Package: symMCD (via r-universe)

June 2, 2026

Type Package

Title Symmetrized MCD

Version 0.6

Date 2026-03-31

Imports Rcpp (>= 0.11.0)

LinkingTo Rcpp, RcppArmadillo

Description Provides implementations of origin-based and symmetrized minimum covariance determinant (MCD) estimators, together with supporting utility functions.

License GPL (>= 2)

NeedsCompilation yes

Author Klaus Nordhausen [aut, cre] (ORCID: <https://orcid.org/0000-0002-3758-8501>), Christophe Croux [aut] (ORCID: <https://orcid.org/0000-0003-2912-3437>)

Maintainer Klaus Nordhausen <klaus.nordhausenr@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-04-03 17:00:02 UTC

RemoteUrl https://github.com/cran/symMCD

RemoteRef HEAD

RemoteSha 351d4734afd219700fb764c892aa3ad3a6da93c5

Contents

symMCD-package *Symmetrized MCD*

Description

Provides implementations of origin-based and symmetrized minimum covariance determinant (MCD) estimators, together with supporting utility functions.

Details

Package:	symMCD
Type:	Package
Title:	Symmetrized MCD
Version:	0.6
Date:	2026-03-31
Authors@R:	c(person(given = "Klaus", family = "Nordhausen", role = c("aut", "cre"), email = "klaus.nordhausenr@gmail.c
Imports:	Rcpp (>= 0.11.0)
LinkingTo:	Rcpp, RcppArmadillo
Description:	Provides implementations of origin-based and symmetrized minimum covariance determinant (MCD) estimato
License:	GPL (>= 2)
Author:	Klaus Nordhausen [aut, cre] (ORCID: <https://orcid.org/0000-0002-3758-8501>), Christophe Croux [aut] (OR
Maintainer:	Klaus Nordhausen <klaus.nordhausenr@gmail.com>
Archs:	x64

The minimum covariance determinant (MCD) estimator is one of the most popular robust scatter matrix estimators; however, it does not possess the independence property. The symmetrized MCD is a variant of the MCD that does satisfy this property. It is, however, computationally more demanding, as it is based on all pairwise differences of the observations. On the other hand, the symmetrized MCD does not require a location estimate, since it is inherently centered at the origin.

To reduce the computational burden, an approximate variant of the symmetrized MCD is also implemented. This variant does not use all pairwise differences, but instead relies on a subset of size nm, where n denotes the sample size and m is a user-specified integer.

In addition, several helper functions are provided to support the computation and use of the symmetrized MCD.

Index of help topics:

Author(s)

Klaus Nordhausen [aut, cre] (ORCID: <https://orcid.org/0000-0002-3758-8501>), Christophe Croux [aut] (ORCID: <https://orcid.org/0000-0003-2912-3437>)

Maintainer: Klaus Nordhausen <klaus.nordhausenr@gmail.com>

References

Croux, C. and Nordhausen, K. (2026). Symmetrized MCD. *Unpublished manuscript.*

`Maha` *Fast squared Mahalanobis distances*

Description

Computes squared Mahalanobis distances using different choices of location and scatter matrices.

Usage

```
Maha(X, center, scatter)
MahaClassical(X)
MahaOrigin(X)
```

Arguments

`X`	numeric data matrix.
`center`	numeric vector giving the location vector.
`scatter`	numeric matrix giving the scatter matrix.

Details

The function `Maha` allows the user to supply a custom location and scatter matrix in order to compute (pseudo) Mahalanobis distances. This makes it possible to use alternative location and scatter estimates instead of the sample mean vector and covariance matrix. In contrast, the function `MahaClassical` always uses the sample mean and the classical covariance matrix. The function `MahaOrigin` fixes the location at the origin and computes the corresponding covariance matrix with respect to the origin.

The three functions provided here prioritize computational efficiency, but offer fewer options than `mahalanobis`. For example, `mahalanobis` allows the user to provide a pre-inverted scatter matrix.

Value

A numeric vector with the squared distances

See Also

`mahalanobis`

Examples

```
X <- matrix(rnorm(300), ncol = 3)

COV <- cov(X)
MEAN <- colMeans(X)

Maha(X, MEAN, COV)
mahalanobis(X, MEAN, COV)
MahaClassical(X)


MEAN0 <- rep(0, 3)
COV0 <- crossprod(X)/99
MahaOrigin(X)
mahalanobis(X, MEAN0, COV0)
```

`MCDzero` *MCD with respect to the Origin*

Description

Computes the minimum covariance determinant (MCD) estimator with the location fixed at the origin, using either a C++ or a plain R implementation.

Usage

```
MCDzero(X, alpha = 0.5, ns = 500, nc = 10, delta = 0.01)
MCDzeroR(y, alpha = 0.5, ns = 500, nc = 10, delta = 0.01)
```

Arguments

`y`	A numeric data matrix with observations in rows and variables in columns.
`X`	A numeric data matrix with observations in rows and variables in columns.
`alpha`	A numeric value in $(0, 1]$ specifying the fraction of observations to be retained for the MCD computation. Usually in the interval (0.5,1).
`ns`	An integer specifying the number of random initial subsets used by the algorithm.
`nc`	An integer specifying the number of concentration steps performed for each initial subset.
`delta`	A numeric tuning constant used in the reweighting step.

Details

The minimum covariance determinant (MCD) estimator is computed using an iterative algorithm based on random initial subsets, followed by a fixed number of concentration steps. In contrast to the classical MCD, the location is fixed at the origin and is not estimated from the data.

The parameter `alpha` controls the size of the subset retained at each step and thus determines the robustness of the estimator. For each of the `ns` random initial subsets, the algorithm applies `nc` concentration steps to improve the determinant of the scatter estimate. The best solution over all starts is retained.

An optional reweighting step is applied using the tuning constant `delta`, which aims to improve efficiency by downweighting observations with large squared Mahalanobis distances.

Value

For the C++ implementation, a list with the following components:

`SigmaRaw`	The raw MCD scatter matrix with the location fixed at the origin.
`SigmaRe`	The reweighted MCD scatter matrix with the location fixed at the origin.
`IndexOptimal`	An integer vector giving the indices of the observations forming the optimal subset.

For the R implementation, a list with the following components:

`sigma`	The raw MCD scatter matrix with the location fixed at the origin.
`sigmaR`	The reweighted MCD scatter matrix with the location fixed at the origin.

Examples

```
X <- matrix(rnorm(300), ncol=3)
colnames(X) <- LETTERS[1:3]

# settings seeds for comparison
set.seed(1)
mcd0cpp <- MCDzero(X, 0.5)
set.seed(1)
mcd0r <- MCDzeroR(X, 0.5)

mcd0cpp$SigmaRaw
mcd0cpp$SigmaRaw - mcd0r$sigma

mcd0cpp$SigmaRe
mcd0cpp$SigmaRe - mcd0r$sigmaR
```

mdiff *Computes Successive Differences for a Data Set*

Description

Computes a subset of pairwise differences between observations by forming successive differences up to a fixed lag.

Usage

```
mdiff(X, m = 20)
```

Arguments

X	A numeric data matrix with observations in rows and variables in columns.
m	A positive integer specifying the number of successive differences to be computed for each observation.

Details

Let X be a data matrix with n rows, denoted by $x_1, \ldots, x_n$. The function computes a matrix of successive differences consisting of the vectors $x_i - x_{i+j}$, for $j = 1, \ldots, m$.

To ensure that each observation contributes exactly m differences, the first m observations are appended to the end of the data matrix before computing the differences. As a result, the output contains nm difference vectors.

The resulting set of differences forms a subset of all possible pairwise differences and provides a computationally cheaper alternative to using all $n(n-1)/2$ differences. Such constructions are useful, for example, in approximations of symmetrized scatter estimators.

Value

A numeric matrix with nm rows and the same number of columns as X, containing the successive difference vectors.

References

Dumbgen, L. and Nordhausen, K. (2024). *Approximating symmetrized estimators of scatter via balanced incomplete U-statistics*. Annals of the Institute of Statistical Mathematics, **76**, 185–207. <doi:10.1007/s10463-023-00879-1>.

Examples

```
X <- matrix(1:12, ncol=2)
mdiff(X, m=3)
```

symMCD *Symmetrized Minimum Covariance Determinant Estimator*

Description

Computes the symmetrized minimum covariance determinant (MCD) estimator based on pairwise differences of the observations.

Usage

```
symMCD(X, gamma = 0.25, m = NULL, ns = 500, nc = 10, delta = 0.01)
```

Arguments

X	A numeric data matrix with observations in rows and variables in columns.
gamma	A numeric value in $(0, 1]$ specifying the fraction of pairwise differences to be retained for the MCD computation.
m	If NULL, all pairwise differences are used. If a positive integer, only m successive differences per observation are used as an approximation.
ns	An integer specifying the number of random initial subsets used by the algorithm.
nc	An integer specifying the number of concentration steps performed for each initial subset.
delta	A numeric tuning constant used in the reweighting step.

Details

The symmetrized MCD computes a minimum covariance determinant (MCD) estimator based on pairwise differences $x_i - x_j$ with $i > j$. Since these differences are centered by construction, the location is fixed at the origin and no location parameter is estimated or returned. In practice, this corresponds to applying an MCD estimator with respect to the origin to the matrix of pairwise differences; see MCDzero.

In contrast to the classical MCD, the symmetrized MCD possesses the independence property: at the population level, the resulting scatter matrix is diagonal for a random vector with independent components.

The argument gamma controls the fraction of pairwise differences used for the scatter computation. Reasonable values are in the interval $(0.25, 1)$. For a given value of gamma, the exact number of differences to be retained is determined analogously to the implementation in the **robustbase** package, using the function h.alpha.n.

Computing all pairwise differences is computationally expensive. As an approximation, the function therefore allows the use of only m successive differences per observation. In this case, the order of the observations should be random; hence, applying a random permutation to the observations prior to calling symMCD is advisable. While this approximation substantially reduces computational cost, it may be less appropriate when robustness is the primary concern rather than the independence property.

When only a subset of pairwise differences is used, the number of differences employed in the computation is given by $\lfloor nm\gamma \rfloor + 1$, where n denotes the sample size and m the number of successive differences per observation.

Value

A list with the following components:

SigmaRaw	The raw symmetrized MCD scatter matrix.
SigmaRe	The reweighted symmetrized MCD scatter matrix.

References

Croux, C. and Nordhausen, K. (2026). *Symmetrized MCD*. Unpublished manuscript.

Examples

```
# data is small to keep the example fast
X <- matrix(rnorm(150), ncol=3)
colnames(X) <- LETTERS[1:3]

# transformed data to check affine equivariance:
A <- matrix(runif(9),3,3)
Y <- X %*% t(A)

# seed needs to be set for comparison
set.seed(1)
mcdall <- symMCD(X, 0.25)
set.seed(1)
mcdallY <- symMCD(Y, 0.25)

mcdall$SigmaRaw
mcdall$SigmaRe

A %*% mcdall$SigmaRaw %*% t(A) - mcdallY$SigmaRaw
A %*% mcdall$SigmaRe %*% t(A) - mcdallY$SigmaRe



# could do same for incomplete variant

set.seed(123)
mcdm20 <- symMCD(X, 0.25, m = 20)
set.seed(123)
mcdm20Y <- symMCD(Y, 0.25, m = 20)

mcdm20$SigmaRaw
mcdm20$SigmaRe

A %*% mcdm20$SigmaRaw %*% t(A) - mcdm20Y$SigmaRaw
A %*% mcdm20$SigmaRe %*% t(A) - mcdm20Y$SigmaRe
```

Index