Package ‘mvdpd’

September 5, 2026

Type Package

Title Robust DPD Methods for Casewise and Cellwise Contamination

LazyLoad yes

Version 0.1-1

Date 2026-08-23

Description Robust multivariate estimation based on multivariate, composite and componentwise Density Power Divergence (DPD) minimization in multivariate normal distribution for casewise and cellwise contamination. Robust estimation for multivariate ordered gamma model using multivariate and composite DPD minimization. See A. Ghosh, C. Agostinelli, and A. Basu (2026) A Composite Divergence Approach to Robust Multivariate Estimation under Cellwise and Casewise Contamination. <doi:10.48550/arXiv.2608.18914> for full details.

License GPL (>= 2)

Depends R (>= 3.5.0)

Imports MASS

Suggests knitr, robustbase, cellWise, dplyr, ggplot2, reshape2, tidyr

SuggestsNote mostly only because of vignette graphics

VignetteBuilder knitr

NeedsCompilation no

Author Claudio Agostinelli [aut, cre],
Abhik Ghosh [aut],
Ayanendranath Basu [aut]

Maintainer Claudio Agostinelli <claudio.agostinelli@unitn.it>

Repository CRAN

Date/Publication 2026-09-05 14:00:10 UTC

Contents

`mvnormDPD` *Robust estimation for multivariate normal model using DPD based methods*

Description

Robust estimation for multivariate normal model using multivariate (ordinary), composite and componentwise Density Power Divergence (DPD) miminization.

Usage

```
mvnormDPD(x, beta = 0.3,
  method = c("multivariate", "composite", "componentwise"),
  initial = "MAD", ...)
```

Arguments

`x`	`matrix`, `data.frame` or `vector`. Data with columns representing variables.
`beta`	scalar. Robustness tuning parameter of the DPD. `beta=0` provides the maximum likelihood estimator for all methods.
`method`	character. `multivariate` provides the ordinary minimum DPD estimator (Basu et al., 1998), `composite` returns the minimum composite DPD estimator based on pairwise likelihood (Ghosh, Agostinelli, and Basu, 2026) and `componentwise` gives the componentwise minimum DPD estimator (Chakraborty, Basu and Ghosh, 2025).
`initial`	User can provide initial values in a list with three components named: mu, sigma, rho for locations, variances and correlation matrix, respectively, otherwise initial values are computed using `mad` as described in Ghosh, Agostinelli, and Basu (2026).
`...`	Further parameters passed to the methods [see Deatils].

Details

Iterative algorithms are used to compute the estimates for all methods if `beta` >`0`. Additional algorithmic control parameters common to all methods are `tol` representing the tolerance limit so that the algorithm is said to converge if the total aboslute errors across all parameters are less than or equal to it (default value 1e-3), and `max.it` denoting the maximum number of iterations to be tried by the algorithm (default 1000).

For `method="multivariate"` and `method="composite"`, additional control parameters are `w.lower`, `w.upper`. Weights smaller than `w.lower` are set to zero while weights larger than `w.upper` are set to one.

For `method="composite"` at the initial step a univariate filter can be applied setting `univariate.filter=TRUE`. In this case a univariate DPD with beta equals to `univariate.beta` (default 0.99) is computed for each column, data are then standardized and values larger than `q` are set to `NA`.

Value

An object of class `mvdpd`, that for this model is a `list` with components

`mu`	estimated location parameter vector
`Sigma`	estimated covariance matrix
`sigma`	vector of estimated variances
`rho`	estimated correlation matrix
`beta`	robustness tuning parameter used by the DPD
`iter`	number of iterations carried out by the algorithm whenever available
`description`	long version of the DPD method used
`method`	the DPD method
`model`	the statistical model
`data`	the input data
`call`	the actual call

Author(s)

A. Ghosh, C. Agostinelli, and A. Basu.

References

A. Basu, I.R. Harris, N.L. Hjort, and M.C. Jones. Robust and efficient estimation by minimizing a density power divergence. *Biometrika*, 85(3), 549-559, 1998.

A. Basu, A. Ghosh, and L. Pardo. *Statistical Inference based on the Density Power Divergence: The Robustness Perspective*. CRC Press, 2025.

S. Chakraborty, A. Basu, and A. Ghosh. A componentwise estimation procedure for multivariate location and scatter: Robustness, efficiency and scalability. *Journal of Multivariate Analysis*, 105546, 2025.

A. Ghosh, C. Agostinelli, and A. Basu. A Composite Divergence Approach to Robust Multivariate Estimation under Cellwise and Casewise Contamination. *arXiv:2608.18914*, http://arxiv.org/abs/2608.18914, 2026.

Examples

```
mu <- rep(0, 3)
Sigma <- diag(3) * 0.5 + 0.5
set.seed(123)
X <- MASS::mvrnorm(1000, mu, Sigma)
X[1:5, 1]  <- X[1:5, 1] + 5
X[6:10, 2] <- X[6:10, 2] - 10
X[12, 1:2] <- c(-4,8)
colnames(X) <- c("X1","X2","X3")
M <- mvnormDPD(X, beta=0.3, method="multivariate")
C <- mvnormDPD(X, beta=0.3, method="composite")
W <- mvnormDPD(X, beta=0.3, method="componentwise")
rbind(M$mu, C$mu, W$mu)
```

---

mvogamma *The Multivariate Ordered Gamma-Generated Distribution*

Description

Density and random generation for the multivariate ordered gamma-generated distribution with scale parameters given by `delta` and baseline rate equal to `lambda`. The bivariate case coincides with the Mckay bivariate gamma distribution.

Usage

```
dmvogamma(x, delta, lambda, beta=1, log = FALSE)
rmvogamma(n, delta, lambda)
```

Arguments

`x`	vector of quantiles.
`beta`	scalar; see Details.
`n`	number of observations.
`delta`	vector of scale parameters.
`lambda`	scalar rate parameter of the baseline exponential distribution.
`log`	logical; if `TRUE`, densities are given as logarithms.

Details

`delta` and `lambda` are positive parameters. In `dmvogamma` the result is the density power to `beta`.

Value

`dmvogamma` gives the density, `rmvogamma` generates random deviates.

Author(s)

A. Ghosh, C. Agostinelli, and A. Basu.

References

N. Balakrishnan and M.M. Ristic. Multivariate families of gamma-generated distributions with finite or infinite support above or below the diagonal. *Journal of Multivariate Analysis*, 143,194-207, 2016.

See Also

Distributions for other standard distributions.

Examples

```
set.seed(123)
n <- 100
delta <- seq(3, 0.5, by=-0.6)
p <- length(delta)
lambda <- 0.5
x <- rmvogamma(n, delta, lambda)
dmvogamma(x, delta, lambda)
```

---

mvogammaDPD — *Robust estimation for multivariate ordered gamma model using DPD based methods*

Description

Robust estimation for multivariate gamma generated model (Balakrishnan and Ristic, 2016) for ordered positive data using composite Density Power Divergence (DPD) miminization.

Usage

```
mvogammaDPD(x, beta = 0.3,
  method = c("multivariate", "composite"),
  initial = list(delta=NULL, lambda=NULL))
```

Arguments

`x`	`matrix` or `data.frame`. Data with columns representing variables.
`beta`	scalar. Robustness tuning parameter of the DPD. `beta=0` provides the maximum likelihood estimator for all methods, however for a better implementation of the maximum likelihood use the function `mvogammaML`.
`method`	character. `composite` returns the minimum composite DPD estimator based on pairwise likelihood (Ghosh, Agostinelli, and Basu, 2026) othewise the usual `multivariate` version is computed based on the full joint density.
`initial`	User must provide initial values in a list with two components named: delta and lambda, for the scale vector and the baseline rate parameters, respectively.

Value

An object of class `mvdpd`, that for this model is a `list` with components

`delta`	estimated scale parameter vector
`lambda`	estimated baseline rate parameter
`beta`	robustness tuning parameter used by the DPD
`description`	long version of the DPD method used
`method`	the DPD method
`model`	the statistical model
`data`	the input data
`call`	the actual call

Author(s)

A. Ghosh, C. Agostinelli, and A. Basu.

References

N. Balakrishnan and M.M. Ristic. Multivariate families of gamma-generated distributions with finite or infinite support above or below the diagonal. *Journal of Multivariate Analysis*, 143,194-207, 2016.

A. Basu, A. Ghosh, and L. Pardo. *Statistical Inference based on the Density Power Divergence: The Robustness Perspective*. CRC Press, 2025.

A. Ghosh, C. Agostinelli, and A. Basu. A Composite Divergence Approach to Robust Multivariate Estimation under Cellwise and Casewise Contamination. *arXiv2608.18914*, http://arxiv.org/abs/2608.18914, 2026.

See Also

`mvogamma` for the description of the multivariate ordered gamma generated distributions. See also `twocenturies50` for an application of the method to a real dataset about Ultra Marathon Races

Examples

```
set.seed(123)
n <- 80
delta <- seq(3, 0.5, by=-0.6)
p <- length(delta)
lambda <- 0.5
x <- rmvogamma(n=n, delta=delta, lambda=lambda)
mvogammaDPD(x, initial=list(delta=delta, lambda=lambda), beta=0.3)
```

mvogammaML	*Maximum Likelihood estimation for multivariate ordered gamma model*

Description

Maximum Likelihood for a multivariate gamma generated model (Balakrishnan and Ristic, 2016) for ordered positive data.

Usage

```
mvogammaML(x, method=c("multivariate", "composite"),
  initial = list(delta=NULL, lambda=NULL))
```

Arguments

`x`	`matrix` or `data.frame`. Data with columns representing variables.
`method`	character. `composite` returns the maximum composite ML estimator based on pairwise likelihood (Ghosh, Agostinelli, and Basu, 2026), othewise the usual `multivariate` version is computed based on the full joint density.
`initial`	User must provide initial values in a list with two components named: delta and lambda, for the scale vector and the baseline rate parameters, respectively.

Value

An object of class `mvdpd`, that for this model is a `list` with components

`delta`	estimated scale parameter vector
`lambda`	estimated baseline rate parameter
`beta`	robustness tuning parameter set to zero (Maximum Likelihood)
`description`	long version of the ML method used
`method`	equal to 'ML-multivariate'
`model`	the statistical model
`data`	the input data
`call`	the actual call

Author(s)

A. Ghosh, C. Agostinelli, and A. Basu.

References

N. Balakrishnan and M.M. Ristic. Multivariate families of gamma-generated distributions with finite or infinite support above or below the diagonal. *Journal of Multivariate Analysis*, 143,194-207, 2016.

See Also

mvogamma for the description of the multivariate ordered gamma generated distributions and mvogammaDPD for a robust method of estimation

Examples

```
set.seed(123)
n <- 80
delta <- seq(3, 0.5, by=-0.6)
p <- length(delta)
lambda <- 0.5
x <- rmvogamma(n=n, delta=delta, lambda=lambda)
mvogammaML(x, initial=list(delta=delta, lambda=lambda))
```

---

summary.mvdpd *Summary Method for "mvdpd" Objects*

Description

Summary method for R object of class "mvdpd".

Usage

```
 ## S3 method for class 'mvdpd'
summary(object, se = c("sandwich", "exact"),
   null.hypothesis=c(rep(0,p), rep(1,p), rep(0, p*(p-1)/2)),
   print.outliers = FALSE, ...)
```

Arguments

object	an R object of class mvdpd, typically created by mvnormDPD or mvogammaDPD
se	procedure used to evaluate the standard errors
null.hypothesis	the value of the parameters under H_0
print.outliers	logical. If TRUE three tables are also printed, summarizing the presence of cell, couple and row outliers in the dataset
...	potentially more arguments passed to methods

Details

For pbjects with model=="normal" identification of cell, couple and row outliers are also reported by comparing the Mahalanobis distance (univariate, bivariate and row-wise) with the corresponding chi-quare distribution.

Value

`summary(object)` returns an object of S3 class `"summary.mvdpd"`, basically a `list` that contains all the elements from the input `object` of class `mvdpd` and the following elements

`vcov` a list which contains `Sigma`: the asymptotic variance and covariance matrix for all the parameters; `J`: the matrix of gradient of the estimating equations (observed or expected depending of the chosen procedure); `K`: the variance and covariance matrix of the estimating equations (observed or expected depending on the chosen procedure); `se`: standard errors for all the parameters

`zTable` a table with columns: estimated coefficients, standard errors, z-values and p-values

`cellsoutliers` identified cells outliers (available only for the model 'normal')

`couplesoutliers` identified couples outliers (available only for the model 'normal')

`rowsoutliers` identified rows outliers (available only for the model 'normal')

`print.outliers` logical, if `TRUE` the summary of the outliers are printed

Author(s)

A. Ghosh, C. Agostinelli, and A. Basu.

References

A. Basu, I.R. Harris, N.L. Hjort, and M.C. Jones. Robust and efficient estimation by minimizing a density power divergence. *Biometrika*, 85(3), 549-559, 1998.

A. Basu, A. Ghosh, and L. Pardo. *Statistical Inference based on the Density Power Divergence: The Robustness Perspective*. CRC Press, 2025.

S. Chakraborty, A. Basu, and A. Ghosh. A componentwise estimation procedure for multivariate location and scatter: Robustness, efficiency and scalability. *Journal of Multivariate Analysis*, 105546, 2025.

A. Ghosh, C. Agostinelli, and A. Basu. A Composite Divergence Approach to Robust Multivariate Estimation under Cellwise and Casewise Contamination. *arXiv2608.18914*, http://arxiv.org/abs/2608.18914, 2026.

Examples

```
mu <- rep(0, 3)
Sigma <- diag(3) * 0.5 + 0.5
set.seed(123)
X <- MASS::mvrnorm(1000, mu, Sigma)
colnames(X) <- c("X1","X2","X3")
M <- mvnormDPD(X, beta=0.3, method="multivariate")
summary(M, se="sandwich")
summary(M, se="exact", print.outliers=TRUE)
```

twocenturies50 *Two Centuries of Ultra Marathon Races (Male 50-miles races)*

Description

This dataset was obtained from the publicly available data on the website https://www.kaggle.com/datasets/fatihyavuzz/two-centuries-of-um-races, which contains ace results from ultramarathon events held worldwide over approximately two centuries.

We restrict to mens' 50-mile races. For each race event (identified by the event name and date), finishing times were converted to hours and the ten fastest finishers were retained, producing one ordered 10-dimensional observation per race. Events with fewer than ten recorded finishers or with tied finishing times among the top ten were discarded, yielding a collection of complete ordered observations of 4683 races.

Usage

```
data("twocenturies50")
```

Format

A numeric matrix with 4683 rows and 10 columns. Each row contains the best ten recorded finishers in a given race.

Source

https://www.kaggle.com/datasets/fatihyavuzz/two-centuries-of-um-races

Examples

```
data(twocenturies50)
ordered_data <- twocenturies50[order(twocenturies50[,1]),]
n <- nrow(twocenturies50)
p <- ncol(twocenturies50)

# Best 200 races in term of the first finisher (clean data?)
best0 <- ordered_data[1:200,]
best0 <- best0/3600
# Maximum likelihood
ml <- mvogammaML(best0, initial=list(delta=c(18, rep(0.5, 9)),
  lambda=5))
ml



# Maximum composite likelihood (minimum CDPD with beta=0)
cml <- mvogammaDPD(x=best0, beta=0, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
# Minimum composite DPD
cdpd1 <- mvogammaDPD(x=best0, beta=0.1, method="composite",
```

```
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd2 <- mvogammaDPD(x=best0, beta=0.2, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd3 <- mvogammaDPD(x=best0, beta=0.3, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd5 <- mvogammaDPD(x=best0, beta=0.5, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
# Minimum (multivariate) DPD
mdpd1 <- mvogammaDPD(x=best0, beta=0.1, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd2 <- mvogammaDPD(x=best0, beta=0.2, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd3 <- mvogammaDPD(x=best0, beta=0.3, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd5 <- mvogammaDPD(x=best0, beta=0.5, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))

results0.multivariate <- cbind(
  c(ml$delta, ml$lambda),
  c(mdpd1$delta, mdpd1$lambda),
  c(mdpd2$delta, mdpd2$lambda),
  c(mdpd3$delta, mdpd3$lambda),
  c(mdpd5$delta, mdpd5$lambda)
)

results0.composite <- cbind(
  c(cml$delta, cml$lambda),
  c(cdpd1$delta, cdpd1$lambda),
  c(cdpd2$delta, cdpd2$lambda),
  c(cdpd3$delta, cdpd3$lambda),
  c(cdpd5$delta, cdpd5$lambda)
)

colnames(results0.multivariate) <- c("ML", "DPD(0.1)", "DPD(0.2)",
  "DPD(0.3)", "DPD(0.5)")
colnames(results0.composite) <- c("CML", "CDPD(0.1)", "CDPD(0.2)",
  "CDPD(0.3)", "CDPD(0.5)")
rownames(results0.multivariate) <- c(paste0("delta", 1:10), "lambda")
rownames(results0.composite) <- rownames(results0.multivariate)

# Casewise contaminated data
best1 <- ordered_data[c(1:180, (n-19):n),]
best1 <- best1/3600
# Maximum likelihood
ml <- mvogammaML(best1, initial=list(delta=c(18, rep(0.5, 9)),
  lambda=5))
# Maximum composite likelihood (minimum CDPD with beta=0)
cml <- mvogammaDPD(x=best1, beta=0, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
# Minimum composite DPD
cdpd1 <- mvogammaDPD(x=best1, beta=0.1, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd2 <- mvogammaDPD(x=best1, beta=0.2, method="composite",
```

```
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd3 <- mvogammaDPD(x=best1, beta=0.3, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd5 <- mvogammaDPD(x=best1, beta=0.5, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
# Minimum (multivariate) DPD
mdpd1 <- mvogammaDPD(x=best1, beta=0.1, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd2 <- mvogammaDPD(x=best1, beta=0.2, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd3 <- mvogammaDPD(x=best1, beta=0.3, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd5 <- mvogammaDPD(x=best1, beta=0.5, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))

results1.multivariate <- cbind(
  c(ml$delta, ml$lambda),
  c(mdpd1$delta, mdpd1$lambda),
  c(mdpd2$delta, mdpd2$lambda),
  c(mdpd3$delta, mdpd3$lambda),
  c(mdpd5$delta, mdpd5$lambda)
)

results1.composite <- cbind(
  c(cml$delta, cml$lambda),
  c(cdpd1$delta, cdpd1$lambda),
  c(cdpd2$delta, cdpd2$lambda),
  c(cdpd3$delta, cdpd3$lambda),
  c(cdpd5$delta, cdpd5$lambda)
)

colnames(results1.multivariate) <- c("ML", "DPD(0.1)", "DPD(0.2)",
  "DPD(0.3)", "DPD(0.5)")
colnames(results1.composite) <- c("CML", "CDPD(0.1)", "CDPD(0.2)",
  "CDPD(0.3)", "CDPD(0.5)")
rownames(results1.multivariate) <- c(paste0("delta", 1:10), "lambda")
rownames(results1.composite) <- rownames(results1.multivariate)

# Cellwise contaminated data
set.seed(1234)
best2 <- best0
for (i in 1:200) {
  nn <- rbinom(n=1, size=10, prob=0.1)
  if (nn){
    pos <- sample(1:p, size=nn)
    best2[i,pos] <- runif(1, 0, 3)*best0[i,pos]
    best2[i,] <- sort(best2[i,])
  }
}
best2 <- best2/3600
# Maximum likelihood
ml <- mvogammaML(best2, initial=list(delta=c(18, rep(0.5, 9)),
  lambda=5))
```

```
# Maximum composite likelihood (minimum CDPD with beta=0)
cml <- mvogammaDPD(x=best2, beta=0, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
# Minimum composite DPD
cdpd1 <- mvogammaDPD(x=best2, beta=0.1, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd2 <- mvogammaDPD(x=best2, beta=0.2, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd3 <- mvogammaDPD(x=best2, beta=0.3, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
cdpd5 <- mvogammaDPD(x=best2, beta=0.5, method="composite",
  initial=list(delta=ml$delta, lambda=ml$lambda))
# Minimum (multivariate) DPD
mdpd1 <- mvogammaDPD(x=best2, beta=0.1, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd2 <- mvogammaDPD(x=best2, beta=0.2, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd3 <- mvogammaDPD(x=best2, beta=0.3, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))
mdpd5 <- mvogammaDPD(x=best2, beta=0.5, method="multivariate",
  initial=list(delta=ml$delta, lambda=ml$lambda))

results2.multivariate <- cbind(
  c(ml$delta, ml$lambda),
  c(mdpd1$delta, mdpd1$lambda),
  c(mdpd2$delta, mdpd2$lambda),
  c(mdpd3$delta, mdpd3$lambda),
  c(mdpd5$delta, mdpd5$lambda)
)

results2.composite <- cbind(
  c(cml$delta, cml$lambda),
  c(cdpd1$delta, cdpd1$lambda),
  c(cdpd2$delta, cdpd2$lambda),
  c(cdpd3$delta, cdpd3$lambda),
  c(cdpd5$delta, cdpd5$lambda)
)

colnames(results2.multivariate) <- c("ML", "DPD(0.1)", "DPD(0.2)",
  "DPD(0.3)", "DPD(0.5)")
colnames(results2.composite) <- c("CML", "CDPD(0.1)", "CDPD(0.2)",
  "CDPD(0.3)", "CDPD(0.5)")
rownames(results2.multivariate) <- c(paste0("delta", 1:10), "lambda")
rownames(results2.composite) <- rownames(results2.multivariate)
```

Index