Project of Robust Statistics

"Fast cross-validation of high-breakdown resampling methods for PCA"

19 April 2022

Anna Gotti r0875721
Dylan John r0873752
Gunho Lee r0736720
Luka Beverin r0819676

1 Introduction

The following report conducts critical research on the content of the article *"Fast cross-validation of high-breakdown resampling methods for pca"* [Hubert and Engelen, 2007]. The article introduces fast and robust new algorithms to perform the cross-validation (CV) Prediction Error Sum of Squares (PRESS) for Robust Principal Component Analysis (ROBPCA) to cope with the computation complexity derived from the high-breakdown methods for high-dimensional data. The introduced algorithms visualize a robust PRESS curve helping select the principal components that have to be retained, which is the major question in PCA. This report aims at breaking down the aforementioned concepts in detail and describing the advantages of the Robust R-PRESS for the component selection in PCA. Further, we will demonstrate the proposed algorithms on high-dimensional data to exhibit that the computation time is significantly reduced.

1.1 Leave-One-Out-Cross-Validation (LOOCV)

The predictive ability of a fitted model can be tested by its performance on a validation set. If large-sized data is available, you may split the data into two of which one is used as the validation set. However, splitting data contains several disadvantages that could raise doubts about model validity. First of all, data-splitting is not a feasible option when the sample size is limited. Furthermore, the homogeneity of the two sets (training and validation) has to be investigated [Hubert et al., 2005a]. As a result, it could increase the bias of the outcomes and be sensitive to outlying observations since a random split might group every anomaly into one single set.

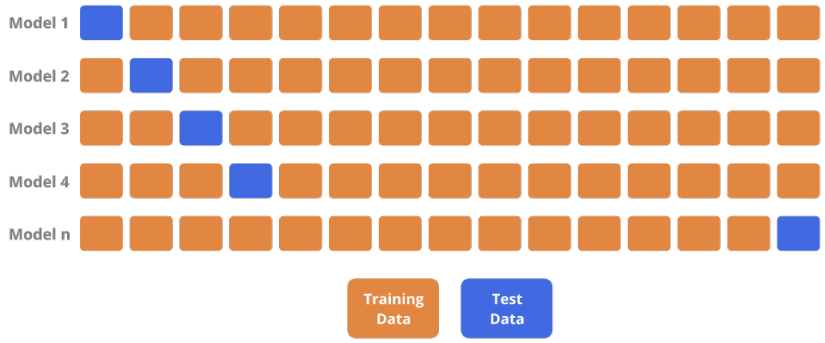


Figure 1: A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (orange) containing all but one observation, and a test set that contains only that observation (blue) [James et al., 2013]

To overcome the described difficulties, a cross-validation (CV) method is recommended when a dataset $\boldsymbol{X}$ contains small sample sizes. In the perspective of *robustness*, the most preferred method is Leave-One-Out Cross-Validation (LOOCV) using each i observation from the set of n data points as the validation set and the remaining observations as the training set. This algorithm has more robust properties than Leave-P-Out Cross-Validation due to its zero randomness. However, the downside of LOOCV is that the procedure is computationally expensive coming from the fact that each observation has to be tested. This issue is intensified when high-breakdown resampling algorithms are utilized. In later sections, we will revisit it together with the fast and robust algorithms to mitigate the problem.

1.2 Principal Component (PC) selections in PCA

PCA is one of the major dimensionality-reduction techniques that is extensively used in a variety of fields. From a statistical point of view, parsimonious models are preferred to ones containing large amounts of variables. Therefore, the objective of PCA is to extract Principal Component (PC) by compressing the original variables based on their similarities (patterns). It leads to a question that how many PCs need to be retained to account for most of the variance in your data without losing critical information. Heretofore, numerous mathematical and graphical methods have been developed to solve the puzzle, but it remains debatable which technique is *ultimately* better than another [Jolliffe, 2002].

The scree plot, also known as the elbow method, visualizes the calculated eigenvalues after eigen-decomposition on the principal components in a descending order, and searches for a drastically curving point (elbow) which decides the maximum number of retained components [Cartell, 1966]. However, selecting a *clear* elbow sounds too abstract and does not provide comprehensible but rather ambiguous evidence [Ledesma et al., 2015]. Another widely-known criteria to choose m PCs is to see a percentage of the total variance explained by the chosen m PCs. The popular rule-of-thumb here is the Kaiser's rule, suggesting that the PCs whose eigenvalues exceed 1 should be retained. The assumption behind the rule is that a component (or $factor$ in Factor Analysis) explaining less *variance* than an original variable is not worth retaining [Kaiser, 1960]. However, this intuitive rule has also been criticized since the *strict* cut-off rule implies, for instance, α component with 1.01 variance would be retained while β with 0.99 variance would not [Kaufman and Dunlap, 2000]. This grows uncertainty towards the $self\text{-}defined$ percentage of total variance (good amounts of variance in other words) [Jolliffe, 2002], and such a fuzzy procedure (e.g: we choose the first two PCs because their variances are larger than 1.) does not seem to be a confident decision rule. Further, classical PCA is based on classical moment measures and as such is not very robust against outliers or extreme observations.

In the next section we introduce the background information of PCA in order to help readers to remind themselves of its properties. In addition we closely look at the PRESS statistics, introduced by Allen [1974], to address the question "How many Principal Components (PC) need to be retained?" and explain why it is a better consideration than the explained techniques above in the context of prediction. Then we will touch upon the Robust PRESS statistics (R-PRESS) by explaining why the original PRESS could (also) be vulnerable to data anomalies.

2 Methods

In this section we first introduce classical results about PCA, followed by the relationship between Principal Component Analysis (PCA) and Singular Value Decomposition (SVD), which is fundamental for detailing each step in the fast Cross Validation (CV) algorithm for high dimensional PCA (sections 4.1 and 4.2 of [Hubert and Engelen, 2007]). In addition, we explain the PRESS with its mathematical properties as well as its usage for component selections in PCA, followed by detailed explanations about the R-PRESS.

2.1 Eigen Decomposition and PCA

We will start by showing a classical result for PCA [Johnson and Wichern, 2002]. Let $\boldsymbol{X}$ denote a $n \times p$ matrix of real data, $\boldsymbol{S}$ its estimated covariance matrix and $\boldsymbol{m}$ the p-dimensional mean vector. Let $\boldsymbol{S}$ have eigenvalues-eigenvector pairs $(\lambda_1, \boldsymbol{e_1}), \dots (\lambda_p, \boldsymbol{e_p})$ where $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$. Applying an eigendecomposition to the symmetric covariance matrix, $\boldsymbol{S}$ is factorized into its canonical form, which is

$$\boldsymbol{S} = V \Lambda V'. \tag{1}$$

with $\boldsymbol{V}$ the square $p \times p$ matrix whose jth column is the eigenvector $\boldsymbol{e_j}$ of $\boldsymbol{S}$ and Λ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\Lambda_{jj} = \lambda_j$.
The jth principal component, is given by the n-dimensional vector

$$Y_j = \boldsymbol{e}_j' \boldsymbol{X}, \qquad j = 1, \dots, p. \tag{2}$$

With this choices, it follows that

$$\begin{aligned} Var(Y_j) &= \boldsymbol{e}_i' \boldsymbol{S} \boldsymbol{e}_i = \lambda_j, \qquad j = 1, \dots, p, \\ Cov(Y_j, Y_k) &= \boldsymbol{e}_j' \boldsymbol{S} \boldsymbol{e}_k = 0, \qquad i \neq k. \end{aligned}$$

A property of PCA is that the sum of the principal component variances is equal to the sum of the original variable variances. Thus, using the proportion of variance explained by each component is a method for selecting k (1.2), with $k < p$, the dimension of the subspace spanned by the loading vector.

Selected k by using an appropriate method, the $n \times k$ score matrix $\boldsymbol{T}$ obtained from the first k principal components, is given by the formula:

$$\boldsymbol{T}_{n,k} = \boldsymbol{X}_{n,p} P_{p,k} = (Y_1 \dots Y_k) \tag{3}$$

where $P_{p,k} = (\boldsymbol{e}_1, \dots, \boldsymbol{e}_p)$ is the $p \times k$ loading matrix. Furthermore, every row of $\boldsymbol{T}_{n,k}$ contains the scores $\boldsymbol{t_i}$, which represents the ith observation in the k-dimensional subspace spanned by the loading vectors.
If data are centered, the ith score is formuled as follows

$$\boldsymbol{t}_i = P_{k,p}'(\boldsymbol{x}_i - \boldsymbol{m}). \tag{4}$$

where $\boldsymbol{x}_i$ is a p-variate vector denoting the ith observation (ith row of $\boldsymbol{X}$).

From formula (3) and obtained $P_{p,k}$ from $\boldsymbol{S}$ eigen decomposition (1), it follows that an estimate for $\boldsymbol{X}$, namely $\hat{\boldsymbol{X}}$, the $n \times p$ predicted values matrix for each observation, is given by

$$\hat{\boldsymbol{X}}_{n,p} = \boldsymbol{T}_{n,k} P_{k,p}'. \tag{5}$$

If data are centered, the prediction for the ith observation $\hat{\boldsymbol{x}}_i$ assumes the following formulation

$$\begin{aligned} \hat{\boldsymbol{x}}_i &= P_{k,p} \boldsymbol{t}_i + \boldsymbol{m} \\ &= P_{k,p} P_{p,k}'(\boldsymbol{x}_i - \boldsymbol{m}) + \boldsymbol{m}. \end{aligned} \tag{6}$$

2.2 Singular Value Decomposition and PCA

In Wall et al. [2002] it is shown that instead of passing through the $\boldsymbol{S}$ eigen decomposition for finding eigenvalues-eigenvectors pairs, an alternative is using SVD on the centered data matrix. Let $\boldsymbol{X}$ now denote

the $n \times p$ matrix of the centered data. The equation for singular value decomposition of $\boldsymbol{X}$ is the following:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}', \tag{7}$$

where $\boldsymbol{U}$ is an $n \times p$, matrix, $\boldsymbol{D}$ is an $p \times p$ diagonal matrix and $\boldsymbol{V}'$ is also a $p \times p$ matrix. The columns of $\boldsymbol{U}$ are called the left singular vectors, the rows of $\boldsymbol{V}'$ contain the elements of the right singular vector while the element of $\boldsymbol{D}$ are only nonzero on the diagonal and are called the singular values. We know that $\boldsymbol{S} = \frac{1}{n}\boldsymbol{X}'\boldsymbol{X}$ and by substituting $\boldsymbol{X}$ as in (7) we get that

$$\boldsymbol{S} = \boldsymbol{V}(\frac{\boldsymbol{D}^2}{n})\boldsymbol{V}'.$$

Following from (1), we can notice that $\boldsymbol{V} = \boldsymbol{V}$ and $\frac{\boldsymbol{D}^2}{n} = \boldsymbol{\Lambda}$. Here explained the equivalence in using SVD for applying classical results of PCA.

2.3 CV PRESS statistics and its usage in PC selection

From formula (6), it follows that the distance between the observed and the fitted observation, defined as the orthogonal distance (OD), in the k-dimensional PCA subspace is expressed by

$$\mathbf{OD}_{i,k} = \| x_i - \hat{x}_{i,k} \| = \| x_i - (P_{p,k}\boldsymbol{t}_i + \boldsymbol{m}) \| \tag{8}$$

With the denotation $\hat{x}_{-i,k}$, that is the fitted value from the PCA model with j PCs without the i-th observation, we express the CV PRESS statistics as follows.

$$\text{PRESS}_j = \frac{1}{n}\sum_{i=1}^{n} \| x_i - \hat{x}_{-i,k} \|^2 = \frac{1}{n}\sum_{i=1}^{n} \mathbf{OD}^2_{-i,k} \tag{9}$$

where $\mathbf{OD_{-i,k}}$ indicates the distance between the original and cross-validated sample without the *ith* observation. Note that this CV PRESS statistics measures the *average* of the squared ODs of all observations.

Wold [1978] introduced R ratio that is comparing the PRESS after fitting j components, with the sum of squared differences between observed and estimated all data points, using $j-1$ components,

$$R = \frac{\text{PRESS}_j}{\sum_{i=1}^{n} \| x_i - {}_{j-1}\hat{x}_{-i,k} \|^2} \tag{10}$$

If the ratio is less than 1, a better prediction is expected by using j instead of $j-1$ PCs.

The W ratio, developed by Eastment and Krzanowski [1982], compares the reduction in PRESS in adding the j-th PC to the model divided by its degree of freedom, with the PRESS after fitting j PCs, also divided by its degrees of freedom. The ratio is given by

$$W = \frac{(\text{PRESS}_{j-1} - \text{PRESS}_j)/df_{j,1}}{\text{PRESS}_j/df_{j,2}} \tag{11}$$

where $df_{j,1}$ and $df_{j,2}$ indicate the degrees of freedom of the numerator and the denominator, respectively. The decision rule is that if W is larger than 1 the inclusion of the j-th PC is acceptable. Compared to the scree plot and the Kaiser's rule, such ratios provide *relative* and *sequential* information between the component (regardless of ordering), and therefore it is more trustworthy. Nevertheless, these two ratios tend to

retain too small amounts of PCs despite the right numbers of PCs chosen in simulation studies [Wold, 1978, Eastment and Krzanowski, 1982]. Adjusting the ratios could solve the underestimation although setting the ratios equal 1 looks *universally* practical [Jolliffe, 2002]. It is worth noting that W ratio selects a more appropriate number of PCs than R ratio [Jolliffe, 2002], and hence we stick to W ratio for the data analysis in section 4.

Coming back to the property of the CV PRESS statistic, we claim that it is still not robust because an outlying observation will be poorly fit and unduly increase the PRESS value. To make it more robust, an added weight is added to the squared errors

$$\text{R} - \text{PRESS}_j = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i \parallel x_i - \hat{x}_{-i,k} \parallel^2 . \tag{12}$$

It is preferred to use the same weights w_i amongst all k components to give equal importance to the squared residuals at each PCA model under consideration. To find the w_i value, first a robust PCA procedure is applied for each k in the full dataset, which gives the loadings, eigenvalues, and fitted values for each model. Then the OD for each observation is found. If an observation has a large OD it is considered an outlier. A cutoff value is hard to determine as the distribution is not known. In [Hubert et al., 2005a] it is shown a good approximation is $c = (\hat{\mu} + \hat{\sigma} z_{0.975})^{3/2}$, where $z_{0.975}^{3/2} = \Phi^{-1}(0.975)$ is the 97.5% quantile of the Gaussian distribution and $\hat{\mu}$ and $\hat{\sigma}$ are the robust estimates of the mean and standard deviation of the OD.

Each observation in each k is given a weight $w_{i,k}$ of 0 if $\text{OD}_{i,k} > c$, and given a weight $w_{i,k}$ of 1 otherwise. A global weight is constructed as the minimum of all w_i over $k = 1, ..., k_{max}$. The global weight w_i is computed for each observation i. As soon as it is an outlier at one value of k, it thus becomes zero,

$$w_{i,min} = \min_k (w_{i,k}). \tag{13}$$

R-PRESS is then used to find the eigenvalues with the maximum variance, measured as the predicted error sum of squares. R-PRESS is more robust than PRESS because it uses weights to minimise the impacts of outliers or extreme values, which could artificially inflate the sum of squares values. The values for $\hat{x}_{-i,k}$ are estimated by the Fast CV algorithm outlined in section 2.5.

2.4 The full Robust PCA algorithm

In this section we detail and explain each step of the full robust PCA algorithm as described in Hubert and Engelen [2007]. When the number of variables p is smaller than the number of observations or rows n, the Minimum Covariance Determinant (MCD) estimator is used. The first k eigenvectors of the MCD covariance matrix, sorted in descending order of the eigenvalues, give us the robust loadings. However, when we have high-dimensional variables ($p > n$), we can no longer use the MCD because the determinant of a covariance matrix of $h < p$ observations will always be zero and thus cannot be minimized. The ROBPCA method circumvents this problem by combining projection pursuit ideas in the high-dimensional space with MCD estimation in a lower dimensional subspace.

(1) First, a singular value decomposition (SVD) is performed on the data such that all the observations are projected onto the space spanned by the data themselves. This results in a representation of the data with n rows and at most $n - 1$ columns without losing information of the data.

Detail

- The algorithm starts by assuming that our original data is stored in a n x p data matrix $X = X_{n,p}$ where n is the number of rows (or objects) and p represents the original number of variables. The goal of the first step is to work in a reduced data space without losing any information.
- The equation for SVD decomposition of X is the following

$$X = UDV^T,$$

  where all the matrix are defined as in (7).
- The subspace in which we are now working is the one spanned by the columns of V - the right singular vectors of the SVD. In other words, the score matrix $T_{n,k}$ becomes the representation of the ith observation in the k-dimensional subspace and it usually replaces X in the calculations that follow.
- The benefit of this step is that when the number of variables vastly exceeds the number of observations, the score matrix yields a huge dimension reduction without losing information.

Reducing the data space to the affine subspace by the n observations is mostly useful when the number of features greatly exceed the number of observations. Although the first step is not always necessary for $n < p$, it is still possible for the observations to span less than the whole p-dimensional space. It is important to note that step 1 is not intended to retain the first eigenvectors of covariance matrix $X_{n,p}$, otherwise we would be doing regular non-robust PCA.

(2) Next a measure of outlyingness is computed for every point. This is done by projecting all the observations on many univariate directions through two data points. If the sample size is small, all directions can be considered, otherwise at most 250 directions are taken. A robust center and robust scale (of the projected data) are computed and the standardized distance of each observation to the center is determined for all the directions. For each point the largest distance, which is called the outlyingness, is retained. The h points with smallest outlyingness form the initial H-subset H_0 (sorted by outlyingness).

Detail

- In this step the algorithm is trying to find the $h < n$ 'least outlying' data points. To find the h points, the Stahel-Donoho outlyingness measure is used.
- For every data point $\boldsymbol{x_i}$ (a row of $\boldsymbol{X}$), the outlyingness is calculated:

$$\text{out}(\boldsymbol{x_i}, \boldsymbol{X}) = \sup_{\boldsymbol{v} \in B} \frac{|\boldsymbol{x_i}'\boldsymbol{v} - m(\boldsymbol{x_i}'\boldsymbol{v})|}{s(\boldsymbol{x_i}'\boldsymbol{v})}$$

  where $m(.)$ and $s(.)$ are the robust univariate Minimum Covariance Determinant (MCD) estimators of location and scale and B is a set of maximum 250 random directions drawn through two data points.
- On every direction $\boldsymbol{v}$, a robust center and scale of the projected data points $\boldsymbol{x_i}'\boldsymbol{v}$ is computed, namely the univariate MCD estimator of location and scale.
- In other words, for each direction $\boldsymbol{v} \in B$, we project the n data points $\boldsymbol{x}_i$ on $\boldsymbol{v}$ and compute their robustly standardized absolute residual.
- The h data points with smallest outlyingness are then retained. These points form the initial H-subset H_0 (sorted by outlyingness).

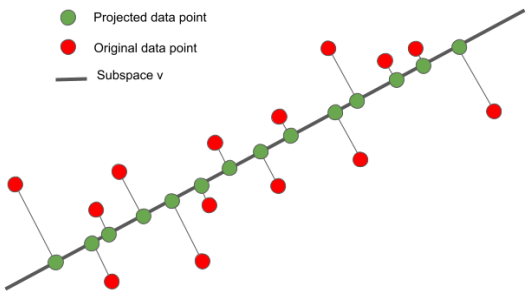


Figure 2: Schematic display of projection of points onto a subspace

(3) A further dimension reduction is obtained by projecting the data on the k-dimensional subspace spanned by the eigenvectors that belong to the k largest eigenvalues of S_0, which is the empirical covariance matrix of the observations in H_0. Afterwards a first re-weighting step is included.

Detail

- A co-variance matrix of subset h is calculated, named S_0

$$\hat{\boldsymbol{u}_0}(\boldsymbol{X}) = \frac{1}{h} \sum_{i \in H_0} \boldsymbol{x}_i \tag{14}$$

$$S_0 = \frac{1}{h-1} \sum_{i \in H_0} (\boldsymbol{x}_i - \hat{\boldsymbol{u}_0}(\boldsymbol{X}))(\boldsymbol{x}_i - \hat{\boldsymbol{u}_0}(\boldsymbol{X}))' \tag{15}$$

- The covariance matrix S_0 is used to determine the number of principal components to retain, k, for further analysis. This step serves as a dimension reduction.
- Once k is chosen (usually by looking at a scree plot), we reduce the dimension by projecting all data on the k-dimension subspace V_0 spanned by the first k eigenvectors of the robust covariance estimator S_0.

(4) In the last stage, a robust center and covariance matrix are computed within the obtained k-dimensional subspace. To this end, the re-weighted MCD estimator is applied to the projected data. From these computations, we retain two h-subsets that will be used in the fast CV algorithm. The first one is $\boldsymbol{H_{freq}}$ that contains the h observations which are most frequently selected in the whole resampling part (steps 1–3 of the MCD algorithm). Secondly, we denote $\boldsymbol{H_1}$ as the h-subset which yields the lowest objective function (see step 4 of the MCD algorithm). The final principal components are the back-transformed eigenvectors of the MCD covariance estimate and are used as the columns of $\boldsymbol{P_{p,k}}$. The back-transformed center $\boldsymbol{m}$ serves as the robust center of the data. The estimated value $\boldsymbol{x}_{i,k}$ can then again be obtained as in (6).

Detail

- The last stage of ROBPCA algorithm consists of projecting the data points onto the k-dimensional subspace spanned by the k largest eigenvectors of S_0 and of computing their center and shape by means of the reweighted MCD estimator.
- The eigenvectors of this scatter matrix then determine the robust principal components, and the MCD location estimate serves as a robust center.

According to Hubert et al. [2005b], the FAST-MCD algorithm was chosen to work under the hood of ROBPCA because it is currently the only algorithm that can deal with exact fit situations - which occur when we encounter a direction v whereby the projected observations have a robust scale equal to zero.

2.5 The fast CV algorithm

Staying in the high-dimensionality context ($p > n$), cross-validation (CV) techniques may be applied to obtain a much more computationally efficient algorithm. The fast CV algorithm avoids the two types of resamplings involved in the naive CV through the computation of a full ROBPCA procedure for each observations in the data set.

As a starting point ROBPCA method is performed on the full data set with $k = k_{max}$ and retain H_0, H_1 and H_{freq}. For each observation $i = 1, ..., n$ the following steps are repeated:

(1) Remove sample i from the data

Detail

- Again, the algorithm starts by assuming that our original data is stored in a n x p data matrix $X = X_{n,p}$ where n is the number of rows and p represents the original number of variables. The first step aims at excluding the ith row from matrix X.

(2) We now have to find $h - 1$ observations with smallest outlyingness. For this, we just update H_0 in a similar way as for the MCD method. If the ith case belongs to H_0, we set $H_{-i,0} = H_0/\{\mathbf{x_i}\}$. Else, the point with the largest outlyingness is deleted from H_0.

Detail

- This step avoids to obtain H_0 by applying step 2 of the ROBPCA algorithm which involves the computation of the outlyingness measures for each point. Resampling is therefore substituted with the following updating method for H_0:

$$H_{-i,0} = \begin{cases} H_0/\{\mathbf{x_i}\} & \text{if the } i\text{th case belongs to } H_0 \\ H_0/\{\mathbf{x_{highout}}\} & \text{otherwise} \end{cases}$$

  where x_i is the ith observation and $x_{highout}$ is the point with the largest outlyingness.

(3) Next, the data are projected on the k_{max}-dimensional subspace spanned by the k_{max} dominant eigenvectors of the empirical covariance matrix of the observations in $H_{-i,0}$. This subspace is found by applying a SVD on $H_{-i,0}$, thereby using the kernel version for data with more variables than cases. Note that doing so, we obtain the same results as if we would first perform an SVD on the reduced data set without sample i, yielding $H_{-i,0} \subset \mathbf{R}^d$. The reason is that SVD in the first stage of ROBPCA is done without loss of information, i.e. all singular values are retained.

Detail

- The subspace in which we are now working is the one spanned by the columns of V, containing the right singular vectors of the SVD.
- The equation for SVD decomposition of $X_{H_{-i,0}}$, where $X_{H_{-i,0}}$ is the data matrix $(h-1 \times p)$ corresponding to the points contained in $H_{-i,0}$, is substituted by a kernel version [Wu et al., 1997].

  For such data sets with many variables and fewer objects, the classic algorithms for PCA based on SVD decomposition becomes very inefficient, because the size of the associated matrix $X^T_{H_{-i,0}} X_{H_{-i,0}}$ $(p \times p)$ is very large. Based on the kernel matrix $X_{H_{-i,0}} X^T_{H_{-i,0}}$ $(h-1 \times h-1)$, the kernel algorithms are developed. They yield the same principal components but, when the number of variables is higher than the number of objects, they are faster than the corresponding classic algorithms.
- From the kernel algorithms we can find a corresponding SVD decomposition of $X_{H_{-i,0}}$:

$$X_{H_{-i,0}} = U_{-i,0} D_{-i,0} V^T_{-i,0}$$

- From the previous decomposition we can find the k_{max} dominant eigenvectors of the empirical covariance matrix of $X_{H_{-i,0}}$ and the projected observations in $H_{-i,0}$ on the k_{max}-dimensional subspace $V_{-i,0}$.

(4) Finally, the reweighted MCD method is applied on the projected observations. However, we only consider three (h-1)-subsets to start C-steps from, instead of drawing many random k_{max}-subsets, namely $H_{-i,0}$, $H_{-i,1}$ and $H_{-i,freq}$. The updates of H_1 and H_{freq} are obtained analogously to that of H_0: if the ith case belongs to H_1, resp. H_{freq}, it is removed from those h-subsets. Otherwise, the observation with largest robust distance, respectively, with smallest frequency, is taken away. This yields a robust center m and covariance matrix S.

Detail

- The $(h-1 \times k_{max})$-dimensional projected observations matrix $T_{H_{-i,0}}$ is obtained as follows:

$$T_{H_{-i,0}} = X_{H_{-i,0}} V_{-i,0}$$

- Respectively, applying step (1), (2) and (3) to H_1 and H_{freq}, we obtain $T_{H_{-i,1}}$ and $T_{H_{-i,freq}}$, where $H_{-i,1}$ and $H_{-i,freq}$ are updated similarly to the updating pattern of $H_{-i,0}$ at step (2):

$$H_{-i,1} = \begin{cases} H_1/\{\mathbf{x_i}\} & \text{if the } i\text{th case belongs to } H_1 \\ H_1/\{\mathbf{x_{highout1}}\} & \text{otherwise} \end{cases}$$

$$H_{-i,freq} = \begin{cases} H_{freq}/\{\mathbf{x_i}\} & \text{if the } i\text{th case belongs to } H_{freq} \\ H_{freq}/\{\mathbf{x_{lessfreq}}\} & \text{otherwise} \end{cases}$$

  where x_i is the ith observation, $x_{highout1}$ is the point with the largest robust distance among H_1 and $x_{lessfreq}$ is the point with the smallest frequency among H_{freq}.

- An MCD method is applied on the projected observations, which do not suffer anymore of the high-dimensionality issue, using $H_{-i,0}$, $H_{-i,1}$ and $H_{-i,freq}$ as initial subsets of the C-steps algorithm. Such a choice helps in avoiding a resampling step over all possible $(h-1)$-subsets. In the end, the MCD procedure returns a robust k_{max}-dimensional center m_{-i} and $k_{kmax} \times k_{max}$ scatter matrix S_{-i}.

(5) For each k, the loading matrix $P_{p,k}$ is now obtained as the (backtransformed) k dominant eigenvectors of S. Also the center m_{-i} is transformed to the original p-dimensional data space. The CV fitted value $\hat{x}_{-i,k}$ is then calculated as $x_{-i,k} = P_{k,p} P'_{p,k}(x_i - m) + m$.

Detail

- The last step of the algorithm returns a loading matrix $P_{-i,k}$ for each $k < k_{max}$ dimension.

- $P_{-i,k}$ is obtained by back-transforming the first k eigenvectors of the robust covariance matrix S_{-i}. Also the center m_{-i} is back-transformed to a p-dimensional vector M_{-i}.

- The CV fitted value $\hat{x}_{-i,k}$ is then calculated according to the given formula:

$$\hat{x}_{-i,k} = P_{-i,k} P_{-i,k}^T (x_i - M_{-i}) + M_{-i}.$$

3 Simulation study

The goal of this section is to reproduce the simulation results of section 5.2 from Hubert and Engelen [2007] and comment on the differences between PRESS and R-PRESS. Simulations are implemented in MATLAB with the help of functions obtained from LIBRA - a Matlab library for Robust Analysis. Equivalent to the paper, we generate data from a mixture of multivariate distributions:

$$(1-\epsilon)N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \epsilon N_p(\tilde{\boldsymbol{\mu}}, \boldsymbol{\Sigma})$$

where ϵ is the contamination parameter. We consider two simulated datasets containing a higher number of features than number of observations ($p > n$), with each dataset being separated into two classes; a model with outliers ($\epsilon = 10\%$) and another without outliers ($\epsilon = 0$). The two data contaminated data sets are described below:

1. n=30, $p = 50$, Σ=diag(10, 7.5, 5, 3, . . .) and $\tilde{\boldsymbol{u}}$ = (0, 0, 0, 0, 0, 50, 0,..., 0)'
2. n=100, $p = 500$, Σ=diag(10, 7.5, 5, 3, . . .) and $\tilde{\boldsymbol{u}}$ = (0, 0, 0, 0, 0, 50, 0,..., 0)'

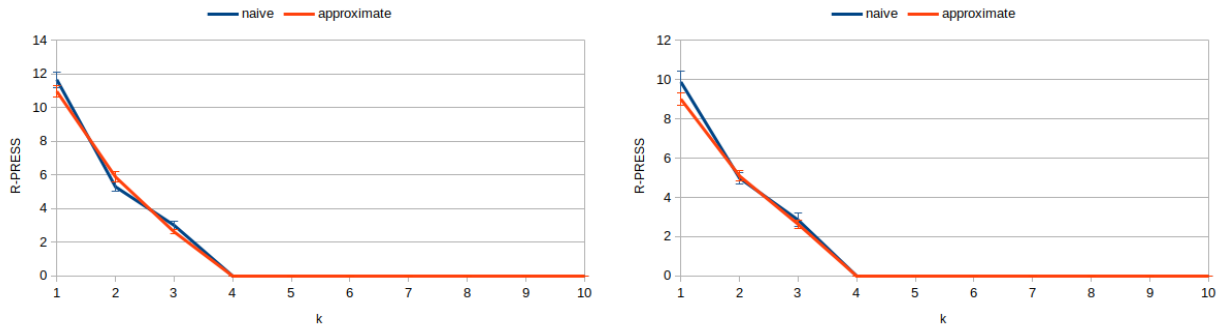
Uncontaminated data has mean vector $\boldsymbol{u} = \mathbf{0}$. The first three components of Σ are set in such a way that they explain a large proportion of variation, while the remaining components are negligibly small numbers. To obtain R-PRESS curves as seen in the paper, for each data set we generated 10 random iterations, each with its own unique seed number. Having unique seed numbers is suited for the reproducibility and comparability of results between and within data sets. To add bars on the R-PRESS curve for each k, some modifications were made to the files `robpca.m` and `cvRobpca` so that R-PRESS values can be extracted and used for further analysis. These modifications were needed since the function `robpca` returns a press plot and not the R-PRESS values for each k. Once these changes were made, we were able to replicate the fast cv method for each generated data set.

Reproducing the naive method was more cumbersome than initially expected. Unlike the approximate method, there was no built in function that could be tinkered with to obtain the much needed R-PRESS values. The steps used to replicate the naive cross-validated ROBPCA method is summarised below:

1. Apply ROBPCA method for each k
2. For each $k = 1, 2, .., k_{max}$, do the following
3. Remove observation i from original data matrix
4. Apply ROBPCA on data with removed ith observation
5. Calculate $\text{OD}_{-i,k}$, the distance between the original data and the (CV) estimated sample
6. Repeat steps 3-5 for all n observations
7. Calculate the kth CV PRESS-value

We first applied the robust PCA method for each k (on the full data set, without CV), thereby obtaining loadings, eigenvalues and fitted values for each model. Due to the code changes made for the fast cv method,

we were also able to extract the weights that are required in the R-PRESS formula (see equation 12). Thereafter, we designed an algorithm that removed n times an observation i from the data set and calculates the CV PRESS-value for every $k = 1,...,k_{\max}$.



(a) (b)

Figure 3: The ROBPCA R-PRESS curves for the first simulation setting with n=30, p=50 (a) without outliers; and (b) with 10% contamination.

The first four eigenvalues are noticeably larger than the remaining ones. Figures 3a and 3b display the R-PRESS curves as well as the corresponding confidence intervals for the two generated datasets. It is apparent that there is very little between the two methods as both curves almost overlap each other entirely. The error bars in the original paper were considerably larger than the ones that we obtained. We suspect that this is due to a smaller variation between our generated data sets.

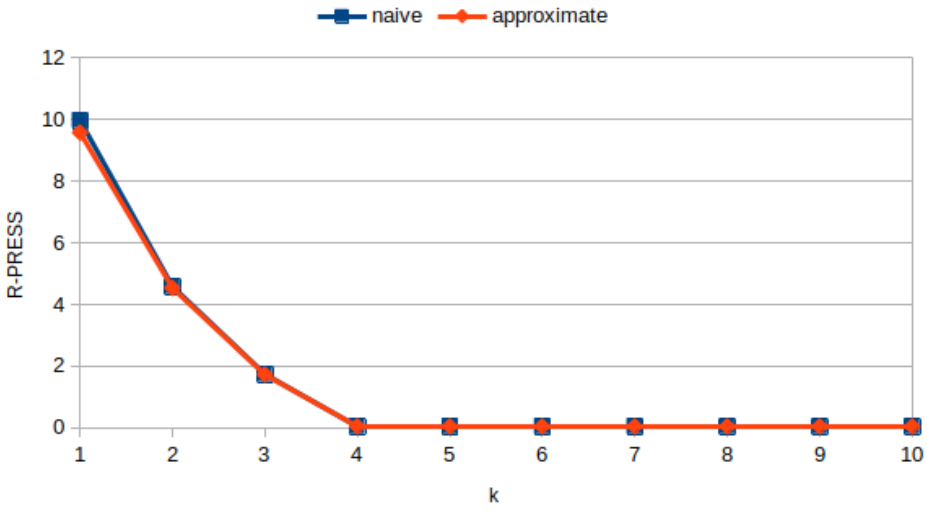


Figure 4: The ROBPCA R-PRESS curves for the second simulation setting with n=100, p=500 and 10% contamination.

The R-PRESS curve for the larger simulated data (see Figure 4) indicates that the naive and approximate methods are almost identical when analysing their performance over 10 random generated data sets. The

approximate approach to CV has resulted in a great reduction in computing time and yet has not severely affected performance or reliability. The same conclusion is drawn for the first simulation setting - with and without outliers.

Table 1 reports the running time of the naive and fast ROBPCA algorithm's. There is a clear difference in running time between the two methods, further demonstrating the importance and need for fast algorthims. However, whilst replicating the results we noticed that both algorithms were much faster than what was announced in the paper. For instance, the authors reported that the naive algorithm took on average 1265.7 to execute, whereas our algorithm ran in under 30 seconds. A significant improvement in computer time is hypothesised to be largely due to technological advancements since 2007.

n	p	Approximate	Naive
30	50	0.92 (0.01)	29.3 (1.3)
100	500	1.2 (0.04)	212.7 (6.4)

Table 1: Average computation times and standard deviations (between brackets) in seconds for the approximate and the naive cross-validated ROBPCA method when applied to data with zero contamination

Figure 5 compares CV PRESS and R-PRESS for the second simulation setting in a naive manner. The PRESS statistic has an increased value for $k = 1, 2$ and 3 since each outlying observation will be badly fitted, even when robust PCA is used. The R-PRESS statistic overcomes this issue by giving equal importance to the squared residuals at each PCA model under consideration, and as a result the residuals are decreased.

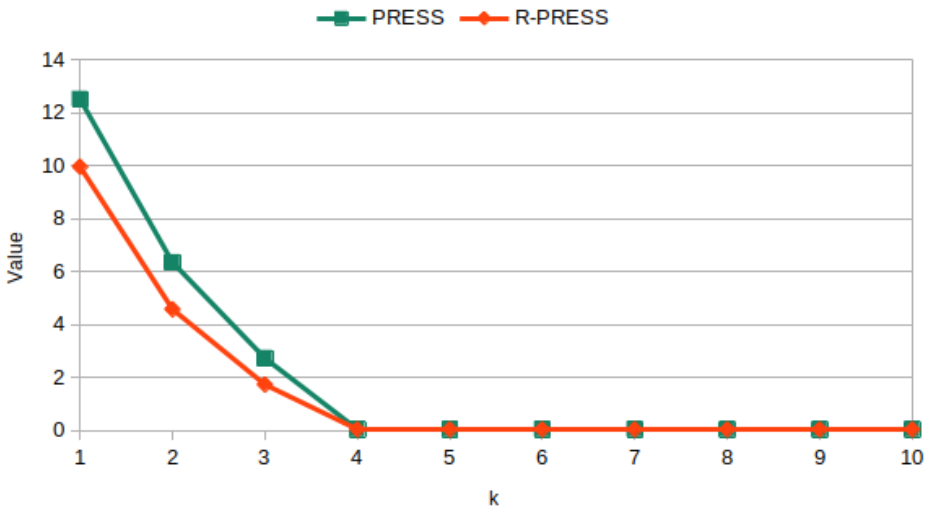


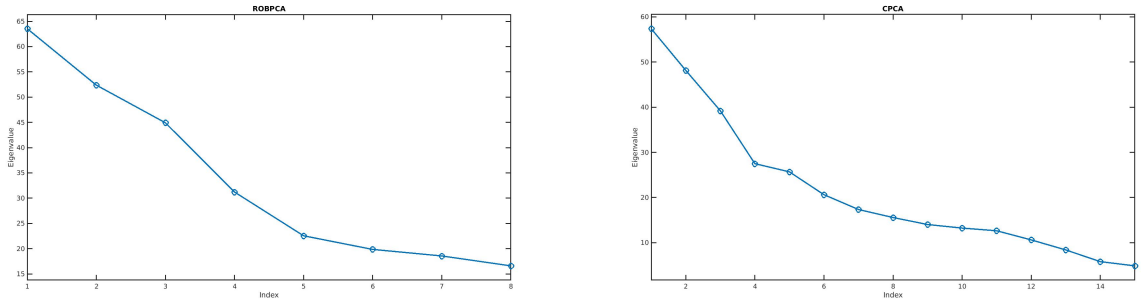
Figure 5: R-PRESS (orange line) and PRESS (green line) curves for the second simulation setting with n=100, p=500 and 10% contamination.

This simulation study demonstrates the accuracy and the gain in computation time of the fast CV algorithms introduced by Hubert and Engelen [2007].

4 Data analysis

An exploratory analysis is conducted on a gait dataset found on UCI Machine Learning repository. The dataset contains the walking parameters of 16 participants measured over 30-meter walks. There are 321 measurements divided into 3 main groups: basic parameters, temporary parameters, and spatial parameters. Since each participant walked 3 times (total n = 48), we took only the first round of walking for each participant so as to remove dependence between observations. Indexing of the observations remained the same and the data are scaled to have unit variance.

We start by performing a ROBPCA analysis on the scaled high-dimensional gait dataset with default value of $h = 0.75$. Figure 6a visualizes the eigenvalues of the components obtained from the ROBPCA procedure on the data versus the number of the component. As mentioned earlier, the scree plot is primarily used to determine the number of principal components to retain. Figure 6b suggests retaining four principal components, whereas a resolution is not so clear in Figure 6a because there is no distinct elbow in the curve. We therefore resort to inspecting the proportion of variance accounted for by each additional component.



(a) (b)

Figure 6: Scree plots of the gait dataset with (a) ROBPCA; and (b) CPCA

The percentage of variance explained by the first 10 components of the ROBPCA procedure is:

0.19733	0.35996	0.49944	0.59637	0.66641	0.72806	0.7857	0.83723

The first six principal components are chosen to compute in further analysis since they account for 72.8% of the variation in the data. However, when performing Classical PCA (CPCA) on the same data set, the resulting first six components amount to 68.1% cumulative variance.

If we are interested to see how well the ROBPCA model is suited for prediction in the gait dataset, analysis of the CV PRESS (or R-PRESS) is a better alternative to the popular scree plot. PRESS curves are drawn in Figure 7 and serve as an additional exploratory tool for selecting the number of principal components to retain. The curve in blue was obtained using the approximate method of Hubert and Engelen [2007]. In the simulation study of section 3, the approximate method was shown to be as accurate as the naive one. It can be seen again that the naive and approximate curves are very similar, even when applied to a real data set. The orange line plots the non-robust CV PRESS values, which is larger than the other curves (R-

PRESS) for all components k. Again, a result that appeared in a simulation study has reverberated through to analysis of real data - that is, the PRESS value is increased in the presence of anomalies.

With respect to choosing the number of principal components to retain for prediction purposes, we apply the W criterion approach of Eastment and Krzanowski [1982] to the approximate CV R-PRESS values (green line). The inclusion of the second and third component is associated with $W > 0.9$. Although the criterion suggests that W should be greater than 1, the rule is not clear-cut. Since for all components after the 4th the W statistic becomes progressively smaller, we decide it is worthwhile to keep at least the third principal component.

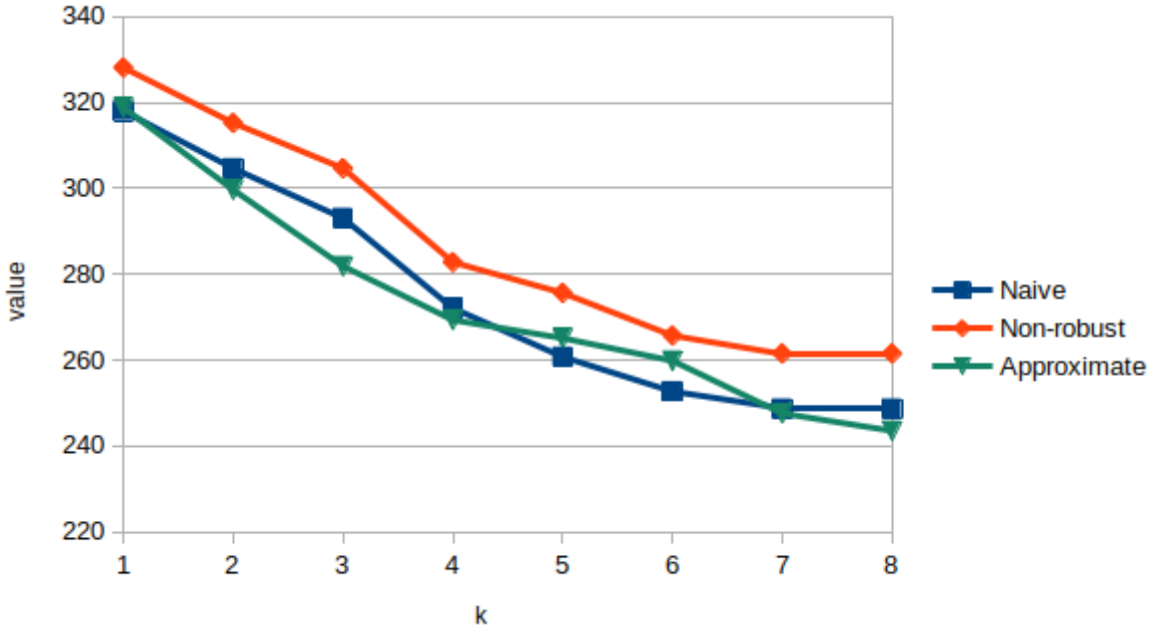
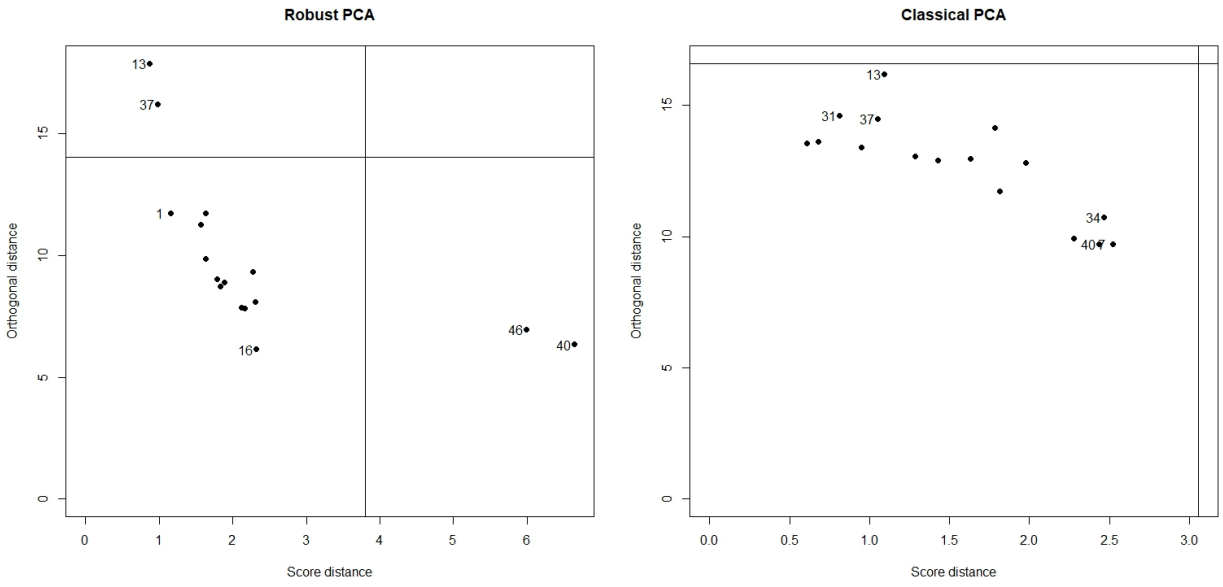


Figure 7: The R-PRESS and PRESS curve for the gait dataset obtained with ROBPCA. The R-PRESS curve is split into two methods; approximate (green) and naive (blue)

The computation time of the naive method on the gait dataset was roughly **18 seconds**, while the approximate method took only **0.62 seconds**. Clearly, a huge reduction in time is achieved when using the approximate CV method.

Based on the scree plots obtained with ROBCPA and CPCA we decided to retain $k = 6$ and $k = 4$ components respectively for outlier analysis. The corresponding diagnostic plots are displayed in Figure **??**.



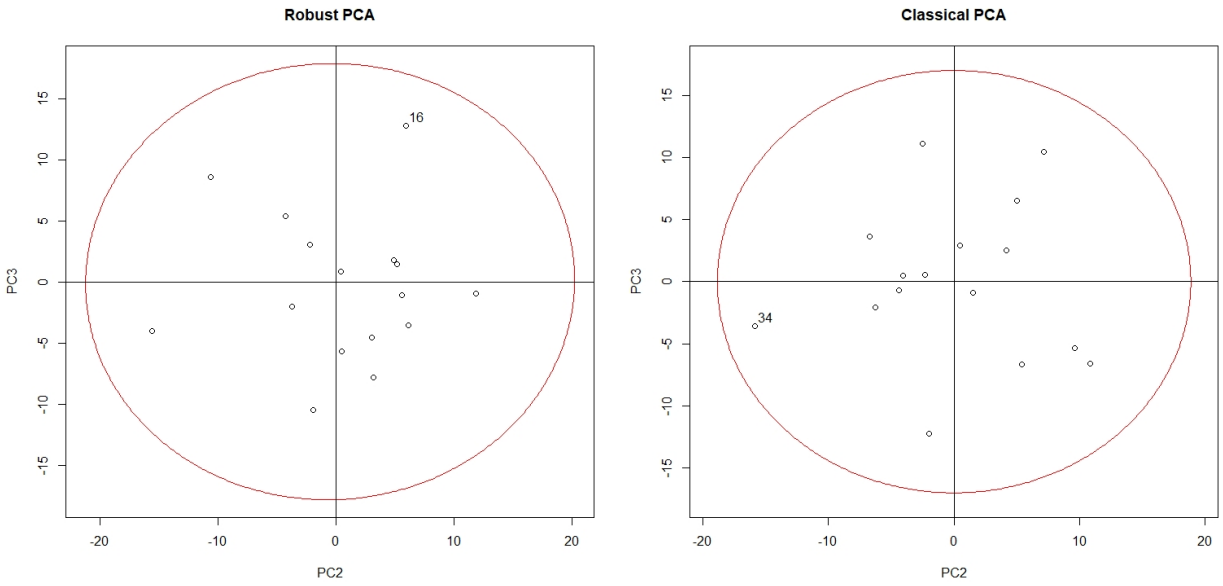
(a) (b)

Figure 8: Outlier maps of the gait dataset obtained with (a) ROBPCA (k=6); and (b) CPCA (k=4)

Figures 8a and 8b plot score distances against the orthogonal distance. The horizontal lines are constructed to differentiate between data points with small and large orthogonal distances. On the other hand, vertical lines separate points with small and large score distances. As a result, the outlier map of the data identifies three different types of outliers:

1. good leverage points
2. orthogonal outliers
3. bad leverage

The CPCA method does not flag any outliers since every data point is located in low score-distance and low orthogonal-distance. On the other hand, the ROBPCA procedure detects 4 outliers with the substantially increased score *or* orthogonal distances. Observations 40 and 46 are regarded as good leverage points since they lie close to the PCA subspace but far from the regular observations, whereas observations 13 and 37 are classified as orthogonal outliers because of their large orthogonal distance to the PCA subspace (see Figure 8a). In both methods, no observations are found to be bad leverage points. The most noticeable outcome of this comparative analysis is that the mentioned observations are found to be outliers in ROBPCA but CPCA. Although observation 13 is close to the cut-off orthogonal distance in CPCA, the outcome is significant.



(a) (b)

Figure 9: Score plots of the gait dataset obtained with (a) ROBPCA (second PC plotted versus third PC); and (b) CPCA (second PC plotted versus third PC)

In reality, researchers may not be satisfied with removing outlying observations, especially in the context of our dataset, where there are very few observations ($n = 16$). Instead, those who are conducting the study might be interested in flagging atypical observations as soon as possible so that the patient can repeat the 30-meter long course. By using the ROBPCA procedure, researchers are able to identify outlying measurements during analysis which are otherwise hidden in classical methods.

5 Conclusion

This paper reviews the leave-one-out cross validation in the context of Principal Component Analysis and emphasises the advantages of utilizing the CV PRESS to assess the predictive power of a PCA model. Compared to the scree plot and the Kaiser's rule, we find that the PRESS provides more mathematical and relative (between each component) affirmation for the component selection, and hence it is more robust. Nevertheless, the PRESS is not completely safe from potential outliers, which leads to the creation of the Robust PRESS by introducing the weights w. Next, the paper develops more robust methods for estimating the mean, covariance, and prinicple components of normal and high-dimensional data. The fast CV algorithm utilises the MCD method to estimate the covariance matrix and center of the data without the observation i. The principle components this yields are then used in R-PRESS to estimate the explained variance and aid in choosing how many principle components to retain. In the case of high-dimensional data, ROBPCA is combined with projection pursuit ideas and the MCD estimator, and then again R-PRESS is used to estimate the amount of variance explained by the components. The computing time is greatly increased by updating the mean and covariance matrix in MCD, or outlyingness in ROBPCA, instead of recomputing

them for the entire dataset.

We performed ROBPCA on the gait dataset, which revealed four outlying observations. The R-PRESS curves of the naive and fast CV algorithms were very similar, further demonstrating that the fast CV algorithm is as accurate as its naive counterpart, even when applied to real data. In summary, four components were retained for CPCA and six for ROBPCA, resulting in a classical explanation percentage of 53.6% and robust explanation percentage of 72.8%. Analysis of the diagnostic plots (see Figure 8) revealed that CPCA did not flag any outliers while ROBPCA does so for observations 13, 37, 40 and 46. For prediction purposes, it is worthwhile to retain at least the first three components in the ROBCA procedure.

Simulations, specifically those in section 3 were done in Matlab. The code for the naive CV method can be found in the file `naive.m`. The code file `get_weights.m` is purposed to extract the weights from the already defined `robpca` function. Data analysis of the gait dataset is performed in Matlab and R, which can be found in the q5.m (or q5.R) file. In R, the `rrcov` package was the main package used.

References

M. Hubert and S. Engelen. Fast cross-validation of high-breakdown resampling methods for pca. *Computational Statistics Data Analysis*, 51:5013–5024, 06 2007. doi: 10.1016/j.csda.2006.08.031.

M. Hubert, P.J. Rousseeuw, and K. Vanden Branden. Robpca: A new approach to robust principal components analysis. *Technometrics*, 47:64–79, 2 2005a. doi: 10.1198/004017004000000563.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

I. T. Jolliffe. *Principal component analysis*. Springer series in statistics. Springer, New York (N.Y.), 2nd ed. edition, 2002. ISBN 0387954422.

B. R. Cartell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1:245–276, 04 1966. doi: 10.1207/s15327906mbr0102_10.

R. Ledesma, P. Valero-Mora, and G. Macbeth. The scree test and the number of factors: a dynamic graphics approach. *The Spanish Journal of Psychology*, 18, 06 2015. doi: 10.1017/sjp.2015.13.

H. Kaiser. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20:141–151, 04 1960. doi: 10.1177/001316446002000116.

J. Kaufman and W. Dunlap. Determining the number of factors to retain: A windows-based fortran-imsl program for parallel analysis. *Behavior research methods, instruments, computers : a journal of the Psychonomic Society, Inc*, 32:389–95, 09 2000.

D. M. Allen. The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, 16:125–127, 02 1974. doi: 10.2307/1267500.

R. A. Johnson and D. W. Wichern. *Applied multivariate statistical analysis*. Prentice Hall, 5. ed edition, 2002. ISBN 0130925535.

M. E. Wall, A. Rechtsteiner, and L. M. Rocha. Singular value decomposition and principal component analysis. 2002.

S. Wold. Cross-validatory estimation of components in factor and principal components models. *Technometrics*, 20:397–405, 11 1978. doi: 10.1080/00401706.1978.10489693.

H. Eastment and W. Krzanowski. Cross-validatory choice of the number of components from a principal component analysis. *Technometrics*, 24:73–77, 02 1982. doi: 10.1080/00401706.1982.10487712.

M. Hubert, P. J. Rousseeuw, and Vanden B. K. Robpca: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005b.

W. Wu, D.L. Massart, and S. de Jong. Kernel-pca algorithms for wide data part ii: Fast cross-validation and application in classification of nir data. *Chemometrics and Intelligent Laboratory Systems*, 37(2):271–280, 1997.