Package ‘robustmatrix’

October 17, 2024

Type Package

Title Robust Matrix-Variate Parameter Estimation

Version 0.1.3

Maintainer Marcus Mayrhofer <marcus.mayrhofer@tuwien.ac.at>

Description Robust covariance estimation for matrix-valued data and data with Kronecker-covariance structure using the Matrix Minimum Covariance Determinant (MMCD) estimators and outlier explanation using and Shapley values.

License GPL-3

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo,

Imports Rcpp, stats, Rdpack

Suggests knitr, rmarkdown, roxygen2, gridExtra, dplyr, forcats, ggnewscale, ggplot2, ggrepel, tibble, tidyr

RoxygenNote 7.3.2

Repository CRAN

RdMacros Rdpack

VignetteBuilder knitr

Depends R (>= 4.0.0)

NeedsCompilation yes

Author Marcus Mayrhofer [aut, cre],
Una Radojičić [aut],
Peter Filzmoser [aut]

Date/Publication 2024-10-17 12:10:06 UTC

Contents

`clean_prob_mmcd` *Probability of obtaining at least one clean h-subset in the `mmcd` function.*

Description

Probability of obtaining at least one clean h-subset in the `mmcd` function.

Usage

```
clean_prob_mmcd(p, q, n_subsets = 500, contamination = 0.5)
```

Arguments

`p`	number of rows.
`q`	number of columns.
`n_subsets`	number of elemental h-substs (default is 500).
`contamination`	level of contamination (default is 0.5).

Value

Probability of obtaining at least one clean h-subset in the `mmcd` function.

`cstep` *C-step of Matrix Minimum Covariance Determinant (MMCD) Estimator*

Description

This function is part of the FastMMCD algorithm (Mayrhofer et al. 2024).

Usage

```
cstep(
  X,
  alpha = 0.5,
  h_init = -1L,
  init = TRUE,
  max_iter = 100L,
  max_iter_MLE = 100L,
  lambda = 0,
  adapt_alpha = TRUE
)
```

Arguments

`X`	a 3d array of dimension (p, q, n), containing n matrix-variate samples of p rows and q columns in each slice.
`alpha`	numeric parameter between 0.5 (default) and 1. Controls the size $h \approx alpha * n$ of the h-subset over which the determinant is minimized.
`h_init`	Integer. Size of initial h-subset. If smaller than 0 (default) size is chosen automatically.
`init`	Logical. If TRUE (default) elemental subsets are used to initialize the procedure.
`max_iter`	upper limit of C-step iterations (default is 100)
`max_iter_MLE`	upper limit of MLE iterations (default is 100)
`lambda`	a smooting parameter for the rowwise and columnwise covariance matrices.
`adapt_alpha`	Logical. If TRUE (default) alpha is adapted to take the dimension of the data into account.

Value

A list containing the following:

`mu`	Estimated $p \times q$ mean matrix.
`cov_row`	Estimated p times p rowwise covariance matrix.
`cov_col`	Estimated q times q columnwise covariance matrix.
`cov_row_inv`	Inverse of `cov_row`.
`cov_col_inv`	Inverse of `cov_col`.
`md`	Squared Mahalanobis distances.
`md_raw`	Squared Mahalanobis distances based on *raw* MMCD estimators.
`det`	Value of objective function (determinant of Kronecker product of rowwise and columnwise covariane).
`dets`	Objective values for the final h-subsets.
`h_subset`	Final h-subset of *raw* MMCD estimators.
`iterations`	Number of C-steps.

See Also

mmcd

Examples

```
n = 1000; p = 2; q = 3
mu = matrix(rep(0, p*q), nrow = p, ncol = q)
cov_row = matrix(c(1,0.5,0.5,1), nrow = p, ncol = p)
cov_col = matrix(c(3,2,1,2,3,2,1,2,3), nrow = q, ncol = q)
X <- rmatnorm(n = 1000, mu, cov_row, cov_col)
ind <- sample(1:n, 0.3*n)
X[,,ind] <- rmatnorm(n = length(ind), matrix(rep(10, p*q), nrow = p, ncol = q), cov_row, cov_col)
par_mmle <- mmle(X)
par_cstep <- cstep(X)
distances_mmle <- mmd(X, par_mmle$mu, par_mmle$cov_row, par_mmle$cov_col)
distances_cstep <- mmd(X, par_cstep$mu, par_cstep$cov_row, par_cstep$cov_col)
plot(distances_mmle, distances_cstep)
abline(h = qchisq(0.99, p*q), lty = 2, col = "red")
abline(v = qchisq(0.99, p*q), lty = 2, col = "red")
```

---

darwin — *DARWIN (Diagnosis AlzheimeR WIth haNdwriting)*

Description

The DARWIN (Diagnosis AlzheimeR WIth haNdwriting) dataset comprises handwriting samples from 174 individuals. Among them, 89 have been diagnosed with Alzheimer's disease (AD), while the remaining 85 are considered healthy subjects (H). Each participant completed 25 handwriting tasks on paper, and their pen movements were recorded using a graphic tablet. From the raw handwriting data, a set of 18 features was extracted.

Usage

```
data(darwin)
```

Format

An array of dimension (p, q, n), comprising $n = 174$ observations, each represented by a $p = 18$ times $q = 25$ dimensional matrix. The observed parameters are:

- Total Time
- Air Time
- Paper Time
- Mean Speed on paper
- Mean Acceleration on paper
- Mean Acceleration in air

- Mean Jerk on paper
- Pressure Mean
- Pressure Variance
- Generalization of the Mean Relative Tremor (GMRT) on paper
- GMTR in air
- Mean GMRT
- Pendowns Number
- Max X Extension
- Max Y Extension
- Dispersion Index

Source

UC Irvine Machine Learning Repository - DARWIN - doi:10.24432/C55D0K

References

Cilia ND, De Stefano C, Fontanella F, Di Freca AS (2018). "An experimental protocol to support cognitive impairment diagnosis by using handwriting analysis." *Procedia Computer Science*, **141**, 466–471.
Cilia ND, De Gregorio G, De Stefano C, Fontanella F, Marcelli A, Parziale A (2022). "Diagnosing Alzheimer's disease from on-line handwriting: a novel dataset and performance benchmarking." *Engineering Applications of Artificial Intelligence*, **111**, 104822.

`matrixShapley` *Outlier explanation based on Shapley values for matrix-variate data*

Description

`matrixShapley` decomposes the squared matrix Mahalanobis distance (`mmd`) into additive outlyingness contributions of the rows, columns, or cell of a matrix (Mayrhofer and Filzmoser 2023; Mayrhofer et al. 2024).

Usage

```
matrixShapley(X, mu = NULL, cov_row, cov_col, inverted = FALSE, type = "cell")
```

Arguments

`X`	a 3d array of dimension (p, q, n), containing n matrix-variate samples of p rows and q columns in each slice.
`mu`	a $p \times q$ matrix containing the means.
`cov_row`	a $p \times p$ positive-definite symmetric matrix specifying the rowwise covariance matrix

`cov_col`	a $q \times q$ positive-definite symmetric matrix specifying the columnwise covariance matrix
`inverted`	Logical. FALSE by default. If TRUE `cov_row` and `cov_col` are supposed to contain the inverted rowwise and columnwise covariance matrices, respectively.
`type`	Character. Either "row", "col", or "cell" (default) to compute rowwise, columnwise, or cellwise Shapley values.

Value

Rowwise, columnwise, or cellwise Shapley value(s).

References

Mayrhofer M, Filzmoser P (2023). "Multivariate outlier explanations using Shapley values and Mahalanobis distances." *Econometrics and Statistics*.

Mayrhofer M, Radojičić U, Filzmoser P (2024). "Robust covariance estimation and explainable outlier detection for matrix-valued data." *arXiv preprint arXiv:2403.03975*.

See Also

`mmd`.

Examples

```
n = 1000; p = 2; q = 3
mu = matrix(rep(0, p*q), nrow = p, ncol = q)
cov_row = matrix(c(5,2,2,4), nrow = p, ncol = p)
cov_col = matrix(c(3,2,1,2,3,2,1,2,3), nrow = q, ncol = q)
X <- rmatnorm(n = 1000, mu, cov_row, cov_col)
distances <- mmd(X, mu, cov_row, cov_col)
```

---

mmcd — *The Matrix Minimum Covariance Determinant (MMCD) Estimator*

Description

`mmcd` computes the robust MMCD estimators of location and covariance for matrix-variate data using the FastMMCD algorithm (Mayrhofer et al. 2024).

Usage

```
mmcd(
  X,
  nsamp = 500L,
  alpha = 0.5,
  lambda = 0,
  max_iter_cstep = 100L,
```

```
  max_iter_MLE = 100L,
  max_iter_cstep_init = 2L,
  max_iter_MLE_init = 2L,
  adapt_alpha = TRUE,
  reweight = TRUE,
  scale_consistency = "quant",
  outlier_quant = 0.975,
  nthreads = 1L
)
```

Arguments

`X`	a 3d array of dimension (p, q, n), containing n matrix-variate samples of p rows and q columns in each slice.
`nsamp`	number of initial h-subsets (default is 500).
`alpha`	numeric parameter between 0.5 (default) and 1. Controls the size $h \approx alpha * n$ of the h-subset over which the determinant is minimized.
`lambda`	a smooting parameter for the rowwise and columnwise covariance matrices.
`max_iter_cstep`	upper limit of C-step iterations (default is 100)
`max_iter_MLE`	upper limit of MLE iterations (default is 100)
`max_iter_cstep_init`	upper limit of C-step iterations for initial h-subsets (default is 2)
`max_iter_MLE_init`	upper limit of MLE iterations for initial h-subsets (default is 2)
`adapt_alpha`	Logical. If TRUE (default) alpha is adapted to take the dimension of the data into account.
`reweight`	Logical. If TRUE (default) the reweighted MMCD estimators are computed.
`scale_consistency`	Character. Either "quant" (default) or "mmd_med". If "quant", the consistency factor is chosen to achieve consistency under the matrix normal distribution. If "mmd_med", the consistency factor is chosen based on the Mahalanobis distances of the observations.
`outlier_quant`	numeric parameter between 0 and 1. Chi-square quantile used in the reweighting step.
`nthreads`	Integer. If 1 (default), all computations are carried out sequentially. If larger then 1, C-steps are carried out in parallel using `nthreads` threads. If < 0, all possible threads are used.

Details

The MMCD estimators generalize the well-known Minimum Covariance Determinant (MCD) (Rousseeuw 1985; Rousseeuw and Driessen 1999) to the matrix-variate setting. It looks for the h observations, $h = \alpha * n$, whose covariance matrix has the smallest determinant. The FastMMCD algorithm is used for computation and is described in detail in (Mayrhofer et al. 2024). NOTE: The procedure depends on *random* initial subsets. Currently setting a seed is only possible if `nthreads = 1`.

Value

A list containing the following:

- `mu` — Estimated $p \times q$ mean matrix.
- `cov_row` — Estimated p times p rowwise covariance matrix.
- `cov_col` — Estimated q times q columnwise covariance matrix.
- `cov_row_inv` — Inverse of `cov_row`.
- `cov_col_inv` — Inverse of `cov_col`.
- `md` — Squared Mahalanobis distances.
- `md_raw` — Squared Mahalanobis distances based on *raw* MMCD estimators.
- `det` — Value of objective function (determinant of Kronecker product of rowwise and columnwise covariane).
- `alpha` — The (adjusted) value of alpha used to determine the size of the h-subset.
- `consistency_factors` — Consistency factors for raw and reweighted MMCD estimators.
- `dets` — Objective values for the final h-subsets.
- `best_i` — ID of subset with best objective.
- `h_subset` — Final h-subset of *raw* MMCD estimators.
- `h_subset_reweighted` — Final h-subset of *reweighted* MMCD estimators.
- `iterations` — Number of C-steps.
- `dets_init_first` — Objective values for the `nsamp` initial h-subsets after `max_iter_cstep_init` C-steps.
- `subsets_first` — Subsets created in subsampling procedure for large `n`.
- `dets_init_second` — Objective values of the 10 best initial subsets after executing C-steps until convergence.

References

Mayrhofer M, Radojičić U, Filzmoser P (2024). "Robust covariance estimation and explainable outlier detection for matrix-valued data." *arXiv preprint arXiv:2403.03975*.

Rousseeuw P (1985). "Multivariate Estimation With High Breakdown Point." *Mathematical Statistics and Applications Vol. B*, 283-297. doi:10.1007/9789400954380_20.

Rousseeuw PJ, Driessen KV (1999). "A Fast Algorithm for the Minimum Covariance Determinant Estimator." *Technometrics*, **41**(3), 212-223. doi:10.1080/00401706.1999.10485670.

See Also

The `mmcd` algorithm uses the `cstep` and `mmle` functions.

Examples

```
n = 1000; p = 2; q = 3
mu = matrix(rep(0, p*q), nrow = p, ncol = q)
cov_row = matrix(c(1,0.5,0.5,1), nrow = p, ncol = p)
cov_col = matrix(c(3,2,1,2,3,2,1,2,3), nrow = q, ncol = q)
X <- rmatnorm(n = n, mu, cov_row, cov_col)
ind <- sample(1:n, 0.3*n)
X[,,ind] <- rmatnorm(n = length(ind), matrix(rep(10, p*q), nrow = p, ncol = q), cov_row, cov_col)
par_mmle <- mmle(X)
par_mmcd <- mmcd(X)
distances_mmle <- mmd(X, par_mmle$mu, par_mmle$cov_row, par_mmle$cov_col)
distances_mmcd <- mmd(X, par_mmcd$mu, par_mmcd$cov_row, par_mmcd$cov_col)
plot(distances_mmle, distances_mmcd)
abline(h = qchisq(0.99, p*q), lty = 2, col = "red")
abline(v = qchisq(0.99, p*q), lty = 2, col = "red")
```

---

mmd *Matrix Mahalanobis distance*

Description

Matrix Mahalanobis distance

Usage

```
mmd(X, mu, cov_row, cov_col, inverted = FALSE)
```

Arguments

X	a 3d array of dimension (p, q, n), containing n matrix-variate samples of p rows and q columns in each slice.
mu	a $p \times q$ matrix containing the means.
cov_row	a $p \times p$ positive-definite symmetric matrix specifying the rowwise covariance matrix
cov_col	a $q \times q$ positive-definite symmetric matrix specifying the columnwise covariance matrix
inverted	Logical. FALSE by default. If TRUE cov_row and cov_col are supposed to contain the inverted rowwise and columnwise covariance matrices, respectively.

Value

Squared Mahalanobis distance(s) of observation(s) in X.

Examples

```
n = 1000; p = 2; q = 3
mu = matrix(rep(0, p*q), nrow = p, ncol = q)
cov_row = matrix(c(1,0.5,0.5,1), nrow = p, ncol = p)
cov_col = matrix(c(3,2,1,2,3,2,1,2,3), nrow = q, ncol = q)
X <- rmatnorm(n = 1000, mu, cov_row, cov_col)
ind <- sample(1:n, 0.3*n)
X[,,ind] <- rmatnorm(n = length(ind), matrix(rep(10, p*q), nrow = p, ncol = q), cov_row, cov_col)
distances <- mmd(X, mu, cov_row, cov_col)
plot(distances)
abline(h = qchisq(0.99, p*q), lty = 2, col = "red")
```

---

mmle — *Maximum Likelihood Estimation for Matrix Normal Distribtuion*

Description

`mmle` computes the Maximum Likelihood Estimators (MLEs) for the matrix normal distribution using the iterative flip-flop algorithm (Dutilleul 1999).

Usage

```
mmle(X, max_iter = 100L, lambda = 0, silent = FALSE)
```

Arguments

`X`	a 3d array of dimension (p, q, n), containing n matrix-variate samples of p rows and q columns in each slice.
`max_iter`	upper limit of iterations.
`lambda`	a smooting parameter for the rowwise and columnwise covariance matrices.
`silent`	Logical. If FALSE (default) warnings and errors are printed.

Value

A list containing the following:

`mu`	Estimated $p \times q$ mean matrix.
`cov_row`	Estimated p times p rowwise covariance matrix.
`cov_col`	Estimated q times q columnwise covariance matrix.
`cov_row_inv`	Inverse of `cov_row`.
`cov_col_inv`	Inverse of `cov_col`.
`norm`	Forbenius norm of squared differences between covariance matrices in final iteration.
`iterations`	Number of iterations of the mmle procedure.

References

Dutilleul P (1999). "The mle algorithm for the matrix normal distribution." *Journal of Statistical Computation and Simulation*, **64**(2), 105-123. doi:10.1080/00949659908811970.

See Also

For robust parameter estimation use `mmcd`.

Examples

```
n = 1000; p = 2; q = 3
mu = matrix(rep(0, p*q), nrow = p, ncol = q)
cov_row = matrix(c(1,0.5,0.5,1), nrow = p, ncol = p)
cov_col = matrix(c(3,2,1,2,3,2,1,2,3), nrow = q, ncol = q)
X <- rmatnorm(n = 1000, mu, cov_row, cov_col)
par_mmle <- mmle(X)
```

`n_subsets_mmcd` *Number of subsets that are required to obtain at least one clean h-subset in the* `mmcd` *function with probability* `prob`.

Description

Number of subsets that are required to obtain at least one clean h-subset in the `mmcd` function with probability `prob`.

Usage

```
n_subsets_mmcd(p, q, prob = 0.99, contamination = 0.5)
```

Arguments

`p`	number of rows.
`q`	number of columns.
`prob`	probability (default is 0.99).
`contamination`	level of contamination (default is 0.5).

Value

Number of subsets that are required to obtain at least one clean h-subset in the `mmcd` function with probability `prob`.

`rmatnorm` *Simulate from a Matrix Normal Distribution*

Description

Simulate from a Matrix Normal Distribution

Usage

```
rmatnorm(n, mu = NULL, cov_row, cov_col)
```

Arguments

`n`	the number of samples required.
`mu`	a $p \times q$ matrix containing the means.
`cov_row`	a $p \times p$ positive-definite symmetric matrix specifying the rowwise covariance matrix
`cov_col`	a $q \times q$ positive-definite symmetric matrix specifying the columnwise covariance matrix

Value

If $n = 1$ a matrix with p rows and q columns, o otherwise a 3d array of dimensions (p, q, n) with a sample in each slice.

Examples

```
n = 1000; p = 2; q = 3
mu = matrix(rep(0, p*q), nrow = p, ncol = q)
cov_row = matrix(c(5,2,2,4), nrow = p, ncol = p)
cov_col = matrix(c(3,2,1,2,3,2,1,2,3), nrow = q, ncol = q)
X <- rmatnorm(n = 1000, mu, cov_row, cov_col)
X[,,9] #printing the 9th sample.
```

`weather` *Glacier weather data – Sonnblick observatory*

Description

Weather data from Austria's highest weather station, situated in the Austrian Central Alps on the glaciated mountain "Hoher Sonnblick", standing 3106 meters above sea level.

Usage

```
data(weather)
```

Format

An array of dimension (p, q, n), comprising $n = 136$ observations, each represented by a $p = 5$ times $q = 12$ dimensional matrix. Observed parameters are monthly averages of

- air pressure (AP)
- precipitation (P)
- sunshine hours (SH)
- temperature (T)
- proportion of solid precipitation (SP)

from 1891 to 2022.

Source

Datasource: GeoSphere Austria - https://data.hub.geosphere.at

Index