymbol{p}_1 = -\frac{1}{2}(1, 1, 1, 1, 0, 0, 0, 0, 0, 0)'$ and $\boldsymbol{p}_2 = -\frac{1}{2}(0, 0, 0, 0, 1, 1, 1, 1, 0, 0)'$. Importantly, the difference between the first and second eigenvalue is large enough such that the methods can clearly determine that $\boldsymbol{p}_1$ is the loading vector of the first PC. When taking a_1 and a_2 closer together, the difference between the first two eigenvalues gets smaller, so it becomes more difficult for the PCA method to identify which of the first two eigenvectors corresponds to the first PC. We also need to make sure that a_2 is large enough, otherwise the difference between the second and third eigenvalue is too small. This can again cause problems because the PCA method can sometimes select the third eigenvector as the loading vector corresponding to the second PC, making our bias criterion become difficult to interpret. With our choices for a_1 and a_2, the difference between the eigenvalues is large enough to avoid these problems. We take $\boldsymbol{V} = \text{diag}(100, \ldots, 100, 25, \ldots, 25, 4, 4)$, so the variables in a group have the same variance. For each simulated scenario, we generate 500 datasets following the above scheme to thoroughly characterize the behavior of the methods.

Figure 2 shows a heat map of the absolute values of one dataset from our simulation setting with $p = 10$, $n = 100$ and $\varepsilon = 0.2$. The outliers are visible as the observations with values taking a dark blue color. Despite being fairly easy to identify on a heat map, we shall see that these can pose difficulties for sparse PCA methods that are not highly robust. We note that the configurations we use to evaluate the methods considered in the paper are known to be particularly challenging for them, while they are capable of easily identifying outliers in other configurations that are not clearly revealed by a heat map.

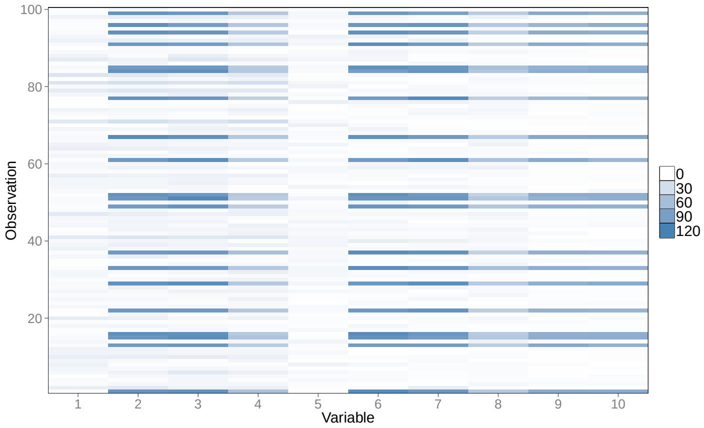


Figure 2: Heat map of absolute value of simulated data with $p = 10$, $n = 100$ and $\varepsilon = 0.2$. Outliers are visible in dark blue.

We also look at a high-dimensional setting with $p = 500$ and $k = 2$. In contrast to the low-dimensional setting, the first two groups consist of $b = 20$ variables each, which results in 40 useful variables and 460 noise variables. In the new setting, the eigenvalues are $18.1, 10.5$, 1 (460 times), 0.5 (19 times) and 0.1 (19 times), where we take $a_1 = 0.9$ and $a_2 = 0.5 < a_1$ again. The first two sparse eigenvectors are given by $\boldsymbol{p}_1 = -\frac{1}{\sqrt{20}}\boldsymbol{q}_1$ and $\boldsymbol{p}_2 = -\frac{1}{\sqrt{20}}\boldsymbol{q}_2$ with $\boldsymbol{q}_1 \in \mathbb{R}^{500}$ a vector with the first 20 elements equal to one and zero elsewhere, and $\boldsymbol{q}_2 \in \mathbb{R}^{500}$ a vector with the second 20 elements equal to one and zero elsewhere. We use the same variances for the groups as before: 100 for group 1, 25 for group 2 and 4 for group 3. For each scenario, we now generate 100 datasets following the high-dimensional scheme to keep computations reasonable.

To compare the robustness of the methods, we look at the 2nd principal angle between the subspace spanned by the two dominant eigenvectors of the correlation matrix $\boldsymbol{R}$

and the subspace spanned by the columns of the estimated loadings matrix (the PCA subspace), as was also done in Hubert et al. (2005) and Todorov and Filzmoser (2013). We compute this angle using the algorithm of Björck and Golub (1973). This angle lies between 0 and $\frac{\pi}{2}$, and we divide it by $\frac{\pi}{2}$ to get values between 0 and 1. In the remainder we will refer to the standardized version as the "angle". It is clear that we want values close to 0.

All simulations were performed in R 3.1.1 using following functions: `prcomp` (CPCA), `PcaHubert` (ROBPCA) from the *rrcov* package (Todorov and Filzmoser 2009) and `SPcaGrid` (SRPCA and SCoTLASS) from *rrcovHD* (Todorov 2014). We used self-written functions for ROSPCA based on the code for `PcaHubert`. For ROSPCA and ROBPCA the parameter α is set to 0.5, yielding maximal robustness. First, we compare the estimation of the PCA subspace and the degree of sparsity attained. Then, we discuss the behavior of the λ selection step of these algorithms following our BIC criterion (6) for ROSPCA and SCoTLASS, and the BIC criterion of Croux et al. (2013) for SRPCA.

3.2 Results of the simulation study

3.2.1 Subspace estimation

We start with the low-dimensional simulations ($p = 10$). For each simulation setting and each sparse method we report two results as boxplots. On the left is a boxplot of the angle values corresponding to a model fitted by a method with λ selected using the previously discussed criteria. We consider following grid of λ values: $\{0, 0.02, \ldots, 2.5\}$. The boxplot on the right is based on the minimal angle value attained by each method over the same range of λ values. These results provide two insights. First, the boxplot based on the minimal angle values gives a sense of the performance of each method if λ were selected to give the fit closest to the real structure of the data possible for that method. Secondly, this boxplot and the boxplot to its left, based on results from models using λ values selected by a criterion, together give a sense of how successful the information criterion is in selecting an optimal value of λ for the method. For CPCA and ROBPCA, we only have the boxplot of the angle values corresponding to the fitted model.

Figures 3, 4 and 5 show boxplots on datasets of increasing size n and contamination rate ε. Mean values are indicated with blue diamonds. As expected, bias decreases and the angles become less dispersed when n increases. SCoTLASS reports the best results for $\varepsilon = 0$ but performs very badly when contamination is present. Also of note, the boxplots corresponding to models based on selected λ values are only slightly higher than the boxplots based on the minimal angle values, showing that the λ selection problem is tractable for SCoTLASS under these settings. Over all contamination levels, ROSPCA shows a low mean and median bias, even for the case where $\varepsilon = 0.4$. Like SCoTLASS when it is applied to uncontaminated data, the boxplots based on selected λ values and the minimal angles tend to be close, meaning that for ROSPCA, λ is typically selected accurately. At small sample sizes, quite some variability is still present in the estimates, but this decreases substantially at larger sample sizes. In contrast to ROSPCA, SRPCA returns distinctly higher biases, even for the best possible λ value. Its bias at outlier-free data only becomes reasonably small when n is very large. Furthermore, the difference in the boxplot pairs for SRPCA reveals that the BIC selection criterion proposed by Croux et al. (2013) yields angles that are on average quite distinct from the optimal ones that could be obtained. CPCA is outperformed by the sparse methods SCoTLASS and ROSPCA at outlier-free data, and completely breaks down at contaminated ones. ROBPCA shows an increased bias when contamination is present. A closer look at the results revealed that the method did correctly identify the outliers, but it was not able to discover the sparse structure of the data as well as ROSPCA does.

Consider now the high-dimensional simulations where $p = 500$. We now consider the following grid of λ values: $\{0, 0.02, \ldots, 1.2\}$. For SRPCA with $n = 500$, we decreased the grid with λ values up to 0.6 instead of 1.2 to keep computations reasonable. Figures 6, 7 and 8 show the results for several sample sizes. As before, the bias and the dispersion of the angle becomes smaller when the sample size n increases. On uncontaminated data, the selection of λ is not successful for SCoTLASS (the BIC from Croux et al. (2013)

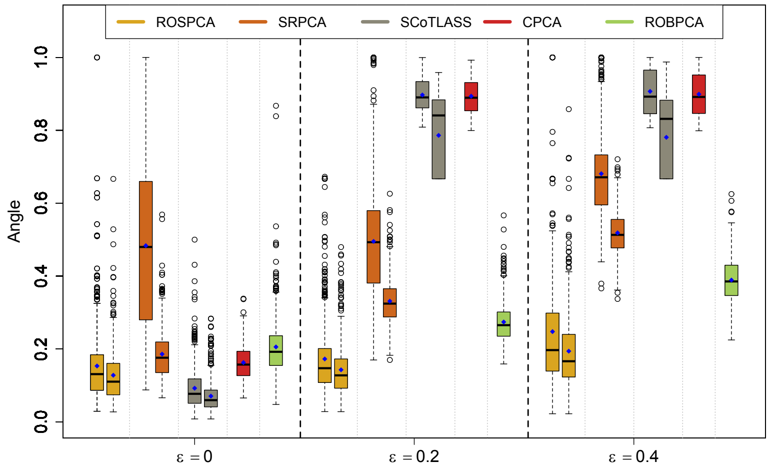


Figure 3: Angle values of ROSPCA, SRPCA, SCoTLASS, CPCA and ROBPCA at $p = 10$ and $\varepsilon = \{0, 0.2, 0.4\}$ for $n = 50$.

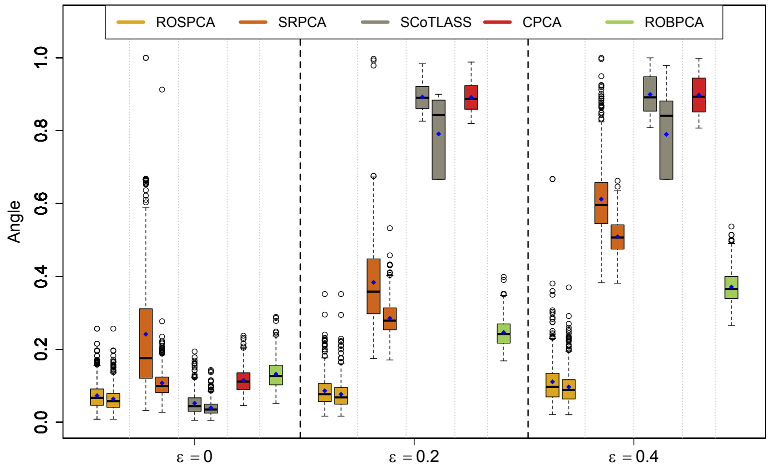


Figure 4: Angle values of ROSPCA, SRPCA, SCoTLASS, CPCA and ROBPCA at $p = 10$ and $\varepsilon = \{0, 0.2, 0.4\}$ for $n = 100$.

returns even slightly worse results). However, the minimum angle boxplot shows that SCoTLASS can perform well, and ROSPCA attains similar performance to SCoTLASS's optimal performance in both boxplots. SRPCA shows very poor performance even when outliers are not present when λ is selected, and has worse results for the minimal angle values as well, indicating that intrinsically it may not be as accurate as SCoTLASS or ROSPCA. CPCA and ROBPCA have a comparable behavior, which is inferior to the

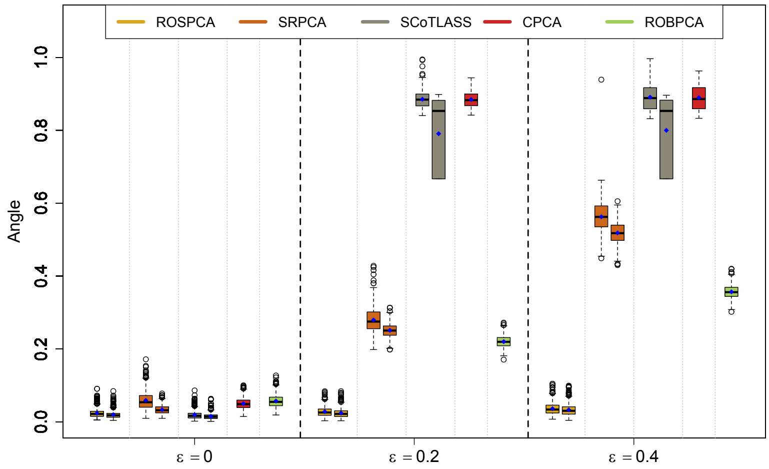


Figure 5: Angle values of ROSPCA, SRPCA, SCoTLASS, CPCA and ROBPCA at $p = 10$ and $\varepsilon = \{0, 0.2, 0.4\}$ for $n = 500$.

sparse methods. When contamination is introduced, SCoTLASS performs very poorly, as expected, while the optimal performance of SRPCA and ROSPCA is only slightly worse than when the data is not contaminated, and ROSPCA continues to show successful λ selection. When $\varepsilon = 0.4$, SRPCA does however show higher bias than for lower ε, unlike ROSPCA. CPCA is no longer reliable, whereas the performance of ROBPCA remains stable.

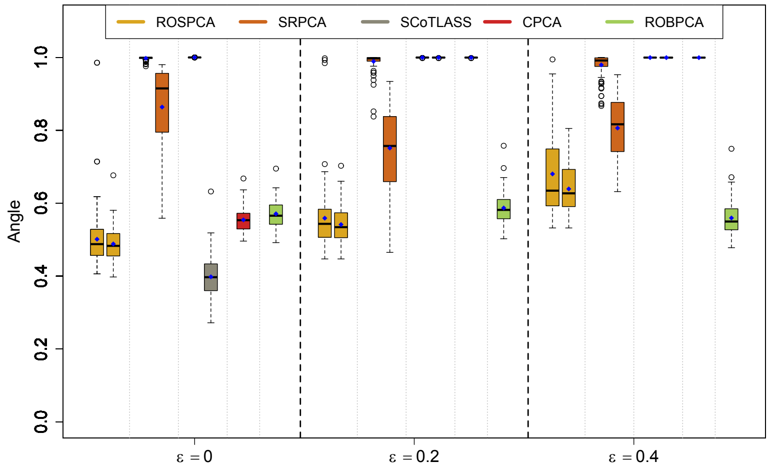


Figure 6: Angle values of ROSPCA, SRPCA, SCoTLASS, CPCA and ROBPCA at $p = 500$ and $\varepsilon = \{0, 0.2, 0.4\}$ for $n = 50$.

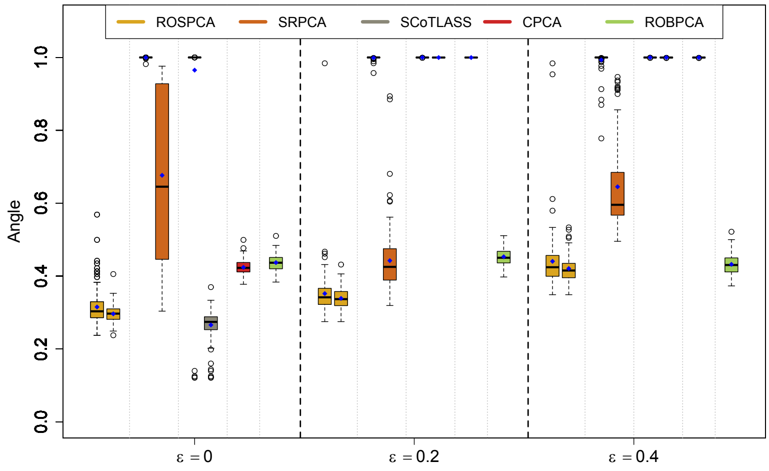


Figure 7: Angle values of ROSPCA, SRPCA, SCoTLASS, CPCA and ROBPCA at $p = 500$ and $\varepsilon = \{0, 0.2, 0.4\}$ for $n = 100$.

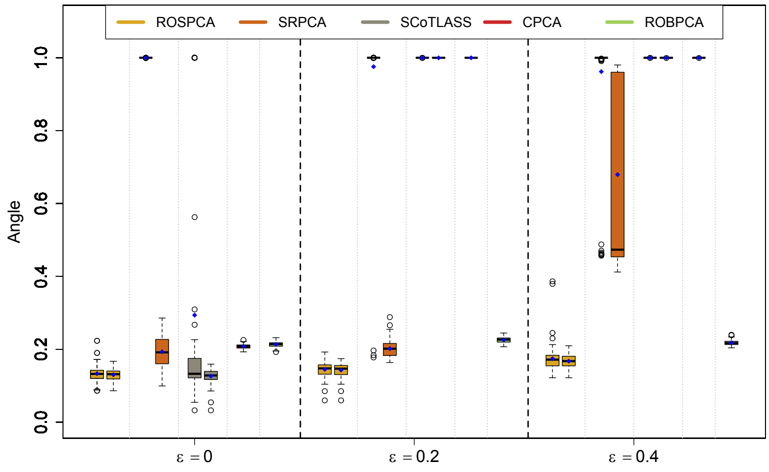
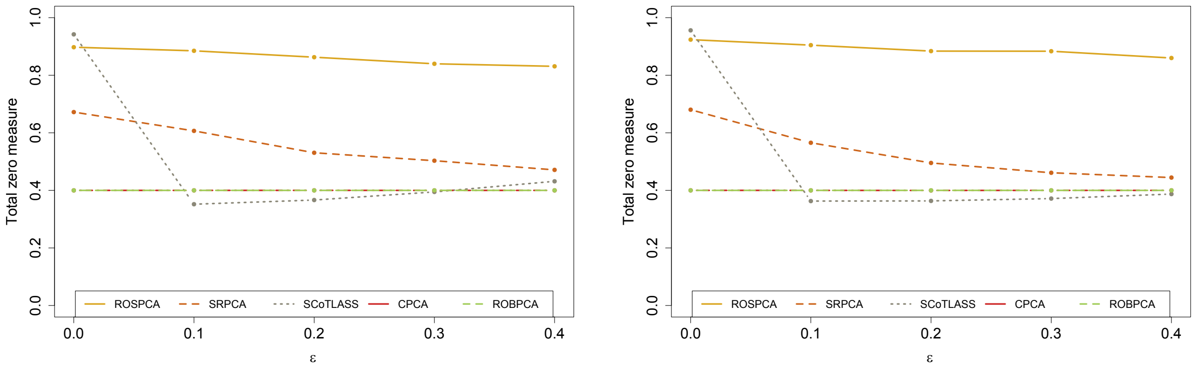


Figure 8: Angle values of ROSPCA, SRPCA, SCoTLASS, CPCA and ROBPCA at $p = 500$ and $\varepsilon = \{0, 0.2, 0.4\}$ for $n = 500$.

3.2.2 Sparsity

In addition to estimating a model that is not influenced by outliers, it is also important to estimate the correct sparsity. The *zero measure* is one way to compare how correctly each of the methods estimates the sparse $\boldsymbol{P}$. For each element of $\boldsymbol{P}$, it is equal to one if the estimated and true value are both zero or both non-zero, and 0 otherwise. We then take the average zero measure over all elements of $\boldsymbol{P}$ and all 500 simulations which we call the *total zero measure*. We need to specify when an element is “equal to zero” because it can be that an element of $\boldsymbol{P}$ is very small but different from zero. We say that all elements with an absolute value smaller than 10^{-5} are “equal to zero”.

In Figures 9a and 9b, we see that ROSPCA accurately discerns the sparse structure of $\boldsymbol{P}$, even when $n = 50$ and $\varepsilon = 0.4$. SRPCA steadily demonstrates weaker performance as ε increases, whereas SCoTLASS performs well for $\varepsilon = 0$, and uniformly poorly for higher values of ε. The zero measure plots for larger sample sizes are very similar to the plot for $n = 100$. These results show that ROSPCA not only gives robust PCA estimates but is also better at detecting the sparse structure of the data. CPCA and ROBPCA hardly yield zero loading elements, so their zero measure is almost constantly equal to 40%, which is the percentage of non-zero entries in $\boldsymbol{P}$.


(a) $n = 50$

(b) $n = 100$

Figure 9: Total zero measure of ROSPCA, SRPCA, SCoTLASS, CPCA and ROBPCA for (a) $n = 50$ and (b) $n = 100$.

The zero measure is less useful in the high-dimensional setting because perfect sparsity for all zero loadings is more difficult to achieve. This results in zero measures that are comparatively more difficult to interpret than those shown in Figures 9a and 9b, since two methods may appear to give similar results by this measure, while a close inspection of the loadings reveals substantial differences.

3.2.3 The λ selection performance of ROSPCA, SRPCA and SCoTLASS

As explained in Section 2.6, we use the BIC-type criterion (6) to select the sparsity parameter λ of ROSPCA and SCoTLASS (since no criterion is proposed in Jolliffe et al. (2003)). For SRPCA, we use the BIC proposed by Croux et al. (2013). We looked at 101 (equidistant) values of λ over the interval in which complete sparsity is attained: $[0, 2.5]$, i.e. $\{0, 0.02, \ldots, 2.48, 2.5\}$. To provide insight into the role of robustness in this process, we introduce $\varepsilon = 20\%$ contamination. In Figure 10, we display the quantile plots of the angle values obtained by these methods over the 500 simulated datasets for $n = 100$ and $\varepsilon = 0.2$ as a function of λ. It depicts the median (solid lines) and first (dotted lines) and third quartile (dashed lines) of angle values for a given λ value over the 500 simulations.

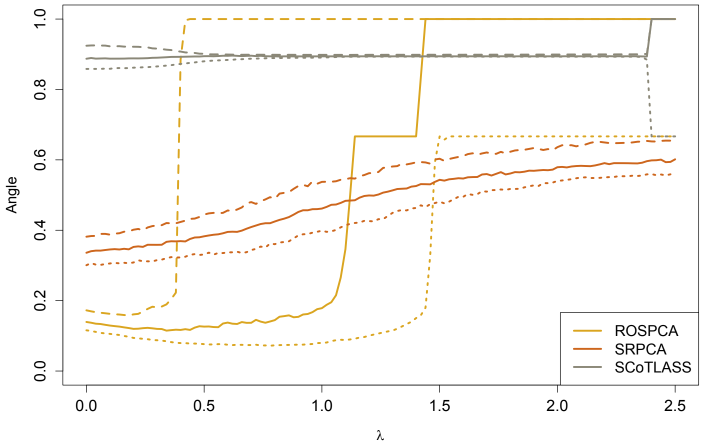


Figure 10: Quantile plots of the angle values for ROSPCA, SRPCA and SCoTLASS as a function of λ.

Examining the angle values corresponding to fits for each of the methods using different values of λ reveals a pattern correlated with the robustness of the methods. The angle values for SCoTLASS, tend to be fairly constant and high across the range of λ. This reflects the fact that the models are all influenced by outliers, and in comparison the sparsity of the model has very little impact of the angle. The quantile plot for SRPCA is not as flat as that of SCoTLASS and is considerably lower, but shows a steadily increasing angle value as λ is increased. Since this method is robust, it can attain decent fits with non-sparse models, but including sparsity makes it vulnerable to missing the outliers and finding a worse fit. This has the consequence that even though the true data is sparse, a full SRPCA model attains the lowest angle value since it allows for the most accurate outlier screening.

The quantile plot for SRPCA illustrates a trade-off between robustness and sparseness, where we find that contamination due to outliers tends to dominate the inaccuracy due to using a non-sparse model on sparse data (which is why the full SRPCA model has the lowest angle). The ROSPCA quantile plot shows that it is possible to account for both the sparse structure of the data and the outliers. For ROSPCA, the lowest value of λ (0 in our case) does not correspond to the lowest angle value. Rather, this is achieved by a sparse model, as we would expect. This is possible because ROSPCA has initially separated the outlier detection and sparsity steps before combining insights from both to return the final model. The first and third quantiles show that there is some variation in the angle values returned by ROSPCA for different values of λ, but the figures in Section 3.2.1 show that the value of λ selected by the BIC criterion is consistently close to the value of λ returning the minimal angle for each simulation.

4 REAL DATA EXAMPLE

In this section we illustrate the behavior of ROSPCA and SRPCA on the glass dataset introduced in Hubert et al. (2005). It consists of Electron Probe X-ray Microanalysis (EPXMA) spectra over $p = 750$ wavelengths and 180 collected glass samples (Lemberge et al. 2000). Although the non-sparse ROBPCA performs well on this dataset, employing a sparse method may be interesting because when one consults the full loadings, the data actually appears to have a sparse structure. Figure 11 shows a heatmap of the absolute values of the centered data matrix where we used the componentwise median. We only plotted the wavelengths with numbers 120-400 because the rest of them are mostly non-informative (due to the sparse structure of the data). As noted in Hubert et al. (2005), two groups of outliers can be clearly identified in this dataset: the last 38 observations that were measured after the spectrometer was cleaned and calcium outliers with high values for two groups of wavelengths between 300 and 370.

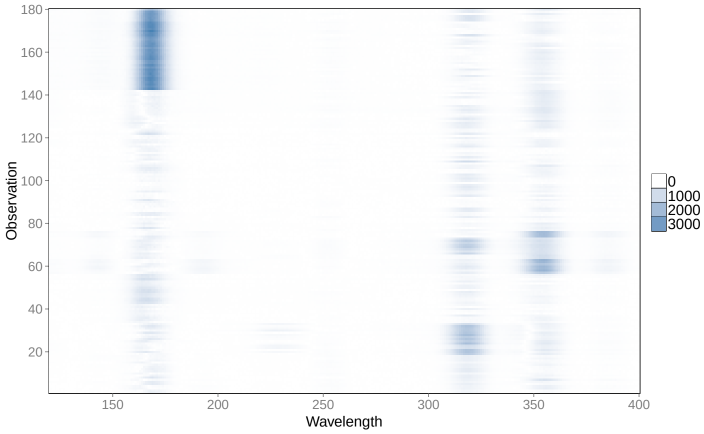


Figure 11: Heat map of the Glass data.

With a sparse robust PCA analysis we hope to achieve outlier detection results comparable to ROBPCA while also obtaining sparse loadings that reflect the atomic structure of the glass samples. We do not standardize the data because all variables are expressed in the same units. The non-robustness of SCoTLASS means that it cannot reliably address the outliers present, so results are omitted.

Selecting the number of components to use in a sparse PCA model is more complicated than in classical PCA due to the inclusion of λ, which varies with k, but must also be selected. In Jolliffe et al. (2003), rather than providing a criterion for selecting k that accounts for sparsity, the authors apply the cumulative percent variation (CPV) criterion to a non-sparse PCA model. Then, they discuss the influence of a range of λ values over a model using that particular value of k. In Croux et al. (2013), the authors fit a robust, non-sparse PCA model with many components and then use those eigenvalues to select k for the sparse, robust model. Similarly, we use the eigenvalues of the robust, non-sparse PCA model described in Step 1 of ROSPCA. Since the SVD is computed on uncontaminated observations, we obtain eigenvalues for all possible $\min(p, n-1)$ components. We use the scree plot corresponding to these eigenvalues to select the number of components to retain, but automatic criteria such as the CPV can also be used.

The scree plot for ROSPCA (Figure 12) indicates that three or four components are sufficient to model the data well, and we select four components. Additionally we set the parameter $\alpha = 0.5$ to obtain maximal robustness. Hence $h_0 = \lceil 0.5 \times 180 \rceil + 1 = 91$. We also select $k = 4$ for SRPCA after consulting the scree plot for SRPCA with $\lambda = 0$.

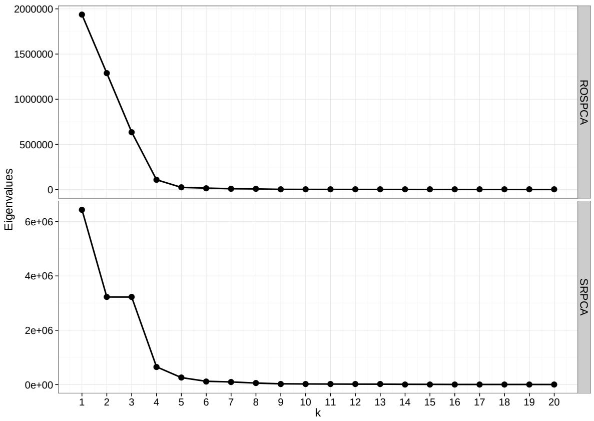


Figure 12: Scree plots for ROSPCA and SRPCA ($\lambda = 0$).

Next, we perform the λ selection step for ROSPCA using our proposed BIC, and for SRPCA using the BIC of Croux et al. (2013). This yields λ values of 0.96 and 72.7, respectively. The running time for ROSPCA using $\lambda = 0.96$ was 146s, whereas SRPCA had a running time of 419s. For comparison we also include the ROBPCA results. As its scree plot is identical to that of ROSPCA (since the singular values are computed on the same subset of observations), we also use $k = 4$ components.

From the fitted models we can produce outlier maps showing the score distance and orthogonal distance of the observations in the dataset. We normalize these diagnostic plots by dividing each of the distances by its cut-off to make the results visually comparable across methods. This gives us Figure 13. All three methods indicate the post-cleaning observations (orange) as bad leverage points, but SRPCA does not show the same discriminatory power as ROSPCA and ROBPCA. These two methods also clearly find several other orthogonal outliers and bad leverage points. This is useful for the practitioner because it provides a clear message that these observations warrant further investigation. Ignoring the boundary cases, we have indicated this set of outliers, as detected by ROSPCA, as open blue circles. Obviously ROBPCA identifies these outliers as well, but SRPCA rather declares them as ambiguous border cases with only larger score distances. Next, we compared the heatmap of the data in Figure 11 with these outlier maps, and noticed that almost all open blue circles correspond to calcium outliers which were highlighted on the heatmap. The three open blue circles that are close to the cut-off line for the score distances on the diagnostic plot of ROSPCA are however not clearly visible on the heatmap. Only a closer inspection of the raw data revealed that they are outlying on variables 215–245. Our robust multivariate analysis was able to detect this abnormal behavior at once.

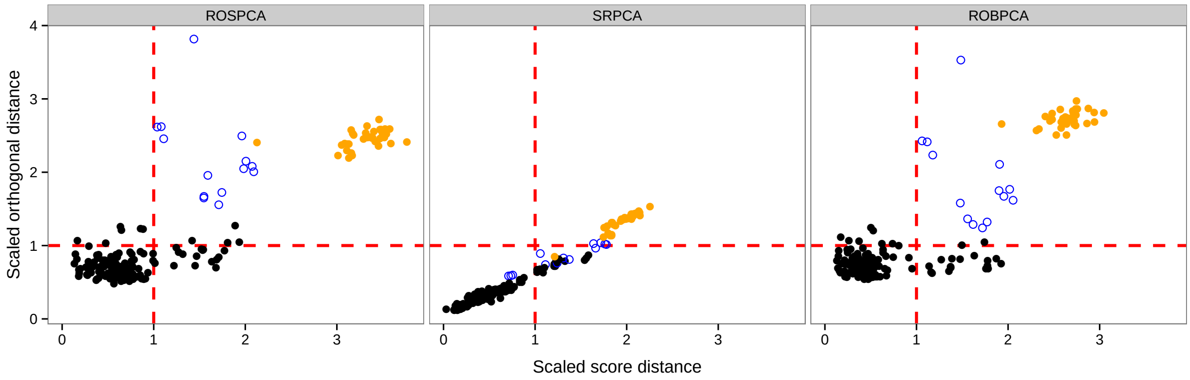


Figure 13: Scaled outlier maps of ROSPCA (with $\lambda = 0.96$), SRPCA ($\lambda = 72.7$) and ROBPCA on the glass data. The orange points correspond to the measurements after the window has been cleaned. The open blue circles correspond to the other outliers identified by ROSPCA.

To study the sparsity, we plot the loadings of each of the methods in Figure 14 and tabulate the sparsity of each in Table 1. Unsurprisingly, ROBPCA produces the least sparse loadings, with only 13 variables with all loadings less than the threshold of 10^{-5}. Nonetheless, the loadings are instructive as they give a sense of the full structure of the data and where sparsity might be obtained. Specifically, three groups of wavelengths (155–185, 310–335, 336–370) are particularly relevant. SRPCA attains the greatest sparsity, but given the poor outlier detection performance, it is likely that as we saw in the simulation studies, the λ selection procedure has been influenced by contamination. The sensitivity of the λ selection step to outliers underscores the need for a highly robust method. ROSPCA obtains loadings similar to those of ROBPCA, but with the important distinction that loadings ROBPCA assigned small values to are now assigned no weight, resulting in 200 excluded variables. This increases the interpretability of the resulting model, while retaining accuracy. We note that a practitioner may choose a larger λ in an ad hoc way to further increase the sparsity of ROSPCA and that for a value of λ giving similar sparsity to that of SRPCA, ROSPCA still identifies the outliers correctly.

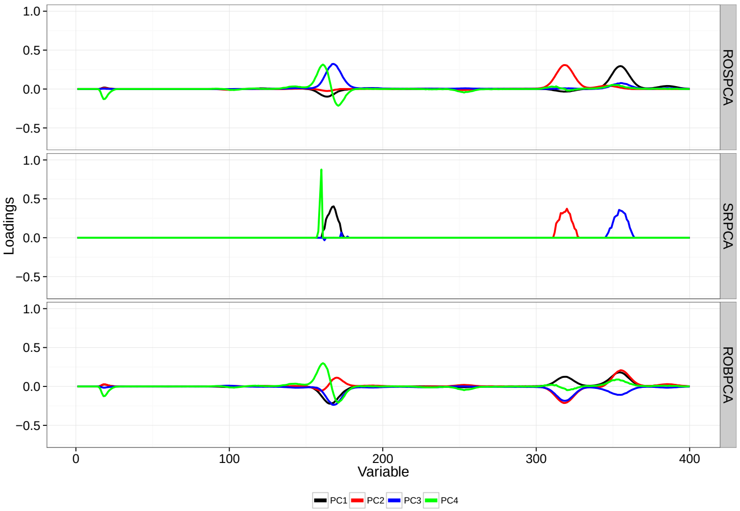


Figure 14: Loadings of ROSPCA (with $\lambda = 0.96$), SRPCA ($\lambda = 72.7$) and ROBPCA on the glass data. Loadings on wavelengths with indices above 400 were small for all methods and are excluded from the plot.

Table 1: Number of non-zero loadings (larger than 10^{-5}) for each method per PC. The bottom row is the number of variables that have zero loadings (smaller than 10^{-5}) on all 4 PCs.

	ROSPCA	SRPCA	ROBPCA
PC1	359	14	733
PC2	272	17	735
PC3	491	34	737
PC4	408	4	736
No. of excluded variables	200	696	13

Finally, we also compare the obtained loadings using the angle measure, results are shown in Table 2. We see that the ROSPCA and ROBPCA subspaces are similar and that the SRPCA subspace differs a lot from the other two subspaces. One could also visually deduce these conclusions from inspecting Figure 14.

The results for the glass dataset reinforce our findings from the simulations. Since the outliers are in two groups, we find that SRPCA does well at detecting the more obvious post-cleaning ones, but struggles to find the more nuanced calcium outliers. As in the

Table 2: Angle between the obtained loadings for the Glass data using ROSPCA, ROBPCA and SRPCA.

	ROSPCA- ROBPCA	ROBPCA- SRPCA	SRPCA- ROSPCA
Angle	0.040	0.731	0.725

simulations, ROSPCA both detects the outliers accurately and finds a plausible sparse structure.

5 CONCLUSIONS

We have detailed a new approach for sparse, robust Principal Component Analysis, ROSPCA, that is a modification of ROBPCA. Unlike existing methods for sparse PCA, ROSPCA prioritizes the detection of the outliers rather than giving robustness and sparsity equal weight. Our results indicate that this approach is warranted. We observe that by first detecting and neutralizing the outliers, ROSPCA is able to fit the sparse structure of the majority of the data with high accuracy. In comparisons with existing methods, we find that ROSPCA consistently obtains the best performance.

In addition to good robustness and sparsity properties, ROSPCA is also computationally faster. One of the most important steps in performing a robust PCA analysis is the selection of the λ parameter. A single execution of ROSPCA is faster than one of SRPCA, but this advantage is compounded when selecting λ since the robustness step only needs to be performed once.

This work opens the door to the development of sparse robust methods for high-dimensional data, such as sparse robust discriminant analysis, sparse partial least squares regression, and for skew-adjusted sparse PCA. Extensions of the ROBPCA based methods, as in Vanden Branden and Hubert (2005), Hubert and Vanden Branden (2003) and Hubert et al. (2009) will be studied. A theoretical study of the influence function of ROSPCA, extending the results of Debruyne and Hubert (2009), is also an interesting challenge for future research.

ACKNOWLEDGEMENT

Mia Hubert acknowledges the financial support of KU Leuven grant GOA/12/14 and of the IAP Research Network P7/06 of the Belgian Science Policy.

REFERENCES

Björck, Å. and Golub, G. H. (1973), "Numerical Methods for Computing Angles Between Linear Subspaces," *Mathematics of Computation*, 27, 579–594.

Cadima, J. and Jolliffe, I. T. (1995), "Loadings and Correlations in the Interpretation of Principal Components," *Journal of Applied Statistics*, 22, 203–214.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011), "Robust Principal Component Analysis?," *Journal of the ACM*, 58, 1–37.

Croux, C. and Ruiz-Gazen, A. (2005), "High Breakdown Estimators for Principal Components: the Projection-Pursuit Approach Revisited," *Journal of Multivariate Analysis*, 95, 206–226.

Croux, C., Filzmoser, P., and Oliveira, M. R. (2007), "Algorithms for Projection-Pursuit Robust Principal Component Analysis," *Chemometrics and Intelligent Laboratory Systems*, 87, 218–225.

Croux, C., Filzmoser, P., and Fritz, H. (2013), "Robust Sparse Principal Component Analysis," *Technometrics*, 55, 202–214.

Debruyne, M. and Hubert, M. (2009), "The influence function of the Stahel-Donoho covariance estimator of smallest outlyingness," *Statistics & Probability Letters*, 79, 275–282.

Engelen, S., Hubert, M., and Vanden Branden, K. (2005), “A Comparison of Three Procedures for Robust PCA in High Dimensions,” *Austrian Journal of Statistics*, 34, 117–126.

Filzmoser, P., Fritz, H., and Kalcher, K. (2014), *pcaPP: Robust PCA by Projection Pursuit.* URL: `http://CRAN.R-project.org/package=pcaPP`, version 1.9-50.

Hubert, M. and Vanden Branden, K. (2003), “Robust Methods for Partial Least Squares Regression,” *Journal of Chemometrics*, 17, 537–549.

Hubert, M., Rousseeuw, P. J., and Verboven, S. (2002), “A Fast Method for Robust Principal Components With Applications to Chemometrics,” *Chemometrics and Intelligent Laboratory Systems*, 60, 101–111.

Hubert, M., Rousseeuw, P. J., and Vanden Branden, K. (2005), “ROBPCA: A New Approach to Robust Principal Component Analysis,” *Technometrics*, 47, 64–79.

Hubert, M., Rousseeuw, P. J., and Verdonck, T. (2009), “Robust PCA for Skewed Data and Its Outlier Map,” *Computational Statistics & Data Analysis*, 53, 2264 –2274.

Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003), “A Modified Principal Component Technique Based on the LASSO,” *Journal of Computational and Graphical Statistics*, 12, 531–547.

Lemberge, P., De Raedt, I., Janssens, K. H., Wei, F., and Van Espen, P. J. (2000), “Quantitative Z-Analysis of the 1617th Century Archaelogical Glass Vessels using PLS Regression of EPXMA and μ-XRF Data,” *Journal of Chemometrics*, 14, 751–763.

Li, G. and Chen, Z. (1985), “Projection-Pursuit Approach to Robust Dispersion Matrices and Principal Components: Primary Theory and Monte Carlo,” *Journal of the American Statistical Association*, 80, 759–766.

Locantore, N., Marron, J. S., Simpson, D. G., Tripoli, N., Zhang, J. T., and Cohen, K. L. (1999), “Robust Principal Component Analysis for Functional Data,” *Test*, 8, 1–73.

R Core Team (2014), *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. URL: `http://www.R-project.org/`.

Rousseeuw, P. J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871–880.

Rousseeuw, P. J. and Croux, C. (1993), "Alternatives to the Median Absolute Deviation," *Journal of the American Statistical Association*, 88, 1273–1283.

Todorov, V. (2014), *rrcovHD: Robust Multivariate Methods for High Dimensional Data.* URL: `http://CRAN.R-project.org/package=rrcovHD`, version 0.2-3.

Todorov, V. and Filzmoser, P. (2009), "An Object-Oriented Framework for Robust Multivariate Analysis," *Journal of Statistical Software*, 32, 1–47.

Todorov, V. and Filzmoser, P. (2013), "Comparing Classical and Robust Sparse PCA" in *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, (eds.) R. Kruse, M. R. Berthold, C. Moewes, M. Á. Gil, P. Grzegorzewski, and O. Hryniewicz, Advances in Intelligent Systems and Computing, 190, pp. 283–291.

Vanden Branden, K. and Hubert, M. (2005), "Robust Classification in High Dimensions Based on the SIMCA Method," *Chemometrics and Intelligent Laboratory Systems*, 79, 10–21.

Zhou, Z., Li, X., Wright, J., Candès, E. J., and Ma, Y. (2010), "Stable Principal Component Pursuit," *Proceedings of International Symposium on Information Theory.*

Zou, H., Hastie, T., and Tibshirani, R. (2006), "Sparse Principal Component Analysis," *Journal of Computational and Graphical Statistics*, 15, 265–286.