

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

Mia Hubert, Sanne Engelen

*Department of Mathematics
Katholieke Universiteit Leuven
W. De Croylaan 54, B-3001 Leuven, Belgium*

Abstract

Cross-validation (CV) is a very popular technique for model selection and model validation. The general procedure of leave-one-out CV is to exclude one observation from the data set, to construct the fit of the remaining observations and to evaluate that fit on the item that was left out. In classical procedures such as least-squares regression or kernel density estimation, easy formulas can be derived to compute this cross-validated fit or the residuals of the removed observations. However, when high-breakdown resampling algorithms are used, it is no longer possible to derive such closed-form expressions. High-breakdown methods are developed to obtain estimates that can withstand the effects of outlying observations. Fast algorithms are presented for leave-one-out CV when using a high-breakdown method based on resampling, in the context of robust covariance estimation by means of the MCD estimator and robust Principal Component Analysis. A robust PRESS curve is introduced as an exploratory tool to select the number of principal components. Simulation results and applications on real data show the accuracy and the gain in computation time of these fast cross-validation algorithms.

Key words: Cross-validation, Robustness, MCD, ROBPCA, PRESS, Fast Algorithms

Email addresses: {mia.hubert,sanne.engelen}@wis.kuleuven.be,
Tel: +32 16322023, Fax: + 32 16322831. (Mia Hubert, Sanne Engelen).
URL: <http://wis.kuleuven.be/stat/robust.html> (Mia Hubert, Sanne Engelen).

1 Introduction

Cross-validation (CV) is a very popular technique for model selection and model validation. It is used when a validation set is not available, or when the data set is too small to split into a training and a validation set. This type of data is often encountered in chemometrics and bio-informatics, where data sampling can be very expensive or time consuming. The general procedure of leave-one-out CV is to exclude one observation from the set of n data points, to construct the fit of the remaining observations and to evaluate that fit on the item that was left out. Also m -fold CV can be considered in which case not a single observation but a group of $\lfloor n/m \rfloor$ items is withheld at once. Cross-validation is both used in parametric and nonparametric settings such as linear regression, classification, and density estimation. In these examples the technique can be applied, for example, to select the optimal set of regressors, to evaluate a classification rule, to estimate the prediction error or to find the optimal kernel bandwidth, see e.g. [Wu et al., 1997b, McQuarrie and Tsai, 1998, Shao, 1993, Hubert and Engelen, 2004].

The main computational effort of cross-validation is the determination of the fit based on $n - 1$ or $n - \lfloor n/m \rfloor$ data points. In classical procedures such as least-squares regression or kernel density estimation, easy formulas can be derived to compute this cross-validated fit or the residuals of the removed observations.

However, when high-breakdown resampling algorithms are used, it is no longer possible to derive such closed-form expressions. High-breakdown methods are developed to obtain estimates that can withstand the effects of outlying observations. Some typical examples are the LTS estimator [Rousseeuw, 1984] and S-estimators [Rousseeuw and Yohai, 1984] for multiple regression, the MCD estimator [Rousseeuw, 1984] and S-estimators [Davies, 1987] for multivariate location and covariance estimation. As the objective function of these methods is not convex, its minimum is searched over a set of possible fits that are based on resampling from the observations. Consequently, their computation time is much longer than for classical convex optimization methods, despite the development of fast algorithms such as the FAST-MCD algorithm [Rousseeuw and Van Driessen, 1999]. The computational complexity becomes really impractical if we want to apply cross-validation to one of these robust methods as this requires computing them at least m times and at most n times.

In this paper we propose how CV can still be performed quickly when using a high-breakdown method based on resampling. We will focus on leave-one-out cross validation (LOO-CV). A motivation to concentrate on LOO-CV is to ensure the robustness of the overall procedure. If we would apply m -fold CV in case of a large proportion of outliers in the data, the reduced data set of size

$n - \lfloor n/m \rfloor$ could contain an even larger proportion of outliers (in the worst case even more than 50%) which might cause the robust procedure to break down. Finally, our fast CV algorithms are based on the estimates for the full data set. When only one sample at a time is removed, we will show in this paper that a good approximation for the estimate of the reduced data set (of size $n - 1$) can be constructed. When more cases are simultaneously taken away, similar approximate techniques could be applied but they might be less precise.

In this paper, we concentrate on leave-one-out CV in the context of covariance estimation and Principal Component Analysis (PCA). As PCA is concerned with dimension reduction, one of the major issues is the determination of the optimal number of principal components. Several graphical and numerical techniques are developed for that purpose such as the scree plot (which displays the eigenvalues in decreasing order) or by considering the percentage of the total variance that is captured by the first k components. But if we are mainly interested to see how well the PCA model is suitable for prediction, the (cross-validation) PRedicted Error Sum of Squares (PRESS) is a better alternative.

In Section 2 we give the precise definition of the PRESS value and we construct a robust version. Next, we discuss two robust methods for PCA and we explain how approximate cross-validated PRESS values can be computed. For low-dimensional data, robust components can be derived as the eigenvectors of a robust covariance matrix, such as the MCD estimator. This will be outlined in Section 3. Section 4 is concerned with the ROBPCA method [Hubert et al., 2005] for high-dimensional data. The performance of the resulting PRESS values is investigated by means of several simulations in Section 5. Throughout we illustrate the new algorithms on two real data sets.

2 A robust PRESS-value

Principal Components Analysis is a dimension reduction technique applied to p -dimensional data \mathbf{x}_i ($i = 1, \dots, n$). It seeks for a $p \times k$ loading matrix $P_{p,k}$ and an estimate of the center of the data \mathbf{m} such that the scores \mathbf{t}_i , which represent the i th observation in the k -dimensional subspace spanned by the loading vectors:

$$\mathbf{t}_i = P'_{k,p}(\mathbf{x}_i - \mathbf{m}), \quad (1)$$

contain most of the variation in the data. In classical PCA, \mathbf{m} is given by the mean of the data and the k loading vectors are the eigenvectors of the covariance matrix of the data that belong to the k largest eigenvalues. Details on the construction of \mathbf{m} and P for robust PCA methods are given in Section 3.1 for the MCD-algorithm and in Section 4.1 for the ROBPCA method.

These procedures additionally provide estimates of the variances of the scores, the so-called eigenvalues $L_{k,k} = \text{diag}(l_1, \dots, l_k)$. An estimate of \mathbf{x}_i in the k -dimensional PCA-space is then given by

$$\hat{\mathbf{x}}_{i,k} = P_{p,k}\mathbf{t}_i + \mathbf{m} = P_{p,k}P'_{k,p}(\mathbf{x}_i - \mathbf{m}) + \mathbf{m}, \quad (2)$$

where the second equality follows from (1). The OD is defined as the distance between the observed and the fitted observation in the k -dimensional PCA subspace:

$$\text{OD}_{i,k} = \|\mathbf{x}_i - \hat{\mathbf{x}}_{i,k}\|. \quad (3)$$

If the i th observation is removed from the sample to construct a PCA model with k components from the $n - 1$ other cases, we denote $\hat{\mathbf{x}}_{-i,k}$ as the corresponding fitted value. The cross-validated PRESS-value now measures how much \mathbf{x}_i and $\hat{\mathbf{x}}_{-i,k}$ differ by taking the average of the squared orthogonal distances (or squared residuals) over all observations [Jolliffe, 1986]:

$$\text{PRESS}_k = \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \hat{\mathbf{x}}_{-i,k}\|^2 = \frac{1}{n} \sum_{i=1}^n \text{OD}_{-i,k}^2, \quad (4)$$

with $\text{OD}_{-i,k} = \|\mathbf{x}_i - \hat{\mathbf{x}}_{-i,k}\|$ the distance between the original and the (cross-validated) estimated sample. Note that equivalently the Root Mean Squared Error of Prediction $\text{RMSEP}_k = \sqrt{\text{PRESS}_k}$ can be considered. This PRESS statistic is however not robust, even if a robust PCA method is used to obtain the estimated values $\hat{\mathbf{x}}_{-i,k}$. This is because an outlying observation will be badly fitted by a robust procedure. Hence it will have an unusually large residual which would unduly increase the PRESS value. Therefore we define a robust PRESS-value by adding weights w_i to the squared errors:

$$\text{R-PRESS}_k = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i \|\mathbf{x}_i - \hat{\mathbf{x}}_{-i,k}\|^2. \quad (5)$$

Note that we prefer to use weights w_i which are the same among all the components k , in order to give equal importance to the squared residuals at each PCA model under consideration ($k = 1, \dots, k_{\max}$).

In order to determine the weights w_i , we use the following procedure. First, we apply the robust PCA method for each k (on the full data set, without cross-validation), which provides loadings, eigenvalues and fitted values for each model. Note that for high-dimensional data, we can apply the ROBPCA- k_{\max} procedure of [Engelen et al., 2005] in order to obtain the solutions for every k by one single run of the algorithm. Then for each observation its orthogonal distance (OD) is computed.

Outliers are those observations which have an unusually large OD. The cutoff value for the orthogonal distances are difficult to determine as their exact dis-

tribution is unknown. In [Hubert et al., 2005] it is shown that a good approximation is obtained by using $c = (\hat{\mu} + \hat{\sigma}z_{0.975})^{3/2}$, where $z_{0.975} = \Phi^{-1}(0.975)$ is the 97.5% quantile of the Gaussian distribution and $\hat{\mu}$ and $\hat{\sigma}$ are robust estimates of the mean and standard deviation of the OD.

For each k we can now give zero weight $w_{i,k}$ to an observation if its $\text{OD}_{i,k} > c$. The observations for which $\text{OD}_{i,k} \leq c$ are assigned a weight $w_{i,k} = 1$. This is a hard-rejection rule. A global weight w_i is then constructed by taking the minimum of all $w_{i,k}$ over $k = 1, \dots, k_{\max}$:

$$w_{i,\min} = \min_k(w_{i,k}). \quad (6)$$

In the following sections we concentrate on the computation of $\hat{\mathbf{x}}_{-i,k}$ in robust PCA. The Minimum Covariance Determinant (MCD) estimator [Rousseeuw, 1984] is a highly robust method to estimate multivariate location and scatter parameters when the number of variables p is smaller than half the number of samples. The principal components can be considered as the eigenvectors of the robust covariance estimate. ROBPCA, ROBust Principal Components Analysis, on the other hand is a technique where projection pursuit ideas are combined with the MCD estimator and is in particular appropriate for high-dimensional data [Hubert et al., 2005].

In order to explain how the calculation of the PRESS values can be speeded up for MCD and ROBPCA, we first describe the original algorithms. Then we present how time improvements can be made to end up with a fast CV procedure. Examples are included at the same time to illustrate the differences between the original and the faster methods. In Section 3 we start with the description of the MCD method. Section 4 is devoted to the ROBPCA algorithm.

3 The MCD method

3.1 The full algorithm

The objective of the raw MCD is to find $h > \frac{n}{2}$ observations out of n whose covariance matrix has the smallest determinant. Its breakdown value is $\lfloor n - h + 1 \rfloor / n$, hence the number h determines the robustness of the estimator. For its computation, the FAST-MCD algorithm [Rousseeuw and Van Driessen, 1999] can be used. It roughly proceeds as follows :

- (1) Construct many random $(p + 1)$ -subsets, which are enlarged to initial

h -subsets using a C-step as explained in the next step. If it is computationally feasible all the possible $(p + 1)$ -subsets are used. Else 500 $(p + 1)$ -subsets are drawn.

- (2) Within each h -subset, two C-steps are performed. According to [Rousseeuw and Van Driessen, 1999] two C-steps are sufficient to distinguish h -subsets that lead to a robust solution from those that contain outliers. Basically a C-step consists of computing first the classical center \mathbf{m}_h and the classical covariance matrix S_h of the h observations. Then the robust distance (which depends on \mathbf{m}_h and S_h) of each point is computed as :

$$\text{RD}_{\mathbf{m}_h, S_h}(\mathbf{x}_i) = \sqrt{(\mathbf{x}_i - \mathbf{m}_h)' S_h^{-1} (\mathbf{x}_i - \mathbf{m}_h)}. \quad (7)$$

A new h -subset is formed by the h observations with smallest robust distance. The initial h -subsets are constructed by performing C-steps on the $(p + 1)$ -subsets from step 1 of this algorithm.

- (3) For the 10 h -subsets with the best value for the objective function (the determinant of S_h), C-steps are performed until convergence. As studied in [Rousseeuw and Van Driessen, 1999], considering only 10 h -subsets is sufficient to find the optimal h -subset. For large data sets, other time saving techniques can be applied.
- (4) The raw estimates of location \mathbf{m}_{raw} and scatter S_{raw} are the classical mean and classical covariance matrix (multiplied with a consistency factor) of the h -subset H_1 with lowest objective function.
- (5) Next, a reweighting step is applied for efficiency purposes. Every observation is given a weight based on its robust distance $\text{RD}_{\mathbf{m}_{\text{raw}}, S_{\text{raw}}}(\mathbf{x}_i)$. When this robust distance exceeds $\sqrt{\chi_{p,0.975}^2}$, the weight of case i is set equal to 0 and else to 1. The classical mean and covariance matrix of the weighted observations are the final robust center \mathbf{m}_{MCD} and scatter matrix S_{MCD} .
- (6) Finally, the principal components are defined as the k eigenvectors of S_{MCD} which belong to the k largest eigenvalues of S_{MCD} . These principal components are columnwise stored in the loading matrix $P_{p,k}$. Analogously to (2), an estimate of \mathbf{x}_i in the k -dimensional space spanned by these components is given by

$$\hat{\mathbf{x}}_{i,k} = P_{p,k} P_{p,k}' (\mathbf{x}_i - \mathbf{m}_{\text{MCD}}) + \mathbf{m}_{\text{MCD}}. \quad (8)$$

3.2 The fast cross-validated algorithm

In order to compute the robust R-PRESS value (5), we need \mathbf{m}_{-i} and S_{-i} , which are estimates of the covariance matrix and center of the data set without observation i . In the naive approach the FAST-MCD algorithm is fully applied n times. This takes a lot of time, as the algorithm is based on random resampling (see step 1). Therefore we have developed an approximate estimate

for \mathbf{m}_{-i} and S_{-i} . First note that \mathbf{m}_{-i} and S_{-i} are the MCD estimates for a data set with $n - 1$ observations. In order to retain a breakdown value as close as possible to the breakdown value of the full MCD, which is $\lfloor n - h + 1 \rfloor / n$, we define the raw MCD estimator on $n - 1$ points as the mean and covariance matrix of the $h - 1$ observations with smallest covariance determinant.

We start by performing the MCD algorithm on the whole data set. We store the h -subset H_1 , the center \mathbf{m}_{raw} and the covariance matrix S_{raw} before weighting. We do not consider this preliminary step as part of the cross-validation procedure as the full MCD results are required anyway.

Then we repeat the following steps for each case $i = 1, \dots, n$:

- (1) Remove observation i from the set of n observations.
- (2) Now we have to find the $(h - 1)$ -subset $H_{-i,1}$ with lowest objective function. Instead of obtaining it by resampling, we just use an update of H_1 . This yields an approximate, but a very fast solution. When H_1 contains the i th observation, we take the remaining $(h - 1)$ points of H_1 . On the other hand, when sample i does not belong to H_1 , the $(h - 1)$ points of H_1 with the smallest robust distance $\text{RD}_{\mathbf{m}_{\text{raw}}, S_{\text{raw}}}(\mathbf{x}_i)$ are used to construct $H_{-i,1}$. Denote \mathbf{x}_r as the observation which has been removed from H_1 , or $H_{-i,1} = H_1 \setminus \{\mathbf{x}_r\}$. Remark that for all observations outside H_1 , $H_{-i,1}$ needs to be computed only once.
- (3) Next, we compute $\mathbf{m}_{-i,1}$ and $S_{-i,1}$ which are the mean and covariance matrix of the $(h - 1)$ points from $H_{-i,1}$. This can be performed quickly using updates of \mathbf{m}_{raw} and S_{raw} :

$$\begin{aligned}\mathbf{m}_{-i,1} &= \frac{n}{n-1} \left(\mathbf{m}_{\text{raw}} - \frac{1}{n} \mathbf{x}_r \right), \\ S_{-i,1} &= \frac{n-1}{n-2} S_{\text{raw}} - \frac{n-1}{n(n-2)} \left((\mathbf{m}_{-i,1} - \mathbf{x}_r)(\mathbf{m}_{-i,1} - \mathbf{x}_r)' \right).\end{aligned}$$

- (4) To improve this solution, we apply C-steps starting from $\mathbf{m}_{-i,1}$ and $S_{-i,1}$, yielding $\mathbf{m}_{-i,\text{raw}}$ and $S_{-i,\text{raw}}$.
- (5) Finally, we perform a reweighting step based on $\mathbf{m}_{-i,\text{raw}}$ and $S_{-i,\text{raw}}$ as described in step 5 of the MCD algorithm. This yields $\mathbf{m}_{-i,\text{MCD}}$ and $S_{-i,\text{MCD}}$ whose first k principal components can be stored in $P_{-i,k}$. As in equation (8), an estimate of $\hat{\mathbf{x}}_{-i,k}$ is then given by:

$$\hat{\mathbf{x}}_{-i,k} = P_{-i,k} P_{-i,k}' (\mathbf{x}_i - \mathbf{m}_{-i}) + \mathbf{m}_{-i}. \quad (9)$$

The R-PRESS value (5) can then be computed from the set of $\hat{\mathbf{x}}_{-i,k}$ estimates. The important steps which accelerate the approximate procedure are thus the update of H_1 (which avoids full resampling) and the use of update formulas for $\mathbf{m}_{-i,1}$ and $S_{-i,1}$.

3.3 The rat data

We illustrate the performance of our fast algorithm on a real data set. The rat data [Wen et al., 1998] contain the expression of $n = 112$ genes in the development of the central nervous system from Sprague-Dawley albino rats. Measurements were taken on $p = 9$ different time points. The first five variables correspond with the embryonic days 11 until 21. The following three variables represent the first 14 postnatal days. The last variable corresponds with adulthood. Note that in [Wen et al., 1998] the authors have normalized the data by dividing each row by its maximum value. This approach is however not very robust as it depends heavily on the largest value measured for each gene. Hence we did not apply this normalization and performed the analysis on the raw data. Applying the MCD estimator on this data set yielded several genes with higher expression levels [Hubert and Engelen, 2004].

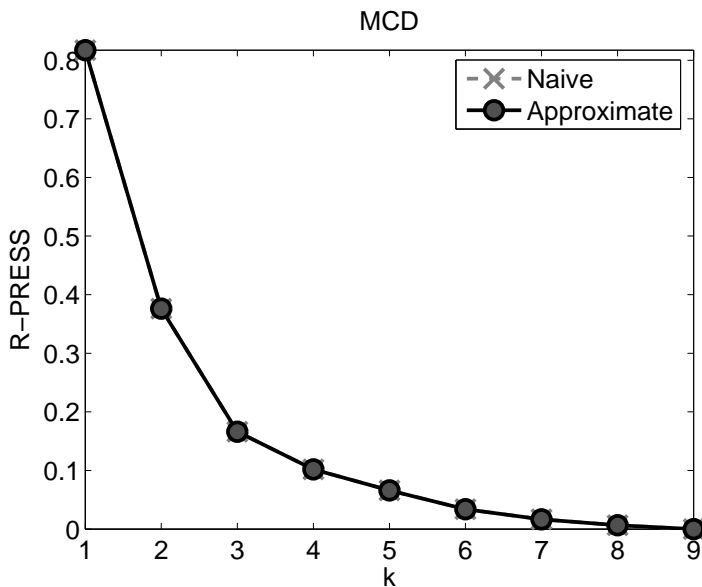


Figure 1. The R-PRESS curves for the rat data.

Figure 1 contains the R-PRESS curve obtained with the approximate algorithm. On this plot we have also drawn the R-PRESS curve obtained in the naive way, i.e. by removing each observation and computing the full MCD algorithm on the remaining data points. Both curves almost coincide, hence our approximation works very well here. The approximate cross-validated MCD is thus as accurate as the naive one, but it is much faster. The computation of the whole R-PRESS curve only requires 2.06 seconds versus 378.25 seconds for the naive cross-validation.

4 The ROBPCA method

4.1 The full algorithm

For high-dimensional regressors ($p > n$) we can not use the MCD anymore because the determinant of a covariance matrix of $h < p$ observations will always be zero and thus can not be minimized. The ROBPCA method [Hubert et al., 2005] handles this problem by combining projection pursuit ideas in the high-dimensional space with MCD estimation in a lower dimensional subspace.

Four major steps can be distinguished in the ROBPCA procedure:

- (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.
- (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).
- (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 reweighting step is included.
- (4) In the last stage, a robust center and covariance matrix are computed within the obtained k -dimensional subspace. To this end, the reweighted MCD estimator is applied to the projected data. From these computations, we retain two h -subsets that will be used in the fast cross-validated algorithm. The first one is H_{freq} that contains the h observations which are most frequently selected in the whole resampling part (steps 1 to 3 of the MCD algorithm). Secondly, we denote 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 $P_{p,k}$. The back-transformed center \mathbf{m} serves as the robust center of the data. The estimated value $\mathbf{x}_{i,k}$ can then again be obtained as in (2).

4.2 The fast cross-validated algorithm

Naive cross-validation would consist of removing n times an observation i from the data set and of calculating the whole ROBPCA procedure for every $k = 1, \dots, k_{\max}$. As we have seen in the description of Section 4.1, the full ROBPCA algorithm uses two types of resampling. The first one is required to compute the outlyingness of each point, whereas the second one is part of the MCD algorithm. This makes naive cross-validation extremely time consuming. As for the cross-validated MCD estimator, our faster method avoids both resampling steps by using information from the full analysis. Again we will now look for optimal subsets of size $h - 1$ when an observation is removed from the data set.

We start by performing ROBPCA on the whole data set with $k = k_{\max}$ and retain H_0, H_1 and H_{freq} . Note that H_0 does not depend on the choice of k_{\max} .

Then we repeat the following steps for each observation $i = 1, \dots, n$.

- (1) Remove sample i from the data.
- (2) We now have to find the $h - 1$ observations with smallest outlyingness. For this, we just update H_0 in a similar way as for the MCD method. If the i th case belongs to H_0 , we set $H_{-i,0} = H_0 \setminus \{\mathbf{x}_i\}$. Else, the point with the largest outlyingness is deleted from H_0 .
- (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 singular value decomposition on $H_{-i,0}$, thereby using the kernel version for data with more variables than cases [Wu et al., 1997a]. Note that doing so, we obtain the same results as if we would first perform an SVD on the reduced data set with sample i , yielding a data set of dimension $d = \min\{n - 2, p\}$ and then would compute the k_{\max} -dimensional subspace spanned by $H_{-i,0} \subset \mathbb{R}^d$. The reason is that the SVD in the first stage of ROBPCA is done without loss of information, i.e. all singular values are retained.
- (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 $H_{-i,\text{freq}}$. The updates of H_1 and H_{freq} are obtained analogously to that of H_0 : if the i th 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 \mathbf{m} and covariance matrix S .
- (5) For each k , the loading matrix $P_{p,k}$ is now obtained as the (backtransformed) k dominant eigenvectors of S . Also the center \mathbf{m} is transformed

to the original p -dimensional data space. The cross-validated fitted value $\hat{\mathbf{x}}_{-i,k}$ is then calculated according to (9).

This approximate cross-validated algorithm does not solely avoid the resampling steps, it also projects the data immediately on a k_{\max} -dimensional subspace. This avoids to recompute the MCD estimator in $k = 1, 2, \dots$ up to k_{\max} dimensions. In [Engelen et al., 2005] it is shown that this approximation yields very accurate results compared to the original ROBPCA method if k_{\max} is chosen not too large. We often take $k_{\max} = 10$ but this can be modified depending on the sample size.

4.3 The egg data

The cross-validation approach for ROBPCA is illustrated on the egg data set, which is kindly provided by Bart Kemps from the Faculty of Agricultural and Applied Biological Sciences of the Katholieke Universiteit Leuven [Kemps et al., 2006].

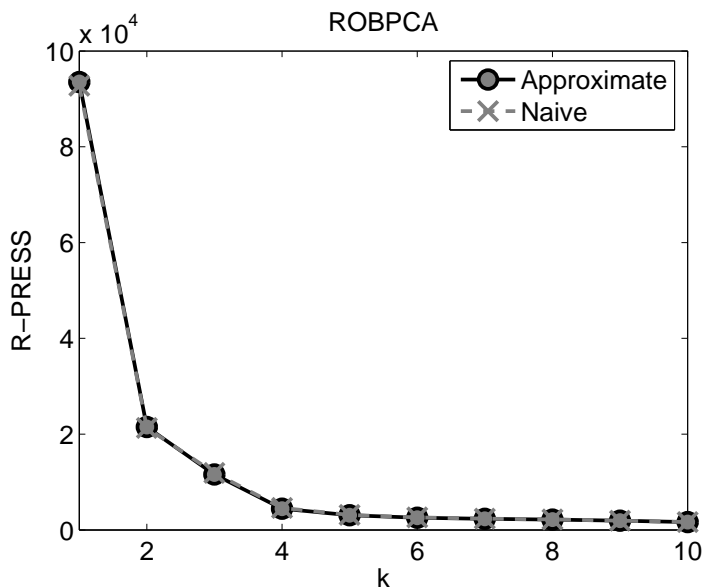


Figure 2. The R-PRESS curves for the egg data.

This data set contains near-infrared (NIR) transmission spectra over $p = 753$ wavelengths from 307nm to 933nm of $n = 727$ eggs. Applying ROBPCA on the full data set revealed several outliers. As we want to compare our fast cross-validation approach with the naive one, we illustrate the results on a subset of 80 observations, randomly drawn out of the whole data set, such that it was still possible to carry out the naive approach. Then we verified that this subset still contained outliers. Figure 2 exposes the R-PRESS values for the ROBPCA method. Here we see that the approximate approach is very

close to the curve obtained with naive CV. The gain in computation time is again striking with 3790 seconds for the naive versus only 12 seconds for the fast algorithm. We have also run the fast algorithm on the whole set of 727 observations. This took 2839 seconds, which is even less than the running time for the naive method on 80 samples. This shows that the approximate cross-validation method can even be used in larger data sets.

5 Simulations

In this section we present the results of several simulations which are performed to investigate the accuracy of our approximate methods and the time reduction on simulated data. To evaluate the simulation results, we compare the R-PRESS curves and the computation time as in the previous examples.

5.1 The MCD estimator

To test the performance of the cross-validated MCD algorithm, we have generated low-dimensional data from a mixture of multivariate normal distributions:

$$(1 - \epsilon)N_p(\boldsymbol{\mu}, \Sigma) + \epsilon N_p(\tilde{\boldsymbol{\mu}}, \Sigma). \quad (10)$$

We have considered models without outliers ($\epsilon = 0$) as well as contaminated models ($\epsilon = 10\%$). The center was taken as $\boldsymbol{\mu} = \mathbf{0}$. We report the results of two simulation settings:

- (1) $n = 30, p = 5, \Sigma = \text{diag}(10, 8.5, 6, 2, 0.1)$ and $\tilde{\boldsymbol{\mu}} = (0, 0, 0, 50, 0)'$. The choice of Σ implies that the first three components explain 92% of the total variation, and the first four components 99.6%.
- (2) $n = 100, p = 10, \Sigma = \text{diag}(10, 9, 7.5, 5, \dots)$, where the dots indicate negligibly small numbers. In case of contamination, $\tilde{\boldsymbol{\mu}} = (0, 0, 0, 0, 0, 75, 0, \dots, 0)'$.

For every situation, we have generated 50 data sets. The presented curves show the average of the R-PRESS values over these 50 data sets. Moreover, we have added bars on the R-PRESS curves for each k defined by

$$\text{mean}(\text{R-PRESS}_k) \pm 2 \frac{\text{std}(\text{R-PRESS}_k)}{50},$$

as a measure of the variation of the obtained simulation results. The computation times are also averaged.

Figure 3 contains the R-PRESS curves for the first simulation setting, and Figure 4 for the second one. We see again that the naive and the approximate

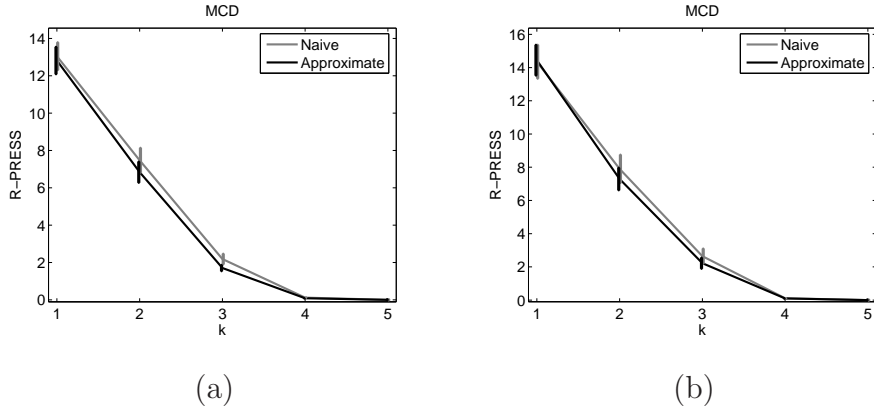


Figure 3. The MCD R-PRESS curves for the first simulation setting with $n = 30$, $p = 5$ (a) without outliers; and (b) with 10% contamination.

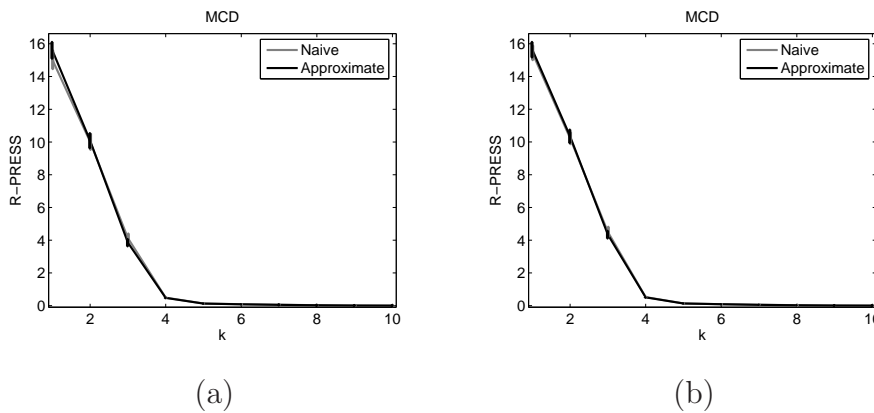


Figure 4. The MCD R-PRESS curves for the second simulation setting with $n = 100$, $p = 10$ (a) without outliers; and (b) with 10% contamination.

curves are very close to each other. The confidence bars indicates that both algorithms have a similar variability and for most values of k there is an overlap between the confidence intervals of the naive and the approximate method. Moreover, we notice that the R-PRESS values are indeed robust, as their values do not change a lot when the data set contains outliers.

The average computation times and their standard errors are reported in Table 1. We see that the average duration of one simulation is roughly 80 to 100 times smaller with the approximate than with the naive cross-validation. A huge time reduction is thus obtained.

Table 1

Average computation times and standard deviations (between brackets) in seconds for the approximate and the naive cross-validated MCD method.

n	p	Approx.	Naive
30	5	1.32 (0.07)	134.84 (6.57)
100	10	3.52 (0.13)	365.79 (19.75)

5.2 ROBPCA

To investigate the performance of the ROBPCA procedure, we have again generated data from the contamination model (10) with $\boldsymbol{\mu} = \mathbf{0}$ and with 0% or 10% contamination. But now, two high-dimensional situations are considered where the number of observations is smaller than the number of variables:

- (1) $n = 30, p = 50, \Sigma = \text{diag}(10, 7.5, 5, 3, \dots)$ and $\tilde{\boldsymbol{\mu}} = (0, 0, 0, 0, 0, 50, 0, \dots, 0)'$
- (2) $n = 100, p = 500, \Sigma = \text{diag}(10, 7.5, 5, 3, \dots)$ and $\tilde{\boldsymbol{\mu}} = (0, 0, 0, 0, 0, 50, 0, \dots, 0)'$.

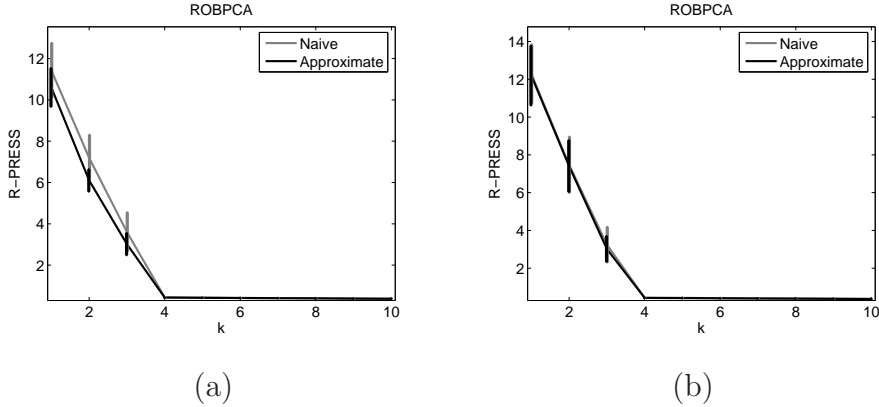


Figure 5. The ROBPCA R-PRESS curves for the first simulation setting with $n = 30, p = 50$ (a) without outliers; and (b) with 10% contamination.

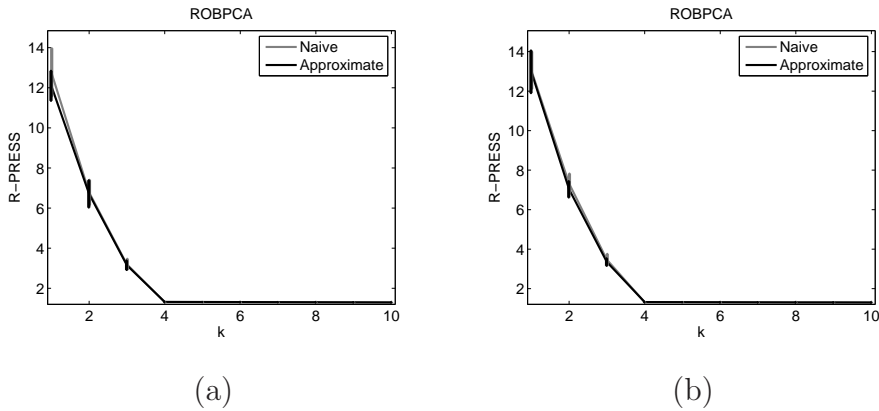


Figure 6. The ROBPCA R-PRESS curves for the second simulation setting with $n = 100, p = 500$ (a) without outliers; and (b) with 10% contamination.

In both settings, the first four eigenvalues are considerably larger than the remaining ones. To obtain R-PRESS curves, we have generated 10 data sets (as the naive method was too time-consuming to consider more).

Figures 5 and 6 show the R-PRESS curves together with the confidence limits for the two settings. It can again be seen that the naive and the approximate curves are very similar and have a similar variation.

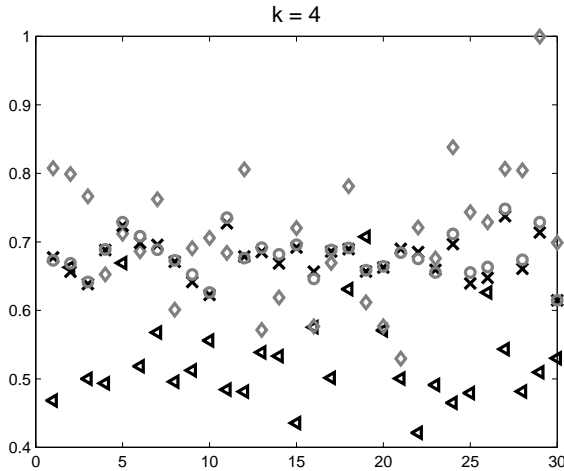


Figure 7. For each observation from a data set with $n = 30$, $p = 50$ and $k = 4$, the orthogonal distance based on ROBPCA applied to the full data set (triangle), the cross-validated distance based on the naive algorithm (circle) and on the approximate algorithm (cross) and orthogonal distances from a validation set (diamond).

Table 2

Mean and standard deviation of the orthogonal distances plotted in Figure 7.

	No CV	Test	Approx.	Naive
mean	0.53	0.71	0.68	0.68
stddev.	0.07	0.10	0.03	0.03

To illustrate the closeness of the approximate and naive algorithms, we have also displayed for each observation the orthogonal distances resulting from both procedures. For this we consider one data set from the first simulation setting and select $k = 4$. The situation without contamination is depicted in Figure 7. On this plot we have drawn for each case $i = 1, \dots, n$ its orthogonal distance $OD_{i,k} = \|\mathbf{x}_i - \hat{\mathbf{x}}_{i,k}\|$ as defined in (3) (marked with a triangle), its cross-validated orthogonal distance $OD_{-i,k} = \|\mathbf{x}_i - \hat{\mathbf{x}}_{-i,k}\|$ obtained with the naive algorithm (marked with a circle) as well as the one obtained with the fast method (marked with a cross). First we see again that both cross-validated orthogonal distances do not differ very much. Next we see that the orthogonal distances $OD_{i,k}$, which are a byproduct of ROBPCA applied to the full data set, are all much smaller than their cross-validated values. To have an indication which residuals are the most appropriate to use for prediction, we have finally generated a validation set of 30 uncontaminated cases and we have as well plotted their orthogonal distance (with a diamond) on Figure 7. We see that the orthogonal distances of the validation set are much closer to the cross-validated OD than to the ordinary OD of the training set. This nicely illustrates that the cross-validated residuals are better for prediction. Table 2 contains the mean and standard deviation of the orthogonal distances for the four procedures, from which the same conclusions can be distracted.

In Figure 8(a) we consider the contaminated data set with three clear outliers. It can be seen that their orthogonal distances almost fall together. To have a better view on the regular points in this data set, we have plotted their residuals in Figure 8(b) and tabulated the summary statistics in Table 3. The same conclusion can be drawn as for Figure 7.

Finally, Table 4 reports the differences in running time between the naive and fast ROBPCA algorithm. Here, the gain in computer time is even more striking than the benefit obtained for the MCD method.

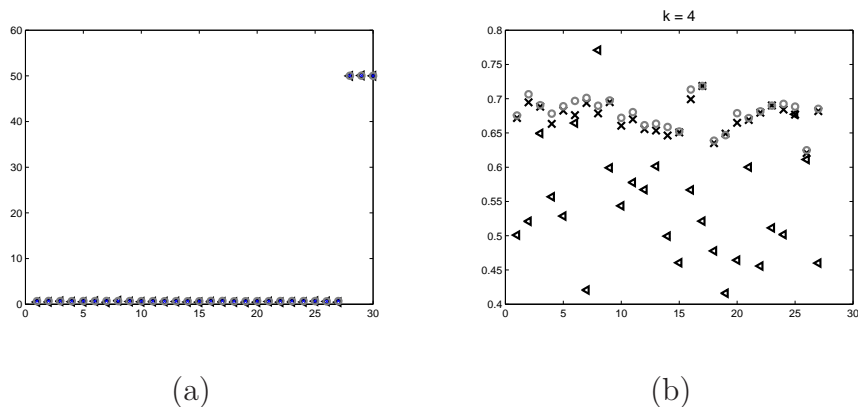


Figure 8. Orthogonal distances from a data set with $n = 30$, $p = 50$ and $k = 4$ and three outliers: (a) all observations and (b) the regular observations.

Table 3

Mean and standard deviation of the orthogonal distances plotted in Figure 8(b).

	No CV	Approx.	Naive
mean	0.54	0.67	0.68
stddev.	0.08	0.02	0.02

Table 4

Average computation times and standard deviations (between brackets) in seconds for the approximate and the naive cross-validated ROBPCA method.

n	p	Approx.	Naive
30	50	1.72 (0.02)	1265.7 (84.0)
100	500	19.03 (0.10)	5191.1 (36.8)

6 Discussion and Conclusion

We have constructed fast algorithms to perform cross-validation on high-breakdown estimators for robust covariance estimation (MCD) and principal components analysis (ROBPCA). These algorithms allow a fast computation of a robust PRESS curve, that can be used to select the number of principal

components. This is very important as PCA is often only used as the first step in a data analysis. We have applied our fast methods to construct a robust method for the classification of high-dimensional data [Vanden Branden and Hubert, 2005] and to select the number of latent variables in principal component regression and partial least squares regression (see [Engelen and Hubert, 2005]).

Other types of reweighting schemes could also be used instead of taking the minimum as proposed in (6). A first idea is to replace the minimum by the median, which has already been considered in [Engelen and Hubert, 2005]:

$$w_{i,\text{median}} = \text{median}_k(w_{i,k}). \quad (11)$$

In case of the minimum, all the observations that are outlying in at least one of the models receive zero weight, whereas with the median definition we only exclude observations that are outlying with respect to the majority of the models under consideration. This is more appropriate when many $w_{i,k}$ become outlying, for example if k_{max} is chosen very large.

Smoother weights can also be considered with $w_{i,k} \sim 1/\text{OD}_{i,k}$ such that the outliers are not completely ignored but a (small) weight is assigned to them. This has the advantage that outliers which are not very extreme are still partially taken into account. This approach has been considered in [Ronchetti and Staudte, 1994] and [Ronchetti et al., 1997] for model selection in linear regression with M-estimators.

Yet another procedure to obtain robust PRESS values could consist of computing trimmed squared errors in which case a fixed proportion (e.g. $\alpha = 50\%, 75\%, \dots$) of the smallest prediction errors are averaged. Such an R-PRESS_k curve then yields a clear interpretation, as it indicates which model fits 100 $\alpha\%$ of the data the best. Next, it could be interesting to consider R-PRESS_k curves for a range of α -values (Ruben Zamar, personal communication).

To conclude, we note that all our functions are implemented in MATLAB and are part of LIBRA: Matlab Library for Robust Analysis [Verboven and Hubert, 2005] which can be downloaded at <http://wis.kuleuven.be/stat/robust.html>.

References

- L. Davies. Asymptotic behavior of S-estimators of multivariate location parameters and dispersion matrices. *The Annals of Statistics*, 15:1269–1292, 1987.
- S. Engelen and M. Hubert. Fast model selection for robust calibration. *Analytica Chimica Acta*, 544:219–228, 2005.

- S. Engelen, M. Hubert, and K. Vanden Branden. A comparison of three procedures for robust PCA in high dimensions. *Austrian Journal of Statistics*, 34:117–126, 2005.
- M. Hubert and S. Engelen. Robust PCA and classification in biosciences. *Bioinformatics*, 20:1728–1736, 2004.
- M. Hubert, P.J. Rousseeuw, and K. Vanden Branden. ROBPCA: a new approach to robust principal components analysis. *Technometrics*, 47:64–79, 2005.
- I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 1986.
- B.J. Kemps, F.R. Bamelis, B. De Ketelaere, K. Mertens, K. Tona, E. M. Decuyper, and J. G. De Baerdemaeker. Visible transmission spectroscopy for the assessment of egg freshness. *Journal of science of food and agriculture*, 2006. Accepted.
- A.D.R. McQuarrie and C.L. Tsai. *Regression and Time Series Model Selection*. World Scientific Publishing, Singapore, 1998.
- E. Ronchetti, C. Field, and W. Blanchard. Robust linear model selection by cross-validation. *Journal of the American Statistical Association*, 92:1017–1023, 1997.
- E. Ronchetti and R.G. Staudte. A robust version of Mallows’s C_p . *Journal of the American Statistical Association*, 89:550–559, 1994.
- P.J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880, 1984.
- P.J. Rousseeuw and K. Van Driessen. A fast algorithm for the Minimum Covariance Determinant estimator. *Technometrics*, 41:212–223, 1999.
- P.J. Rousseeuw and V.J. Yohai. Robust regression by means of S-estimators. In J. Franke, W. Härdle, and R.D. Martin, editors, *Robust and Nonlinear Time Series Analysis*, pages 256–272, New York, 1984. Lecture Notes in Statistics No. 26, Springer-Verlag.
- J. Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88:486–494, 1993.
- K. Vanden Branden and M. Hubert. Robust classification in high dimensions based on the SIMCA method. *Chemometrics and Intelligent Laboratory Systems*, 79:10–21, 2005.
- S. Verboven and M. Hubert. LIBRA: a Matlab library for robust analysis. *Chemometrics and Intelligent Laboratory Systems*, 75:127–136, 2005.
- X. Wen, S. Fuhrman, G.S. Michaels, D.B. Carr, S. Smith, J.L. Barker, and R. Somogyi. Large-scale temporal gene expression mapping of the central nervous system development. *Proceedings of the National Academy of Sciences of the USA*, 95:334–339, 1998.
- W. Wu, D.L. Massart, and S. de Jong. The kernel PCA algorithms for wide data. Part I: Theory and algorithms. *Chemometrics and Intelligent Laboratory Systems*, 36:165–172, 1997a.
- 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*, 37:271–280, 1997b.