ROBPCA: A New Approach to Robust Principal Component Analysis

Mia HUBERT

Department of Mathematics
Katholieke Universiteit Leuven
B-3001 Leuven, Belgium
(*mia.hubert@wis.kuleuven.ac.be*)

Peter J. ROUSSEEUW

Department of Mathematics and Computer Science
University of Antwerp
Antwerp, Belgium
(*peter.rousseeuw@ua.ac.be*)

Karlien VANDEN BRANDEN

Department of Mathematics
Katholieke Universiteit Leuven
B-3001 Leuven, Belgium
(*karlien.vandenbranden@wis.kuleuven.ac.be*)

We introduce a new method for robust principal component analysis (PCA). Classical PCA is based on the empirical covariance matrix of the data and hence is highly sensitive to outlying observations. Two robust approaches have been developed to date. The first approach is based on the eigenvectors of a robust scatter matrix such as the minimum covariance determinant or an S-estimator and is limited to relatively low-dimensional data. The second approach is based on projection pursuit and can handle high-dimensional data. Here we propose the ROBPCA approach, which combines projection pursuit ideas with robust scatter matrix estimation. ROBPCA yields more accurate estimates at noncontaminated datasets and more robust estimates at contaminated data. ROBPCA can be computed rapidly, and is able to detect exact-fit situations. As a by-product, ROBPCA produces a diagnostic plot that displays and classifies the outliers. We apply the algorithm to several datasets from chemometrics and engineering.

KEY WORDS: High-dimensional data; Principal component analysis; Projection pursuit; Robust methods.

1. INTRODUCTION

Principal component analysis (PCA) is a popular statistical method that tries to explain the covariance structure of data by means of a small number of components. These components are linear combinations of the original variables, and often allow for interpretation and better understanding of the different sources of variation. Because PCA is concerned with data reduction, it is widely used for the analysis of high-dimensional data, which are frequently encountered in chemometrics, computer vision, engineering, genetics, and other domains. PCA is then often the first step of the data analysis, followed by discriminant analysis, cluster analysis, or other multivariate techniques. It is thus important to find those principal components that contain most of the information.

In the classical approach, the first component corresponds to the direction in which the projected observations have the largest variance. The second component is then orthogonal to the first component and again maximizes the variance of the data points projected on it. Continuing in this way produces all of the principal components, which correspond to the eigenvectors of the empirical covariance matrix. Unfortunately, both the classical variance (which is being maximized) and the classical covariance matrix (which is being decomposed) are very sensitive to anomalous observations. Consequently, the first components are often attracted toward outlying points, and may not capture the variation of the regular observations. Therefore, data reduction based on classical PCA (CPCA) becomes unreliable if outliers are present in the data.

The goal of robust PCA methods is to obtain principal components that are not influenced much by outliers. A first group of methods is obtained by replacing the classical covariance matrix by a robust covariance estimator. Maronna (1976) and Campbell (1980) proposed using affine-equivariant M-estimators of scatter for this purpose, but these cannot resist many outliers. More recently, Croux and Haesbroeck (2000) used positive-breakdown estimators, such as the minimum covariance determinant (MCD) method (Rousseeuw 1984) and S-estimators (Davies 1987; Rousseeuw and Leroy 1987). The result is more robust, but unfortunately is limited to small to moderate dimensions. To see why this is so, consider, for example, the MCD estimator, defined as the mean and the covariance matrix of the h observations (out of the whole data set of size n) whose covariance matrix has the smallest determinant. If p denotes the number of variables in our dataset, then the MCD estimator can be computed only if $p < h$; otherwise, the covariance matrix of any h-subset has zero determinant. By default, h is about $.75n$, and it may be chosen as small as $.5n$; in any case, p may never be larger than n. A second problem is the computation of these robust estimators in high dimensions. Today's fastest algorithms (Woodruff and Rocke 1994; Rousseeuw and Van Driessen 1999) can handle up to about 100 dimensions, whereas in some fields, like chemometrics, data with dimensions in the thousands need to be analyzed.

A second approach to robust PCA uses projection pursuit (PP) techniques (Li and Chen 1985; Croux and Ruiz-Gazen 1996; Hubert, Rousseeuw, and Verboven 2002). These tech-

TECHNOMETRICS, FEBRUARY 2005, VOL. 47, NO. 1
DOI 10.1198/004017004000000563

niques maximize a robust measure of spread to obtain consecutive directions on which the data points are projected. This idea has also been generalized to common principal components (Boente, Pires, and Rodrigues 2002). It yields transparent algorithms that can be applied to datasets with many variables and/or many observations.

In Section 2 we propose the ROBPCA method, which attempts to combine the advantages of both approaches. We also describe an accompanying diagnostic plot that can be used to detect and classify possible outliers. We analyze several real datasets from chemometrics and engineering in Section 3. In Section 4 we investigate the performance and robustness of ROBPCA through simulations. Finally, in Section 5 we outline potential applications of ROBPCA in other types of multivariate data analysis.

2. THE ROBPCA METHOD

2.1 Description

The proposed ROBPCA method combines ideas of both PP and robust covariance estimation. The PP part is used for the initial dimension reduction. Some ideas based on the MCD estimator are then applied to this lower-dimensional data space. The combined approach yields more accurate estimates than the raw PP algorithm, as we discuss in Section 4.

The complete description of the ROBPCA method is quite involved and thus is relegated to the Appendix; here is a rough sketch of how it works. We assume that the original data are stored in an $n \times p$ data matrix $\mathbf{X} = \mathbf{X}_{n,p}$, where n denotes the number of objects and p denotes the original number of variables. The ROBPCA method then proceeds in three major steps. First, the data are preprocessed such that the transformed data are lying in a subspace whose dimension is at most $n - 1$. Next, a preliminary covariance matrix $\mathbf{S}_0$ is constructed and used for selecting the number of components k that will be retained in the sequel, yielding a k-dimensional subspace that fits the data well. Then the data points are projected on this subspace where their location and scatter matrix are robustly estimated, from which its k nonzero eigenvalues $l_1, \ldots, l_k$ are computed. The corresponding eigenvectors are the k robust principal components.

In the original space of dimension p, these k components span a k-dimensional subspace. Formally, writing the (column) eigenvectors next to one another yields the $p \times k$ matrix $\mathbf{P}_{p,k}$ with orthogonal columns. The location estimate is denoted by the p-variate column vector $\hat{\boldsymbol{\mu}}$ and called the robust center. The scores are the entries of the $n \times k$ matrix

$$\mathbf{T}_{n,k} = (\mathbf{X}_{n,p} - \mathbf{1}_n \hat{\boldsymbol{\mu}}')\mathbf{P}_{p,k}, \tag{1}$$

where $\mathbf{1}_n$ is the column vector with all n components equal to 1. Moreover, the k robust principal components generate a $p \times p$ robust scatter matrix $\mathbf{S}$ of rank k given by

$$\mathbf{S} = \mathbf{P}_{p,k}\mathbf{L}_{k,k}\mathbf{P}'_{p,k}, \tag{2}$$

where $\mathbf{L}_{k,k}$ is the diagonal matrix with the eigenvalues $l_1, \ldots, l_k$.

Like classical PCA, the ROBPCA method is location and orthogonal equivariant. That is, when a shift and/or an orthogonal transformation (e.g., a rotation or a reflection) is applied to the data, the robust center is also shifted, and the loadings are rotated accordingly. Hence the scores do not change under this type of transformation. Let $\mathbf{A}_{p,p}$ define an orthogonal transformation; thus $\mathbf{A}$ is of full rank and $\mathbf{A}' = \mathbf{A}^{-1}$, and $\hat{\boldsymbol{\mu}}_{\mathbf{x}}$ and $\mathbf{P}_{p,k}$ are the ROBPCA center and loading matrix for the original $\mathbf{X}_{n,p}$. Then the ROBPCA center and loadings for the transformed data $\mathbf{X}\mathbf{A}' + \mathbf{1}_n\mathbf{v}'$ are equal to $\mathbf{A}\hat{\boldsymbol{\mu}}_{\mathbf{x}} + \mathbf{v}$ and $\mathbf{A}\mathbf{P}$. Consequently, the scores remain the same under these transformations,

$$\begin{aligned}\mathbf{T}(\mathbf{X}\mathbf{A}' + \mathbf{1}_n\mathbf{v}') &= \big(\mathbf{X}\mathbf{A}' + \mathbf{1}_n\mathbf{v}' - \mathbf{1}_n(\mathbf{A}\hat{\boldsymbol{\mu}}_{\mathbf{x}} + \mathbf{v})'\big)\mathbf{A}\mathbf{P} \\ &= (\mathbf{X} - \mathbf{1}_n\hat{\boldsymbol{\mu}}'_{\mathbf{x}})\mathbf{P} = \mathbf{T}(\mathbf{X}).\end{aligned}$$

Although these properties seem very natural for a PCA method, they are not shared by some other robust PCA estimators, such as the resampling by half-means and the smallest half-volume methods of Egan and Morgan (1998).

2.2 Diagnostic Plot

As is the case for many robust methods, the purpose of a robust PCA is twofold: (1) to find those linear combinations of the original variables that contain most of the information, even if there are outliers, and (2) to flag outliers and to determine their type.

To see that there can be different types of outliers, consider Figure 1, where $p = 3$ and $k = 2$. Here we can distinguish between four types of observations. The *regular observations* form one homogeneous group that is close to the PCA subspace. Next, we have *good leverage points*, which lie close to the PCA space but far from the regular observations, such as the observations 1 and 4 in Figure 1. We can also have *orthogonal outliers*, which have a large orthogonal distance to the PCA space but cannot be seen when we look only at their projection on the PCA space, like observation 5. The fourth type of data points are the *bad leverage points*, which have a large orthogonal distance and whose projection on the PCA subspace is remote from the typical projections, such as observations 2 and 3.

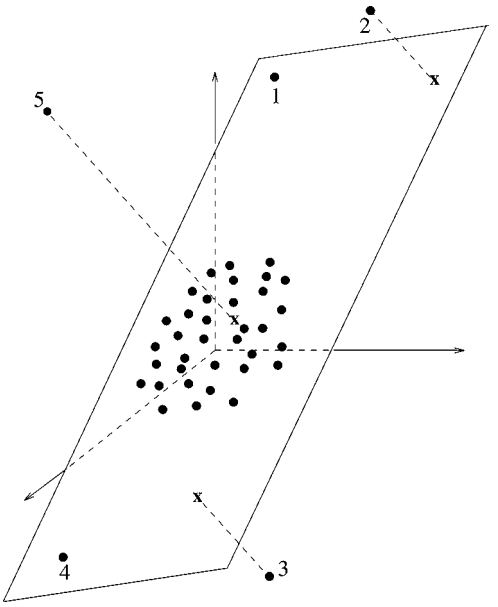

Figure 1. Different Types of Outliers When a Three-Dimensional Dataset Is Projected on a Robust Two-Dimensional PCA Subspace.

To distinguish between regular observations and the three types of outliers for higher-dimensional data, we construct a *diagnostic plot* or *outlier map*. On the horizontal axis we plot the *robust score distance*, SD_i, of each observation, given by

$$SD_i = \sqrt{\sum_{j=1}^{k} \frac{t_{ij}^2}{l_j}}, \tag{3}$$

where the scores t_{ij} are obtained from (1). If $k = 1$, then we prefer to plot the (signed) standardized score $t_i/\sqrt{l_1}$. On the vertical axis of the diagnostic plot we display the *orthogonal distance*, OD_i, of each observation to the PCA subspace, defined as

$$OD_i = \|\mathbf{x}_i - \hat{\boldsymbol{\mu}} - \mathbf{P}_{p,k}\mathbf{t}_i'\|, \tag{4}$$

where the ith observation is denoted as the p-variate column vector $\mathbf{x}_i$ and $\mathbf{t}_i'$ is the ith row of $\mathbf{T}_{n,k}$.

To classify the observations, we draw two cutoff lines. The cutoff value on the horizontal axis is $\sqrt{\chi^2_{k,.975}}$ when $k > 1$ and $\pm\sqrt{\chi^2_{1,.975}}$ when $k = 1$ (because the squared Mahalanobis distances of normally distributed scores are approximately χ^2_k distributed). The cutoff value on the vertical axis is more difficult to determine, because the distribution of the orthogonal distances is not known exactly. However, a scaled chi-squared distribution $g_1\chi^2_{g_2}$ gives a good approximation of the unknown distribution of the squared orthogonal distances (Box 1954). Nomikos and MacGregor (1995) used the method of moments to estimate the two unknown parameters g_1 and g_2. We prefer to follow a robust approach. We use the Wilson–Hilferty approximation for a chi-squared distribution. This implies that the orthogonal distances to the power 2/3 are approximately normally distributed with mean $\mu = (g_1g_2)^{1/3}(1 - \frac{2}{9g_2})$ and variance $\sigma^2 = \frac{2g_1^{2/3}}{9g_2^{1/3}}$. We obtain estimates $\hat{\mu}$ and $\hat{\sigma}^2$ using the univariate MCD. The cutoff value on the vertical axis then equals $(\hat{\mu} + \hat{\sigma}z_{.975})^{3/2}$, with $z_{.975} = \Phi^{-1}(.975)$ as the 97.5% quantile of the Gaussian distribution.

Note the analogy of this diagnostic plot with the plot of Rousseeuw and Van Zomeren (1990) for robust regression. There the vertical axis gives the standardized residuals obtained with a robust regression method, with cutoff values at -2.5 and 2.5 (because for normally distributed data, roughly 1% of the standardized residuals fall outside that interval). Here, we prefer horizontal and vertical cutoff values that both have an exceeding probability of 2.5%.

3. EXAMPLES

Here we illustrate the ROBPCA method and the diagnostic plot on several real datasets. We also compare the results from ROBPCA with four other PCA methods: classical PCA (CPCA), RAPCA (Hubert et al. 2002), and spherical (SPHER) and ellipsoidal (ELL) PCA (Locantore et al. 1999). The latter three methods are also robust and designed for high-dimensional data.

Li and Chen (1985) proposed the idea of PP for PCA, but their algorithm has a high computational cost. More attractive methods were developed by Croux and Ruiz-Gazen (1996), but in high dimensions these algorithms still have numerical inaccuracy. Consequently, Hubert et al. (2002) developed RAPCA, a fast two-step algorithm that searches for the direction on which the projected observations have the largest robust scale, then removes this dimension and repeats.

The spherical and ellipsoidal PCA methods provide a very fast algorithm for performing robust PCA. After robustly centering the data, the observations are projected on a sphere (in SPHER PCA) or an ellipse (in ELL PCA). The principal components are then derived as the eigenvectors of the covariance matrix of these projected data points. SPHER and ELL do not yield estimates of the eigenvalues, which makes it impossible to compute score distances. Therefore, in the examples we also applied the MCD estimator on the scores to compute robust distances in the PCA subspace.

3.1 Car Data

Our first example is the low-dimensional car dataset, which is available in S–PLUS as the data frame cu.dimensions. For $n = 111$ cars, $p = 11$ characteristics were measured, including the length, width, and height of the car. We first looked at pairwise scatterplots of the variables, and computed pairwise Spearman rank correlations $\rho_S(X_i, X_j)$. This preliminary analysis already indicated that there are high correlations among the variables, for example, $\rho_S(X_1, X_2) = .83$ and $\rho_S(X_3, X_9) = .87$. Hence PCA seems to be an appropriate method for finding the most important sources of variation in this dataset.

When applying ROBPCA to these data, an important choice that we need to make is how many principal components to keep. We make this choice using the eigenvalues $\tilde{l}_1 \geq \tilde{l}_2 \geq \cdots \geq \tilde{l}_r$ of $\mathbf{S}_0$ with $r = \text{rank}(\mathbf{S}_0)$, as obtained in the second stage of the algorithm [see also (A.4) in the App.]. We can use these eigenvalues in various ways. We can look at the *scree plot*, which is a graph of the (monotone decreasing) eigenvalues (Joliffe 1986). We can also use a selection criterion to choose k such that

$$\sum_{j=1}^{k} \tilde{l}_j \Big/ \sum_{j=1}^{r} \tilde{l}_j \approx 90\%, \tag{5}$$

or, for instance, such that

$$\frac{\tilde{l}_k}{\tilde{l}_1} \geq 10^{-3}. \tag{6}$$

Here we decided to retain $k = 2$ components based on criterion (5), because $(\tilde{l}_1 + \tilde{l}_2)/\sum_1^{11} \tilde{l}_j = 94\%$.

Figure 2(a) shows the resulting diagnostic plot. We can distinguish a group of orthogonal outliers (labeled 103–104, 107, 109, and 111) and two groups of bad leverage points (cases 102, 105–106, 108, and 110 and observations 25, 30, 32, 34, and 36). A few good leverage points are also visible (6 and 96). If we look at the measurements, we notice that the 5 most important bad leverage points (25, 30, 32, 34, and 36) have the value -2 on 4 of the 11 original variables, namely $X_6 =$ Rear.Hd, $X_8 =$ Rear.Seat, $X_{10} =$ Rear.Shld, and $X_{11} =$ luggage. None of the other observations share this property. The observations 102–111 have the value -2 for the last variable $X_{11} =$ luggage, and observation 109 has the value -3.

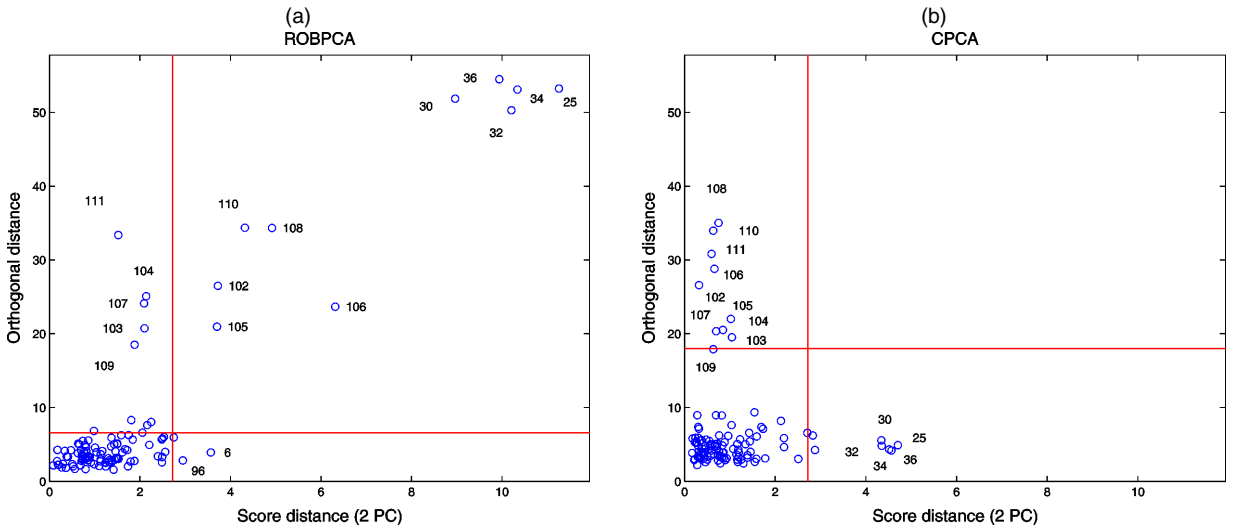

Figure 2. Diagnostic Plots of the Car Dataset Based on (a) Two Robust Principal Components and (b) Two Classical Principal Components.

We compare this robust result with a CPCA analysis. The first two components account for 85% of the total variance. The diagnostic plot in Figure 2(b) looks completely different from the robust plot in (a), although the same set of outliers is detected. The most striking difference is that the group of bad leverage points from ROBPCA is converted into good leverage points. This shows how the subspace found by CPCA is attracted toward these bad leverage points.

Some differences between ROBPCA and CPCA are also visible in the plot of the scores (t_{i1}, t_{i2}) for all $i = 1, \ldots, n$. Figure 3(a) shows the score plot of ROBPCA, together with the 97.5% tolerance ellipse, which is defined as the set of vectors in $\mathbb{R}^2$ whose score distance is equal to $\sqrt{\chi^2_{2,.975}}$. Data points that fall outside the tolerance ellipse are by definition the good and bad leverage points. We clearly see how well the robust tolerance ellipse encloses the regular data points. Figure 3(b) is the score plot obtained with CPCA. The corresponding tolerance ellipse is highly inflated toward the outliers 25, 30, 32, 34, and 36. The resulting eigenvectors are not lying in the direction of the highest variability of the other points. We also see how the second eigenvalue of CPCA is blown up by the same set of outliers.

We also performed robust PCA on this low-dimensional dataset using the eigenvectors and eigenvalues of the MCD covariance matrix. The resulting diagnostic plot was almost identical to the ROBPCA plot and thus is not included. The other robust methods also detected the same set of outliers.

3.2 Octane Data

Our second example is the octane dataset described by Esbensen, Schönkopf, and Midtgaard (1994). This dataset contains near-infrared (NIR) absorbance spectra over $p = 226$ wavelengths of $n = 39$ gasoline samples with certain octane numbers. It is known that six of the samples (25, 26, and 36–39) contain added alcohol. Both the classical scree plot shown in

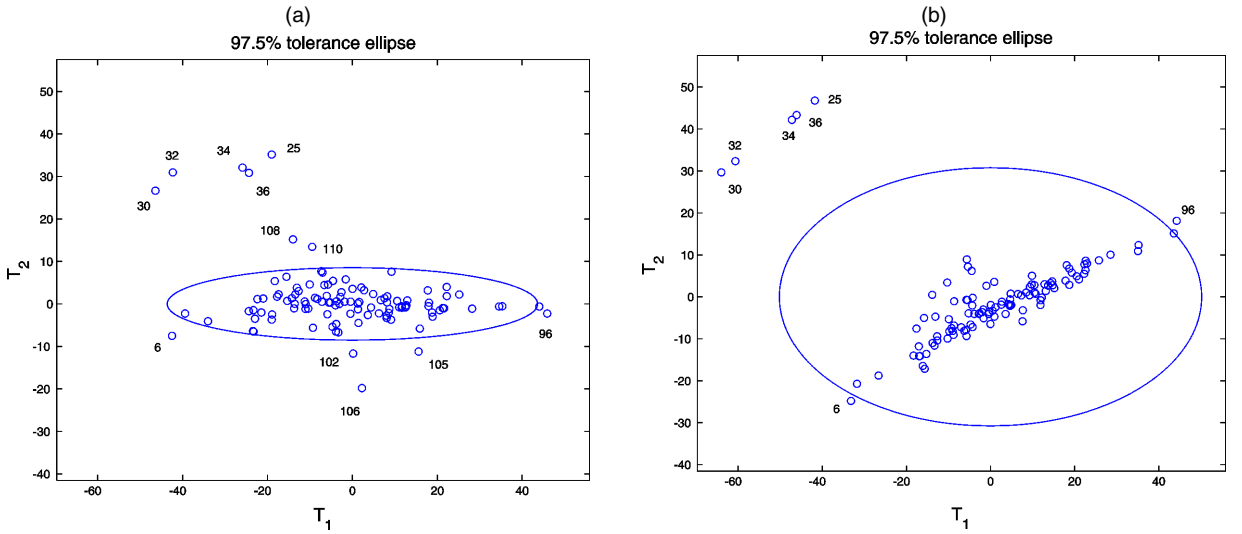

Figure 3. Score Plots With the 97.5% Tolerance Ellipse of the Car Dataset for (a) ROBPCA and (b) CPCA.

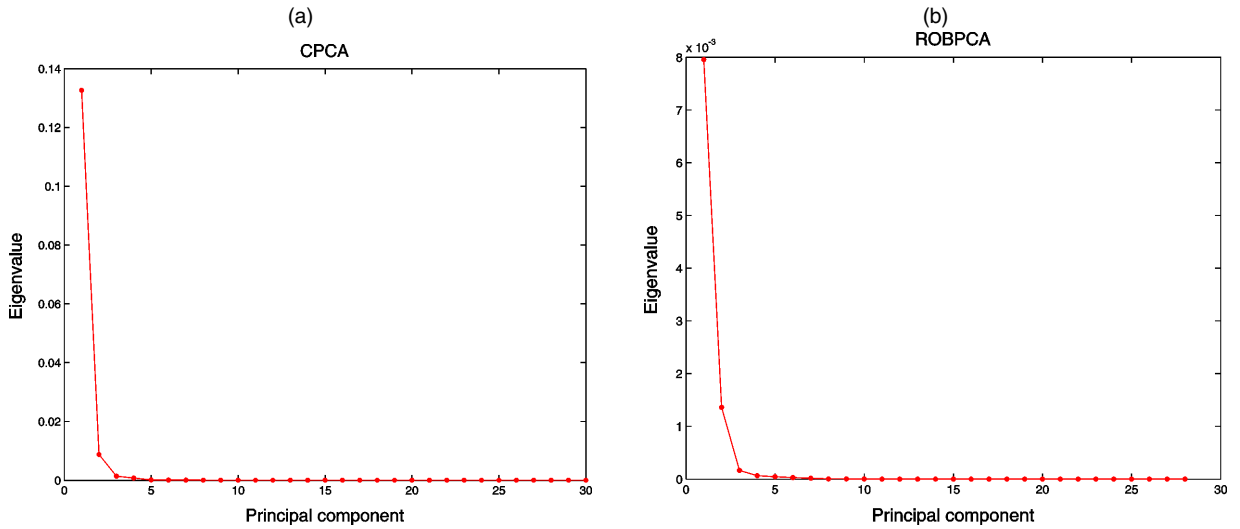

Figure 4. Scree Plots of the Octane Dataset With (a) CPCA and (b) ROBPCA.

Figure 4(a) and the ROBPCA scree plot shown in Figure 4(b) suggest retaining two principal components.

The CPCA diagnostic plot, given in Figure 5(a), shows that the classical analysis detects only the outlying spectrum 26,

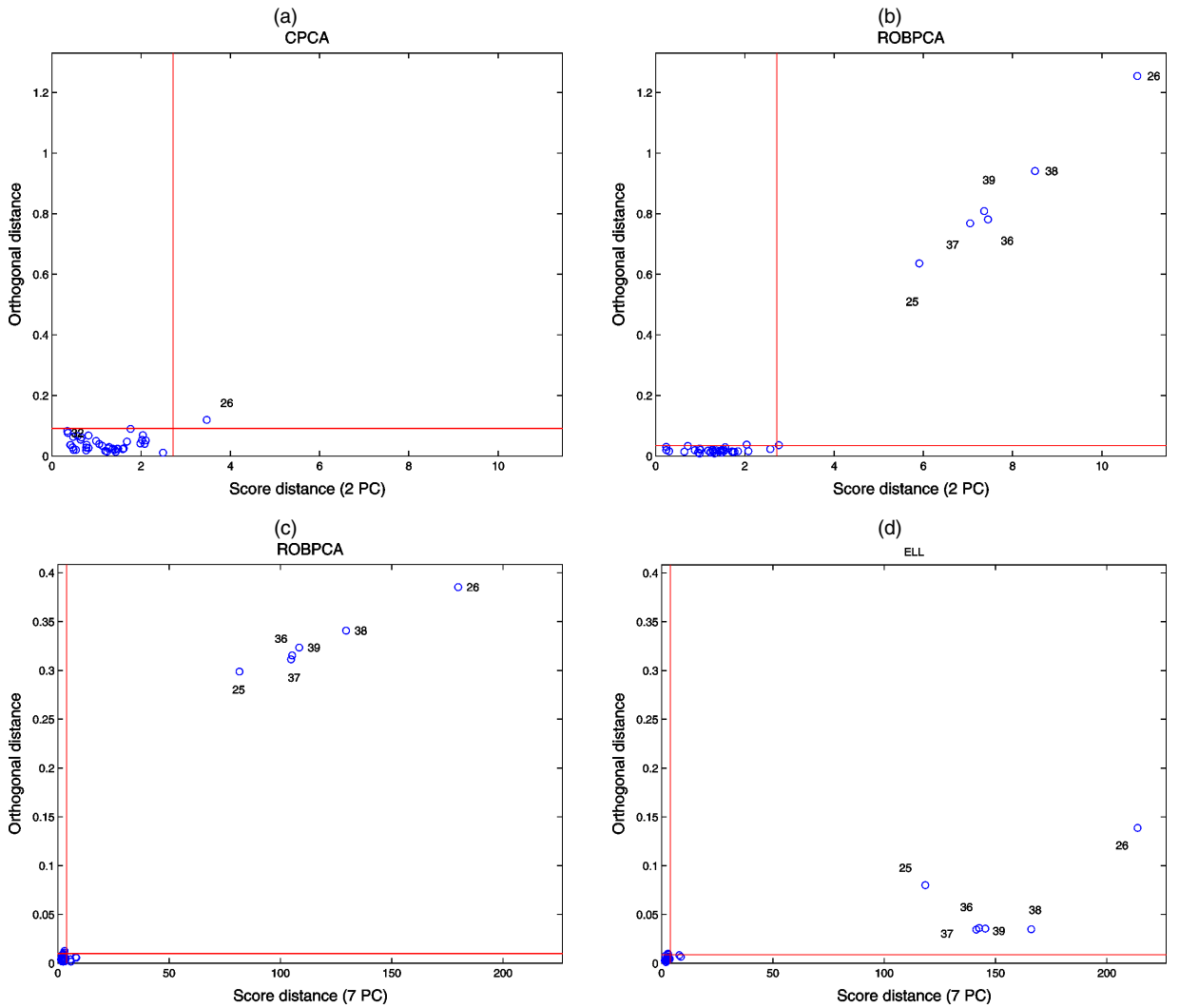

Figure 5. Diagnostic Plots of the Octane Dataset Based on (a) Two CPCA Principal Components, (b) Two ROBPCA Principal Components, (c) Seven ROBPCA Principal Components, and (d) Seven ELL Principal Components.

which does not stick out much above the border line. In contrast, we immediately clearly spot the six samples with added alcohol on the ROBPCA diagnostic plot in Figure 5(b). The first principal component from the CPCA is clearly attracted by the six outliers, yielding a classical eigenvalue of .13. In contrast, the first robust eigenvalue l_1 is only .01.

Next, we wondered whether the robust loadings would be influenced by the outlying spectra if we retained more than two components. To avoid the curse of dimensionality with $n = 39$ observations, it is generally advised that $n > 5k$ (see Rousseeuw and Van Zomeren 1990), so we considered $k_{\max} = 7$. From the robust diagnostic plot in Figure 5(c), we see that the outliers are still very far from the estimated robust subspace.

The diagnostic plots of RAPCA, SPHER, and ELL were similar to Figure 5(b) for $k = 2$. But when we selected $k = 7$ components with ELL, we see from Figure 5(d) that the outliers have a much lower orthogonal distance. This illustrates their leverage effect on the estimated principal components.

3.3 Glass Spectra

Our third dataset consists of EPXMA spectra over $p = 750$ wavelengths collected on 180 different glass samples (Lemberge, De Raedt, Janssens, Wei, and Van Espen 2000). The chemical analysis was performed using a Jeol JSM 6300 scanning electron microscope equipped with an energy-dispersive Si(Li) X-ray detection system (SEM–EDX).

We first performed ROBPCA with default value of $h = .75n = 135$; however, the diagnostic plots revealed a large number of outliers. Therefore, we analyzed the dataset a second time with $h = .70n = 126$. Three components were retained for CPCA and ROBPCA, yielding a classical explanation percentage of 99% and a robust explanation percentage [see (5)] of 96%. We then obtained the diagnostic plots in Figure 6. From the classical diagnostic plot in Figure 6(a), we see that CPCA does not find important outliers. In contrast, the ROBPCA plot in Figure 6(b) clearly distinguishes two major groups in the data, a smaller group of bad leverage points, a few orthogonal outliers, and the isolated case 180 in between the two major groups. A high-breakdown method such as ROBPCA treats the smaller group with cases 143–179 as one set of outliers. Later, it turned out that the window of the detector system had been cleaned before the last 38 spectra were measured. As a result, less radiation (X-rays) was absorbed and more could be detected, resulting in higher X-ray intensities. Looking at the

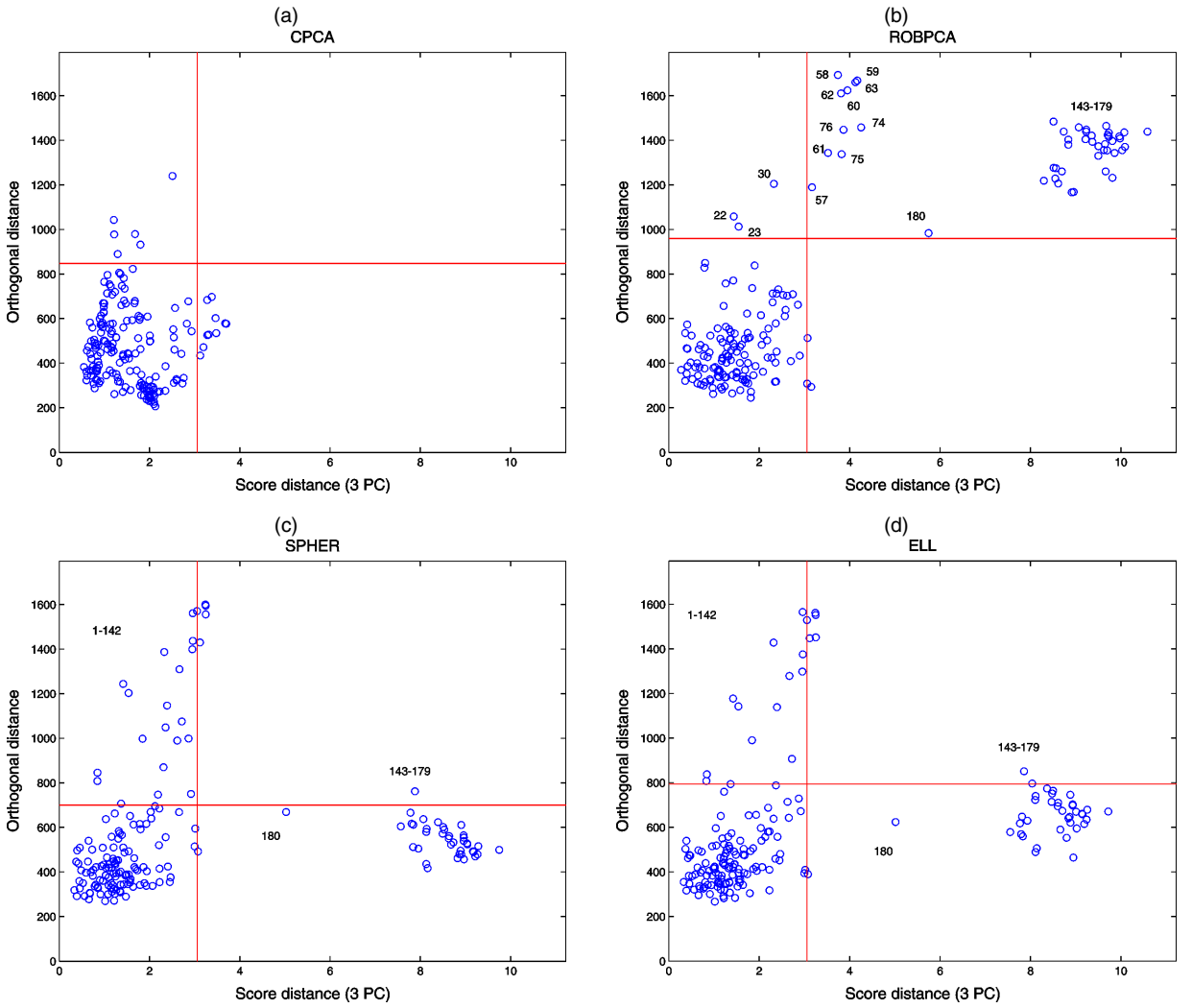

Figure 6. Diagnostic Plots of the Glass Dataset Based on Three Principal Components Computed With (a) CPCA, (b) ROBPCA, (c) SPHER, and (d) ELL.

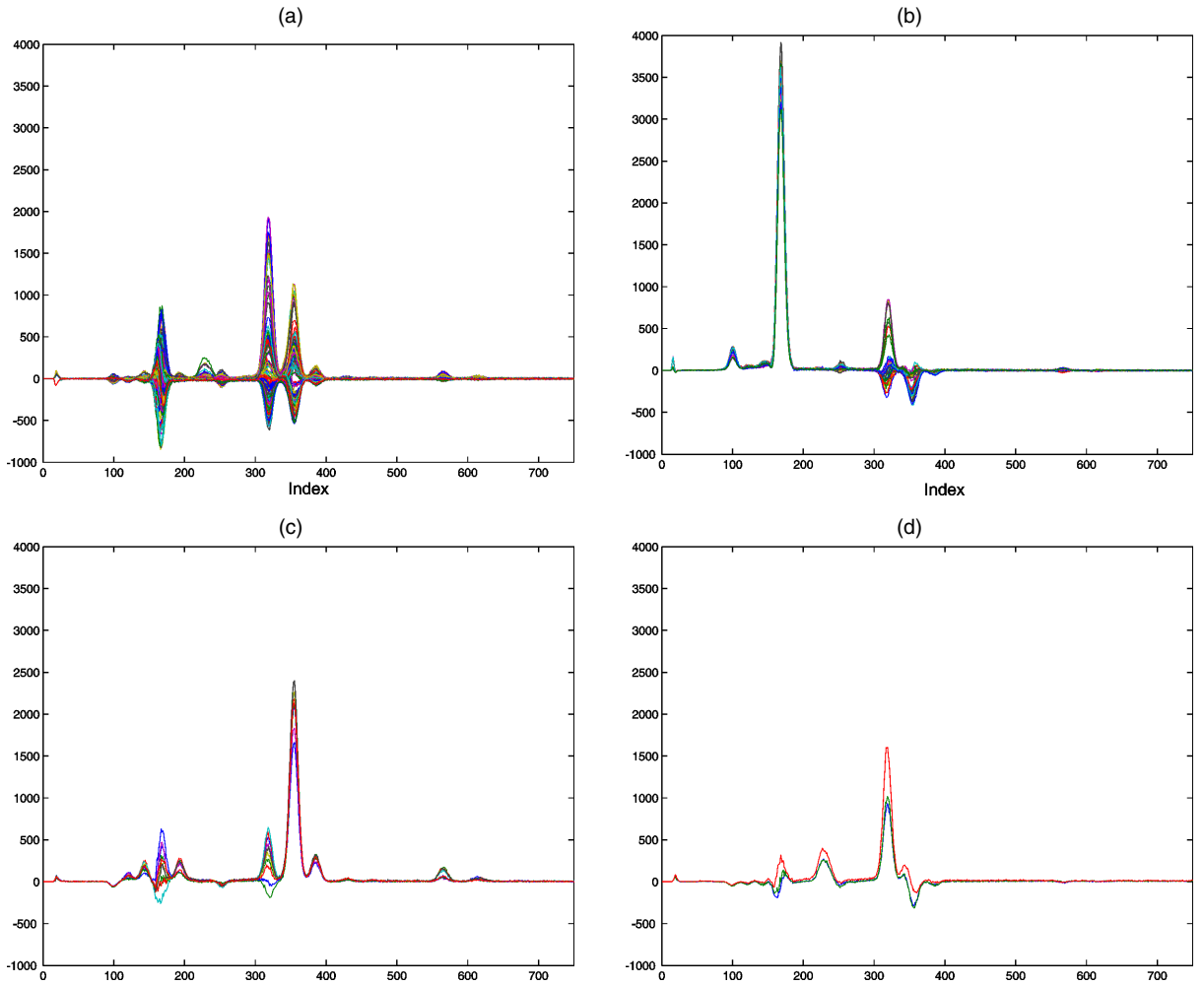

Figure 7. The Glass Dataset. (a) Regular observations; (b) bad leverage points 143–179; (c) bad leverage points 57–63 and 74–76; and (d) orthogonal outliers 22, 23, and 30.

spectra, we can indeed observe these differences. The regular samples, shown in Figure 7(a), clearly have lower measurements at channels 160–175 than did samples 143–179 of Figure 7(b). The spectrum of case 180 (not shown) was somewhat in between. Note that instead of plotting the raw data, we first robustly centered the spectra by subtracting the univariate MCD location estimator from each wavelength. Doing so allow us to observe more of the variability that is present in the data.

The other bad leverage points, (57–63) and (74–76), are samples with high concentrations of calcic. Figure 7(c) shows that their calcic alpha peak (around channels 340–370) and calcic beta peak (channels 375–400) is higher than for the other glass vessels. The orthogonal outliers (22, 23, and 30), the spectra of which are shown in Figure 7(d), are boundary cases, although they have larger measurements at channels 215–245. This might indicate a larger concentration of phosphor.

RAPCA yielded a diagnostic plot similar to the ROBPCA plot. SPHER and ELL are also able to detect the outliers, as shown in Figures 6(c) and 6(d), but they turn the bad leverage points into good leverage points and orthogonal outliers.

4. SIMULATIONS

We conducted a simulation study to compare the performance and the robustness of ROBPCA with the four other principal component methods introduced in Section 3: CPCA, RAPCA, SPHER, and ELL. We generated 1,000 samples of size n from the contamination model

$$(1-\varepsilon)\mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma}) + \varepsilon \mathrm{N}_p(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{\Sigma}})$$

or

$$(1-\varepsilon)t_5(\mathbf{0}, \mathbf{\Sigma}) + \varepsilon t_5(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{\Sigma}})$$

for different values of n, p, ε, $\mathbf{\Sigma}$, $\tilde{\boldsymbol{\mu}}$, and $\tilde{\mathbf{\Sigma}}$. That is, $n(1-\varepsilon)$ of the observations were generated from the p-variate Gaussian distribution $\mathrm{N}_p(\mathbf{0}, \mathbf{\Sigma})$ or the p-variate elliptical $t_5(\mathbf{0}, \mathbf{\Sigma})$ distribution, and $n\varepsilon$ of the observations were generated from $\mathrm{N}_p(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{\Sigma}})$ or from $t_5(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{\Sigma}})$.

Note that the consistency factor in the FAST–MCD algorithm, which is used within ROBPCA, is constructed under the assumption that the regular observations are normally distributed. Then the denominator equals $\chi^2_{k,1-\alpha}$, the $(1-\alpha)$th quantile of the chi-squared distribution with k degrees of freedom. Hence the best results of the simulations with t_ν (and here $\nu = 5$) are obtained by replacing the denominator with $k((\nu-2)/\nu)F_{k,\nu,1-\alpha}$, with $F_{k,\nu,1-\alpha}$ the $(1-\alpha)$th quantile of the F distribution with k and ν degrees of freedom. However, in real examples, any foreknowledge of the true underlying distribution is mostly unavailable. Therefore (and also to make a fair

comparison with RAPCA), we did not adjust the consistency factor.

In Tables 1 and 2 and Figures 8–12 we report some typical results obtained in the following situations:

1. $n = 100$, $p = 4$, $\mathbf{\Sigma} = \text{diag}(8, 4, 2, 1)$, and $k = 3$ [because then $(\sum_1^3 \lambda_i)/(\sum_1^4 \lambda_i) = 93.3\%$].
   (4a) $\varepsilon = 0$ (no contamination).
   (4b) $\varepsilon = 10\%$ or $\varepsilon = 20\%$, $\tilde{\boldsymbol{\mu}} = f_1 \mathbf{e}_4 = (0, 0, 0, f_1)'$, and $\tilde{\mathbf{\Sigma}} = \mathbf{\Sigma}/f_2$, where $f_1 = 6, 8, 10, \ldots, 20$ and $f_2 = 1$ or $f_2 = 15$.
2. $n = 50$, $p = 100$, $\mathbf{\Sigma} = \text{diag}(17, 13.5, 8, 3, 1, .095, \ldots, .002, .001)'$, and $k = 5$ [because here $(\sum_1^5 \lambda_i)/(\sum_1^{100} \lambda_i) = 90.3\%$].
   (100a) $\varepsilon = 0$ (no contamination).
   (100b) $\varepsilon = 10\%$ or $\varepsilon = 20\%$, $\tilde{\boldsymbol{\mu}} = f_1 \mathbf{e}_6$, and $\tilde{\mathbf{\Sigma}} = \mathbf{\Sigma}/f_2$, where $f_1 = 6, 8, 10, \ldots, 20$ and $f_2 = 1$ or $f_2 = 15$.

Note that $\varepsilon = 0\%$ also corresponds to $f_1 = 0$ and $f_2 = 1$. The subspace spanned by the first k eigenvectors of $\mathbf{\Sigma}$ is denoted by $\mathbf{E}_k = \text{span}\{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$, with $\mathbf{e}_j$ the jth column of $\mathbf{I}_{p,k}$.

The settings (4a) and (4b) consider low-dimensional data ($p = 4$) of not too small a size, $n = 100$, whereas in (100a) and (100b) we generate high-dimensional data, with $n = 50$ being rather small and even less than $p = 100$. In settings (4b) and (100b), the contaminated data are shifted by a distance f_1 in the direction of the $(k + 1)$th principal component. We started with $f_1 = 6$; otherwise, the outliers could not be distinguished from the regular data points. The factor f_2 determines how strongly the contaminated data are concentrated. Rocke and Woodruff (1996) showed that shifted outliers with the same covariance structure as the regular points are the most difficult to detect. This situation corresponds with $f_2 = 1$. Note that because of the orthogonal equivariance of the ROBPCA method, we need consider only diagonal covariance matrices.

For each simulation setting, we summarized the results of each estimation procedure (CPCA, RAPCA, SPHER, ELL, and ROBPCA) as follows:

1. For each method, we considered the maximal angle between $\mathbf{E}_k$ and the estimated PCA subspace, which is spanned by the columns of $\mathbf{P}_{p,k}$. Krzanowski (1979) proposed a measure for calculating this angle, which we denote by
$$\text{maxsub} = \arccos(\sqrt{\lambda_k}),$$
where λ_k is the smallest eigenvalue of $\mathbf{I}'_{k,p}\mathbf{P}_{p,k}\mathbf{P}'_{k,p}\mathbf{I}_{p,k}$. It represents the largest angle between a vector in $\mathbf{E}_k$ and the vector most parallel to it in the estimated PCA subspace. To standardize this value, we have divided it by $\frac{\pi}{2}$.
2. We compute the proportion of variability that is explained by the estimated eigenvalues. We do this by comparing the sum of the k largest eigenvalues with the sum of all p known eigenvalues. We report the mean proportion of explained variability,
$$\frac{1}{1{,}000} \sum_{l=1}^{1{,}000} \frac{\hat{\lambda}_1^{(l)} + \hat{\lambda}_2^{(l)} + \cdots + \hat{\lambda}_k^{(l)}}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_p},$$
where $\hat{\lambda}_j^{(l)}$ is the estimated value of λ_j at the lth replication. It would be more elegant if the denominator also contained the estimated eigenvalues, but ROBPCA and RAPCA estimate only the first k eigenvalues. We report these results for the settings without contamination and also for a specific situation with 10% contamination ($f_1 = 10, f_2 = 1$). Because SPHER and ELL estimate only the principal components and not their eigenvalues, we do not include these methods in the comparison.
3. For the k largest eigenvalues, we also compute the mean squared error (MSE), defined as
$$\text{MSE}(\hat{\lambda}_j) = \frac{1}{1{,}000} \sum_{l=1}^{1{,}000} \left(\hat{\lambda}_j^{(l)} - \lambda_j\right)^2.$$
We report the results only for $f_2 = 1$, because they were very similar for $f_2 = 15$.

Table 1. Simulation Results of maxsub in Settings (4a) and (100a) When There Is No Contamination

Distribution	n	p	CPCA	RAPCA	SPHER	ELL	ROBPCA
Normal	100	4	.094	.160	.127	.087	.176
	50	100	.215	.707	.272	.213	.282
t_5	100	4	.130	.183	.127	.086	.133
	50	100	.308	.701	.272	.213	.311

The ideal value of maxsub and MSE in the tables and figures is thus 0. For the mean proportion of explained variability, the optimal values are 93.3% for low-dimensional data and 90.3% for high-dimensional data.

Table 1 reports the simulation results of maxsub for the settings (4a) and (100a). We see that elliptical PCA yields the best results for maxsub when there is no contamination. For low-dimensional data, the results for the other methods are more or less comparable, whereas for high-dimensional data, RAPCA is clearly the less efficient approach.

From Table 2, we see that CPCA provides the best mean proportion of explained variability when there is no contamination in the data. RAPCA attains higher values than ROBPCA for both distributions. When contamination is added to the data, the eigenvalues obtained with CPCA are overestimated, resulting in estimated percentages even larger than 100%! The robust methods are much less sensitive to the outliers, but RAPCA also attains a value larger than 100% at the contaminated low-dimensional normal distribution. Note that when the consistency factor in ROBPCA is adapted to the t_5 distribution, the results improve substantially. For the low-dimensional data, we obtain 80% without contamination and 82.8% with contamination, whereas in high dimensions the mean percentages of explained variability are 69.3% and 69.4%.

Table 2. Simulation Results of the Mean Proportion of Explained Variability When There Is No Contamination and With 10% Contamination ($f_1 = 10$ and $f_2 = 1$)

	Multivariate normal			Multivariate t_5		
	CPCA	RAPCA	ROBPCA	CPCA	RAPCA	ROBPCA
$n = 100$, $p = 4$						
$\varepsilon = 0\%$	93.4%	94.7%	83.9%	98.7%	72.1%	60.2%
$\varepsilon = 10\%$	147.8%	112.9%	88.5%	135.2%	86.8%	67.5%
$n = 50$, $p = 100$						
$\varepsilon = 0\%$	91.6%	83.6%	79.5%	99.2%	65.1%	57.3%
$\varepsilon = 10\%$	109.4%	86.1%	79.3%	110.9%	66.9%	56.7%

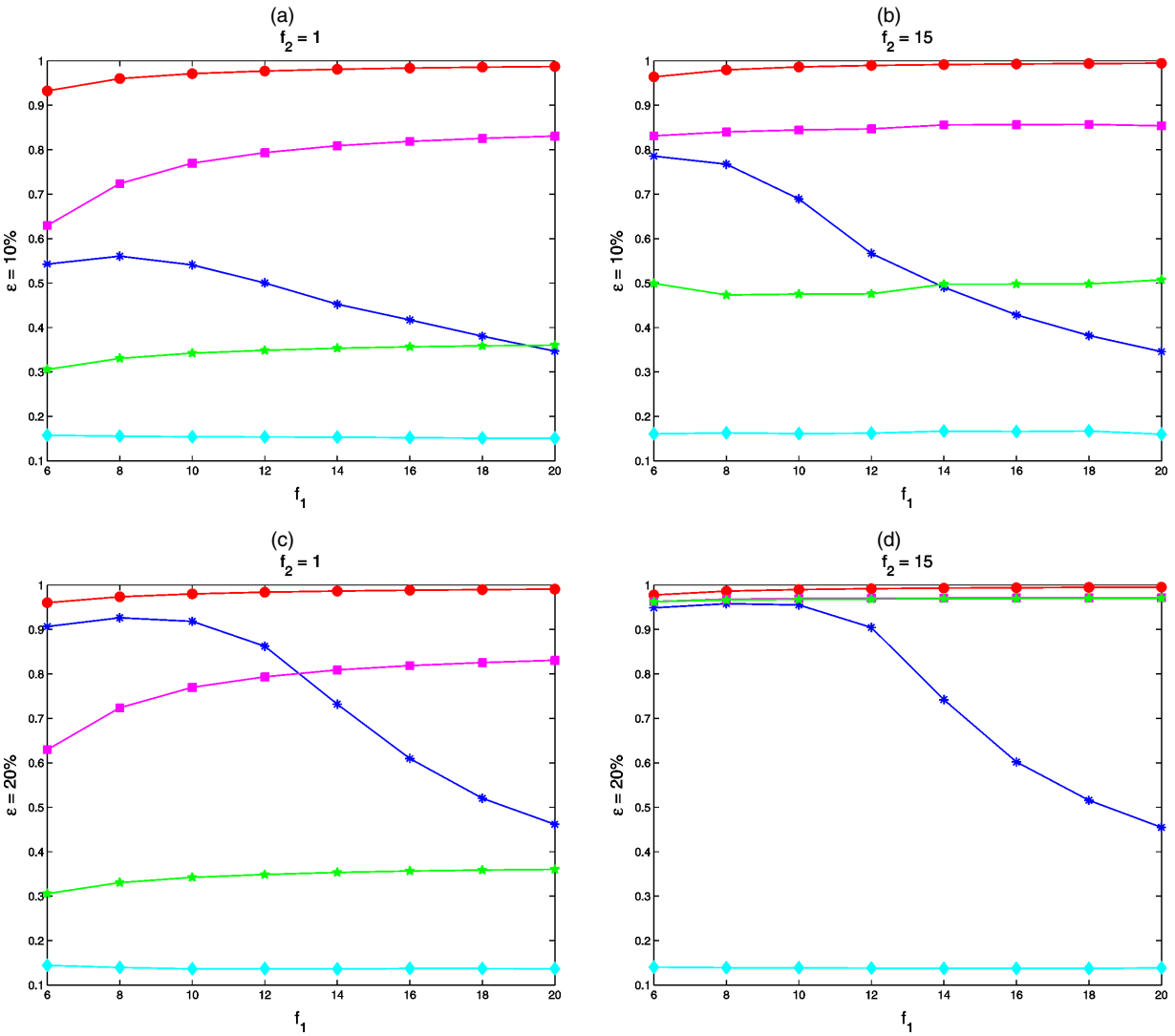

Figure 8. The maxsub Value of the Low-Dimensional Multivariate Normal Data for (a) $\varepsilon = 10\%$ and $f_2 = 1$, (b) $\varepsilon = 10\%$ and $f_2 = 15$, (c) $\varepsilon = 20\%$ and $f_2 = 1$, and (d) $\varepsilon = 20\%$ and $f_2 = 15$. The curves represent the results for CPCA (•), SPHER (■), ELL (⋆), RAPCA (∗), and ROBPCA (♦).

The results of the maxsub measure for simulations (4b) and (100b) are summarized in Figures 8–11. In every situation, CPCA clearly fails and provides the worst possible result, because maxsub is always very close to 1. This implies that the estimated PCA subspace has been attracted by outliers in such a way that at least one principal component is orthogonal to $\mathbf{E}_k$. RAPCA, SPHER, and ELL are also clearly influenced by the outliers, most strongly when the data are high-dimensional or when there is a high percentage of contamination. In all situations, ROBPCA outperforms the other methods. ROBPCA attains high values for maxsub at the long-tailed t_5 only when f_1 is between 6 and 8. This is because in this case the outliers are not yet very well separated from the regular data group. The other methods also fail in such a situation. As soon as the contamination lies somewhat further, ROBPCA is capable of distinguishing the outliers, and maxsub remains almost constant.

Finally, we summarize some results for the MSEs of the eigenvalues in Figure 12. The figure displays the ratio of the MSEs of CPCA versus ROBPCA and RAPCA versus ROBPCA, for the normally distributed data with $\varepsilon = 10\%$ contamination and $f_2 = 1$. Figures 12(a) and 12(b) show the results for the low-dimensional data, whereas Figures 12(c) and 12(d) present the results for the high-dimensional ones. Comparing CPCA and ROBPCA, we see that the MSE of the first CPCA eigenvalue increases strongly when the contamination is shifted further away from the regular points. Also, the MSEs of the other CPCA eigenvalues are much larger than those of ROBPCA. Only $\mathrm{MSE}(\hat{\lambda}_2)$ and $\mathrm{MSE}(\hat{\lambda}_3)$ in Figure 12(c) are of the same order of magnitude.

Figures 12(b) and 12(d) demonstrate the superiority of ROBPCA over RAPCA. For high-dimensional data, the differences are most prominent in the fifth eigenvalue. This explains the bad results for maxsub obtained with RAPCA in this situation. The first four eigenvalues (and their eigenvectors) are well estimated, but the fifth eigenvalue is clearly attracted by the outliers.

4.1 Computation Time

Although ROBPCA is slower than the other methods discussed in this article, its computation time is still very short. Our Matlab implementation requires only 3.19 seconds for the

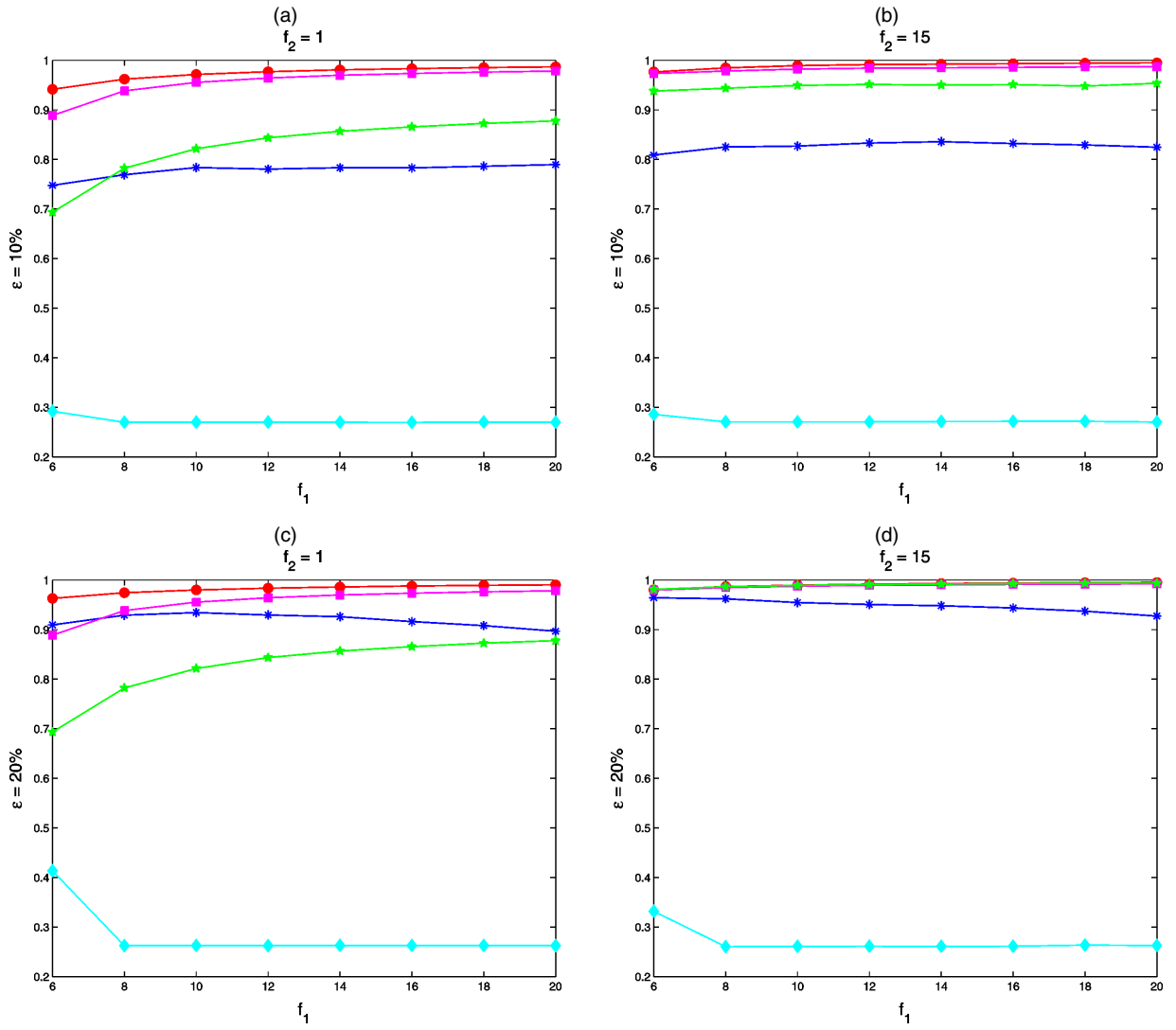

Figure 9. The maxsub Value of the High-Dimensional Multivariate Normal Data for (a) $\varepsilon = 10\%$ and $f_2 = 1$, (b) $\varepsilon = 10\%$ and $f_2 = 15$, (c) $\varepsilon = 20\%$ and $f_2 = 1$, and (d) $\varepsilon = 20\%$ and $f_2 = 15$. The curves represent the results for CPCA (●), SPHER (■), ELL (⋆), RAPCA (∗), and ROBPCA (♦).

car dataset ($n = 111$, $p = 11$), 3.06 seconds for the octane dataset ($n = 39, p = 226$), and 4.16 seconds for the glass dataset ($n = 180, p = 750$) on a 2.40-GHz Pentium IV processor.

Figure 13(a) shows the mean CPU time in seconds over 100 runs for varying low-dimensional normal data. The sample sizes vary from 50 to 5,000, and p is relatively small ($p = 4$ or $p = 10$). We see that the computation time is linear in n and k. From Figure 13(b), the same conclusion can be drawn for high-dimensional datasets. We also looked at the effect of varying p while holding $n = 100$ and $k = 4$ constant. In this case the mean CPU time was 3.2 seconds for $p = 10$ and increased to only 4.3 seconds for $p = 3{,}000$.

5. CONCLUSION AND OUTLOOK

We have constructed a fast and robust algorithm for PCA of high-dimensional data. The algorithm first applies PP techniques in the original data space. These results are then used to project the observations into a subspace of small to moderate dimension. Within this subspace, ideas of robust covariance estimation are the applied. Throughout, we have the ability to detect exact fit situations and to reduce the dimension accordingly. Simulations and applications to real data demonstrate that this ROBPCA algorithm yields very robust estimates when the data contains outliers. The associated diagnostic plot is a useful graphical tool that allows one to visualize and classify the outliers.

As mentioned in Section 1, data analysis often starts with PCA. We have used a robust PCA before applying a robust discriminant analysis technique (Hubert and Van Driessen 2003; Hubert and Engelen 2004) and a robust method for logistic regression (Rousseeuw and Christmann 2003). In addition, using ROBPCA in robust principal components regression (Hubert and Verboven 2003) and robust partial least squares (Hubert and Vanden Branden 2003) has been investigated. The ROBPCA method thus opens a door to practical robust multivariate calibration and to the analysis of regression data with both outliers and multicollinearity. To select the number of principal components based on the predictive power of the model, fast methods of cross-validation have been developed (Engelen and Hubert 2004).

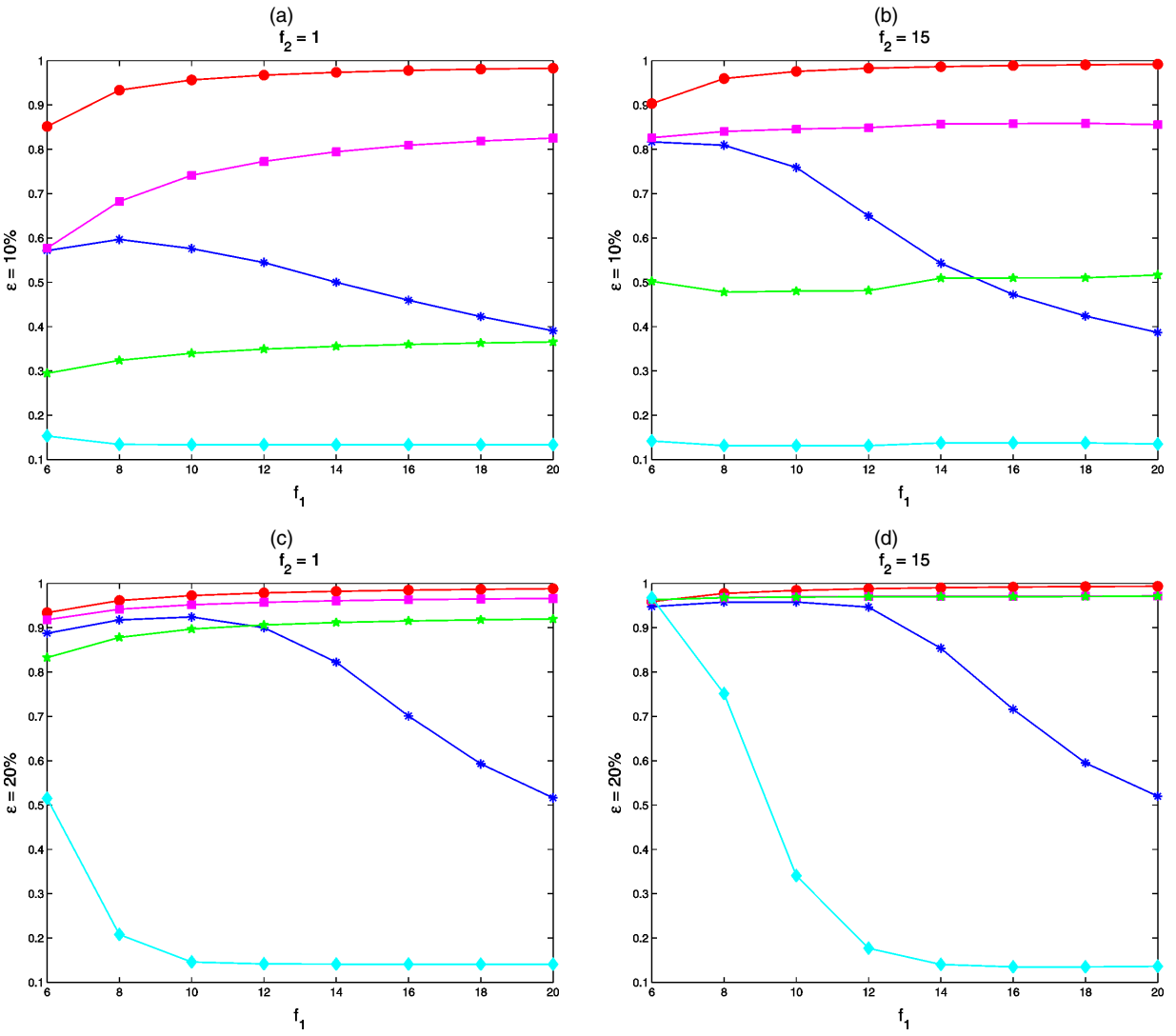

Figure 10. The maxsub Value of the Low-Dimensional Multivariate t_5 Data for (a) $\varepsilon = 10\%$ and $f_2 = 1$, (b) $\varepsilon = 10\%$ and $f_2 = 15$, (c) $\varepsilon = 20\%$ and $f_2 = 1$, and (d) $\varepsilon = 20\%$ and $f_2 = 15$. The curves represent the results for CPCA (●), SPHER (■), ELL (⋆), RAPCA (∗), and ROBPCA (⧫).

The Matlab program *robpca* and auxiliary functions are available at the websites *http://www.agoras.ua.ac.be/* and *http://www.wis.kuleuven.ac.be/stat/robust.html* as part of LIBRA: the Matlab Library for Robust Analysis (Verboven and Hubert 2004). Also stand-alone S–PLUS and R implementations can be downloaded from these websites.

ACKNOWLEDGMENTS

The authors thank the associate editor and two referees for their constructive remarks on the first version of the manuscript. They also thank Oxana Rodionova for providing the octane dataset, Pascal Lemberge for providing the glass dataset and Steve Marron for providing the Matlab code for the spherical and elliptical PCA methods.

APPENDIX: DETAILED ROBPCA ALGORITHM

Here we describe the ROBPCA method in detail, following the sketch in Section 2.

Stage 1. As proposed by Hubert et al. (2002), we start by reducing the data space to the affine subspace spanned by the n observations. This is especially useful when $p \geq n$, but even when $p < n$, the observations may span less than the whole p-dimensional space. A convenient way to do this is by a singular value decomposition of the mean-centered data matrix, yielding

$$\mathbf{X}_{n,p} - \mathbf{1}_n \hat{\boldsymbol{\mu}}_0' = \mathbf{U}_{n,r_0} \mathbf{D}_{r_0,r_0} \mathbf{V}_{r_0,p}', \tag{A.1}$$

where $\hat{\boldsymbol{\mu}}_0$ is the classical mean vector, $r_0 = \operatorname{rank}(\mathbf{X}_{n,p} - \mathbf{1}_n \hat{\boldsymbol{\mu}}_0')$, $\mathbf{D}$ is an $r_0 \times r_0$ diagonal matrix, and $\mathbf{U}'\mathbf{U} = \mathbf{I}_{r_0} = \mathbf{V}'\mathbf{V}$, where $\mathbf{I}_{r_0}$ is the $r_0 \times r_0$ identity matrix. When $p > n$, we carry out the decomposition in (A.1) using the kernel approach based on computing the eigenvectors and eigenvalues of $(\mathbf{X} - \mathbf{1}_n \hat{\boldsymbol{\mu}}_0')(\mathbf{X} - \mathbf{1}_n \hat{\boldsymbol{\mu}}_0')'$ (Wu, Massart, and de Jong 1997). Because the latter matrix has n rows and columns, its decomposition can be obtained faster than can the decomposition of the $p \times p$ matrix $(\mathbf{X} - \mathbf{1}_n \hat{\boldsymbol{\mu}}_0')'(\mathbf{X} - \mathbf{1}_n \hat{\boldsymbol{\mu}}_0')$.

Without losing any information, we now work in the subspace spanned by the r_0 columns of $\mathbf{V}$. That is, $\mathbf{Z}_{n,r_0} = \mathbf{UD}$

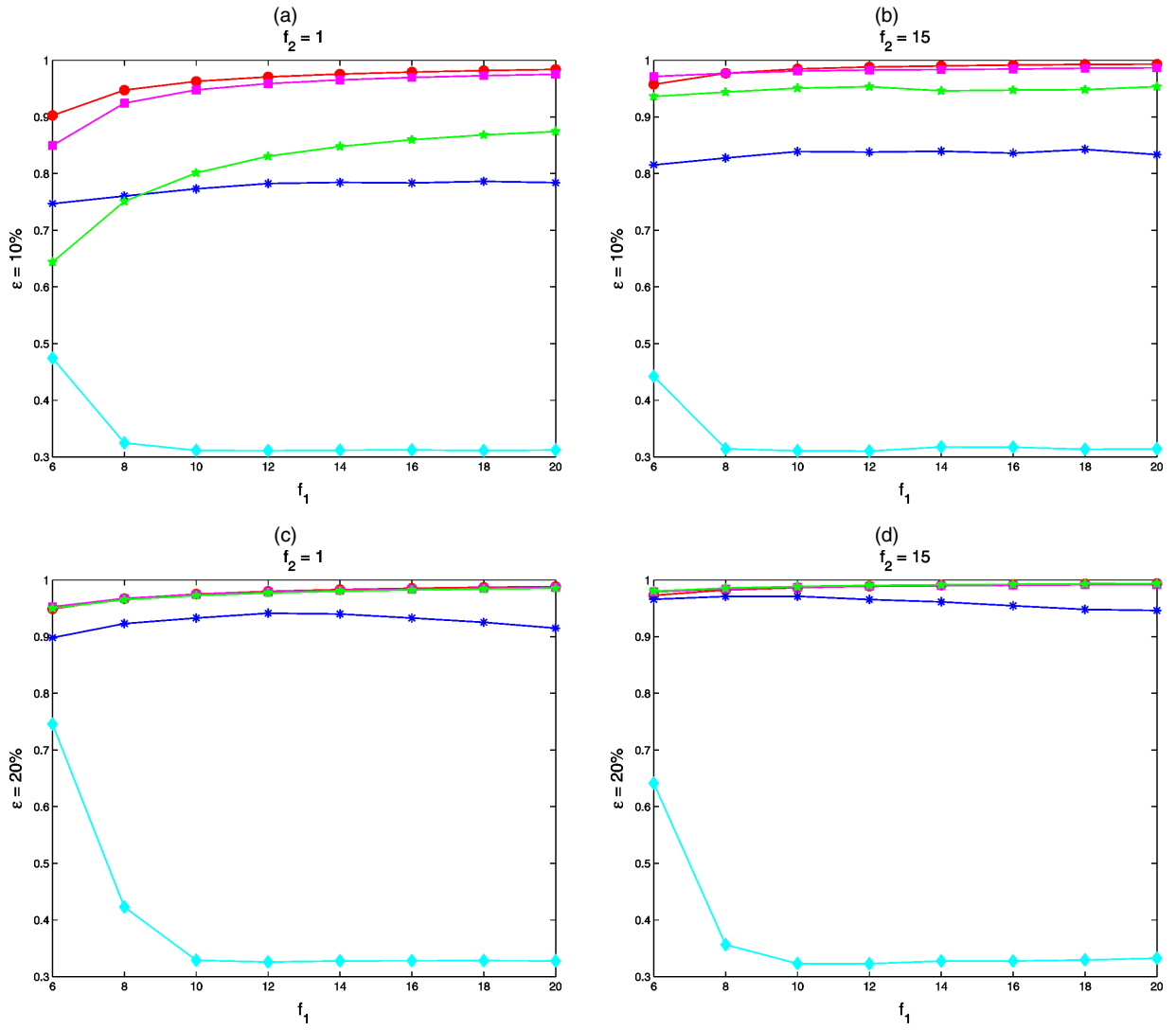

Figure 11. The maxsub Value of the High-Dimensional Multivariate t_5 Data for (a) $\varepsilon = 10\%$ and $f_2 = 1$, (b) $\varepsilon = 10\%$ and $f_2 = 15$, (c) $\varepsilon = 20\%$ and $f_2 = 1$, and (d) $\varepsilon = 20\%$ and $f_2 = 15$. The curves represent the results for CPCA (•), SPHER (■), ELL (⋆), RAPCA (∗), and ROBPCA (♦).

becomes our new data matrix. Note that this singular value decomposition is just an affine transformation of the data. We do not use it to retain only the first eigenvectors of the covariance matrix of $\mathbf{X}_{n,p}$; this would imply that we were performing CPCA, which is of course not robust. Here we merely represent the data in its own dimensionality.

Stage 2. In this stage we try to find the $h < n$ "least outlying" data points. We then use their covariance matrix to obtain a preliminary subspace of dimension k_0. The value of h can be chosen by the user, but $n - h$ should exceed the number of outliers in the dataset. Moreover, h needs to be larger than $[(n + k_0 + 1)/2]$, for reasons that are explained in stage 3 of the algorithm. Because we do not know the number of outliers or k_0 at this moment, we take $h = \max\{[\alpha n], [(n + k_{\max} + 1)/2]\}$, where $k_{\max}$ represents the maximal number of components that will be computed and is set to 10 by default. The parameter α can be chosen as any real value between .5 and 1. The higher the α, the more efficient the estimates will be for uncontaminated data. But setting a lower value for α will increase the robustness of the algorithm for contaminated samples. Our default, which is also used in the simulations, is $\alpha = .75$.

To find the h "least outlying" data points, we proceed as follows:

1. For each data point $\mathbf{x}_i$, we compute its outlyingness. The Stahel–Donoho affine-invariant outlyingness (Stahel 1981; Donoho 1982) is defined as

$$\text{outl}_A(\mathbf{x}_i) = \max_{\mathbf{v} \in B} \frac{|\mathbf{x}_i'\mathbf{v} - \text{med}(\mathbf{x}_j'\mathbf{v})|}{\text{mad}(\mathbf{x}_j'\mathbf{v})}, \qquad \text{(A.2)}$$

where B contains all non-0 vectors, $\text{med}(\mathbf{x}_j^t\mathbf{v})$ is the median of $\{\mathbf{x}_j'\mathbf{v}, j = 1, \ldots, n\}$, and $\text{mad}(\mathbf{x}_j'\mathbf{v}) = \text{med}|\mathbf{x}_j'\mathbf{v} - \text{med}(\mathbf{x}_j'\mathbf{v})|$. In a PCA analysis we need only an orthogonally invariant measure, so we can restrict the set B to all directions through two data points. If $\binom{n}{2} > 250$, then we take at random 250 directions from B. Moreover, we replace the median and the mad in (A.2) by the univariate MCD location and scale estimator (Rousseeuw 1984), denoted by t_{MCD} (resp. s_{MCD}). These estimators are defined as the mean (resp. the standard deviation) of the h observations with smallest variance. The estimators

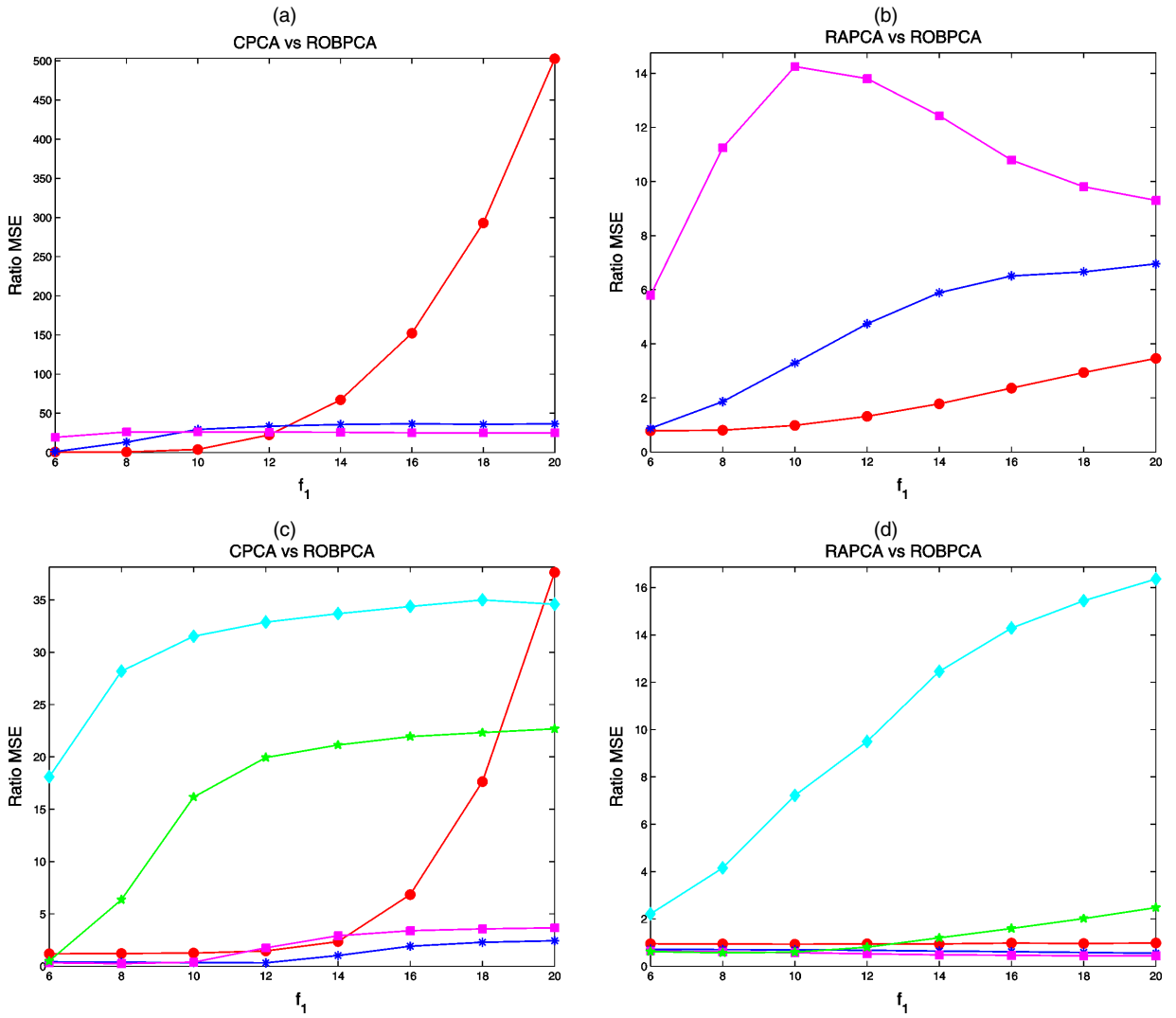

Figure 12. The Ratio of the MSEs of the Normal Data, $\varepsilon = 10\%$, and $f_2 = 1$ for (a) CPCA versus ROBPCA and (b) RAPCA versus ROBPCA for the Low-Dimensional Data and (c) CPCA versus ROBPCA and (d) RAPCA versus ROBPCA for the High-Dimensional Data. The curves represent the ratio of the MSE of $\hat{\lambda}_1$ (•), $\hat{\lambda}_2$ (■), $\hat{\lambda}_3$ (⋆), $\hat{\lambda}_4$ (∗), and $\hat{\lambda}_5$ (♦).

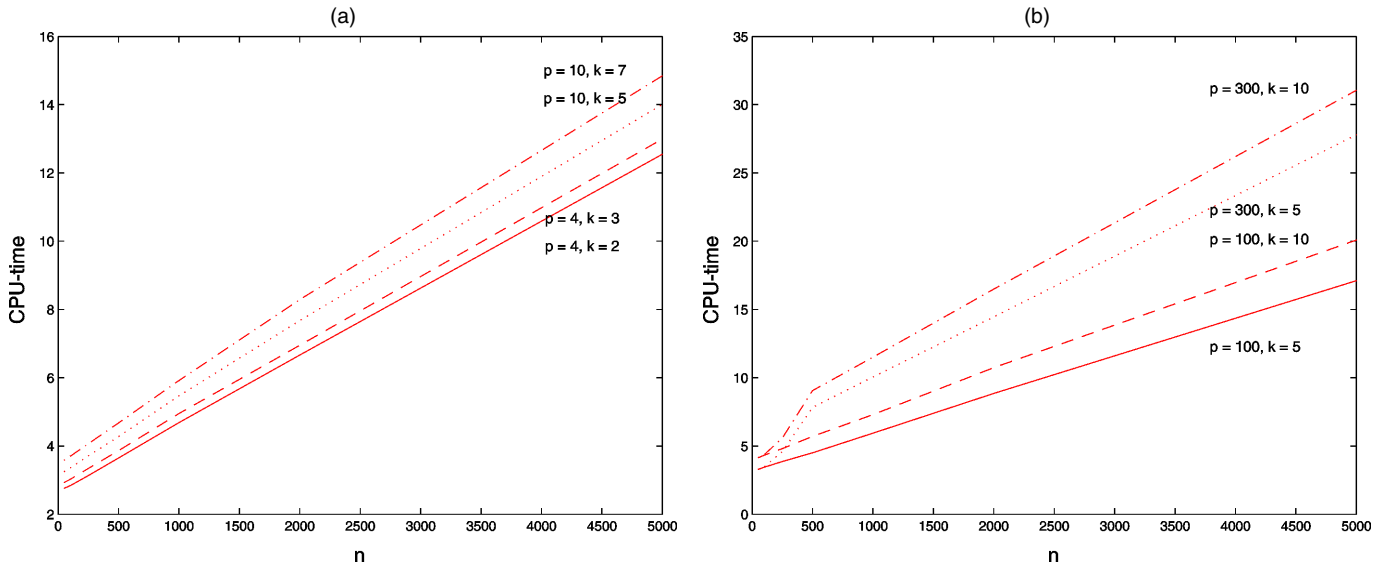

Figure 13. Mean CPU Time (in seconds) Over 100 Runs of ROBPCA for (a) Low-Dimensional Data and (b) High-Dimensional Data.

t_{MCD} and s_{MCD} can be easily computed in $O(n\log(n))$ time (Rousseeuw and Leroy 1987, p. 171). Summarizing, for each direction $\mathbf{v} \in B$, we project the n data points $\mathbf{x}_i$ on $\mathbf{v}$ and compute their robustly standardized absolute residual, $|\mathbf{x}_i'\mathbf{v} - t_{MCD}(\mathbf{x}_j'\mathbf{v})|/s_{MCD}(\mathbf{x}_j'\mathbf{v})$. This leads to the orthogonally invariant outlyingness,

$$\text{outl}_O(\mathbf{x}_i) = \max_{\mathbf{v}\in B} \frac{|\mathbf{x}_i'\mathbf{v} - t_{MCD}(\mathbf{x}_j'\mathbf{v})|}{s_{MCD}(\mathbf{x}_j'\mathbf{v})}. \quad \text{(A.3)}$$

a. When all robust scales s_{MCD} are nonzero, we can compute $\text{outl}_O(\mathbf{x}_i)$ for all data points and consider the h observations with smallest outlyingness. Their indices are stored in the set H_0.
b. When we encounter a direction $\mathbf{v}$ in which the projected observations have zero robust scale [i.e., $s_{MCD}(\mathbf{x}_j'\mathbf{v}) = 0$], we have in fact found a hyperplane $H_\mathbf{v}$ orthogonal to $\mathbf{v}$ that contains h observations. This is called an "exact fit" situation. When this happens, we project all of the data points on $H_\mathbf{v}$, thereby reducing the true dimension by one. To perform this projection, we apply the reflection step, described in detail by Hubert et al. (2002). The reflection step starts by reflecting all data such that the normalized vector $\mathbf{v}/\|\mathbf{v}\|$ coincides with the first basis vector $\mathbf{e}_1$. The projection on the orthogonal complement of $\mathbf{v}$ then simply corresponds to removing the first coordinate of each data point. We then repeat the search for the h least outlying data points in $H_\mathbf{v}$; that is, we return to step 1.

Note that the exact fit situation can occur more than once, in which case we reduce the working dimension sequentially. We end up with a dataset in some dimension $r_1 \le r_0$ and a set H_0 indexing the h data points with smallest outlyingness. For convenience, we still denote our lower-dimensional data points by $\mathbf{x}_i$.

2. We now consider $\hat{\boldsymbol{\mu}}_1$ and $\mathbf{S}_0$ the mean and covariance matrix of the h observations in H_0. We follow the convention that the eigenvalues of any scatter matrix are sorted in descending order and the eigenvectors are indexed accordingly. This means that the eigenvector $\mathbf{v}_1$ corresponds to the largest eigenvalue, $\mathbf{v}_2$ corresponds to the second largest eigenvalue, and so on. The spectral decomposition of $\mathbf{S}_0$ is denoted by

$$\mathbf{S}_0 = \mathbf{P}_0\mathbf{L}_0\mathbf{P}_0', \quad \text{(A.4)}$$

with $\mathbf{L} = \text{diag}(\tilde{l}_1, \ldots, \tilde{l}_r)$ and $r \le r_1$.

The covariance matrix S_0 is used to decide how many principal components $k_0 \le r$ will be retained in the further analysis. We can do this in various ways; for instance, we can look at the scree plot, which is a graph of the (monotone-decreasing) eigenvalues, or we can use a selection criterion such as (5) or (6). See also Engelen and Hubert (2004) for a method based on cross-validation.

3. Finally, we project the data points on the subspace spanned by the first k_0 eigenvectors of $\mathbf{S}_0$. To implement this step, we set

$$\mathbf{X}^*_{n,k_0} = \left(\mathbf{X}_{n,r_1} - \mathbf{1}_n\hat{\boldsymbol{\mu}}_1'\right)\mathbf{P}_{r_1,k_0},$$

where $\mathbf{P}_{r_1,k_0}$ consists of the first k_0 columns of $\mathbf{P}_0$ in (A.4).

Stage 3. In the third stage of the algorithm, we robustly estimate the scatter matrix of the data points in $\mathbf{X}^*_{n,k_0}$ using the MCD estimator. Recall that for this, we need to find h data points whose covariance matrix has minimal determinant. Because in general we cannot consider all h-subsets, we must rely on approximate algorithms. Here we slightly adapt the FAST–MCD algorithm of Rousseeuw and Van Driessen (1999) by taking advantage of the result of stage 2, which used the outlyingness measure (A.3):

1. We first apply C-steps, starting from the $\mathbf{x}_i^*$ with $i \in H_0$ (the index set H_0 was obtained in step 1 in stage 2). The C-step was proposed by Rousseeuw and Van Driessen (1999), who gave it a fundamental role in the fast computation of the MCD estimator. It is defined as follows: Let $\mathbf{m}_0$ and $\mathbf{C}_0$ be the mean and the covariance matrix of the h points in H_0. Then:

a. If $\det(\mathbf{C}_0) > 0$, then we compute the robust distances of all data points with respect to $\mathbf{m}_0$ and $\mathbf{C}_0$, denoted as

$$d_{\mathbf{m}_0,\mathbf{C}_0}(i) = \sqrt{(\mathbf{x}_i^* - \mathbf{m}_0)'\mathbf{C}_0^{-1}(\mathbf{x}_i^* - \mathbf{m}_0)} \quad \text{for } i = 1, \ldots, n. \quad \text{(A.5)}$$

We then define the subset H_1 as the (indices of the) h points with smallest robust distances $d_{\mathbf{m}_0,\mathbf{C}_0}(i)$. We then use this subset to compute $\mathbf{m}_1$, $\mathbf{C}_1$, and all robust distances $d_{\mathbf{m}_1,\mathbf{C}_1}(i)$. Rousseeuw and Van Driessen (1999) proved that always $\det(\mathbf{C}_1) \le \det(\mathbf{C}_0)$. We continue updating the subset until the determinant of the covariance matrix no longer decreases.

b. if at some iteration step $m = 0, 1, \ldots,$ a covariance matrix $\mathbf{C}_m$ is found to be singular, then we project the data points on the lower-dimensional space spanned by the eigenvectors of $\mathbf{C}_m$ that correspond to its nonzero eigenvalues, and we continue the C-steps inside that space.

On convergence, we obtain a data matrix, still denoted by $\mathbf{X}^*_{n,k_1}$, with $k_1 \le k_0$ variables, and indices of the final h-subset, which are stored in the set H_1.

2. We now apply the FAST–MCD algorithm to $\mathbf{X}^*_{n,k_1}$. This algorithm draws many random subsets of size $(k_1 + 1)$ out of $\mathbf{X}^*$. In each subset, the mean and the covariance matrix are computed. Then the robust distances (A.5) with respect to this mean and covariance matrix are obtained for all observations. Next, the $(k_1 + 1)$-subset is enlarged to an h-subset by considering the h observations with smallest robust distances. This h-subset is then used to start C-steps. Note that the FAST–MCD algorithm generates quasi-random h-subsets, whereas in step 1 of stage 3 C-steps are applied starting from one specific h-subset H_1.

The FAST–MCD algorithm contains several time-saving techniques. For instance, it does not apply C-steps until convergence for each h-subset under consideration. Instead, it carries out two C-steps for each, selects the 10 best results, and only iterates fully starting from these. Moreover, when n is large, the algorithm constructs several nonoverlapping representative subsets of 300–600 cases. It first applies C-steps to those subsets, then uses the best solutions as starts for C-steps in the union of

those subsets. In the end, the best solutions are iterated in the whole dataset.

Whereas FAST–MCD draws 500 random subsets of size $(k_1 + 1)$ by default, we use 250 random subsets in the ROBPCA algorithm. This reduced the computation time considerably, and in simulations had almost no effect on the estimates. This is because we already used the outlyingness measure (A.3), which gave us a very good initial h-subset, allowing us to draw fewer random subsets.

In this step of the ROBPCA algorithm, we also could have used other algorithms for robust covariance estimation. One reason we chose the FAST–MCD algorithm was that to the best of our knowledge, it is currently the only algorithm that can deal with exact fit situations. When these situations occur, the algorithm reduces the working dimension.

The final data set is denoted by $\tilde{\mathbf{X}}_{n,k}$ with $k \leq k_1$. Let $\hat{\boldsymbol{\mu}}_2$ and $\mathbf{S}_1$ denote the mean and covariance matrix of the h-subset found in step 1, and let $\hat{\boldsymbol{\mu}}_3$ and $\mathbf{S}_2$ denote the mean and covariance matrix found by the FAST–MCD algorithm. If $\det(\mathbf{S}_1) < \det(\mathbf{S}_2)$, then we continue our computations based on $\hat{\boldsymbol{\mu}}_2$ and $\mathbf{S}_1$. For this, we set $\hat{\boldsymbol{\mu}}_4 = \hat{\boldsymbol{\mu}}_2$ and $\mathbf{S}_3 = \mathbf{S}_1$. Otherwise, we let $\hat{\boldsymbol{\mu}}_4 = \hat{\boldsymbol{\mu}}_3$ and $\mathbf{S}_3 = \mathbf{S}_2$.

3. Based on $\hat{\boldsymbol{\mu}}_4$ and $\mathbf{S}_3$, we compute a reweighted mean and covariance matrix to increase the statistical efficiency. First, we multiply $\mathbf{S}_3$ by a consistency factor, c_1, to make the estimator unbiased at normal distributions. As proposed by Rocke and Woodruff (1996), we use the consistency factor of Rousseeuw and Van Driessen (1999), adapted with the hth quantile of the robust distances instead of their median, so that

$$c_1 = \frac{\{d^2_{\hat{\mu}_4,S_3}\}_{(h)}}{\chi^2_{k,h/n}},$$

with $\{d^2_{\hat{\mu}_4,S_3}\}_{(1)} \leq \{d^2_{\hat{\mu}_4,S_3}\}_{(2)} \leq \cdots \leq \{d^2_{\hat{\mu}_4,S_3}\}_{(n)}$. Let d_i $(i = 1, \ldots, n)$ be the robust distances of all observations with respect to $\hat{\boldsymbol{\mu}}_4$ and $c_1\mathbf{S}_3$, let w be a weight function, and put $w_i = w(d_i)$ for all i. Then the center and scatter of the data are estimated by

$$\hat{\boldsymbol{\mu}}_5 = \frac{\sum_{i=1}^n w_i \tilde{\mathbf{x}}_i}{\sum_{i=1}^n w_i}$$

and

$$\mathbf{S}_4 = \frac{\sum_{i=1}^n w_i (\tilde{\mathbf{x}}_i - \hat{\boldsymbol{\mu}}_5)(\tilde{\mathbf{x}}_i - \hat{\boldsymbol{\mu}}_5)'}{\sum_{i=1}^n w_i - 1}.$$

In our implementation we use "hard rejection" by taking $w(d_i) = I(d_i \leq \sqrt{\chi^2_{k,.975}})$, where I denotes the indicator function.

The spectral decomposition of $\mathbf{S}_4$ can be written as $\mathbf{S}_4 = \mathbf{P}_2\mathbf{L}_2\mathbf{P}_2'$, where the columns of $\mathbf{P}_2 = \mathbf{P}_{k,k}$ contain the eigenvectors of $\mathbf{S}_4$ and $\mathbf{L}_2 = \mathbf{L}_{k,k}$ is the diagonal matrix with the corresponding eigenvalues. The final scores are now given by

$$\mathbf{T}_{n,k} = (\tilde{\mathbf{X}}_{n,k} - \mathbf{1}_n\hat{\boldsymbol{\mu}}_5')\mathbf{P}_2. \qquad (A.6)$$

4. The last step transforms the columns of $\mathbf{P}_2$ back to $\mathbb{R}^p$, yielding the final robust principal components $\mathbf{P}_{p,k}$. The final robust center $\hat{\boldsymbol{\mu}}$ is obtained by transforming $\hat{\boldsymbol{\mu}}_5$ back to $\mathbb{R}^p$, and the final p-dimensional robust scatter matrix $\mathbf{S}$ of rank k is given by (2). The scores (A.6) can be written as the equivalent formula (1) in $\mathbb{R}^p$. Note that the robust score distance, SD_i, of (3) can be computed in the k-dimensional PCA space by the equivalent formula $SD_i = \sqrt{(\tilde{\mathbf{x}}_i - \hat{\boldsymbol{\mu}}_5)'\mathbf{S}_4^{-1}(\tilde{\mathbf{x}}_i - \hat{\boldsymbol{\mu}}_5)}$, which saves computation time in high-dimensional applications.

[Received February 2002. Revised November 2003.]

REFERENCES

Boente, G., Pires, A. M., and Rodrigues, I. (2002), "Influence Functions and Outlier Detection Under the Common Principal Components Model: A Robust Approach," *Biometrika*, 89, 861–875.

Box, G. E. P. (1954), "Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems: Effect of Inequality of Variance in One-Way Classification," *The Annals of Mathematical Statistics*, 25, 290–302.

Campbell, N. A. (1980), "Robust Procedures in Multivariate Analysis I: Robust Covariance Estimation," *Applied Statistics*, 29, 231–237.

Croux, C., and Haesbroeck, G. (2000), "Principal Components Analysis Based on Robust Estimators of the Covariance or Correlation Matrix: Influence Functions and Efficiencies," *Biometrika*, 87, 603–618.

Croux, C., and Ruiz-Gazen, A. (1996), "A Fast Algorithm for Robust Principal Components Based on Projection Pursuit," in *COMPSTAT 1996, Proceedings in Computational Statistics*, ed. A. Prat, Heidelberg: Physica-Verlag, pp. 211–217.

Davies, L. (1987), "Asymptotic Behavior of S-Estimators of Multivariate Location and Dispersion Matrices," *The Annals of Statistics*, 15, 1269–1292.

Donoho, D. L. (1982), "Breakdown Properties of Multivariate Location Estimators," Ph.D. qualifying paper, Harvard University.

Egan, W. J., and Morgan, S. L. (1998), "Outlier Detection in Multivariate Analytical Chemical Data," *Analytical Chemistry*, 70, 2372–2379.

Engelen, S., and Hubert, M. (2004), "Fast Cross-Validation in Robust PCA," in *COMPSTAT 2004, Proceedings in Computational Statistics*, ed. J. Antoch, Heidelberg: Springer-Verlag, pp. 989–996.

Esbensen, K. H., Schönkopf, S., and Midtgaard, T. (1994), *Multivariate Analysis in Practice*, Trondheim: Camo.

Hubert, M., and Engelen, S. (2004), "Robust PCA and Classification in Biosciences," *Bioinformatics*, 20, 1728–1736.

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

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

Hubert, M., and Van Driessen, K. (2004), "Fast and Robust Discriminant Analysis," *Computational Statistics and Data Analysis*, 45, 301–320.

Hubert, M., and Verboven, S. (2003), "A Robust PCR Method for High-Dimensional Regressors," *Journal of Chemometrics*, 17, 438–452.

Joliffe, I. T. (1986), *Principal Component Analysis*, New York: Springer-Verlag.

Krzanowski, W. J. (1979), "Between-Groups Comparison of Principal Components," *Journal of the American Statistical Association*, 74, 703–707.

Lemberge, P., De Raedt, I., Janssens, K. H., Wei, F., and Van Espen, P. J. (2000), "Quantitative Z-Analysis of the 16th–17th Century Archaeological Glass Vessels Using PLS Regression of EPXMA and μ-XRF Data," *Journal of Chemometrics*, 14, 751–763.

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

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

Maronna, R. A. (1976), "Robust M-Estimators of Multivariate Location and Scatter," *The Annals of Statistics*, 4, 51–67.

Nomikos, P., and MacGregor, J. F. (1995), "Multivariate SPC Charts for Monitoring Batch Processes," *Technometrics*, 37, 41–59.

Rocke, D. M., and Woodruff, D. L. (1996), "Identification of Outliers in Multivariate Data," *Journal of the American Statistical Association*, 91, 1047–1061.

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

Rousseeuw, P. J., and Christmann, A. (2003), "Robustness Against Separation and Outliers in Logistic Regression," *Computational Statistics and Data Analysis*, 43, 315–332.

Rousseeuw, P. J., and Leroy, A. (1987), *Robust Regression and Outlier Detection*, New York: Wiley.

Rousseeuw, P. J., and Van Driessen, K. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, 41, 212–223.

Rousseeuw, P. J., and Van Zomeren, B. C. (1990), "Unmasking Multivariate Outliers and Leverage Points," *Journal of the American Statistical Association*, 85, 633–651.

Stahel, W. A. (1981), "Robust Estimation: Infinitesimal Optimality and Covariance Matrix Estimators," Ph.D. thesis, ETH, Zürich.

Verboven, S., and Hubert, M. (2004), "LIBRA: A Matlab Library for Robust Analysis," *Chemometrics and Intelligent Laboratory Systems*, doi:10.1016/j.chemolab.2004.06.003.

Woodruff, D. L., and Rocke, D. M. (1994), "Computable Robust Estimation of Multivariate Location and Shape in High Dimension Using Compound Estimators," *Journal of the American Statistical Association*, 89, 888–896.

Wu, W., Massart, D. L., and de Jong, S. (1997), "The Kernel PCA Algorithms for Wide Data. Part I: Theory and Algorithms," *Chemometrics and Intelligent Laboratory Systems*, 36, 165–172.