Package ‘DetMCD’

May 19, 2018

Type Package

Title Implementation of the DetMCD Algorithm (Robust and Deterministic Estimation of Location and Scatter)

Version 0.0.5

Date 2018-05-13

Depends robustbase, pcaPP

Suggests mvtnorm, MASS

LinkingTo Rcpp, RcppEigen

Description Implementation of DetMCD, a new algorithm for robust and deterministic estimation of location and scatter. The benefits of robust and deterministic estimation are explained in Hubert, Rousseeuw and Verdonck (2012) <doi:10.1080/10618600.2012.672100>.

License GPL (>= 2)

LazyLoad yes

Maintainer Vakili Kaveh <vakili.kaveh.email@gmail.com>

Author Vakili Kaveh [aut, cre],
Mia Hubert [ths]

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-05-19 11:49:57 UTC

R topics documented:

DetMCD-package 2
DetMCD 2
DetMCD_CS 5
DetMCD_RW 6
DetMCD_SP 7
inQn 7
plot.DetMCD 8
quanff 10
xtractR_M 11

Index **12**

DetMCD-package *Deterministic and Robust Algorithms for Data Analysis.*

Description

This packages contains various robust and deterministic algorithms for data analysis.

Details

Package:	DetMCD
Type:	Package
Version:	0.0.1
Date:	2012-09-19
Depends:	matrixStats, pcaPP (>= 1.8-1), robustbase, MASS
License:	GPL (>= 2)
LazyLoad:	yes

Index:

```
DetMCD                  Robust and deterministic estimation of location
                        and scatter.
DetMCD-package           Robust and Deterministic Algothms for Data
                        Analysis
plot.DetMCD          Diagnostic plots for DetMCD
DetMCD_CS               Internal function for DetMCD
DetMCD_RW               Internal function for DetMCD
DetMCD_SP               Internal function for DetMCD
xtractR_M               Internal function for DetMCD
quanff                  Internal function for DetMCD
```

Author(s)

Kaveh Vakili [aut, cre],

Maintainer: Kaveh Vakili <vakili.kaveh.email@gmail.com>

DetMCD *Robust and Deterministic Location and Scatter Estimation via DetMCD*

Description

Computes a robust and deterministic multivariate location and scatter estimate with a high breakdown point, using the `DetMCD` (Deterministic Minimum Covariance Determinant) algorithm.

Usage

```
DetMCD(X,h=NULL,alpha=0.75,scale_est="Auto",tol=1e-07)
```

Arguments

`X` a numeric matrix or data frame. Missing values (NaN's) and infinite values (Inf's) are allowed: observations (rows) with missing or infinite values will automatically be excluded from the computations.

`alpha` Ignored if `h!=NULL`. (Possibly vector of) numeric parameter controlling the size of the subsets over which the determinant is minimized, i.e., `alpha*n` observations are used for computing the determinant. Allowed values are between 0.5 and 1 and the default is 0.75.

`h` numeric integer parameter controlling the size of the subsets over which the determinant is minimized, i.e., `h` observations are used for computing the determinant. Allowed values are between `[(n+p+1)/2]` and `n` and the default is NULL.

`scale_est` a character string specifying the variance functional. Possible values are "qn", "tau" and 'Auto". Default value `"Auto"` is to use the Qn estimator for data with less than 1000 observations, and to use the tau-scale for data sets with more observations. But one can also always use the Qn estimator `"qn"` or the tau scale `"tau"`.

`tol` a small positive numeric value to be used for determining numerical 0.

Details

DetMCD computes the MCD estimator of a multivariate data set in a deterministic way. This estimator is given by the subset of h observations with smallest covariance determinant. The MCD location estimate is then the mean of those h points, and the MCD scatter estimate is their covariance matrix. The default value of h is roughly 0.75n (where n is the total number of observations), but the user may choose each value between n/2 and n. Based on the raw estimates, weights are assigned to the observations such that outliers get zero weight. The reweighted MCD estimator is then given by the mean and covariance matrix of the cases with non-zero weight.

To compute the MCD estimator, six initial robust h-subsets are constructed based on robust transformations of variables or robust and fast-to-compute estimators of multivariate location and shape. Then C-steps are applied on these h-subsets until convergence. Note that the resulting algorithm is not fully affine equivariant, but it is often faster than the FAST-MCD algorithm which is affine equivariant. Note that this function can not handle exact fit situations: if the raw covariance matrix is singular, the program is stopped. In that case, it is recommended to apply the FastMCD function.

The MCD method is intended for continuous variables, and assumes that the number of observations n is at least 5 times the number of variables p. If p is too large relative to n, it would be better to first reduce p by variable selection or robust principal components (see the functions PcaHubert).

Value

A list with components:

`raw.center` The raw MCD location of the data.

`raw.cov` The raw MCD covariance matrix (multiplied by a consistency factor).

`crit` The determinant of the raw MCD covariance matrix.

`raw.rd` The robust distance of each observation to the raw MCD center, relative to the raw MCD scatter estimate.

`raw.wt` Weights based on the estimated raw covariance matrix 'raw.cov' and the estimated raw location 'raw.center' of the data. These weights determine which observations are used to compute the final MCD estimates.

`center` The robust location of the data, obtained after reweighting.

`cov` The robust covariance matrix, obtained after reweighting.

`h` The number of observations that have determined the MCD estimator, i.e. the value of h.

`which.one` The identifier of the initial shape estimate which led to the optimal result.

`best` The subset of h points whose covariance matrix has minimal determinant.

`weights` The finale vector of weights.

`rd` The robust distance of each observation to the final, reweighted MCD center of the data, relative to the reweighted MCD scatter of the data. These distances allow us to easily identify the outliers.

`rew.md` The Mahalanobis distance of each observation (distance from the classical center of the data, relative to the classical shape of the data).

`X` Same as the X in the call to DetMCD, without rows containing missing or infinite values.

`alpha` The vector of values of alpha used in the algorithm.

`scale_est` The vector of scale estimators used in the estimates (one of `tau2` or `qn`.

Author(s)

Vakili Kaveh (includes section of the help file from the LIBRA implementation).

References

Hubert, M., Rousseeuw, P.J. and Verdonck, T. (2012), "A deterministic algorithm for robust location and scatter", Journal of Computational and Graphical Statistics, Volume 21, Number 3, Pages 618–637.

Verboven, S., Hubert, M. (2010). Matlab library LIBRA, Wiley Interdisciplinary Reviews: Computational Statistics, 2, 509–515.

Examples

```
## generate data
set.seed(1234)  # for reproducibility
alpha<-0.5
n<-101
p<-5
#generate correlated data
D<-diag(rchisq(p,df=1))
```

```
W<-matrix(0.9,p,p)
diag(W)<-1
W<-D
V<-chol(W)
x<-matrix(rnorm(n*p),nc=p)
x<-scale(x)


result<-DetMCD(x,scale_est="tau",alpha=alpha)
plot(result, which = "dd")

#compare to robustbase:
result<-DetMCD(x,scale_est="qn",alpha=alpha)
resultsRR<-covMcd(x,nsamp='deterministic',scalefn=qn,alpha=alpha)
#should be the same:
result$crit
resultsRR$crit


#Example with several values of alpha:
alphas<-seq(0.5,1,l=6)
results<-DetMCD(x,scale_est="qn",alpha=alphas)
plot(results, h.val = 2, which = "dd")
```

DetMCD_CS *DetMCD_CS*

Description

Internal function. Computes the Csteps for the DetMCD algorithm.

Usage

```
DetMCD_CS(Data,scale_est,h,out1)
```

Arguments

`Data` a numeric matrix or data frame without missing values.

`scale_est` a character string specifying the variance functional. Possible values are "qn" for the Qn or "tau" for the tau scale.

`h` a vector of integers (between n/2 and n).

`out1` A list. Typically the result of a call to DetMCD_SP.

Value

returns a list.

Author(s)

Vakili Kaveh

See Also

`DetMCD`,`DetMCD_SP`.

`DetMCD_RW` *DetMCD_RW*

Description

Internal function. Carries the re-weighting part of the DetMCD algorithm.

Usage

```
DetMCD_RW(ll,hlst,Xw,out2,scale_est,alpha)
```

Arguments

`ll`	integer in `1:6`.
`hlst`	a vector of integers between in (n/2,n).
`Xw`	a n by p data matrix.
`out2`	a list. Typically the result of a call to `"DetMCD_CS"`.
`scale_est`	a character string specifying the variance functional. Possible values are `"qn"` for the Qn or `"tau"` for the tau scale.
`alpha`	a vector of values in [1/2,1].

Value

returns a list.

Author(s)

Vakili Kaveh

See Also

`DetMCD`,`DetMCD_CS`.

`DetMCD_SP` *DetMCD_SP*

Description

Internal function. Computes the starting points for the DetMCD algorithm.

Usage

```
DetMCD_SP(Data,scale_est,tol)
```

Arguments

`Data` a numeric matrix or data frame without missing values.

`scale_est` a character string specifying the variance functional. Possible values are `"qn"` for the Qn or `"tau"` for the tau scale.

`tol` a small positive numeric value to be used for determining numerical 0.

Value

returns a list.

Author(s)

Vakili Kaveh

See Also

`DetMCD`.

`inQn` *Test function for the qn*

Description

Test function for the qn used in DetR.

Usage

```
inQn(x)
```

Arguments

`x` Vector of 2 or more numbers. Should contain no ties.

Value

the value of the qn estimator of scale.

Author(s)

Kaveh Vakili

References

see `pcaPP::qn` and citation("pcaPP").

Examples

```
set.seed(123) #for reproductibility
x<-rnorm(101)
inQn(x)
#should be the same:
pcaPP::qn(x)
```

---

plot.DetMCD *Robust Diagnostic Plots For DetMCD*

Description

Shows the Mahalanobis distances based on robust and classical estimates of the location and the covariance matrix in different plots. The following plots are available:

- index plot of the robust and mahalanobis distances
- distance-distance plot
- Chisquare QQ-plot of the robust and mahalanobis distances
- plot of the tolerance ellipses (robust and classic)
- Scree plot - Eigenvalues comparison plot

This function is a minimally modified adaptation of `"robustbase::covPlot"`. See `citation("robustbase")`.

Usage

```
## S3 method for class 'DetMCD'
plot(x,h.val=1,
     which = c("all", "dd", "distance", "qqchi2",
               "tolEllipsePlot", "screeplot"),
     classic = FALSE, ask = (which == "all" && dev.interactive()),
     cutoff = NULL, id.n, labels.id = rownames(x), cex.id = 0.75,
     label.pos = c(4,2), tol = 1e-07, ...)
```

Arguments

`x` For the `plot()` method, a `DetMCD` object, typically result of `DetMCD`.

`h.val` An integer in `1:length(DetMCD_object$h)` indicating for which of the values of `h` the diagnostic plot should be shown.

`which` string indicating which plot to show. See the *Details* section for a description of the options. Defaults to `"all"`..

`classic` whether to plot the classical distances too. Defaults to `FALSE`..

`ask` logical indicating if the user should be *ask*ed before each plot, see `par(ask=.)`. Defaults to `which == "all" && dev.interactive()`.

`cutoff` the cutoff value for the distances.

`id.n` number of observations to be identified by a label. If not supplied, the number of observations with distance larger than `cutoff` is used.

`labels.id` vector of labels, from which the labels for extreme points will be chosen. `NULL` uses observation numbers.

`cex.id` magnification of point labels.

`label.pos` positioning of labels, for the left half and right half of the graph respectively (used as `text(.., pos=*)`).

`tol` tolerance to be used for computing the inverse, see `solve`. Defaults to `tol = 1e-7`.

`...` Further arguments passed to the plot function.

Details

These functions produce several plots based on the robust and classical location and covariance matrix. Which of them to select is specified by the attribute `which`. The `plot` method for `"mcd"` objects is calling `covPlot()` directly, whereas `covPlot()` should also be useful for plotting other (robust) covariance estimates. The possible options are:

`distance` index plot of the robust distances

`dd` distance-distance plot

`qqchi2` a qq-plot of the robust distances versus the quantiles of the chi-squared distribution

`tolEllipsePlot` a tolerance ellipse plot, via tolEllipsePlot

`screeplot` an eigenvalues comparison plot - screeplot

The Distance-Distance Plot, introduced by Rousseeuw and van Zomeren (1990), displays the robust distances versus the classical Mahalanobis distances. The dashed line is the set of points where the robust distance is equal to the classical distance. The horizontal and vertical lines are drawn at values equal to the cutoff which defaults to square root of the 97.5% quantile of a chi-squared distribution with p degrees of freedom. Points beyond these lines can be considered outliers.

References

P. J. Rousseeuw and van Zomeren, B. C. (1990). Unmasking Multivariate Outliers and Leverage Points. *Journal of the American Statistical Association* **85**, 633–639.

P. J. Rousseeuw and K. van Driessen (1999) A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **41**, 212–223.

See Also

DetMCD

Examples

```
data(Animals, package ="MASS")
brain <- Animals[c(1:24, 26:25, 27:28),]
detmcd <- DetMCD(log(brain))

plot(detmcd, which = "distance", classic = TRUE)# 2 plots
plot(detmcd, which = "dd")
plot(detmcd, which = "tolEllipsePlot", classic = TRUE)
op <- par(mfrow = c(2,3))
plot(detmcd)## -> which = "all" (5 plots)
par(op)
```

---

quanff *quanff*

Description

Internal function. Converts alpha values to h values.

Usage

```
quanff(alpha,n,p)
```

Arguments

alpha	a value in [1/2,1].
n,p	integers.

Value

returns an integer.

Author(s)

Vakili Kaveh

References

Hubert, M., Rousseeuw, P.J. and Verdonck, T. (2012), "A deterministic algorithm for robust location and scatter", Journal of Computational and Graphical Statistics, in press.

Examples

```
quanff(0.75,n=100,p=5);
```

xtractR_M *xtractR_M*

Description

Internal function. Formats the output for the DetMCD algorithm.

Usage

```
xtractR_M(out2,X)
```

Arguments

`out2`	A list. Typically the result of a call to DetMCD_RW.
`X`	a numeric matrix or data frame without missing values.

Value

returns a list.

Author(s)

Vakili Kaveh

See Also

`DetMCD`,`DetMCD_RW`.

Index

∗Topic **deterministic**
DetMCD, 2
DetMCD_CS, 5
DetMCD_RW, 6
DetMCD_SP, 7
quanff, 10
xtractR_M, 11
∗Topic **hplot**
plot.DetMCD, 8
∗Topic **multivariate**
DetMCD, 2
DetMCD_CS, 5
DetMCD_RW, 6
DetMCD_SP, 7
inQn, 7
plot.DetMCD, 8
quanff, 10
xtractR_M, 11
∗Topic **package**
DetMCD-package, 2
∗Topic **robust**
DetMCD, 2
DetMCD_CS, 5
DetMCD_RW, 6
DetMCD_SP, 7
inQn, 7
plot.DetMCD, 8
quanff, 10
xtractR_M, 11

DetMCD, 2, *6*, *7*, *9–11*
DetMCD-package, 2
DetMCD_CS, 5, *6*
DetMCD_RW, 6, *11*
DetMCD_SP, *6*, 7
dev.interactive, *9*

inQn, 7

par, *9*
plot.DetMCD, 8

quanff, 10

solve, *9*

text, *9*
tolEllipsePlot, *9*

xtractR_M, 11