Session 5: Robust principal component analysis
Winter course, CMStatistics 2016

Mia Hubert, Peter Rousseeuw, Stefan Van Aelst

Department of Mathematics
KU Leuven, Belgium

December 6–7, 2016

KU LEUVEN

Outline of the course

- 1. General notions of robustness
- 2. Robustness for univariate data
- 3. Robust multivariate methods
- 4. Robust regression
- 5. Robust principal component analysis
- 6. Inference
- 7. Multivariate and functional depth
- 8. High dimensional data and sparsity
- 9. Cellwise outliers

Principal component analysis: Outline

1. Classical PCA
2. Outlier detection in PCA
3. Robust PCA based on a robust covariance estimator
4. Robust PCA based on projection pursuit
5. Spherical robust PCA
6. ROBPCA, based on projection pursuit and the MCD
7. Robust PCA for skewed data

Classical PCA

- Consider a dataset $X_{n,p} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ with $\boldsymbol{x}_i \in \mathbb{R}^p$
- We assume that the variables are continuous.
- Sometimes p is very large: $p \gg 500$ and/or $p > n$.
- The main objective of PCA is to reduce the dimension of the data set without losing too much information.
- One looks for a k-dimensional subspace of $\mathbb{R}^p$ (with $k \ll \min(n, p)$) such that the projection of the data on this subspace contains most of the information of the original p-dimensional data.
- We thus search for a center $\hat{\boldsymbol{\mu}}$ and a loading matrix $P_{p,k}$ (of size $p \times k$) such that the k-dimensional scores $\boldsymbol{t}_i$

  $$\boldsymbol{t}_i = (P')_{k,p}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})$$

  are the most informative.

Classical PCA

Classical PCA (CPCA) seeks the directions of maximum variability of the data.

In particular, it computes the loading matrix

$$P_{p,k} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_k]$$

where the first column is chosen as

$$\boldsymbol{p}_1 = \underset{||\boldsymbol{p}||=1}{\text{argmax}} \ \ \text{var}\{\boldsymbol{p}'(\boldsymbol{x}_1 - \bar{\boldsymbol{x}}), \boldsymbol{p}'(\boldsymbol{x}_2 - \bar{\boldsymbol{x}}), \ldots, \boldsymbol{p}'(\boldsymbol{x}_n - \bar{\boldsymbol{x}})\}$$

and all the following columns are chosen sequentially by

$$\boldsymbol{p}_{j+1} = \underset{||\boldsymbol{p}||=1, \boldsymbol{p}\perp\boldsymbol{p}_1, \ldots, \boldsymbol{p}\perp\boldsymbol{p}_j}{\text{argmax}} \ \ \text{var}\{\boldsymbol{p}'(\boldsymbol{x}_1 - \bar{\boldsymbol{x}}), \boldsymbol{p}'(\boldsymbol{x}_2 - \bar{\boldsymbol{x}}), \ldots, \boldsymbol{p}'(\boldsymbol{x}_n - \bar{\boldsymbol{x}})\} \ .$$

Classical PCA

- The solution of this maximization problem yields the loading matrix as the matrix containing the k dominant eigenvectors of the covariance matrix S_n of the data points.
  In particular, the spectral decomposition of S_n yields

  $$S_n = PLP'$$

  with P the $p \times p$ orthogonal matrix containing all p eigenvectors of S_n and L the diagonal matrix with the p eigenvalues $l_1, \ldots, l_p$ in decreasing order.
  The CPCA loading matrix is the matrix $P_{p,k}$ which contains the first k columns of P.
- The eigenvalues l_j equal

  $$l_j = \text{var}\{\boldsymbol{p}_j'(\boldsymbol{x}_1 - \bar{\boldsymbol{x}}), \boldsymbol{p}_j'(\boldsymbol{x}_2 - \bar{\boldsymbol{x}}), \ldots, \boldsymbol{p}_j'(\boldsymbol{x}_n - \bar{\boldsymbol{x}})\} \ .$$

Classical PCA

To select the number of principal components, one typically looks at

- the scree plot, which is a plot of the eigenvalues (in decreasing order)
- the fraction of the total variance explained by the k first principal components:

$$\frac{\sum_{j=1}^{k} l_j}{\sum_{j=1}^{p} l_j}$$

  which is often required to be at least 80%, or 90%,...

Equivariance

- When the variance of the original variables differs a lot between variables, it is recommended to first **standardize** the variables (otherwise the first principal components will be dominated by the variables with largest variance). When the variables are standardized by dividing them by their standard deviation, CPCA comes down to decomposing the **correlation** matrix of the data, instead of the covariance matrix.
- When we apply a robust PCA method, we will standardize the variables by dividing them by the MAD or another robust scale estimator.
- As PCA is sensitive to standardization of the variables, it is NOT affine equivariant. PCA is however **orthogonally equivariant**: when the data are rotated or reflected, the center and the principal components are rotated/reflected accordingly.
- Consequently, any robust PCA method only needs to be orthogonally equivariant. This allows us e.g. to use the L^1-median as robust estimate of the center.

Outlier detection in PCA

1. Classical PCA
2. Outlier detection in PCA
3. Robust PCA based on a robust covariance estimator
4. Robust PCA based on projection pursuit
5. Spherical robust PCA
6. ROBPCA, based on projection pursuit and the MCD
7. Robust PCA for skewed data

Outlier detection in PCA

Any PCA method will result in an estimate of the center $\hat{\boldsymbol{\mu}}$, a loading matrix $P_{p,k}$ with normalized and orthogonal principal components, and a diagonal matrix of eigenvalues $L_{k,k}$.

The orthogonal projection of each observation on the PCA subspace is denoted as $\hat{\boldsymbol{x}}_i \in \mathbb{R}^p$. It is computed as

$$\hat{\boldsymbol{x}}_i = \hat{\boldsymbol{\mu}} + P_{p,k}\boldsymbol{t}_i = \hat{\boldsymbol{\mu}} + P_{p,k}(P')_{k,p}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})$$

Note that $(P')_{k,p}P_{p,k} = I_k$ because the principal components are normalized and orthogonal, but $P_{p,k}(P')_{k,p} \neq I_p$ (unless $k = p$).

We can then consider the **orthogonal distance** of each observation to the k-dimensional subspace:

$$\text{OD}_{i,k} = \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\| = \|\boldsymbol{x}_i - \hat{\boldsymbol{\mu}} - P_{p,k}\boldsymbol{t}_i\|$$

Outlier detection in PCA

To detect outliers with respect to the estimated PCA model, we can identify observations which are outlying

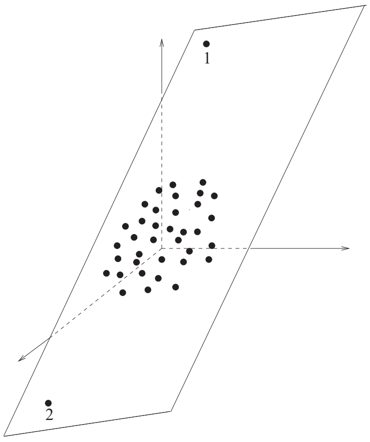
- **relative to** the PCA subspace. We can measure this by computing the orthogonal distance to the PCA subspace.
- **within** the PCA subspace. That is, their projections are outliers in the subspace. We can measure this by computing a robust distance in the k-dimensional PCA subspace. This distance only uses the scores, hence it is called the **score distance**.
  Since the scores are centered, and their variability is estimated by the eigenvalues contained in the $L_{k,k}$ matrix, the score distance is given by:

$$\text{SD}_{i,k} = \sqrt{\boldsymbol{t}_i' L_{k,k}^{-1} \boldsymbol{t}_i} = \sqrt{\sum_{j=1}^{k} \frac{t_{ij}^2}{l_j}}$$

Outlier map

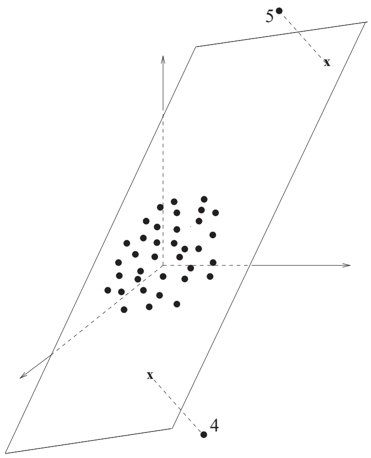
We can distinguish three types of PCA-outliers:

(1) **good PCA leverage points** have an outlying score distance, but a regular orthogonal distance.



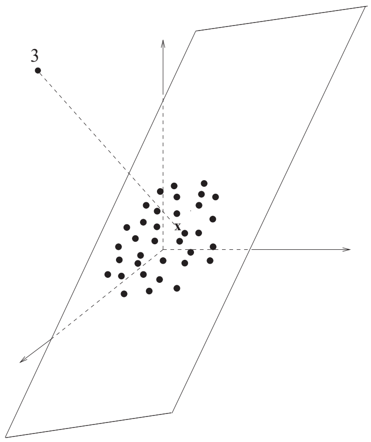
Outlier map

(2) **bad PCA leverage points** have an outlying score distance AND an outlying orthogonal distance.



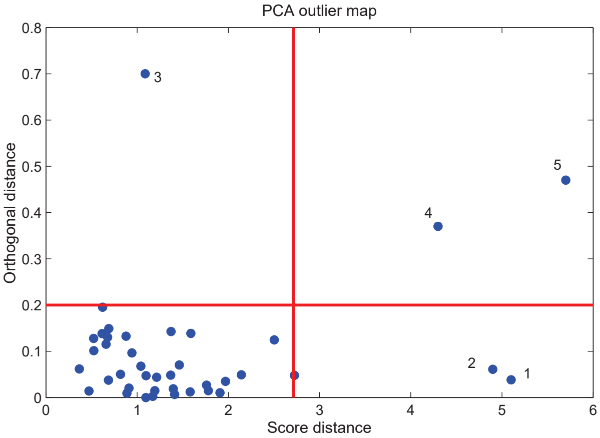
Outlier map

(3) **orthogonal outliers** only have an outlying orthogonal distance.



Outlier map

The PCA outlier map displays the orthogonal distances versus the score distances:



For each type of distance, cut-off values are available to flag outliers (Hubert et al., 2005).

Robust PCA based on a robust covariance estimator

1. Classical PCA
2. Outlier detection in PCA
3. Robust PCA based on a robust covariance estimator
4. Robust PCA based on projection pursuit
5. Spherical robust PCA
6. ROBPCA, based on projection pursuit and the MCD
7. Robust PCA for skewed data

Robust PCA based on a robust covariance estimator

General idea:

- Replace the covariance matrix S_n of X by a robust covariance estimate $\hat{\Sigma}$ such as the MCD, multivariate S, or MM-estimator.
- The robust center corresponds to the robust location estimate associated with $\hat{\Sigma}$.
- The k robust eigenvalues then correspond to the k largest eigenvalues of $\hat{\Sigma}$.
- Take the k corresponding eigenvectors.

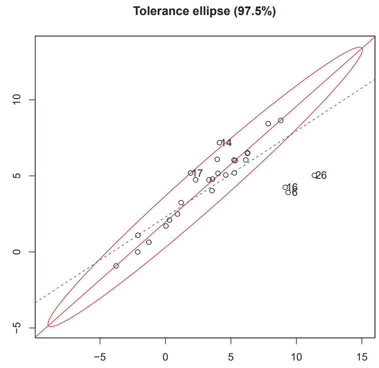
This approach can only be used when $n > 2p$ hence not for high-dimensional data.

Robust covariance-based PCA: Example 1

Example 1: Animals data set $(n = 28, p = 2)$.

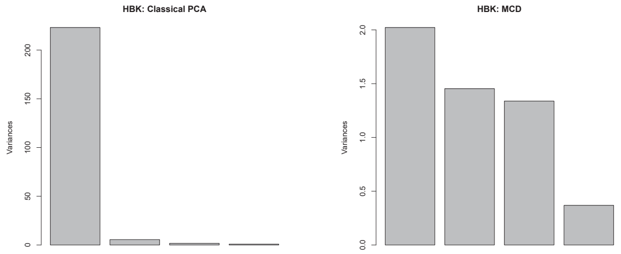
The ellipse is the MCD tolerance ellipse. The red line is the first eigenvector of the MCD covariance matrix. This eigenvector corresponds to the main axis of the tolerance ellipse.

The dotted blue line is the first classical eigenvector.



Robust covariance-based PCA: Example 2

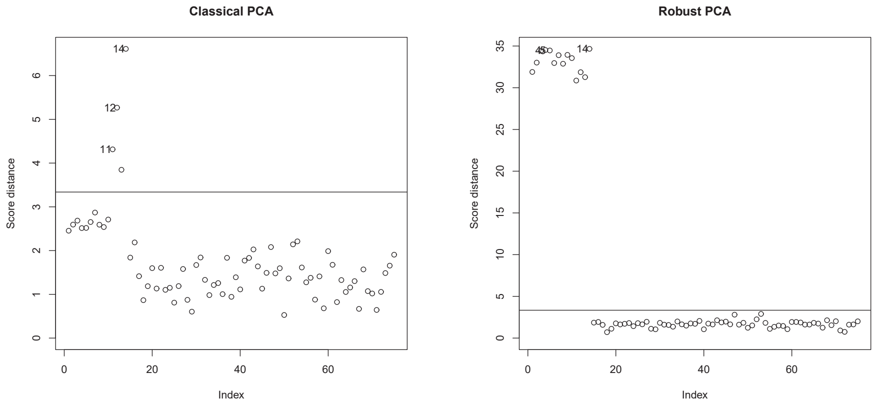
Example 2: Hawkins-Bradu-Kass data set $(n = 75, p = 4)$.
This is an artificial data set with two groups of outliers: observations 1-10 and 11-14. We apply classical PCA and robust PCA based on the MCD estimator with 50% breakdown value. This yields the following scree plots:



The first classical eigenvector already explains 96.5% of the total classical variance. The robust analysis explains 63% of the total variability when $k = 2$ and 93% when $k = 3$.

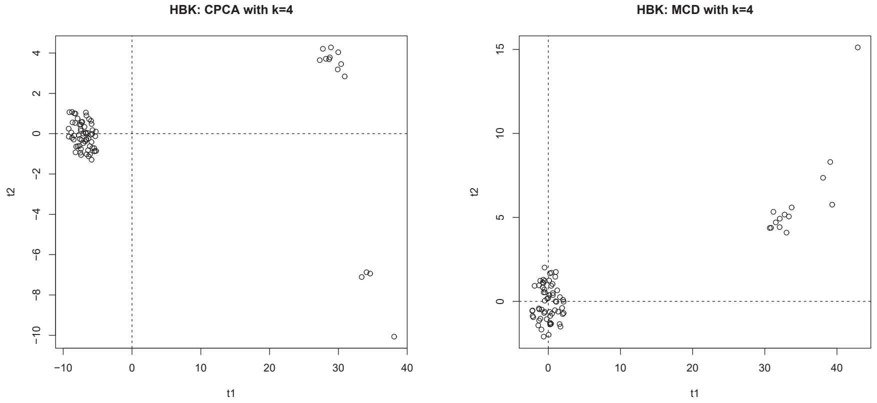
Robust covariance-based PCA: Example 2

When we select all $k = 4$ principal components, we can look at the resulting score distances only:



Robust covariance-based PCA: Example 2

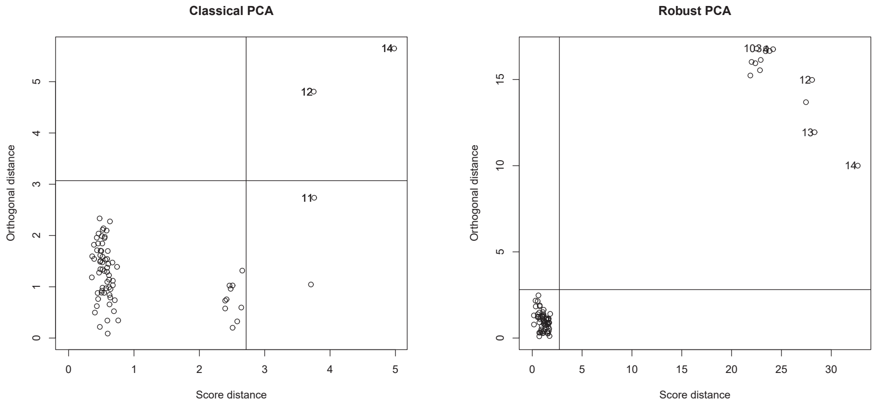
Let us plot the first two scores:



For CPCA always $\bar{\boldsymbol{t}} = \mathbf{0}$, but here (0,0) is not at the center of the regular observations.

Robust covariance-based PCA: Example 2

When we select only $k = 2$ principal components, we can also look at the orthogonal distances:



Robust covariance-based PCA

Code for analyzing this data set in R:

```
> library(rrcov)
> data(hbk)
> pca.hbk50 <- PcaCov(hbk)  # uses MCD with alpha = 0.5
> screeplot(pca.hbk50,main="HBK: MCD")
> plot(pca.hbk50)
> scores.kbk50 <- getScores(pca.hbk50)
> plot(scores.hbk50[,1],scores.hbk50[,2])
> getLoadings(pca.hbk50)
> getEigenvalues(pca.hbk50)
```

Robust PCA based on projection pursuit

1. Classical PCA
2. Outlier detection in PCA
3. Robust PCA based on a robust covariance estimator
4. Robust PCA based on projection pursuit
5. Spherical robust PCA
6. ROBPCA, based on projection pursuit and the MCD
7. Robust PCA for skewed data

Robust PCA based on projection pursuit

Look for directions $\boldsymbol{p}$ such that the data projected on them have the largest spread, but now use a robust measure of univariate spread, e.g. the Q_n estimator.

- Start by robustly estimating the center of the data, e.g. by the L^1 median. This yields $\hat{\boldsymbol{\mu}}$.
- Then search for the k directions $\{\boldsymbol{p}_1, ..., \boldsymbol{p}_k\}$ characterized by:

$$\boldsymbol{p}_{j+1} = \underset{||\boldsymbol{p}||=1, \boldsymbol{p}\perp\boldsymbol{p}_1,\ldots,\boldsymbol{p}\perp\boldsymbol{p}_j}{\text{argmax}} Q_n\{\boldsymbol{p}'(\boldsymbol{x}_1 - \hat{\boldsymbol{\mu}}), \boldsymbol{p}'(\boldsymbol{x}_2 - \hat{\boldsymbol{\mu}}), \ldots, \boldsymbol{p}'(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}})\}$$

- The k robust 'eigenvectors' then correspond to $\{\boldsymbol{p}_1, ..., \boldsymbol{p}_k\}$.
- The k robust 'eigenvalues' l_j then correspond to

$$l_j = (Q_n\{\boldsymbol{p}_j'(\boldsymbol{x}_1 - \hat{\boldsymbol{\mu}}), \boldsymbol{p}_j'(\boldsymbol{x}_2 - \hat{\boldsymbol{\mu}}), \ldots, \boldsymbol{p}_j'(\boldsymbol{x}_n - \hat{\boldsymbol{\mu}})\})^2 \ .$$

The projection pursuit (PP) approach was developed by Li and Chen (1985), Hubert et al. (2002), and Croux and Ruiz-Gazen (2005).

Robust PCA based on projection pursuit

Some advantages of the PP approach:

- It can be used when $p > n$ as it projects the data on lines.
- It is performed sequentially and can be stopped whenever sufficiently many components are obtained.
- The solutions are nested: any j-dimensional PCA subspace is a subspace of all higher-dimensional PCA subspaces found later.
- Fast algorithms are available for its computation:
  - R: the function `PcaGrid` in the `rrcov` package. The function `PcaProj` uses another algorithm.
  - Matlab: the function `rapca` in LIBRA.

Robust PP-based PCA: Bus example

Example: the bus data ($p = 18$ shape features extracted from vehicle silhouettes, $n = 218$). One variable has zero MAD and is removed, hence $p = 17$.

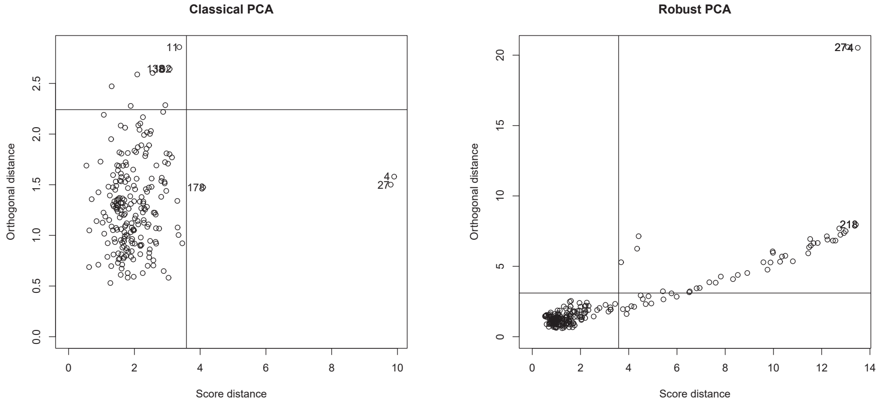
We standardize the data, and apply the projection-based PCA with Q_n as scale estimator. Then 92% of the variability is explained by $k = 5$ components:

```
> pcagrid <- PcaGrid(bus2,method="qn")
> screeplot(pcagrid,main="Projection-based PCA")
> load.grid=getLoadings(pcagrid)
> eigenv.grid=getEigenvalues(pcagrid)
> cumsum(eigenv.grid)/sum(eigenv.grid)

 [1] 0.4418105 0.6475868 0.7938430 0.8635211 0.9241256 0.9544322
 [7] 0.9692368 0.9781212 0.9841466 0.9891732 0.9925678 0.9955517
[13] 0.9969762 0.9982615 0.9994451 0.9997683 1.0000000
```

Robust PP-based PCA: Bus example

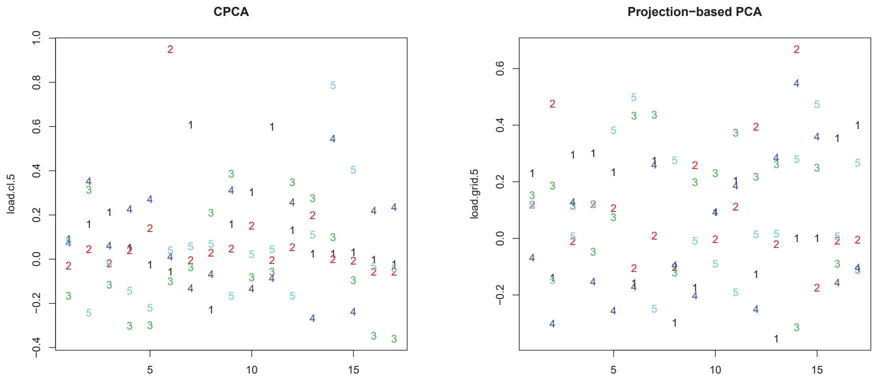
Outlier maps of CPCA and robust PCA (both with $k = 5$):



The extreme bad leverage points found by robust PCA are masked as good leverage points by CPCA.

Robust PP-based PCA: Bus example

Comparison of the loadings:



The first CPCA component is highly influenced by the 7th and 11th variable in the data set. The second CPCA component is influenced by the 6th variable in the data set. These three variables all have many outliers.

Spherical robust PCA

1. Classical PCA
2. Outlier detection in PCA
3. Robust PCA based on a robust covariance estimator
4. Robust PCA based on projection pursuit
5. Spherical robust PCA
6. ROBPCA, based on projection pursuit and the MCD
7. Robust PCA for skewed data

Spherical PCA

Introduced by Locantore et al. (1999).

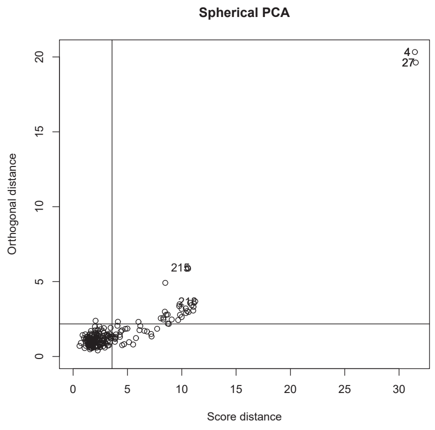
- The data are centered by the L^1-median, denoted as $\hat{\boldsymbol{\mu}}$.
- The data are projected on the unit sphere with center $\hat{\boldsymbol{\mu}}$.
- The robust eigenvectors are computed as the dominant eigenvectors of the covariance matrix of these projected data points, i.e. the eigenvectors of the sign covariance matrix

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})}{\|\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}\|} \frac{(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})'}{\|\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}\|}$$

  with the largest eigenvalues.
- These eigenvalues are not consistent, but they can be replaced by a robust scale^2 of the original data projected on each eigenvector.

Spherical PCA: Bus example

```
> pca.sphere <- PcaLocantore(bus2,k=5)
> plot(pca.sphere,main="Spherical PCA")
```



ROBPCA, based on projection pursuit and the MCD

1. Classical PCA
2. Outlier detection in PCA
3. Robust PCA based on a robust covariance estimator
4. Robust PCA based on projection pursuit
5. Spherical robust PCA
6. ROBPCA, based on projection pursuit and the MCD
7. Robust PCA for skewed data

ROBPCA, based on projection pursuit and the MCD

Main steps (Hubert, Rousseeuw and Vanden Branden, 2005):

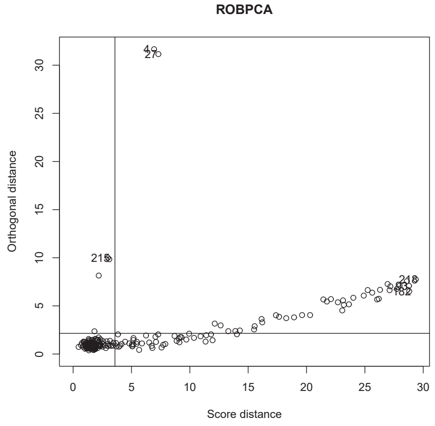
1. Find the $h < n$ ‘least outlying’ data points, with roughly $n/2 < h < n$. For this an orthogonally invariant measure of outlyingness is used, inspired by the Stahel-Donoho outlyingness (SDO):

$$\text{SDO}(\boldsymbol{x}_i) = \max_{\boldsymbol{v} \in B} \frac{|\boldsymbol{x}_i'\boldsymbol{v} - \hat{\mu}_{mcd}(\boldsymbol{x}_j'\boldsymbol{v})|}{\hat{s}_{mcd}(\boldsymbol{x}_j'\boldsymbol{v})}$$

   with $\hat{\mu}_{mcd}$ and $\hat{s}_{mcd}$ the univariate MCD estimators of location and scale. The set B contains 250 directions through two data points, randomly drawn from the data.
2. Set S_h the covariance matrix of the h points with smallest outlyingness. The data are then projected on the k-dimensional subspace spanned by the k dominant eigenvectors of S_h.
3. The location vector and scatter matrix of the projected data are computed with the reweighted MCD estimator. The spectral decomposition of this covariance matrix yields the robust principal components (and eigenvalues).

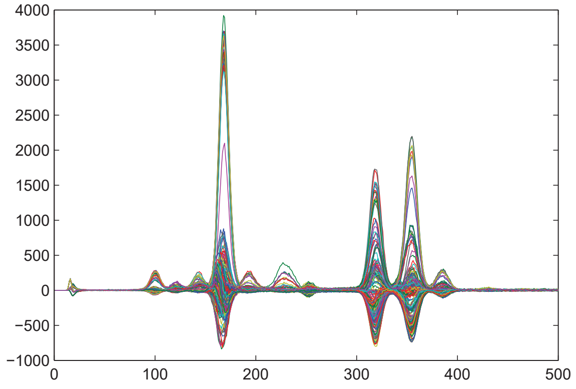
ROBPCA: Bus example

```
> pcaROBPCA <- PcaHubert(bus2, k=5, mcd=FALSE, alpha=0.5)
> plot(pcaROBPCA)
```



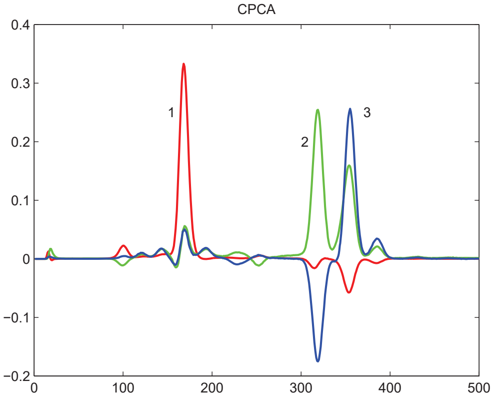
ROBPCA: Glass example

Glass data: $n = 180$ archaeological glass samples (objects) whose spectra have $p = 750$ wavelengths (variables). We only show 500.



ROBPCA: Glass example

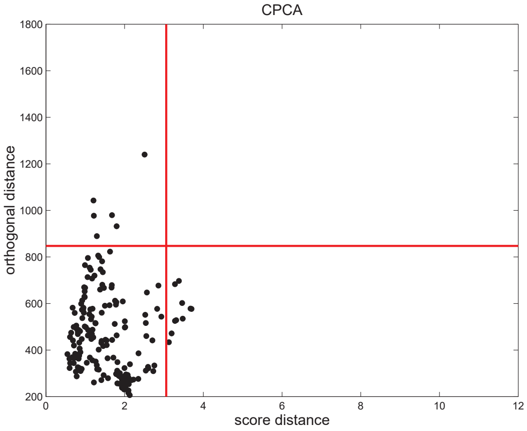
The first 3 basis vectors ("loadings") of classical PCA:



With classical PCA the second and third peaks are mixed up.

ROBPCA: Glass example

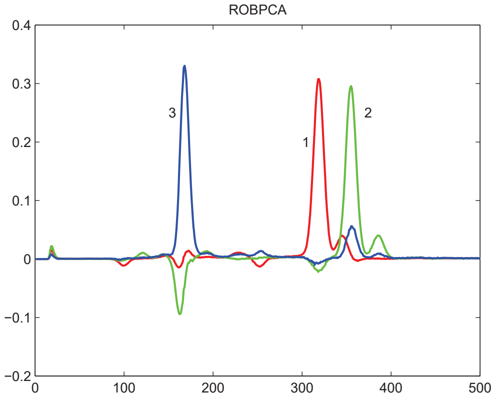
Outlier map from classical PCA:



There would appear to be only mild orthogonal outliers.

ROBPCA: Glass example

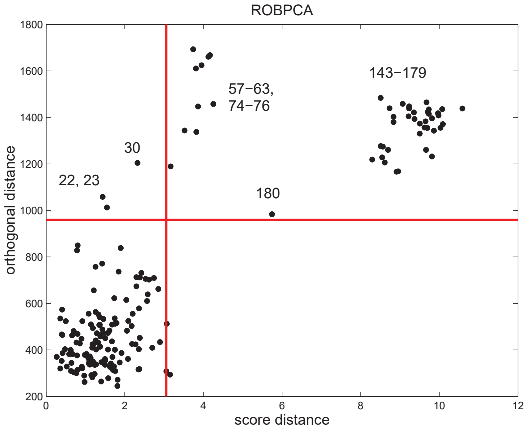
The first 3 basis vectors (“loadings”) of robust PCA:



ROBPCA keeps the peaks more separate.

ROBPCA: Glass example

Outlier map from ROBPCA:



Now we also see bad PCA leverage points!

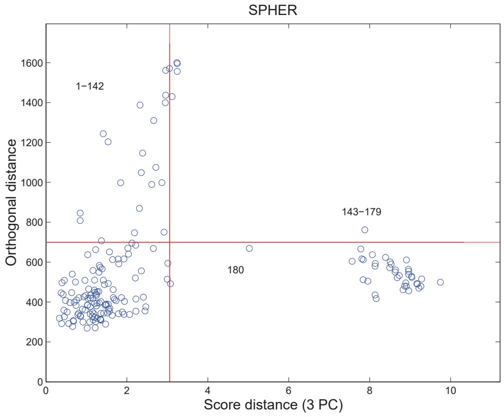
ROBPCA: Glass example

What has caused the outliers in the glass data?

- The window of the detector system was cleaned before the last 38 spectra were measured ⇒ less radiation was absorbed, hence more was detected.
- Observations 57–63 and 74–76 are samples with a large concentration of calcium.
- Observations 22, 23 and 30 are borderline cases (with a larger concentration of phosphor).

ROBPCA: Glass example

Spherical PCA did not find the bad leverage points in this example:



Robust PCA for skewed data

1. Classical PCA
2. Outlier detection in PCA
3. Robust PCA based on a robust covariance estimator
4. Robust PCA based on projection pursuit
5. Spherical robust PCA
6. ROBPCA, based on projection pursuit and the MCD
7. Robust PCA for skewed data

Adjusted outlyingness

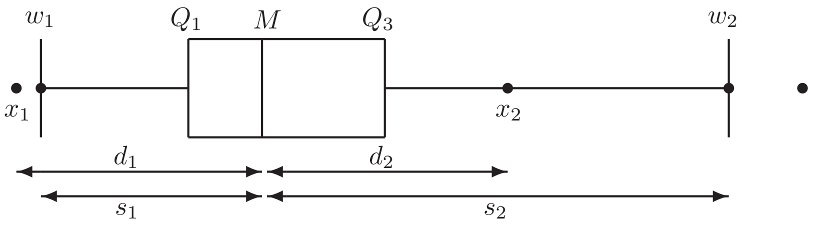
The Stahel-Donoho outlyingness (SDO) assumes symmetry!

Adjusted outlyingness

For univariate data with median M, the adjusted outlyingness is defined as:

$$\text{AO}_i^{(1)} = \text{AO}_i^{(1)}(x_i, X_n) = \frac{|x_i - M|}{(w_2 - M)I[x_i > M] + (M - w_1)I[x_i < M]}$$

with w_1 and w_2 the lower and upper whiskers of the adjusted boxplot.



Adjusted outlyingness

- here $s_1 = M - w_1$ and $s_2 = w_2 - M$.
- $\text{AO}_i^{(1)}(x_1) = d_1/s_1$ and $\text{AO}_i^{(1)}(x_2) = d_2/s_2$.
- Although x_1 and x_2 are at the same distance from the median, x_1 will have a higher adjusted outlyingness because its denominator s_1 is smaller.
- Skewness is thus used to estimate the scale differently on both sides of the median.

For **multivariate data**, the projection pursuit idea can again be used (Brys et al. 2005; Hubert and Van der Veeken 2008):

$$\text{AO}_i = \text{AO}(\boldsymbol{x}_i, X_n) = \sup_{\boldsymbol{a} \in \mathbb{R}^p} \text{AO}^{(1)}(\boldsymbol{a}'\boldsymbol{x}_i, X_n\boldsymbol{a}).$$

In practice: consider $250p$ directions, generated as the direction perpendicular to the subspace spanned by p observations, randomly drawn from the data set.

ROBPCA

Recall the main steps in the ROBPCA method:

- Fix $\frac{n}{2} < h < n$.
- Apply classical PCA on the h data points with smallest SDO_i and retain k components.
- Apply MCD covariance estimator in the subspace: mean and covariance of the h points with smallest robust distance RD_i.
- The outlier map displays the OD versus the SD. Cutoff values for the SD and the OD are based on parametric assumptions.

ROBPCA-AO

ROBPCA for skewed data, based on the adjusted outlyingness:

- Fix $\frac{n}{2} < h < n$.
- Apply classical PCA on the h data points with smallest AO$_i$ and retain k components.
- Apply robust covariance estimator in subspace: mean and covariance matrix of the h points with smallest AO$_i$ (recomputed in the subspace).
- On outlier map: plot AO$_i$ on horizontal axis, and use adjusted boxplot outlier rule for the AO$_i$ and the OD$_i$.

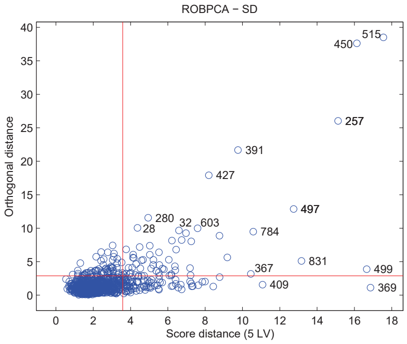
ROBPCA-AO: Example

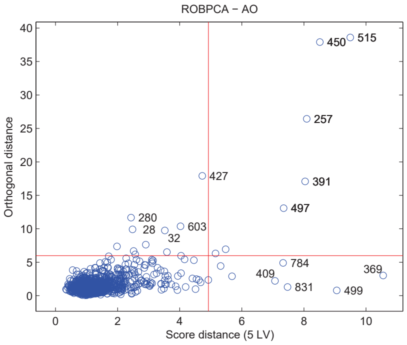
Consumer Expenditure Survey: 869 households and 8 variables collected by U.S. Department of Labor

Variable	Description	MC	p-value
EXP	Total household expenditure	0.21	< 0.00001
FDHO	Food consumed at home	0.17	< 0.00001
FDAW	Food consumed away from home	0.32	< 0.00001
SHEL	Housing and household equipment	0.22	< 0.00001
TELE	Telephone services	0.33	< 0.00001
CLOT	Clothing	0.27	< 0.00001
HEAL	Health care	0.24	< 0.00001
ENT	Entertainment	0.37	< 0.00001

ROBPCA-AO: Consumer Expenditure Survey

- We retained 5 components (88% explained variance)
- ROBPCA (198 observations flagged as outlier)





- ROBPCA-AO flags only 24 observations.