Multivariate Median

Yangming Li

Abstract

The robust estimation of multivariate location is crucial in many application areas. This report reviews the definition for three multivariate median. Concept of multivariate sign and rank are considered. Properties like robustness, affine equivalence are discussed. Simulation studies on computation and outlier detection are carried out.

Keywords: multivariate median; robust; affine equivalence; outlier detection

1. Introduction

Let $\boldsymbol{X_1},\ldots,\boldsymbol{X_n}$ denotes a sample of p dimensional random variables from a distribution F, the multivariate median $\boldsymbol{T}(\boldsymbol{X})$ is a minimization of the criterion function $\mathrm{D}(\boldsymbol{t})$ with respect to $\boldsymbol{t}$. Marginal median is a minimizer for the sum of Manhattan distance, spatial median is a point to which sum of samples' Euclidean norm is smallest. In $\mathbb{R}^p$, for Oja median, the sum of the volume of all p-variate simplix is minimum. Such simplex is determined by the Oja median and any p observations. In summary, for the multivariate median $\boldsymbol{T}$:

$$\boldsymbol{T}_{marginal}(\boldsymbol{X}) = \min_{\boldsymbol{t}} D_1(\boldsymbol{t}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{p}\{|x_{ik} - t_k| - |x_{ik}|\},$$

$$\boldsymbol{T}_{spatial}(\boldsymbol{X}) = \min_{\boldsymbol{t}} D_2(\boldsymbol{t}) = \frac{1}{n}\sum_{i=1}^{n}\{\| \boldsymbol{X}_i - \boldsymbol{t} \| - \| \boldsymbol{X}_i \|\},$$

$$\boldsymbol{T}_{oja}(\boldsymbol{X}) = \min_{\boldsymbol{t}} D_3(\boldsymbol{t}) = \binom{n}{p}^{-1} \sum \frac{1}{p!}\left|\det\begin{pmatrix} 1 & \cdots & 1 & 1 \\ \boldsymbol{X}_{i_1} & \cdots & \boldsymbol{X}_{i_p} & \boldsymbol{t}\end{pmatrix}\right|,$$

where $1 \le i_1 <\ldots< i_p \le n$.

As an extension of median, multivariate median can be found based on three L_1 objective functions. This report will firstly review the concept of sign and rank with corresponding tests in Sect. 2, then discuss the robustness and affine equivalence in Sect 3 and 4. In Sect 5, we will do simulation studies on the algorithms for approximation. As an application, scatter matrices that based on spatial and Oja sign are compared to perform robust outlier detection in Sect 6. The report ends with a short conclusion on these multivariate median in Sect 7.

2. Sign Test and Rank Statistics

Sign and rank are defined in different forms for marginal median, spatial median and Oja median. However, their definitions all come from optimizing the L_1 objective functions and their statistical models coincide. We know that, a convex function is optimized where its derivative equal to 0. Which is true the same for three median. The rank function where $\boldsymbol{R}(\boldsymbol{t}) = \boldsymbol{0}$ both evaluate $\nabla_{\boldsymbol{t}}\mathrm{D}(\boldsymbol{t}) = \boldsymbol{0}$, and is the sign test statistics for $\boldsymbol{H}_0\colon \boldsymbol{T}(\boldsymbol{X}) = \boldsymbol{t}$. For univariate case, a sign function $S(x) = sign(x)$ is $1, 0, -1$ as $x > 0, = 0, < 0$. Marginal sign and spatial sign follow this idea and are defined as:

$$\boldsymbol{msgn}(\boldsymbol{t}) = \boldsymbol{S}_1(\boldsymbol{t}) = \begin{pmatrix} S(t_1) \\ \cdots \\ S(t_p) \end{pmatrix}, \quad \boldsymbol{ssgn}(\boldsymbol{t}) = \boldsymbol{S}_2(\boldsymbol{t}) = \begin{cases} \frac{\boldsymbol{t}}{\|\boldsymbol{t}\|}, if\ \boldsymbol{t} \neq \boldsymbol{0} \\ \boldsymbol{0},\ if\ \boldsymbol{t} = \boldsymbol{0} \end{cases}.$$

The sign function is robust against extreme value, since it is a nonparametric, relative measure of direction. The marginal sign is just the vector of univariate sign, indicating relative location on each dimension. While the spatial sign projects the observations onto a multidimensional sphere. For median $T(F) = t$ being the symmetry center of a distribution, it should solve the rank function $\boldsymbol{R}(\boldsymbol{t}) = ave[\boldsymbol{S}(\boldsymbol{x} - \boldsymbol{t})] = 0$. Since the sign function is the component-wise gradient of the objective function, i.e., $\frac{d}{dx}|x| = S(x)$. The rank that averages sign over independent samples, is the gradient of the objective function. For marginal rank and spatial rank:

$$\boldsymbol{mrnk}_{\mathbb{X}}(\boldsymbol{t}) = \boldsymbol{R}_1(\boldsymbol{t}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{S}_1(\boldsymbol{t} - \boldsymbol{X}_i) = \begin{pmatrix} ave\{S(t_{i1} - x_{i1})\} \\ \cdots \\ ave\{S(t_{ip} - x_{ip})\} \end{pmatrix}$$

$$\boldsymbol{srnk}_{\mathbb{X}}(\boldsymbol{t}) = \mathbf{R}_2(\boldsymbol{t}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{S}_2(\boldsymbol{t} - \boldsymbol{X}_i) = \frac{1}{n}\sum_{i=1}^{n} \frac{\boldsymbol{t} - \boldsymbol{X}_i}{\|\boldsymbol{t} - \boldsymbol{X}_i\|}$$

We can validate that the marginal and spatial rank function to be the gradient function of $D_1(\boldsymbol{t})$ and

$D_2(\boldsymbol{t})$. Hence, for Oja median, same definitions are presented.

We first give the definition of a hyperplane in $\mathbb{R}^p$ for our geometrical understanding:

$$\left\{\boldsymbol{x} \in \mathbb{R}^p : \det\begin{pmatrix} 1 & \cdots & 1 & 1 \\ \boldsymbol{x}_{i_1} & \cdots & \boldsymbol{x}_{i_p} & \boldsymbol{x} \end{pmatrix}\right\}$$

Such $\boldsymbol{x}$ forms a hyperplane I that determined by p points $\{\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_p}\}$ in $\mathbb{R}^p$. Thus, to indicate on which side of hyperplane I, an $\boldsymbol{x}$ is located, we write:

$$S_I(\boldsymbol{x}) = sign\left\{\det\begin{pmatrix} 1 & \cdots & 1 & 1 \\ \boldsymbol{x}_{i_1} & \cdots & \boldsymbol{x}_{i_p} & \boldsymbol{x} \end{pmatrix}\right\} = \nabla_{\boldsymbol{x}}\left|\det\begin{pmatrix} 1 & \cdots & 1 & 1 \\ \boldsymbol{x}_{i_1} & \cdots & \boldsymbol{x}_{i_p} & \boldsymbol{x} \end{pmatrix}\right|$$

Divided the part being taken absolute value by $p!$, we have the volume of the p-variate simplex formed by $p+1$ vertices $\{\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_p}, \boldsymbol{x}\}$ in $\mathbb{R}^p$. And $D_3(\boldsymbol{x})$ is a sum over such values with respect to every hyperplane in $\mathbb{R}^p$. Then the Oja rank is the average of such $S_I(\boldsymbol{x})$. Let $\mathcal{P} = \{p = (i_1, \ldots, i_p) | 1 \le i_1 < \cdots < i_p \le n\}$ to collect such hyperplane. From n observations, $\mathcal{P}$ has $N_{n,p} = \binom{n}{p}$ elements. Thus:

$$\boldsymbol{ornk}_{\mathbb{X}}(\boldsymbol{t}) = \mathbf{R}_3(\boldsymbol{t}) = \frac{1}{N_{n,p}} \sum_{(i_1,\ldots,i_p)\in P_{n,p}} \nabla_{\boldsymbol{t}}\left|\det\begin{pmatrix} 1 & \cdots & 1 & 1 \\ \boldsymbol{x}_{i_1} & \cdots & \boldsymbol{x}_{i_p} & \boldsymbol{t} \end{pmatrix}\right|$$

We can think that for a point $\boldsymbol{t}$ with zero Oja rank, it will be on one side of half of the hyperplane that n samples can form in $\mathbb{R}^p$, and on the other side of the rest. In the meantime, $D_3(\boldsymbol{t})$ is minimized here since the gradient is zero. To illustrate this nonparametric test, we visualize three families on 10 samples from $\boldsymbol{\mathcal{N}}(\mathbf{0}, \boldsymbol{I}_2)$ in Fig.1.

(a) (b) (c)

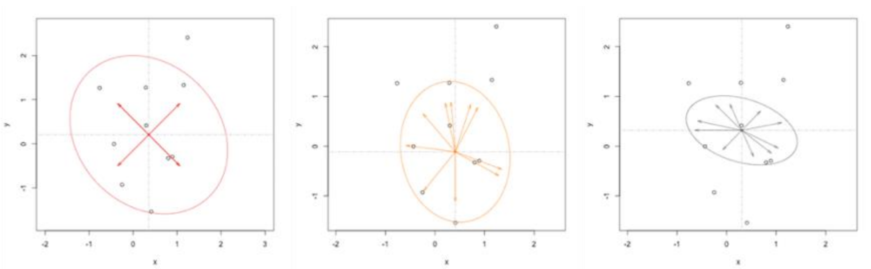


Fig.1 Plots of multivariate median, sign vectors and rank covariance matrices with critical value $\chi^2_{2,0.8}$: (a) Marginal median, (b) Spatial median, (c) Oja median.

By calculating sign and rank, the magnitude information from sample is neglected and our samples would be transformed to a distribution on the surface of a p-dimensional sphere which centered at the estimated median. Intuitively, the distribution of sign $S(\boldsymbol{X} - T(\boldsymbol{X}))$ gives information about the orientation of data. Therefore, the sign covariance matrix, that calculate sample covariance for signs of centered observations is given, which will be used for several applications:

$$\boldsymbol{SCM} = \boldsymbol{Cov}\left(\boldsymbol{S}(\boldsymbol{X} - \boldsymbol{T}(\boldsymbol{X}))\right)$$

3. Robustness

This section explains how robustness of SCM characterize outlier's influence on estimation.

3. 1. Influence function of sign covariance matrix

The condition number (shape) for matrix Σ: $Cond(\Sigma) = \frac{\lambda_{max}}{\lambda_{min}}$, is the ratio of largest and smallest eigenvalue of Σ. We quantify the influence of extreme value by tracking the shape of SCM. The influence function is define as: $\mathcal{IF}\ (\boldsymbol{x}; \Sigma, \mathrm{F}) = \frac{Cond(\tilde{\Sigma})}{Cond(\Sigma)}$, where $\tilde{\Sigma}$ is the SCM of perturbed distribution. If the estimation was on a perfect symmetric sample and one outlier was included, dramatic change in shape represents estimation movement. If the estimator was robust against one outlier, this ratio should not keep changing. Thus, we use 10 samples from $\boldsymbol{\mathcal{N}}(\boldsymbol{0},\ \boldsymbol{I}_3)$, change last sample's x axis value from -500 to 500 and update the condition number ratio. Fig.2 says that Oja median has breakdown point 0, since its ratio keeps increasing. And marginal and spatial median are robust estimators.

3. 2. Influence function on higher dimension

This part shows the robust property of marginal and spatial median by increasing dimension of samples. We also sample 10 observations from $\boldsymbol{\mathcal{N}}(\boldsymbol{0},\ \Sigma\ )$, and get the influence curve following the same method as before. Here we change Σ from $\boldsymbol{I}_2$ to $\boldsymbol{I}_{20}$ step by step. Fig.2 shows that, marginal median and spatial median are still robust. But when the number of dimension grows close to the number of samples, the ratio of condition number may suddenly grow large for spatial median. While for marginal median, this ratio varies with number of dimension.

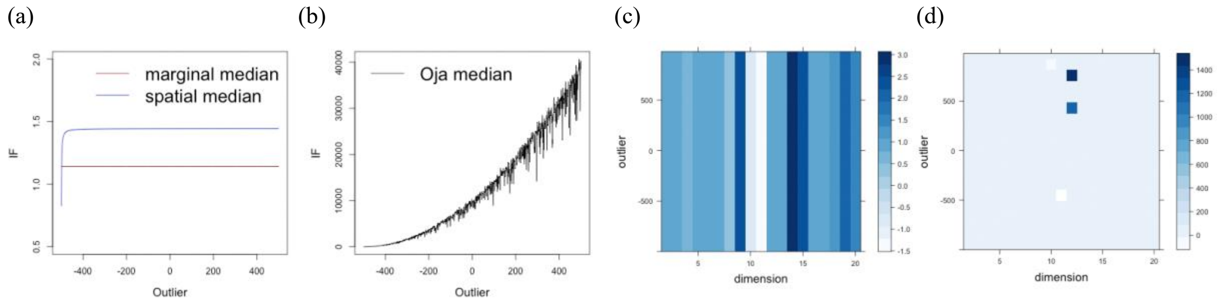


Fig.2: Visualization of influence function: (a) influence curves of ratio for spatial median and marginal median are stable, (b) the shape of Oja's sign covariance matrix keeps changing as outlier moves. To study influence function on different dimension, we have heat-map of ratio against number of samples' dimension for (c) marginal median and for (d) spatial median.

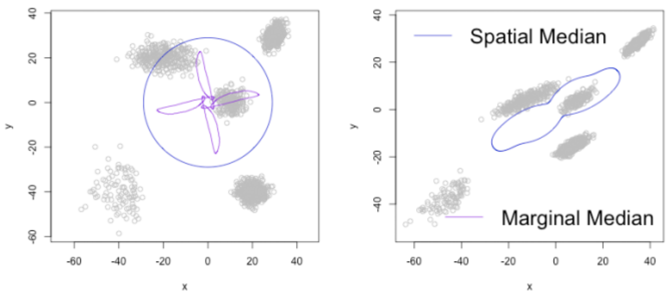
4. Affine Equivalence

Apart from breakdown point, being equivalent under affine transformations is crucial for multivariate estimators, which will be illustrated in Sect 6. In $\mathbb{R}^p$, if an estimator $\boldsymbol{T}$ satisfied that $\boldsymbol{T}(F_{\boldsymbol{Ax+b}}) = \boldsymbol{A}T(F_{\boldsymbol{x}}) + \boldsymbol{b}$, for all full-rank $p \times p$ matrices $\boldsymbol{A}$ and p-vectors $\boldsymbol{b}$, $\boldsymbol{T}$ is affine equivalence.

Marginal median is estimated with respect to each dimension, which is sensitive to component choices. While univariate median is equivalent under scale, only when $\boldsymbol{A}$ is a diagonal matrix with non-zero diagonal elements, marginal median is invariant; Spatial median relies on Euclidean distance, which is preserved under orthogonal transformation. But spatial median may be not invariant under arbitrary transformation of coordinate system. Fig.3 is a visualization for these properties. Oja median is affine equivariant, see Oja H (2013), whose rank function is affine equivariant:

$$\boldsymbol{ornk}_{\mathbb{X}\boldsymbol{A}^T}(\boldsymbol{At}) = \det(\boldsymbol{A})\,(\boldsymbol{A}^{-1})^T \boldsymbol{ornk}_{\mathbb{X}}(\boldsymbol{t})$$

(a) (b)



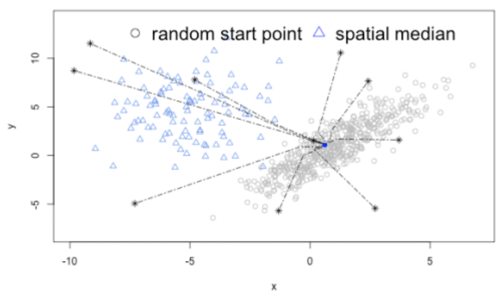


Fig.3: Effects of affine transformation on marginal median and spatial median : (a) locus of marginal and spatial median during rotating a spatial dataset, (b) locus of spatial median during rotating the dataset which has been transformed by a rand matrix.

Fig.4: Simulation of Weiszfeld's algorithm. 10 randomly picked starting points converge at a same point: (0.613, 1.073). The estimation from Gmedian package is (0.803, 0.996)

5. Computation Method

In R programming, the computing marginal median is straightforward. Packages exist for multivariate median and related statistics, like "Gmedian" and "SpatialNP" for spatial median, "OjaNP" for Oja median. This section explains one algorithm for each.

5. 1. Spatial Median

Weiszfeld's algorithm studied by Vardi and Zhang (2000) uses iterative procedure to approximate the minimum of a strictly convex function $D_2(\boldsymbol{t})$. It defines a set of weights that are inversely proportional to the distance from the current estimation to the samples. Then it updates estimation to be the weighted average of the sample according to these weights:

Algorithm I: Computing spatial median via Weiszfeld's algorithm

Step (i) : let vector of mean to be the initial point $\boldsymbol{\mu}^{(0)}$

Step (ii): Iterate k times, in t^{th} iteration:

$$\boldsymbol{\mu}^{(t+1)} = \sum_{i=1}^{n} \boldsymbol{w}_i^{(t)} \boldsymbol{X}_i \,/ \sum_{i=1}^{n} \boldsymbol{w}_i^{(t)},\ \ \boldsymbol{w}_i^{(t)} = \| \boldsymbol{X}_i - \boldsymbol{\mu}^{(t)} \|;$$

If $\boldsymbol{w}_i^{(t)} = 0$, we set $\boldsymbol{\mu}^{(t+1)}$ to any point other than samples

For interpretation, we write $\boldsymbol{srnk}_{\mathbb{X}}(\boldsymbol{\mu}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\boldsymbol{\mu}-\boldsymbol{X}_i}{\|\boldsymbol{\mu}-\boldsymbol{X}_i\|} = 0\ (*)$, so $D_2(\boldsymbol{t})$ is minimized at $\boldsymbol{t} = \boldsymbol{\mu}$ with $\frac{dD_2(\boldsymbol{t})}{dt}|_{\boldsymbol{\mu}} = 0$. While $(*)$ yields an iteration method that $\mu = \left(\sum_{i=1}^{n} \frac{\boldsymbol{X}_i}{\|\boldsymbol{X}_i-\boldsymbol{\mu}\|}\right) / \left(\sum_{i=1}^{n} \frac{1}{\|\boldsymbol{X}_i-\boldsymbol{\mu}\|}\right)$.

Fig. 4 is our simulation result to approximate the spatial median of a contanminated dataset with 500 observations from $\boldsymbol{\mathcal{N}}\left(\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}3 & 4\\5 & 7\end{bmatrix}\right)$ and 100 from $\boldsymbol{\mathcal{N}}\left(\begin{bmatrix}-5\\5\end{bmatrix}, \begin{bmatrix}3 & 0\\0 & 7\end{bmatrix}\right)$. We iterate 5000 times and get similar estimation with one from Gmedian package.

5.2. Oja Median

Oja median can also be viewed as an extension from univariate median. Since one dimensional simplex is the length, Oja median minimize the absolute distance to all samples in $\mathbb{R}^1$. But unlike marginal median, Oja median may not be unique in high dimension. However, the set of Oja median is convex in high dimension, see Ronkainen T and Oja H (2003). Based on the convex property and sign test, the grid-based algorithm can approximate Oja median iteratively:

Algorithm II: Computing Oja median via Grid-based Algorithm

Step (i): Set initial knot distance h, chose tolerance ε and significance level α.

Step (ii): Create a grid G that covers the whole dataset using h.

Step (iii): Randomly choose a set of hyperplanes and remove knots that bellows the significance level of sign test with respect to these hyperplanes.

Step (iv): Repeat Step (iii) until any additional step removes all knots.

Step (v): Build a new grid around left knots using $h/2$. If $h/2 > \varepsilon$: Step (ii)

6. Outlier Detection based on Sign Covariance Matrices

The multivariate median we have discussed so far not only provide a robust location estimation. This section compares robust scatter estimation based on spatial and Oja sign covariance matrix, in terms of detecting outliers from high dimensional samples.

6.1. Methodology

The sample covariance matrix $\boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{X}_i - \overline{\boldsymbol{X}})(\boldsymbol{X}_i - \overline{\boldsymbol{X}})^T$ is sensitive to outliers, where the assumption of multivariate normal fails. Consider eigenvalue decomposition of a covariance matrix $\Sigma = U\Lambda U^T$, a better way to construct Σ depends on robust estimators for U and Λ repectively.

Observations only contribute unit vector to the sign covariance matrix (SCM), SCM well preserves the orientation information of original distribution. By setting the eigenvectors of SCM to be U, new scatter matrix's orientation also gains resistance to outliers. Since the diagonal elements of Λ are the variance along the directions of eigenvectors, we can find Λ through robust scale estimation of $\boldsymbol{X}$'s marginal variance on each direction, see Visuri S and Koivunen V (2000).

Algorithm III: Computing scatter matrix based on sign covariance matrix

Step (i): Find eigenvectors $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p]$ of sign covariance matrix.

Step (ii): Use MAD to find standard deviation of $\boldsymbol{X}$ on $\boldsymbol{u}_i$'s direction. On each direction, $\hat{\boldsymbol{\lambda}}_i = MAD(\boldsymbol{u}_i^T \boldsymbol{X}_1, \ldots, \boldsymbol{u}_i^T \boldsymbol{X}_n)$, then $\widehat{\boldsymbol{\Lambda}} = diag(\boldsymbol{\lambda}_1^2, \ldots, \boldsymbol{\lambda}_p^2)$.

Step (iii): Calculate scatter matrix $\boldsymbol{\Sigma} = \boldsymbol{U}\widehat{\boldsymbol{\Lambda}}\boldsymbol{U}^T$.

Therefore, outlier detection in high dimensional data can be implemented. Based on this estimation of scatter matrix $\boldsymbol{\Sigma}$, and robust location estimation $\boldsymbol{\mu}$, we have robust statistical distance (Mahalanobis distance) $d(\boldsymbol{X}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{(\boldsymbol{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{X} - \boldsymbol{\mu})}$. Compare it to the statistical distance obtained from sample covariance matrix and vector of means, the outliers are those process large difference between these two values.

6.2. A two dimensional case

We first use a simple example to show the importance of robust scatter matrix. We generate 25 uncontaminated samples from $\boldsymbol{\mathcal{N}}\left(\boldsymbol{0}, \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}\right)$. For two contaminated samples, we include 1 and 5 observations from $\boldsymbol{\mathcal{N}}\left(\begin{bmatrix} 15 \\ -10 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}\right)$ respectively. To visualize the effects of outliers, the estimated tolerance ellipses ($\chi^2_{2,0.95}$) based on the sample covariance matrix $\boldsymbol{S}$, the scatter matrix $\boldsymbol{\Sigma}$ based on spatial sign and $\boldsymbol{\Sigma}$ based on Oja sign are plotted in Fig. 5. The centers of ellipses are located at mean vector, spatial median and Oja median. We also give ellipse with theoretical center and covariance.

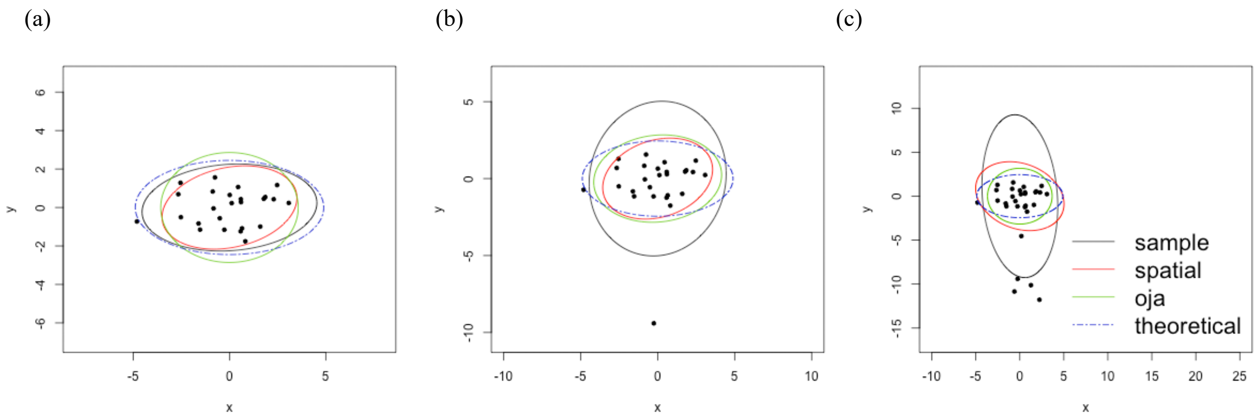


Fig.5: The 95% tolerance ellipses based on the sample covariance matrix, the scatter matrix based on spatial and Oja median, and the theoretical covariance in case of (a) original data (n=25), (b) 1 outlier included, (c) 5 outliers included.

When the data is uncorrupted, we have four similar ellipses. One outlier is enough to rotate and stretch $\boldsymbol{S}$. While spatial scatter matrix performs the best, both its shape and orientation stay similar with theoretical. The Oja scatter matrix is changing in a same trend with $\boldsymbol{S}$, but the bias is much smaller than $\boldsymbol{S}$.

6.3. Outlier detection on "woodmod.dat" (5 dimensions)

When dimensions $p > 3$, we cannot use tolerance ellipsoid to identify outliers. The 5 dimensional dataset "woodmod.dat" from "robust" package is known to have 4 outliers, which can be validated using built-in method. We follow the outlier detection method discussed above, using spatial and Oja sign covariance matrix. Fig.6 shows that, outlier detection that through spatial sign covariance matrix fails, while Oja method successfully identify the outliers.

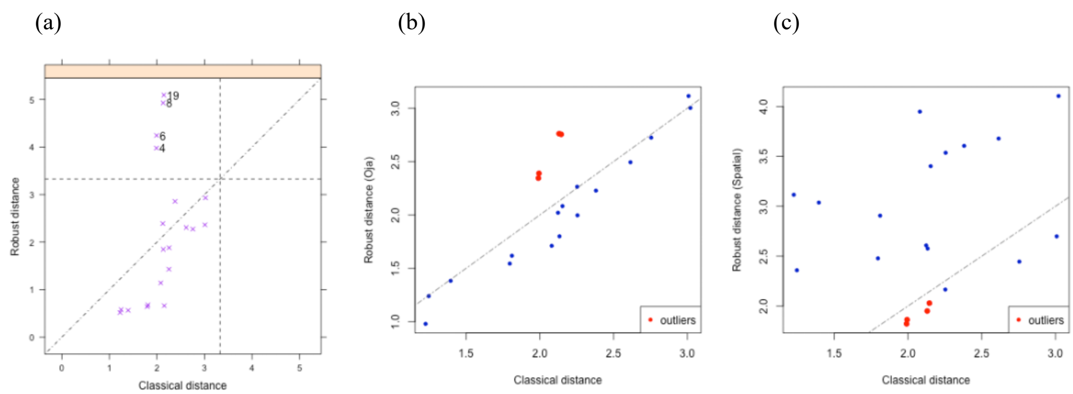


Fig.6: Distance-distance plot of woodmod dataset. Compare to the classical distance using sample mean and sample covariance matrix, the robust distance comes from: (a) covRob function in "robust" package, (b) scatter matrix based on Oja sign covariance matrix, (c) scatter matrix based on spatial sign covariance matrix. Highlighted points in (b) and (c) are true outliers identified in (a).

7. Other Applications

Various kinds of statistical applications rely on different properties of multivariate median. Bruce (1997) used the stochastic expansion of the spatial median for the bootstrap methods to identify extremes. Nealainen (2007) showed spatial median is useful for clustering. Welk (2015) used Oja median to approximate PDE in bivariate case. Visuri (2000) pointed out that scatter matrix based on sign will be useful for PCA, multivariate regression, canonical correlation analysis, factor analysis and discriminant analysis.

8. Conclusion

For multivariate median, marginal median and spatial median are robust estimator with breakdown point 1/2, see Oja H (2013). However, they are not affine equivariant. Oja median has 0 breakdown point, but processes affine equivariant property. Robustness and affine equivariant are all crucial for multivariate studies as our case shows.

9. Reference

[1] Brown B M, Hall P, Young G A (1997) On the effect of inliers on the spatial median. Journal of Multivariate Analysis 63: 88-104.

[2] Brown B. M (1983) Statistical uses of the spatial median. Journal of the Royal Statistical Society 45.1: 25-30

[3] Campbell A (1980) Robust procedures in multivariate analysis: robust covariance estimation. Journal of the Royal Statistical Society 29:231-237

[4] Durre A, Vogel D, Tyler E (2014) The spatial sign covariance matrix with unknown location. Journal of Multivariate Analysis 130:107-117

[5] Hettmansperger P, Mottonen J, Oja H (1999) The geometry of the affine invariant multivariate sign and rank methods. Journal of Nonparametric Statistics 11:271-285

[6] Nevalainen J, Larocque D, Oja H (2007) On the multivariate spatial median for cluster data. The Canadian Journal of Statistics 35:215-231

[7] Oja H (2013) Multivariate median. In: Becker C., Fried R., Kuhnt S. (eds) Robustness and Complex Data Structures. Springer, Berlin, Heidelberg.

[8] Rousseeuw P, Hubert M (2013) High-breakdown estimators of multivariate location and scatter. In: Becker C., Fried R., Kuhnt S. (eds) Robustness and Complex Data Structures. Springer, Berlin, Heidelberg.

[9] Vardi Y, Zhang CH (2000) The multivariate L1-median and associated data depth. Proceedings of the National Academy of Sciences 97:1423-1426

[10] Visuri S, Koivunen V, Oja H (2000) Sign and rank covariance matrices. Journal of Statistical Planning and Inference 91:557-575

[11] Visuri S, Ollila E, Koivunen V (2003) Affine equivariant multivariate rank methods. Journal of Statistical Planning and Inference 114:161-185

[12] Welk M (2015) Partial Differential Equations of Bivariate Median Filters. International Conference on Scale Space and Variational Methods in Computer Vision 53-65.

Appendix (R code)

```
################################################################
# In same order as our plots
#----------------------------------------------
library("MASS")
library("Gmedian")
library("OjaNP")
library("RColorBrewer")
library("robust")
library("SpatialNP")
library("lattice")
#----------------------------------------------
# self implemented function
ellipse <- function(mu, sigma, c2, output=FALSE) {
  # mu: vector
  # sigma: matrix
  # c2 <- qchisq(p,df)
  # output: print the ellipse if TRUE
  # return dot series of ellipse

  es <- eigen(sigma)
  c1 <- sqrt(c2)
  e1 <- es$vec%*%diag(sqrt(es$val))
  theta <- seq(0,2*pi,len=1000)
  v1 <- cbind(c1*cos(theta),c1*sin(theta))
  pts=t(mu+(e1%*%t(v1)))
  # graph
  if(output==TRUE){plot(pts, type="l",xlab="x", ylab="y")}
  return(pts)
}
#----------------------------------------------
# marginal median
mmed <- function(data){
  # data[,i]: all data from ith dimension
  # di: number of dimensions
  di <- dim(sample)[2]
  result <- rep(0,di)
  for (i in 1:di){ result[i] <- median(data[,i]) }
  return(result)
}
# marginal sign
marsign <- function(mm, sample){
  n <- length(mm)
  sign <- rep(0,n)
  for (i in 1:n){
    if (sample[i] < mm[i]){sign[i]<- (-1)}
    if (sample[i] == mm[i]){sign[i]<- 0}
    if (sample[i] > mm[i]){sign[i]<- 1}
  }
  return(sign)
}
# marginal sign covariance matrix
MSCM <- function(mm, sample){
  # mm: marginal median
```

```
    # sample : original sample
    n <- length(mm)
    scm <- matrix(0, nrow=n, ncol=n)
    for (i in 1:dim(sample)[1]) {
        tmp <- marsign(mm,sample[i,])
        scm <- scm + tmp %*% t(tmp)
    }
    return(scm/(dim(sample)[1]))
}
# calculate covariance matrix based on marginal sign
MSCM_covariance <- function(sample){
    m_med <- mmed(sample) # center: spatial median
    scm <- MSCM(m_med, sample) # spatial sign covariance matrix
    U <- eigen(scm)$vector # step 1
    # step 2
    trans <- cbind(rep(0,10),rep(0,10))
    for (i in 1:10){
        trans[i,] <- t(U%*%sample[i,])
    }
    # marginal variance
    lambda <- matrix(c(mad(trans[,1]),0,0,mad(trans[,2])),2,2)
    return(t(U)%*%lambda%*%U)
}
#-----------------------------------------------
# spatial sign covariance matrix
# calculate spatial median: Gmedian(sample)
SSCM <- function(mm, sample){
    # mm: spatial median
    # sample: original sample
    scm <- matrix(c(0,0,0,0),2,2)
    for (i in 1:dim(sample)[1]){
        tmp <- sample[i,]-mm # direction
        long <-sqrt((tmp[1])^2+(tmp[2])^2) # length
        tmp <- tmp/long # unit vector
        scmi <- t(tmp)%*%tmp
        scm <- scm + scmi
    }
    return(scm/(dim(sample)[1]))
}
# calculate covariance matrix based on spatial/oja sign
SCM_covariance <- function(sample,type){
    if (type=="OJA") {scm <- ojaSCM(sample)}
    if (type=="Spatial") {scm <- SCov(sample)}
    U <- eigen(scm)$vector # step 1
    # step 2
    trans <- matrix(0,ncol=dim(sample)[1], nrow=dim(sample)[2])
    for (i in 1:dim(sample)[2]){ # loop every eigenvector
        ui <- U[,i]
        trans[i,] <- ui%*%t(sample)
    }
    # marginal variance
    lambda <- rep(0, dim(sample)[2])
    for (j in 1:dim(sample)[2]) {lambda[j] <- mad(trans[j,])^2}
    # diagonal matrix
    H <- matrix(0, ncol=dim(sample)[2], nrow=dim(sample)[2])
    for (k in 1:dim(sample)[2]) {H[k,k] <- lambda[k]}

    return(U%*%H%*%t(U)) # scatter matrix
```

```
}
#----------------------------------------------
# OJA
OSCM <- function(sample){
  ojaSCM(sample)
}
#----------------------------------------------
# draw sign vector
drawsign <- function(mm, sample, m="marginal", color){
  if (m=="oja") {sign <- ojaSign(sample)}
  for (i in 1:dim(sample)[1]){
    samplei <- sample[i,]
    if (m=="spatial") {direction <- (samplei-mm)/sqrt((mm[1] - samplei[1])^2+(mm[2] - samplei[2])^2)}
    if (m=="marginal") {direction <- marsign(mm, samplei)/sqrt(2)}
    if (m=="oja") {direction <- sign[i,]}
    end <- direction + mm
    arrows(mm[1],mm[2],end[1],end[2],length = 0.1, angle = 15,col=color)
  }
}
#################################################
# Figure 1: draw sign and sign covariance
set.seed(0)
sample <- mvrnorm(10, c(0,0), matrix(c(1,0,0,1),2,2))
c2 <- qchisq(0.8,2)

# marginal
mar <- mmed(sample)
color = "brown2"
png(filename="Marginal_Median_and_Sign.png")
plot(sample, xlab="x", ylab="y", asp=1,xlim=c(-2,3))
points(mar[1],mar[2],pch=20, col=color)
abline(h=mar[2],lty=6,col="grey")
abline(v=mar[1],lty=6,col="grey")
drawsign(mar, sample,"marginal",color)
sigma <- MSCM(mar, sample)
mu <- c(mar[1],mar[2])
ellipse_mar <- ellipse(mu, sigma, c2)
lines(ellipse_mar, col=color, lty=1)
dev.off()

# spatial
color = "darkorange"
l1 <- Gmedian(sample)
png(filename="Spatial_Median_and_Sign.png")
plot(sample, xlab="x", ylab="y", asp=1)
points(l1[1],l1[2],pch=20, col=color)
abline(h=l1[2],lty=6,col="grey")
abline(v=l1[1],lty=6,col="grey")
drawsign(l1, sample, "spatial", col=color)
sigma <- SSCM(l1,sample)
mu <- c(l1[1],l1[2])
ellipse_sp <- ellipse(mu, sigma, c2)
lines(ellipse_sp, col=color, lty=1)
dev.off()

# OJA
color = "dimgray"
oja <- ojaMedian(sample)
```

```
png(filename="OJA_Median_and_Sign.png")
plot(sample, xlab="x", ylab="y", asp=1)
points(oja[1],oja[2],pch=20, col=color)
abline(h=oja[2],lty=6,col="grey")
abline(v=oja[1],lty=6,col="grey")
drawsign(oja, sample, "oja", color)
mu <- c(oja[1],oja[2])
sigma <- OSCM(sample)
ellipse_oja <- ellipse(mu, sigma, c2)
lines(ellipse_oja, col=color, lty=1)
dev.off()
################################################
# Figure 2
# condition number (shape)
cond <- function(matrix){
  value <- eigen(matrix)$values
  return(max(value)/min(value))
}
# change dimension
set.seed(321)
SSCM_ratio <- matrix(0, nrow = 19, ncol = 19)
MSCM_ratio <- matrix(0, nrow = 19, ncol = 19)
for (i in 2:20){
  print(i)
  sigma <- diag(x=1,i,i)
  mu <- rep(0, i)
  sample <- rmvnorm(n = 10, mean = mu, sigma = sigma)
  x <- seq(-1000,1000,110)
  SSCM_cond <- cond(SCov(sample))
  m_med <- mmed(sample)
  MSCM_cond <- cond(MSCM(m_med,sample))
  SSCM_IF <- rep(0, length(x))
  MSCM_IF <- rep(0, length(x))
  for (j in 1:length(x)){
    tmp <- sample[10,]
    tmp[1] <- j
    samplei <- rbind(sample[1:9,], tmp)
    m_medi <- mmed(samplei)
    SSCM_IF[j] <- cond(SCov(samplei))/SSCM_cond
    MSCM_IF[j] <- cond(MSCM(m_medi,samplei))/MSCM_cond
  }
  SSCM_ratio[i-1,] <- SSCM_IF
  MSCM_ratio[i-1,] <- MSCM_IF
}

ratio <- colorRampPalette(brewer.pal(9, "Blues"))(100)
png(filename="SSCM_ratio.png",width = 400, height = 400, units = "px")
levelplot(SSCM_ratio, xlab = list(label="dimension",cex=1.3), ylab = list(label="outlier",cex=1.3),col.regions=ratio,
row.values=seq(2,20,1), column.values = seq(-1000,1000,110),asp=1)
dev.off()
png(filename="MSCM_ratio.png",width = 400, height = 400, units = "px")
levelplot(MSCM_ratio, xlab = list(label="dimension",cex=1.3), ylab = list(label="outlier",cex=1.3),col.regions=ratio,
row.values=seq(2,20,1), column.values = seq(-1000,1000,110),asp=1)
dev.off()

# generate a symmetric sample
set.seed(0)
mu <- c(0,0,0)
```

```
sigma <- matrix(c(1,0,0,0,1,0,0,0,1),3,3)
sample <- rmvnorm(n = 10, mean = mu, sigma = sigma)
x <- seq(-500,500)

OSCM_cond <- cond(ojaSCM(sample))
SSCM_cond <- cond(SCov(sample))
m_med <- mmed(sample)
MSCM_cond <- cond(MSCM(m_med,sample))
OSCM_IF <- rep(0, length(x))
SSCM_IF <- rep(0, length(x))
MSCM_IF <- rep(0, length(x))
for (i in 1:length(x)){
  print(i)
  tmp <- sample[10,]
  tmp[1] <- i
  samplei <- rbind(sample[1:9,], tmp)
  m_medi <- mmed(samplei)
  OSCM_IF[i] <- cond(ojaSCM(samplei))/OSCM_cond
  SSCM_IF[i] <- cond(SCov(samplei))/SSCM_cond
  MSCM_IF[i] <- cond(MSCM(m_medi,samplei))/MSCM_cond
}
png(filename="OJA_IF.png",width = 400, height = 400, units = "px")
plot(x,OSCM_IF, xlim=c(-500,500),type="l",xlab="Outlier",ylab="IF",cex.lab=1.3)
legend("topleft", legend=c("Oja median"),bty="n",lty=1,cex=2)
dev.off()
png(filename="mar_spa_IF.png",width = 400, height = 400, units = "px")
plot(x,SSCM_IF, xlim=c(-500,500),type="l",ylim=c(0.5,2),xlab="Outlier",ylab="IF",col="blue3",cex.lab=1.3)
lines(x, MSCM_IF,col="darkred")
legend("topright", legend=c("marginal median", "spatial median"),col=c("darkred","blue3"), bty="n", lty=1, ncol=1, cex=2)
dev.off()
###################################################
# Figure 3: Affine
set.seed(0)
mu <- rbind(c(-40,-40), c(10,0), c(-20,20),c(30,30),c(20,-40))
Sigma <-
rbind(matrix(c(40,0,0,70),2,2),matrix(c(10,0,0,10),2,2),matrix(c(50,0,0,10),2,2),matrix(c(5,3,2,8),2,2),matrix(c(10,0,0,8),2,2))
pop <- c(100,200,300,400,500)
sample <- cbind(rep(0, sum(pop)),rep(0, sum(pop)))
flag <- 1
for (i in 1:5){
  mui <- mu[i,]
  sig <- Sigma[(2*i-1):(2*i),]
  p <- pop[i]
  sample[flag:(flag+p-1),]<-mvrnorm(p, mui, sig)
  flag <- flag + p
  print(flag)
}

l1 <- Gmedian(sample)
mm <- mmed(sample)

theta <- seq(0,2*pi,len=1000)
t.spatial.rotate <- matrix(nrow = length(theta), ncol = 2)
t.marginal.rotate <- matrix(nrow = length(theta), ncol = 2)
for (i in 1:length(theta)){
  angle <- theta[i]
  A <- rbind(cbind(cos(angle),-sin(angle)),cbind(sin(angle),cos(angle)))
  t.spatial.rotate[i,] <- Gmedian(sample %*% A)
```

```
    t.marginal.rotate[i,] <- mmed(sample %*% A)
    # m <- test %*% A
    # t.oja.rotate[i,] <- ojaMedian(m, alg="exact")
}
color=c("blue3", "blueviolet")
png(filename="rotate.png",width = 400, height = 400, units = "px")
plot(sample,asp=1,col="grey",ylab="y",xlab="x")
lines(t.spatial.rotate,col=color[1])
lines(t.marginal.rotate, col=color[2])
dev.off()

library("pracma")
set.seed(0)
A.rand <- rand(2,2)
for (i in 1:length(theta)){
    angle <- theta[i]
    A <- rbind(cbind(cos(angle),-sin(angle)),cbind(sin(angle),cos(angle)))
    t.spatial.rotate[i,] <- Gmedian(sample %*% A %*% A.rand)
}
test <- sample %*% A.rand
png(filename="affine.png",width = 400, height = 400, units = "px")
plot(test,asp=1,col="grey",ylab="y",xlab="x")
lines(t.spatial.rotate, col=color[1])
legend("topleft", legend=c("Spatial Median"), lty=c(1),col=color[1], bty="n", ncol=1, cex=2)
legend("bottomright", legend=c("Marginal Median"), lty=c(1),col=color[2], bty="n", ncol=1, cex=2)
dev.off()
################################################
# Weiszfeld's algorithm in Fig.4
set.seed(0)
test <- mvrnorm(500, c(1,1), matrix(c(3,4,5,7),2,2))
test <- rbind(test,mvrnorm(100, c(-5,5), matrix(c(3,0,0,7),2,2)))
png(filename="Weiszfeld.png",width = 600, height = 400, units = "px")
plot(test[501:600,],col="cornflowerblue",xlab="x", ylab="y",pch=2,xlim=c(-10,7),ylim=c(-8,14))
points(test[1:500,],col="grey")
legend("topright", legend=c("random start point", "spatial median"),col=c("black","blue"), bty="n", ncol=2, cex=1.8,pch=c(1,2,8,20))
for (k in 1:10){
    u <- matrix(nrow = 5000, ncol = 2)
    u[1,] <- c(runif(1,-10,6),runif(1,-6,12))
    points(u[1,1], u[1,2],pch=8)
    for (i in 2:5000){ # in one iteration
        nu <-c(0,0)
        dn <-0
        for (j in 1:600){
            di <-u[i-1,]-test[j,]
            wi <- sqrt((di[1])^2+(di[2])^2)
            dn <- dn + (1/wi)
            nu <- nu + test[j,]/wi
        }
        ui <- nu/dn
        u[i,]<-ui
    }
    lines(u, lty=6)
    points(u[5000,1], u[5000,2],pch=20,col="blue")
}
G <- Gmedian(test)
# title(sub=paste("estimated spatial median:(",round(u[5000,1],digits = 3),",",round(u[5000,2],digits = 3),")", ",result from Gmedian funciton:(",round(G[1],digits = 3),",",round(G[2],digits = 3),")"), cex=1.8)
dev.off()
```

```
################################################
# Figure 5 outlier in two dimension
# Simulation
set.seed(0)
# generate 10 sample and 5 outlier
sample <- mvrnorm(25, c(0,0), matrix(c(4,0,0,1),2,2))
out <- mvrnorm(5, c(0,-10), matrix(c(1,0,0,5),2,2))
# with 1 outlier
out_1 <- rbind(sample, out[1,])
# with 5 outlier
out_5 <- rbind(sample, out)

# estimation
c2 <- qchisq(0.95,2) # confidence region
# threotical
scatter <- ellipse(c(0,0),matrix(c(4,0,0,1),2,2),c2)
# no outlier
S <- SCM_covariance(sample,"Spatial") # covariance based on spatial sign
O <- SCM_covariance(sample,"OJA")
cov_0 <- ellipse(c(0,0),cov(sample),c2)
s_0 <- ellipse(c(0,0),S,c2)
o_0 <- ellipse(c(0,0),O,c2)
# 1 outlier
S_1 <- SCM_covariance(out_1,"Spatial")
O_1 <- SCM_covariance(out_1,"OJA")
cov_1 <- ellipse(c(0,0),cov(out_1),c2)
s_1 <- ellipse(c(0,0),S_1,c2)
o_1 <- ellipse(c(0,0),O_1,c2)
# 5 outlier
S_5 <- SCM_covariance(out_5,"Spatial")
O_5 <- SCM_covariance(out_5,"OJA")
cov_5 <- ellipse(c(0,0),cov(out_5),c2)
s_5 <- ellipse(c(0,0),S_5,c2)
o_5 <- ellipse(c(0,0),O_5,c2)
################################################
# visualization and save png
png(filename="scm1.png",width = 400, height = 400, units = "px")
color = c("black","red","chartreuse3","blue3")
plot(sample, asp=1, xlim=c(-8,8), pch=20, xlab="x", ylab="y")
lines(cov_0)
lines(s_0, lty=6, col=color[2])
lines(o_0, lty=6, col=color[3])
lines(scatter, lty=6, col=color[4])
dev.off()

png(filename="scm2.png",width = 400, height = 400, units = "px")
plot(out_1, asp=1, xlim=c(-10,10), ylim=c(-10,6), pch=20, xlab="x", ylab="y")
lines(cov_1)
lines(s_1, lty=6, col=color[2])
lines(o_1, lty=6, col=color[3])
lines(scatter, lty=6, col=color[4])
dev.off()

png(filename="scm3.png",width = 400, height = 400, units = "px")
plot(out_5, asp=1, xlim=c(-10,25), ylim=c(-15,12), pch=20, xlab="x", ylab="y")
lines(cov_5)
lines(s_5, lty=6, col=color[2])
lines(o_5, lty=6, col=color[3])
```

```
lines(scatter, lty=6, col=color[4])
legend("bottomright", legend=c("sample","spatial","oja","theoretical"),col=color, bty="n",lty=1,ncol=1,cex=1.8)
dev.off()
##################################################
# # Figure 6
# Use statistical distance to detect outlier
# based on spatial & Oja
data("woodmod.dat")
outlier <- c(4, 6, 8, 19)
XX <- as.matrix(woodmod.dat) # 5 Variables
mean <- c(mean(XX[,1]),mean(XX[,2]),mean(XX[,3]),mean(XX[,4]),mean(XX[,5]))
CovS <- cov(XX)
cov_inverse <- solve(CovS)

l1 <- as.vector(Gmedian(XX))
S_spatial <- SCM_covariance(XX,"Spatial")
S_inverse_spatial <- solve(S_spatial)

d_classical <- rep(0,20)
d_robust <- rep(0,20)

for(i in 1:20){
  d_classical[i] <- sqrt(t(XX[i,]-mean)%*%cov_inverse%*%(XX[i,]-mean))
  d_robust[i] <- sqrt(t(XX[i,]-l1)%*%S_inverse_spatial%*%(XX[i,]-l1))
}

png(filename="outlier1.png",width = 400, height = 400, units = "px")
plot(d_classical, d_robust, xlab="Classical distance", ylab="Robust distance (Spatial)", pch=20, col="blue3")
points(d_classical[outlier], d_robust[outlier], pch=19, col="red")
lines(0:7,0:7, lty=6, col="dimgray")
legend("bottomright", legend=c("outliers"),col=c("red"),pch=20,ncol=1)
dev.off()

oja <- as.vector(ojaMedian(XX))
S_oja <- SCM_covariance(XX,"OJA")
S_inverse_oja <- solve(S_oja)

d_classical <- rep(0,20)
d_robust <- rep(0,20)
for(i in 1:20){
  d_classical[i] <- sqrt(t(XX[i,]-mean)%*%cov_inverse%*%(XX[i,]-mean))
  d_robust[i] <- sqrt(t(XX[i,]-oja)%*%S_inverse_oja%*%(XX[i,]-oja))
}

png(filename="outlier2.png",width = 400, height = 400, units = "px")
plot(d_classical, d_robust, xlab="Classical distance", ylab="Robust distance (Oja)", pch=20, col="blue3")
points(d_classical[outlier], d_robust[outlier], pch=19, col="red")
lines(0:7,0:7, lty=6, col="dimgray")
legend("bottomright", legend=c("outliers"),col=c("red"),pch=20,ncol=1)
dev.off()

png(filename="outlier3.png",width = 400, height = 400, units = "px")
woodm.fm=fit.models(list(Robust="covRob", Classical="covClassic"),
                    data=woodmod.dat)
ddPlot.covfm(woodm.fm, pch=4, col="purple", xlab="Classical distance",
             ylab="Robust distance", id.n=5)
dev.off()
##################################################################
```