Algorithms for Projection-Pursuit Robust Principal Component Analysis

C. Croux[*], P. Filzmoser[†] and M.R. Oliveira[‡]

[*] Christophe Croux, University Centre of Statistics and Faculty of Economics and Applied Economics, K.U. Leuven, Naamsestraat 69, B-3000 Leuven, Belgium; Christophe.Croux@econ.kuleuven.be.

[†] Peter Filzmoser, Dept. of Statistics & Probability Theory, Vienna University of Technology, Wiedner Hauptstraße 8-10, A-1040 Vienna, Austria; P.Filzmoser@tuwien.ac.at.

[‡] M. Rosario Oliveira, Department of Mathematics and Center for Mathematics & its Applications, Instituto Superior Técnico, Lisboa, Portugal; Rosario.Oliveira@math.ist.utl.pt

Algorithms for Projection-Pursuit Robust Principal Component Analysis

Abstract

The results of a standard Principal Component Analysis (PCA) can be affected by the presence of outliers. Hence robust alternatives to PCA are needed. One of the most appealing robust methods for principal component analysis uses the Projection-Pursuit principle. Here, one projects the data on a lower-dimensional space such that a robust measure of variance of the projected data will be maximized. The Projection-Pursuit based method for principal component analysis has recently been introduced in the field of chemometrics, where the number of variables is typically large. In this paper, it is shown that the currently available algorithm for robust Projection-Pursuit PCA performs poor in presence of many variables. A new algorithm is proposed that is more suitable for the analysis of chemical data. Its performance is studied by means of simulation experiments and illustrated on some real data sets.

Keywords: Multivariate Statistics, Optimization, Numerical Precision, Outliers, Robustness, Scale Estimators.

1 Introduction

Projection-Pursuit (PP) methods aim at finding structures in multivariate data by projecting them on a lower-dimensional subspace [1]. PP methods are well suited for the analysis of data sets with a large number of variables, often encountered in chemometrics, since they involve a dimension reduction step. The lower-dimensional subspace, often of dimension one, is selected by maximizing a certain *Projection Index.* Principal Component Analysis (PCA) is an example of the PP approach. If we have n observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, all of them column vectors of dimension p, then the first principal component is obtained by finding the unit vector $\boldsymbol{a}$ which maximizes the variance of the data projected on it:

$$\boldsymbol{a}_1 = \underset{\|\boldsymbol{a}\|=1}{\operatorname{argmax}} S^2(\boldsymbol{a}^t\boldsymbol{x}_1, \ldots, \boldsymbol{a}^t\boldsymbol{x}_n), \tag{1}$$

where S^2 stands for the variance. By projecting the data on the direction $\boldsymbol{a}_1$, we obtain univariate data $\boldsymbol{a}_1^t\boldsymbol{x}_1, \ldots, \boldsymbol{a}_1^t\boldsymbol{x}_n$, in accordance with the Projection-Pursuit idea. Taking the variance as a projection index leads, as well known, to standard PCA. However, the outcome of standard PCA is very sensitive to atypical observations, e.g. gross errors or sources of new variation. Hence there is a need for robust principal components, yielding reliable results in presence of outliers, and describing well the structure of the majority of the data.

One of the most appealing procedures for robust PCA was proposed by Li and Chen [2], and is based on the PP principle. The idea is very simple: instead of taking the variance as a projection index in (1), a robust measure of variance is taken. It was shown in [2] and [3] that these estimators are consistent and asymptotically normal, and recently [4] proved its robustness by computing influence functions and breakdown points. This approach to robust PCA was introduced in the chemometrics literature by [5] and [6], and was shown to be appropriate for use in chemometrical data analysis. Recently, it was included in TOMCAT, a MATLAB toolbox for multivariate calibration techniques [7]. A recent review on robust statistics in data analysis is given in [8].

The topic of this paper is the actual computation of the principal components, i.e. solving the maximization problem in (1). When the projection index is the usual standard deviation, then it is known that $\boldsymbol{a}_1$ is the eigenvector of the sample covariance matrix of the data corresponding to the largest eigenvalue (see e.g. [9]). But for other choices of S, solving (1) is not easy, and one needs to rely on approximative algorithms. A first algorithm was introduced by Li and Chen [2] which, however, was complicated, time consuming, and not made publicly available. In [10] simulated annealing was used, again yielding a slow algorithm that is not publicly available. On the other hand, [11] introduced an algorithm which is very simple, fast to compute and easy to implement. We call it the CR algorithm and consider it as the basis algorithm. Equivalent, but numerically more stable versions were implemented by [5] and [4], see also the discussion in [12]. In this paper we will work with the CR algorithm as described in [4], but [5] and [12] share the same properties, and give exactly the same outcomes, upon rounding error. It was never noticed, however, that the CR algorithm may implode in presence of many variables, making it less suitable for the typical high-dimensional chemometrical applications. This implosion results in a severe downward bias of the scale estimators. This paper will propose a new algorithm not suffering from this problem, being much more precise, while still being computationally efficient. We call this new algorithm the GRID algorithm.

Other proposals, besides the PP approach, have been made for robust PCA, as in [13], [14], or [15]. We would like to emphasize that this paper is not on a comparison between different types of *estimators* for robust PCA, since such a comparison was already made in the aforementioned papers. This paper is on a comparison between different *algorithms* for computing the PP based estimator defined by (1). We recall that the PP approach has a clear definition, is intuitively appealing, and has good statistical and robustness properties. It can be applied if the data set has more variables than observations, a case of special interest in chemometrics. A particular feature is that the PP based approach for PCA allows sequential estimation of the principal components. If we have already computed the $(k-1)$th principal component, then the direction of the kth component, with $1 < k \leq p$, is defined as the unit vector maximizing the index S^2 of the data projected on it, under the condition of being orthogonal to all previously obtained components:

$$\boldsymbol{a}_k = \underset{\|\boldsymbol{a}\|=1, \boldsymbol{a}\perp\boldsymbol{a}_1,\ldots,\boldsymbol{a}\perp\boldsymbol{a}_{k-1}}{\operatorname{argmax}} S^2(\boldsymbol{a}^t\boldsymbol{x}_1, \ldots, \boldsymbol{a}^t\boldsymbol{x}_n). \tag{2}$$

Hence, one is not obliged to compute all p different principal components, but could stop after, for example, $k = 2$ steps, implying a significant reduction in computation time if p is large.

In Section 2 of this paper we briefly review the CR algorithm and in Section 3 we prove that it suffers from implosion in presence of many variables. Section 4 outlines the new GRID algorithm for finding the optimum values in (1) or (2). Section 5 compares the different algorithms through some numerical experiments. It turns out that the GRID algorithm finds, throughout practically all experiments, the highest values for the maximum in (1), at a reasonable computational price. We also compare the performance of the algorithms on some real data examples.

2 Description of the Croux-Ruiz (CR) Algorithm

The input of all algorithms we will discuss is a data matrix $\boldsymbol{X}$ with n rows (observations) and p columns (variables), having the observations $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^t$ in its rows. Let us first state that if there are more variables than observations, hence $p > n$, the first step of any algorithm we discuss is to perform a singular value decomposition, and continue working with the scores on all components in an n dimensional space. This is a standard preprocessing step, yielding no loss of information since we keep the full rank of the data matrix, e.g. [6]. Hence, we may assume without loss of generality that $p \leq n$ when describing the algorithms.

An algorithm will give us a sequence of approximations for the unit vectors defined by (1) and (2), but also a sequence of maximal values for the (robust) variances, which we denote by

$$\lambda_k = S^2(\boldsymbol{a}_k^t\boldsymbol{x}_1, \ldots, \boldsymbol{a}_k^t\boldsymbol{x}_n), \tag{3}$$

for $1 \leq k \leq p$. If we take for S^2 the sample variance, then the values of λ_k are the eigenvalues of the sample covariance matrix, sorted from the largest to the smallest. For a robust PCA, we will consider robust versions of S^2, but we will continue to call the λ_k, as defined by (3), "eigenvalues". Similar as in [4], [5], and [6], we focus on two particular choices: the Median Absolute Deviation (MAD),

$$\mathrm{MAD}(z_1, \ldots, z_n) = 1.48 \operatorname*{med}_j |z_j - \operatorname*{med}_i z_i|, \tag{4}$$

and the first quartile of the pairwise differences between all data points

$$Q(z_1, \ldots, z_n) = 2.22\{|z_i - z_j|; 1 \leq i < j \leq n\}_{(\binom{n}{2}/4)}, \tag{5}$$

where $\{z_1, \ldots, z_n\}$ is any given univariate data set (see [16]). Taking the squared MAD or Q, yields then a robust variance estimator. The reason why the MAD or the Q estimators are taken is that they attain, as a scale estimator, the maximal breakdown point of 50%. The breakdown point is the most popular measure of robustness of an estimator. It measures the smallest fraction of outliers in the data that is needed to drive the scale estimators to their extreme values, i.e. zero for breakdown to zero, and infinity for explosion breakdown.

Before starting the algorithm, we center the data by subtracting the centers of the variables from the columns of the data matrix. Since the arithmetic mean is not robust, the centering is done with the L_1-median [17] or the coordinate-wise median, denoted by $\hat{\boldsymbol{\mu}}(\boldsymbol{X})$. Note that a fast and stable algorithm for computing an L_1 algorithm is described in [18]. We obtain then the centered data $\boldsymbol{x}_i^1 = \boldsymbol{x}_i - \hat{\boldsymbol{\mu}}(\boldsymbol{X})$. Suppose that in step $k-1$, the algorithm returned the vector $\hat{\boldsymbol{a}}_{k-1}$, an approximation for the solution to (2). Then we project the observations on the space spanned by the first $k-1$ optimal directions, by computing

$$\boldsymbol{x}_i^k = \boldsymbol{x}_i^{k-1} - \left(\hat{\boldsymbol{a}}_{k-1}^t \boldsymbol{x}_i^{k-1}\right) \hat{\boldsymbol{a}}_{k-1}, \tag{6}$$

for $1 \leq i \leq n$ and $k > 1$. When searching for the kth optimal direction, the CR algorithm is not optimizing over the whole space of all possible unit vectors, but only considers n trial directions in the set

$$A_{n,k} = \left\{ \frac{\boldsymbol{x}_1^k}{\|\boldsymbol{x}_1^k\|}, \ldots, \frac{\boldsymbol{x}_n^k}{\|\boldsymbol{x}_n^k\|} \right\}. \tag{7}$$

(Elements in $A_{n,k}$ corresponding to a zero denominator are set equal to zero.) The kth eigenvalue is then approximated by

$$\hat{\lambda}_k = \max_{\boldsymbol{a} \in A_{n,k}} S^2(\boldsymbol{a}^t \boldsymbol{x}_1^k, \ldots, \boldsymbol{a}^t \boldsymbol{x}_n^k), \tag{8}$$

and $\hat{\boldsymbol{a}}_k$ is the argument where the maximum in the above equation is attained.

For more details on the algorithm we refer to [4]. This basic and simple algorithm works well for a large sample size relative to the number of variables, because the set of trial directions that the algorithm considers are pointing in the directions where the data are. This algorithm has been successfully used in different applications (e.g. [19], [20], [21], [6], [22]). However, due to its construction, the CR algorithm may implode in situations with a low sample size relative to the number of variables.

3 Implosion of the CR algorithm

In this section we will show that the CR algorithm suffers from severe downward bias. In fact, from a certain order k on, all estimated eigenvalues will be zero, independently of the data. We will formally prove that if $k > n/2$, then $\hat{\lambda}_k = 0$ when using the CR algorithm. This is not only happening when taking for S the MAD or the Q, but for any robust scale measure with a high breakdown point. This implosion of the eigenvalues occurs for *all* data sets, whether there are outliers present or not, whatever the latent structure in the data is. This is an undesirable artefact of the CR algorithm.

For stating the next proposition, we need a condition on the scale estimator S. A scale estimator S has the *m-exact fit property* if $S(Z) = 0$ for any sample Z containing at least m zeros.

Proposition 1 *If a scale estimator S has the m-exact fit property, then the CR algorithm yields at any given data set*

$$\hat{\lambda}_k = 0 \qquad \textit{for } k = m+1, \ldots, p.$$

Proof: Denote $\boldsymbol{X}^k = \{\boldsymbol{x}_1^k, \ldots, \boldsymbol{x}_n^k\}$, where the notation of Section 2 is used. We will first show by induction that

$$\text{(A) } \boldsymbol{X}^k \text{ contains at least } (k-1) \text{ zero vectors } \mathbf{0}.$$

For $k = 1$, the property obviously holds, and suppose now that (A) holds for $k-2$. It follows from equation (6), that if $\boldsymbol{x}_i^{k-1} = \mathbf{0}$, then also $\boldsymbol{x}_i^k = \mathbf{0}$, yielding already $k-2$ zeros. Denote now i' the index corresponding to the maximum over the set $A_{n,k-1}$ in (8), so $\hat{\boldsymbol{a}}_{k-1} = \boldsymbol{x}_{i'}^{k-1}/\|\boldsymbol{x}_{i'}^{k-1}\|$. But then it follows directly from (6) that $\boldsymbol{x}_{i'}^k = \mathbf{0}$. This yields the $(k-1)$th zero, and confirms (A).

From (8) it follows that, in the kth step of the CR algorithm, S is computed on a sample containing $k-1$ zeros, and this for every value of $\boldsymbol{a}$. Hence, we conclude that if S has the m-exact fit property with $m \geq k-1$, then $\hat{\lambda}_k = 0$. □

From the definitions (4) and (5) of the MAD and the Q scale estimators, it is easy to check that they have the property that they are equal to zero as soon as $n/2$ of the observations are equal to each other. Hence Proposition 1 applies for MAD and Q with $m = n/2$. In fact, any other robust scale estimator will suffer from the artefact of the CR algorithm. Indeed, a scale estimator is robust, among others, if its value cannot become arbitrary large in presence of a certain number of outliers. This is formalized by its *explosion breakdown point* (cfr. [23], [16]), which at a given sample $Z = \{z_1, \dots, z_n\}$ is defined as

$$\varepsilon^*(S, Z) = \min\{\frac{m}{n}; \sup_{Z'} S(Z') = \infty\} \tag{9}$$

where Z' is a contaminated sample obtained by replacing any m observations of Z by arbitrary values. This next proposition makes a link between the *explosion breakdown point* of a scale estimator and the exact fit property of a scale estimator. The relation between the exact fit property and the breakdown point of an estimator was established for regression estimators by [24], but not yet for scale estimators. Recall that a scale estimator is said to be scale equivariant if $S(az_1, \dots, az_n) = |a|S(z_1, \dots, z_n)$, for any scalar a and any sample $z_1, \dots, z_n$. Scale equivariance is a basic property that all sensible scale estimators share.

Proposition 2 *If a scale equivariant scale estimator S has breakdown point $\varepsilon^*(S, Z) > (m - n)/n$, then it has the m-exact fit property.*

Proof: Suppose that S does not have the m-exact fit property. Then there exists a sample of the form $Z = \{0, \dots, 0, z_1, \dots, z_{n-m}\}$ such that $S(Z) > 0$. Now replace the last $n - m$ observations, simply by multiplying them by a factor λ. Denote the contaminated sample Z'_λ. Then, due to the scale equivariance, $S(Z'_\lambda) = \lambda S(Z)$. Since $S(Z) > 0$, we can let the scale at the contaminated Z'_λ tend to infinity, simply by letting λ tend to infinity, causing explosion breakdown. Due to the given lower bound on the explosion breakdown point of S, this cannot occur. Hence there is a contradiction, and the m-exact fit property should hold. □

Any robust scale estimator S having the maximal value for the breakdown point, being 50% (e.g. [25]), will according to the above proposition have the m-exact fit property, with $m \approx n/2$. Hence the CR algorithm will also implode for other choices of S having a high breakdown point, and not only for the MAD or Q estimator.

We also stress that the RAPCA algorithm proposed by [5] suffers from exactly the same implosion problem. The reason is that the CR and the RAPCA algorithm are mathematically equivalent, and they differ only upon rounding error. The data deflation step is done differently in [5], but it is immediate to see that property (A) in the proof of proposition 1 also holds for the RAPCA algorithm. Note that the CR algorithm as

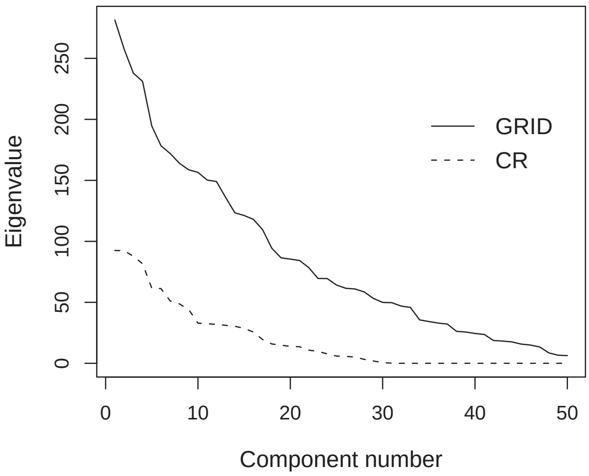

Figure 1: Scree plot of an artifical data set. For the CR algorithm the eigenvalues are strongly biased, and implode after $k = 30$. For the new GRID algorithm no implosion is observed.

described in Section 2 will be faster to compute than the RAPCA algorithm, as was noticed by [6].

In Figure 1 we illustrate the implosion behavior of the CR algorithm. We generate a data set of size $n = 60$ in a space of dimension $p = 50$, similar to the simulation scheme detailed in Section 4, and containing 10 outliers. Figure 1 plots the obtained $\hat{\lambda}_k$ as a function of k, where the MAD was used as a scale estimator. Such a plot is called a *Scree Plot*. We see that all eigenvalues of order larger than $k = 30$ are exactly equal to zero. Of course, one could argue that in practice one is often not interested in the principal components of these higher orders. But we also see that the lower order eigenvalues, although not exactly equal to zero, are subject to severe downward bias to zero. Figure 1 also shows the scree plot when applying the GRID algorithm, which we will explain in the next Section. It is seen that this algorithm is much less vulnerable to downward bias, and finds optimal directions explaining more variability, as shown by their higher values of $\hat{\lambda}_k$.

The downward bias of the CR algorithm will also affect the performance of the outliers detection rule associated with the robust PCA. As robust outlier diagnostics we use a robust version of the Mahalanobis Distances

$$RD_i = \sqrt{(\boldsymbol{x}_i^1)^t \hat{\boldsymbol{\Sigma}}^{-1} \boldsymbol{x}_i^1}, \tag{10}$$

with $\boldsymbol{x}_i^1 = \boldsymbol{x}_i - \hat{\boldsymbol{\mu}}(\boldsymbol{X})$ the centered observations and $\hat{\boldsymbol{\mu}}(\boldsymbol{X})$ the L_1-median. As robust covariance matrix we take

$$\hat{\boldsymbol{\Sigma}} = \sum_{k=1}^{kmax} \hat{\lambda}_k \, \hat{\boldsymbol{a}}_k \hat{\boldsymbol{a}}_k^t,$$

with $kmax$ the number of retained components, i.e. the ones not corresponding to zero (or very small) eigenvalues. Properties of this robust covariance matrix estimator have been discussed in [4], and plugging the above expression in (10) yields the *Robust Distances*

$$RD_i = \sqrt{\sum_{k=1}^{kmax} \frac{\left(\hat{\boldsymbol{a}}_k^t \boldsymbol{x}_i^1\right)^2}{\hat{\lambda}_k}} \qquad (11)$$

for $1 \leq i \leq n$. If the robust distance exceeds the critical value $\chi_{kmax,0.975}$, the observation is flagged as an outlier, see also [5]. For the data set used to construct Figure 1, it follows from inspection of the scree plot that $kmax = 30$ for the CR algorithm, while $kmax = 50$, the maximal value, when using the new GRID algorithm. Recall that this data set contains 10 gross errors, having indices 51 up to 60. Figure 2 plots the robust distances of the observations versus their index, once for the CR algorithm (left panel), and once for the GRID algorithm (right panel). The left panel of Figure 2 shows that the 10 gross-errors are correctly flagged as outliers, but there are also many good observations having a large robust distance, some of them even a larger RD_i than the true outliers. Hence the CR algorithm suffers from the so-called *swamping effect*, meaning that good observations are incorrectly flagged as outliers. This is due to the downward bias of the estimated eigenvalues when using the CR algorithm, making the values in (11) too large. On the other hand, the GRID algorithm clearly succeeds to pinpoint the 10 outliers, and separates well the outliers from the normal observations.

4 The GRID algorithm (GRID)

In this Section we propose the GRID algorithm. It is searching the space of all possible directions more thoroughly. A preliminary version of this algorithm was briefly discussed in [26], where it was applied to the robust continuum regression problem of [12]. As for the CR algorithm, we will first apply a pre-processing step by working with the scores of the singular value decomposition (if $p > n$). In this section, we use the short hand notation $S(\boldsymbol{a}) = S(\boldsymbol{x}_1^t\boldsymbol{a}, \ldots, \boldsymbol{x}_n^t\boldsymbol{a})$, and our aim is to find the $\boldsymbol{a}$ maximizing $S(\boldsymbol{a})$, under the constraint that $\|\boldsymbol{a}\| = 1$. We will first explain in detail the algorithm for finding the first optimal direction $\boldsymbol{a}_1$ in (1).

The basic idea behind the GRID algorithm is that the optimization problem is trivial if $p = 2$. Indeed, if $p = 2$, then the problem reduces to maximizing the function $\theta \rightarrow$

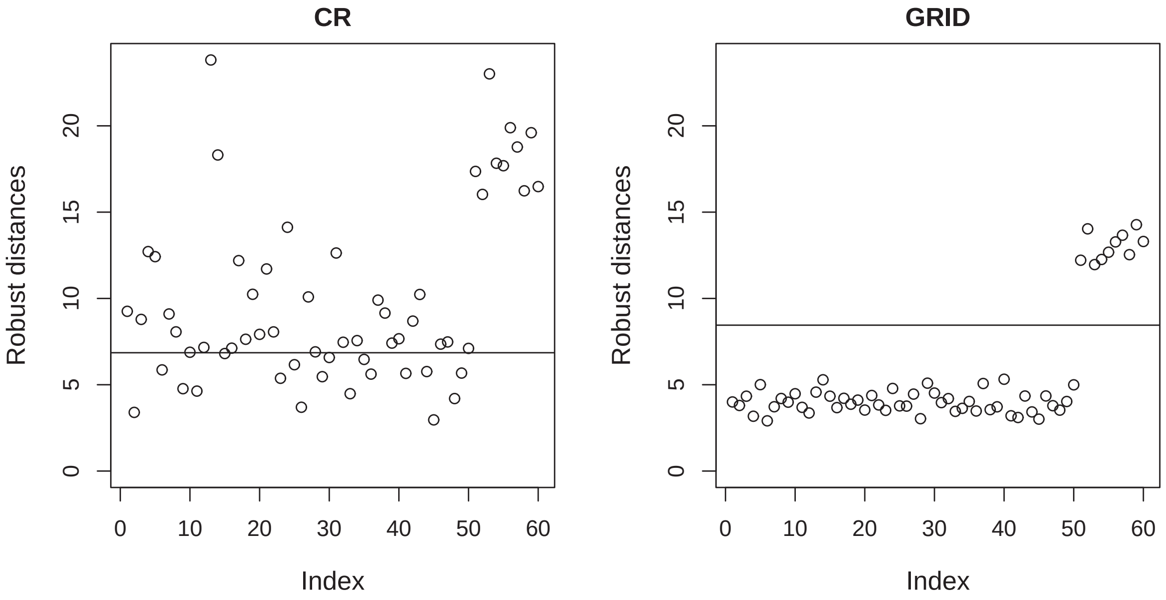

Figure 2: Robust distances versus their indices for the artificial data set of Figure 1. Distances are once computed with the CR algorithm (left panel), and once with the GRID algorithm (right panel).

$S((\cos(\theta), \sin(\theta))^t$ over the interval $[-\pi/2, \pi/2[$. This can be done with an easy grid search, i.e. we divide the interval in $(N_g - 1)$ equal parts, and evaluate the function at the grid points $(-\frac{1}{2} + \frac{j}{N_g})\pi$, for $j = 0, \ldots, N_g - 1$. For N_g large enough, we will get close enough to the solution. Note that a grid search is not asking for differentiability of the function to maximize, and can distinguish a global maximum from a local one.

In the general case, $p > 2$, the idea is to perform an iterative scheme of optimizations in two-dimensional planes. Before starting, we relabel the indices of the columns of the data matrix $\boldsymbol{X}$ in decreasing order of the value of $S_j = S(x_{1j}, \ldots, x_{nj})$, for $j = 1, \ldots, p$. Hence, we may assume that $S(\boldsymbol{e}_1) \geq S(\boldsymbol{e}_2) \geq \ldots \geq S(\boldsymbol{e}_p)$, where $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_p$ are the canonical basis vectors.

1. Start with $\hat{\boldsymbol{a}} = \boldsymbol{e}_1$.

2. For $i = 1, \ldots, N_c$, execute the following cycle.

For $j = 1, \dots, p$

(a) Maximize the objective function in the plane spanned by $\boldsymbol{e}_j$ and $\hat{\boldsymbol{a}}$, by a grid search of the function $\theta \to S(\cos\theta\,\hat{\boldsymbol{a}} + \sin\theta\,\boldsymbol{e}_j)$, where the angle θ is restricted to the interval $[-\pi/(2^i);\pi/(2^i)[$. Denote θ_0 the angle where the maximum is attained over all grid points.

(b) Update $\hat{\boldsymbol{a}} \leftarrow \cos\theta_0\,\hat{\boldsymbol{a}} + \sin\theta_0\,\boldsymbol{e}_j$

So the first direction $\hat{\boldsymbol{a}}$ we are looking at is the one corresponding with the variable having the largest dispersion. During one cycle, a sequence of grid searches over $p-1$ planes is carried out; the jth grid search will update the jth coordinate of $\hat{\boldsymbol{a}}$. When the second cycle starts, it is assumed that the direction $\hat{\boldsymbol{a}}$ is already pointing in the good direction, but still needs local improvement. Hence, the grid search will not search over the whole plane, but only over the halfplane determined by the angles $[-\pi/4, \pi/4[$. After every cycle, the search is limited to a more narrow interval of angles, but keeping the number of grid points N_g constant for every search. As such, the first few cycles will allow to find the region of the search space where the maximum should be, and the subsequent cycles aim at further increasing the precision of the solution.

The subsequent optimal directions are found in a similar way. Once an $\boldsymbol{a}_{k-1}$ is obtained (for $k \geq 2$), a Householder transformation is applied to the data, corresponding to a rotation of the data in such a way that $\boldsymbol{a}_{k-1}$ is transformed into the $(k-1)$th canonical vector. Then the grid algorithm is applied again, but now only using the last $(p-k+1)$ coordinates of the data vectors. As such, a sequence of eigenvalues is obtained and, after inverting the Householder transformations, also a sequence of optimal directions. We refer to [5] for full details on this, and emphasize that due to the easy form of the Householder transformation and its inverse, the additional computational load is marginal.

If $p > n$, then the variables are exactly the scores on the principal components, and the algorithm is equivariant under orthogonal transformations. Furthermore, the algorithm is completely deterministic, and easy to implement. Default choices are $N_g = 10$ and $N_c = 10$, giving good results, also for noisy data sets containing many outliers. These default choices are used throughout the paper.

5 Numerical Experiments

Simulation Experiment

To compare the different algorithms, we perform a simulation study. As a first measure of accuracy of the algorithm we take the value of $\hat{\lambda}_1$, the maximum of the objective function in (1). Since the aim of the algorithm is to attain the highest value in (1), better algorithms provide higher values for $\hat{\lambda}_1$. Also, the higher this value, the more informative, in terms of higher variability, the obtained principal component is. Note that the highest possible value, λ_1 is in general not known, neither is $\boldsymbol{a}_1$. (An exception is the case where the scale measure S equals the standard deviation, then λ_1 and $\boldsymbol{a}_1$ are exactly computable from an eigenvalue analysis of the sample covariance matrix). More generally, the precision of the algorithm for finding the first k optimal directions will be measured by

$$I_k = \sum_{j=1}^{k} \hat{\lambda}_j, \tag{12}$$

being the total variability accounted for by the first k principal components. The efficiency of the CR algorithm with respect to the GRID algorithm is then measured by

$$\text{EFF}_k(\text{CR;GRID}) = \frac{I_k^{\text{CR}}}{I_k^{\text{GRID}}}. \tag{13}$$

It will turn out that throughout all numerical experiments we carried out, the above efficiency was nearly always below 100%.

Two situations are of special interest, namely the case $n > p$ and the case $n \leq p$, where after data compression the latter case corresponds to $n = p$. We take $p = 50$ throughout this experiment, and set the sample size $n = 50$ and $n = 1000$, respectively. Note that the smaller sample size is the most important in chemometrical applications. Data are then generated according to $N(\mathbf{0}, diag(\lambda_1, \ldots, \lambda_p))$ with $\lambda_j = 1/j^2$, yielding a fast decaying scree plot. Since the algorithms are aimed at data containing outliers, further simulations are done for contaminated data: 80% of the observations of a simulated data set come from the same distribution as before, and the remaining 20% of observations are generated due to $N(\mathbf{0}, 10 \cdot diag(\lambda_p, \ldots, \lambda_1))$. Hence the outlier generating distribution has exactly the inverse order of principal components.

We generate 1000 different samples, and compute then the efficiency in (13), for $k = 1$. Boxplots of these 1000 replicates are given in Figure 3, once for $n = 50$, and once for $n = 1000$, for the data with and without contamination. As scale measure, the MAD was taken here. It is seen from Figure 3 that the efficiencies remain below 100% in nearly all cases, showing the superiority of the GRID algorithm over the CR algorithm. The difference becomes more pronounced for the lower sample size, where the median efficiency is about 85%. In presence of outliers, the relative performance of the GRID algorithm further improves, with a median efficiency of about 76% for $n = 50$. Note that there are

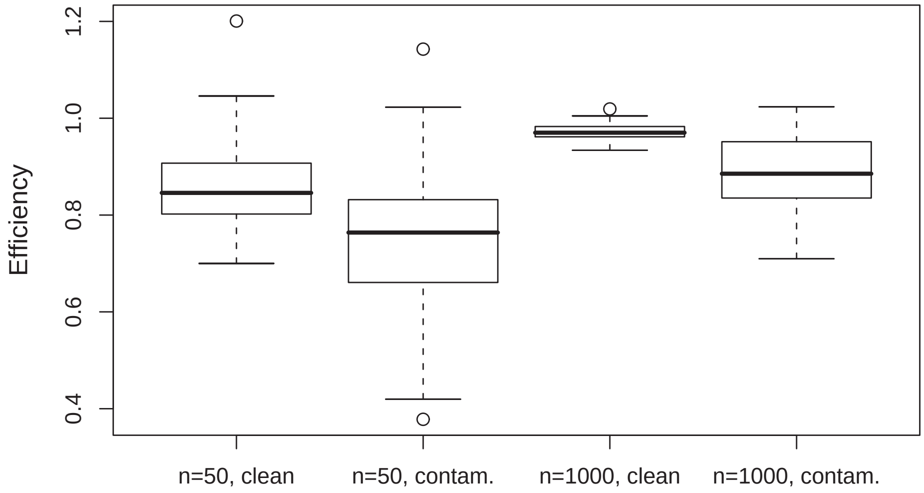

Figure 3: Boxplots of the efficiencies of the CR algorithm with respect to the GRID algorithm, for 1000 simulated samples of size $n = 50$ and $n = 1000$, once without outliers ("clean") and once with outliers ("contaminated").

samples where the efficiency gain by the GRID algorithm is very considerable; for example, for the contaminated case with $n = 50$, efficiencies of only 40% are observed in the boxplots. We conclude from Figure 3 that the GRID algorithm indeed yields much more accurate approximations for the case $n \leq p$. On the other hand, for $n = 1000 > p = 50$, the CR algorithm performs reasonably well, and is not loosing much w.r.t. the GRID algorithm.

Table 1 presents results for the same simulation setup, but with values for the sample size n ranging from 25 to 500, and for both robust scale measures MAD (4), and once for Q (5). Average efficiency measures are reported for $k = 1$ and $k = 5$. The standard errors around these reported averages were always less than 1.4%.

The results in Table 1 confirm the findings from the boxplots in Figure 3. The GRID algorithm is much more accurate than the CR algorithm, in particular for the smaller sample sizes. This efficiency gain is even more important for contaminated data, containing outliers. Moreover, the efficiencies EFF_5 over the first 5 principal components are consistently lower than the values EFF_1, indicating that the inaccuracy of the CR algorithm is cumulating over the additional components. It can also be seen that the efficiency gain when using MAD is more pronounced than for Q. This can be explained by the fact that for the scale estimator Q the objective function (3) to be maximized will

Table 1: Average efficiencies of the CR- with respect to the GRID algorithm, for $p = 50$, several values of n, for the MAD and Q, and for clean and contaminated simulated data.

	clean data				contaminated data			
	MAD		Q		MAD		Q	
	EFF$_1$	EFF$_5$	EFF$_1$	EFF$_5$	EFF$_1$	EFF$_5$	EFF$_1$	EFF$_5$
$n = 25$	71.3	62.0	77.0	68.7	61.0	40.9	66.8	48.1
$n = 50$	85.8	73.5	92.4	82.9	76.2	50.2	85.1	62.0
$n = 100$	89.7	80.8	94.7	90.8	82.2	55.9	88.0	68.3
$n = 200$	94.2	86.6	96.0	94.0	86.1	60.1	89.2	70.3
$n = 500$	96.4	91.8	97.2	95.9	87.7	61.3	89.2	70.3

be much smoother than for MAD. Thus, a more precise algorithm like GRID will be able to identify local jumps close to the optimum whereas CR fails to find these peaks.

The computational effort of the GRID algorithm will be higher than that for the CR algorithm. In fact, the GRID algorithm requires $p \times N_g \times N_c$ evaluations of the scale measure, while the CR algorithm needs only n evaluations (for finding a single optimal direction). Hence, to make the comparison more fair, we generated additional directions to the considered search directions of (7) for the CR algorithm such that the computation time becomes comparable (i.e. the number of evaluations of S is the same). Interestingly, the results of Table 1 remained almost identical. The only exception is the value of EFF$_5$ for $n = 25$ and contaminated data, which increased from 40.9 to 51.2 when using the MAD, and from 48.1 to 54.6 for the Q. Hence, unless the sample size is very small, generating addition random directions is of little use. The reason is that the trial directions of (7) are good candidate directions which cannot easily be improved by simply adding some random directions to (7).

While the above simulation experiment indicates the superior performance of the GRID with respect to the CR algorithm, the efficiencies do not show how close the results are to the true optimum of (3). Unfortunately, there is only one setting were this could be verified, namely when the scale measure is the classical standard deviation. In this case, the maximal value of the objective function is given by the largest eigenvalue of the sample covariance matrix of the data. Thus, we repeated the same simulation exercise as above, comparing the GRID algorithm with the exact solution, and obtained that the computational efficiency of GRID with respect to the exact solution is almost 100% (results available upon request). When the standard deviation serves as projection index, the GRID algorithm retrieves the exact solution upon rounding error.

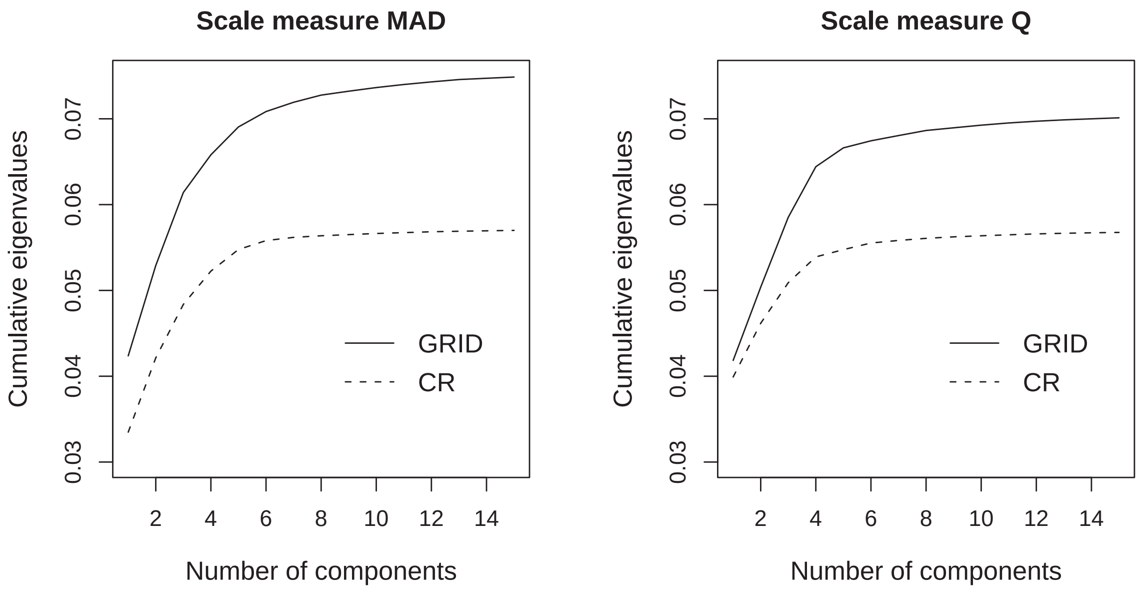

Figure 4: Plot of the cumulated eigenvalues for the 'Gas data', as estimated by the algorithms GRID and CR, once using MAD and once for Q as projection index.

Real Data Examples

We compare the performance of the two algorithms also on two real data sets. The first data set reports the NIR (near-infrared) spectra of $n = 60$ gasoline samples, measured in $2nm$ intervals from $900nm$ to $1700nm$ ($p = 401$). It will be called the 'Gas data' being described in [27]. This is an example where the number of variables p is larger than the number of observations n. The second data set, called 'Car data', is available in S-Plus ([28]), and illustrates a situation where the number of observations is larger than the number of variables. In this example, $p = 11$ variables are evaluated from $n = 111$ different cars.

For both examples we considered the MAD and Q as robust scale measures. The results are presented by plotting the cumulated eigenvalues defined by (8) with respect to k, the number of principal components retained. Recall that (8) measures the variability explained by the first k components. Figure 4 presents the results for the Gas data. The plot suggest that after 6 to 8 components the extra gain in explained variability by adding more components becomes much smaller, suggesting that k about 8 is a good choice. The same trend as seen from the simulations studies is also visible here. The GRID algorithm yields much higher explained variabilities than the CR algorithm, resulting in more informative principal components.

Figure 5 shows the cumulative eigenvalues for the Car data set, and similar conclusions

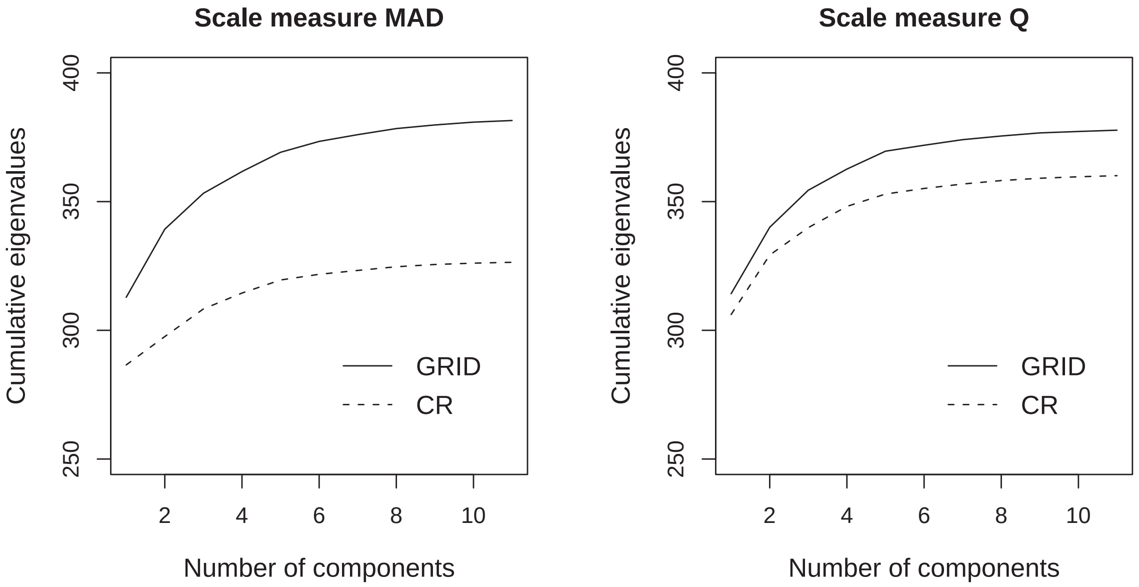

Figure 5: Plot of the cumulated eigenvalues for the ‘Car data’, as estimated by the algorithms GRID and CR, once using MAD and once for Q as projection index.

can be made. Especially for the first few components the GRID algorithm shows much higher contributions to the explained variability than the CR algorithm. This is true for both scale measures, with the gain of the GRID algorithm being largest when the MAD is used.

6 Summary and Conclusions

Principal component analysis based on projection-pursuit (PP) is very attractive in many respects. By using different scale measures as Projection Index, different types of PCA are obtained. In particular, taking a robust scale measure like MAD or Q, yields a robustification of standard PCA. Moreover, it is possible to stop the computation of the principal components at any desired number of components, which is not the case for PCA based on a robust covariance matrix. Especially for high-dimensional data this is an important feature, since the computational effort can be reduced considerably by computing only the first few robust principal components.

We presented a new algorithm, called GRID, for computing the robust principal components and showed that it outperforms the currently available CR algorithm. In practically all simulation situations and examples it turned out that the GRID algorithm leads to higher amounts of explained variability. The differences between the algorithms is substantial for small sample sizes with $n < p$, being the most pertinent case in chemometrical applications. For a large sample size n with respect to the dimension p, the CR algorithm still performs quite well, as can be seen from the simulation results in Section 5. Note that the CR algorithm as proposed by [11] was not aimed at high dimensional applications. For $n >> p$ we still recommend this algorithm as a fast and reasonably accurate way to find the robust principal components, but for $p > n$ it should not be used. In particular, we showed in Section 3 that the CR algorithm suffers from implosion breakdown in the latter case.

Note that the GRID algorithm works for any scale estimator, and no differentiability conditions are needed. Hence it is an "omnibus" algorithm that can serve to maximize any Projection Index. In this paper we focus on the widely used robust scale measures MAD and Q, and demonstrated that for both scale measures the algorithm GRID gives much better approximations to the eigenvalues than the CR algorithm. The choice of the scale estimator to be used must be based on statistical grounds, as discussed in [4]. The MAD is more robust, while the Q has a higher statistical efficiency.

In this paper we did not compare with other robust PCA approaches based on the Projection-Pursuit principle, since several papers have already been devoted to this (see Introduction). This paper is on a comparison of different algorithms for the same estimator, and not on comparing robust PCA estimators of different natures. If the aim of the principal component analysis method is to maximize explained variability, then the PP based approach is the natural estimator to use.

Both the GRID and the CR algorithm have been implemented in the freely available statistical software package R. They can be downloaded from (http://www.R-project.org) as the library *pcaPP*, including as well further documentation for the user.

Acknowledgments

We would like to thank Heinrich Fritz from the Vienna University of Technology for translating the algorithms into C++ and for preparing the R library. We also thank the reviewers for their constructive comments.

References

[1] P.J. Huber, The Annals of Statistics, 13 (1985) 435-525.

[2] G. Li and Z. Chen, Journal of the American Statistical Association, 80 (1985) 759-766.

[3] Cui, H., He, X., and Ng. K.W, Biometrika, 90 (2003) 953-966.

[4] C. Croux and A. Ruiz-Gazen, Journal of Multivariate Analysis, 95 (2005) 206-226.

[5] M. Hubert, P.J. Rousseeuw, and S. Verboven (2002), Chemometrics and Intelligent Laboratory Systems, 60 (2002), 101-111.

[6] I. Stanimirova, B. Walczak, D.L. Massart, and V. Simenov, Chemometrics and Intelligent Laboratory Systems, 71 (2004) 83-95.

[7] M. Daszykowski, S. Serneels, K. Kaczmarek, P. Van Espen, C. Croux, and B. Walczak, Chemometrics and Intelligent Laboratory Systems, to appear.

[8] M. Daszykowski, K. Kaczmarek, Y. Vander Heyden and B. Walczak, Chemometrics and Intelligent Laboratory Systems (In Press), available online 28 August 2006.

[9] R.A. Johnson and D.W. Wichern, Applied Multivariate Statistical Analysis, 4th edition, Prentice Hall, New York, 1998.

[10] Y. Xie, J. Wang, Y. Liang, L. Sun, X. Song, and R. Yu, J. Chemometrics 7 (1993) 527-541.

[11] C. Croux and A. Ruiz-Gazen, in A. Prat (Ed.), Compstat: Proceedings in Computational Statistics, Physica-Verlag, Heidelberg, 1996, pp. 211-216.

[12] S. Serneels, P.Filzmoser, C. Croux, and P.J. Van Espen, Chemometrics and Intelligent Laboratory Systems, 76 (2005), 197-204.

[13] C. Croux and G. Haesbroeck, Biometrika, 87 (2000) 603-618.

[14] R.A. Maronna, Technometrics, 47 (2005) 264-273.

[15] M. Hubert, P.J. Rousseeuw, and K. Vanden Branden, Technometrics, 47 (2005) 64-79.

[16] P.J. Rousseeuw and C. Croux, Journal of the American Statistical Association, 88 (1993) 1273-1283.

[17] A. Weber, Über den Standort der Industrien, Tübingen. English translation by C. Friedrich (1929): Alfred Weber's theory of location of industries. University of Chicago Press, 1909.

[18] O. Hössjer and C. Croux, Nonparametric Statistics 4 (1995) 293-308.

[19] U. Gather, T. Hilker, and C. Becker, in L.T. Fernholz, S. Morgenthaler, and W. Stahel (Eds.), Statistics in Genetics and in the Environmental Sciences, Birkhauser, Basel, 2001, pp. 147-157.

[20] G. Boente, A.M. Pires, and I.M. Rodrigues, Biometrika 89 (2002) 861-875.

[21] P. Filzmoser, Environmetrics 10 (1999) 363-375.

[22] I. Stanimirova, M. Daszykowski, E. Van Gyseghem, F.F. Bensaid, M. Lees, J, Smeyers-Verbeke, D.L. Massart, and Y. Vander Heyden, Analytica Chimica Acta, 552 (2005) 1-12.

[23] D.L. Donoho and P.J. Huber, in P.J. Bickel, K.A. Doksum, and J.L. Hodges (Eds.), A Festschrift for Erich L. Lehmann, Wadsworth, Belmont, CA, 1983, pp. 157-184.

[24] P.J. Rousseeuw and A.M. Leroy, Robust Regression and Outlier Detection. John Wiley, New York, 1987.

[25] C. Croux and P.J. Rousseeuw, Communications in Statistics, Theory and Methods, 21 (1992) 1935-1951.

[26] P. Filzmoser, S. Serneels, C. Croux, and P.J. Van Espen, in M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nürnberger, and W. Gaul (Eds.), From Data and Information Analysis to Knowledge Engineering, Springer, Berlin Heidelberg, 2006, pp. 270-277.

[27] J.H. Kalivas, Chemometrics and Intelligent Laboratory Systems, 37 (1997) 255-259.

[28] S-Plus: Statistical software package, http://www.insightful.com