Package ‘fitHeavyTail’

October 13, 2022

Title Mean and Covariance Matrix Estimation under Heavy Tails

Version 0.1.4

Date 2022-5-11

Description Robust estimation methods for the mean vector, scatter matrix, and covariance matrix (if it exists) from data (possibly containing NAs) under multivariate heavy-tailed distributions such as angular Gaussian (via Tyler's method), Cauchy, and Student's t distributions. Additionally, a factor model structure can be specified for the covariance matrix. The latest revision also includes the multivariate skewed t distribution. The package is based on the papers: Sun, Babu, and Palomar (2014), Sun, Babu, and Palomar (2015), Liu and Rubin (1995), and Zhou, Liu, Kumar, and Palomar (2019).

Maintainer Daniel P. Palomar <daniel.p.palomar@gmail.com>

URL https://CRAN.R-project.org/package=fitHeavyTail,
https://github.com/dppalomar/fitHeavyTail,
https://www.danielppalomar.com,
https://doi.org/10.1109/TSP.2014.2348944,
https://doi.org/10.1109/TSP.2015.2417513

BugReports https://github.com/dppalomar/fitHeavyTail/issues

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

Depends

Imports ICSNP, mvtnorm, ghyp, numDeriv, stats

Suggests knitr, ggplot2, prettydoc, reshape2, rmarkdown, R.rsp, testthat

VignetteBuilder knitr, rmarkdown, R.rsp

NeedsCompilation no

Author Daniel P. Palomar [cre, aut],
Rui Zhou [aut],
Xiwen Wang [aut]

Repository CRAN

Date/Publication 2022-05-11 08:40:02 UTC

R topics documented:

`fitHeavyTail-package` *fitHeavyTail: Mean and Covariance Matrix Estimation under Heavy Tails*

Description

Robust estimation methods for the mean vector, scatter matrix, and covariance matrix (if it exists) from data (possibly containing NAs) under multivariate heavy-tailed distributions such as angular Gaussian (via Tyler's method), Cauchy, and Student's t distributions. Additionally, a factor model structure can be specified for the covariance matrix.

Functions

`fit_Tyler`, `fit_Cauchy`, and `fit_mvt`

Help

For a quick help see the README file: GitHub-README.

For more details see the vignette: CRAN-vignette.

Author(s)

Daniel P. Palomar and Rui Zhou

References

Ying Sun, Prabhu Babu, and Daniel P. Palomar, "Regularized Tyler's Scatter Estimator: Existence, Uniqueness, and Algorithms," IEEE Trans. on Signal Processing, vol. 62, no. 19, pp. 5143-5156, Oct. 2014. <https://doi.org/10.1109/TSP.2014.2348944>

Ying Sun, Prabhu Babu, and Daniel P. Palomar, "Regularized Robust Estimation of Mean and Covariance Matrix Under Heavy-Tailed Distributions," IEEE Trans. on Signal Processing, vol. 63, no. 12, pp. 3096-3109, June 2015. <https://doi.org/10.1109/TSP.2015.2417513>

Chuanhai Liu and Donald B. Rubin, "ML estimation of the t-distribution using EM and its extensions, ECM and ECME," Statistica Sinica (5), pp. 19-39, 1995.

Rui Zhou, Junyan Liu, Sandeep Kumar, and Daniel P. Palomar, "Robust factor analysis parameter estimation," Lecture Notes in Computer Science (LNCS), 2019. <https://arxiv.org/abs/1909.12530>

`fit_Cauchy` — *Estimate parameters of a multivariate elliptical distribution to fit data under a Cauchy distribution*

Description

Estimate parameters of a multivariate elliptical distribution, namely, the mean vector and the covariance matrix, to fit data. Any data sample with NAs will be simply dropped. The estimation is based on the maximum likelihood estimation (MLE) under a Cauchy distribution and the algorithm is obtained from the majorization-minimization (MM) optimization framework. The Cauchy distribution does not have second-order moments and the algorithm actually estimates the scatter matrix. Nevertheless, assuming that the observed data has second-order moments, the covariance matrix is returned by computing the missing scaling factor with a very effective method.

Usage

```
fit_Cauchy(
  X,
  initial = NULL,
  max_iter = 100,
  ptol = 0.001,
  ftol = Inf,
  return_iterates = FALSE,
  verbose = FALSE
)
```

Arguments

`X` Data matrix containing the multivariate time series (each column is one time series).

`initial` List of initial values of the parameters for the iterative estimation method. Possible elements include:

- `mu`: default is the data sample mean,

- cov: default is the data sample covariance matrix,
- scatter: default follows from the scaled sample covariance matrix.

max_iter
: Integer indicating the maximum number of iterations for the iterative estimation method (default is 100).

ptol
: Positive number indicating the relative tolerance for the change of the variables to determine convergence of the iterative method (default is 1e-3).

ftol
: Positive number indicating the relative tolerance for the change of the log-likelihood value to determine convergence of the iterative method (default is Inf, so it is not active). Note that using this argument might have a computational cost as a convergence criterion due to the computation of the log-likelihood (especially when X is high-dimensional).

return_iterates
: Logical value indicating whether to record the values of the parameters (and possibly the log-likelihood if ftol < Inf) at each iteration (default is FALSE).

verbose
: Logical value indicating whether to allow the function to print messages (default is FALSE).

Value

A list containing possibly the following elements:

mu
: Mean vector estimate.

cov
: Covariance matrix estimate.

scatter
: Scatter matrix estimate.

converged
: Boolean denoting whether the algorithm has converged (TRUE) or the maximum number of iterations max_iter has reached (FALSE).

num_iterations
: Number of iterations executed.

cpu_time
: Elapsed CPU time.

log_likelihood
: Value of log-likelihood after converge of the estimation algorithm (if ftol < Inf).

iterates_record
: Iterates of the parameters (mu, scatter, and possibly log_likelihood (if ftol < Inf)) along the iterations (if return_iterates = TRUE).

Author(s)

Daniel P. Palomar

References

Ying Sun, Prabhu Babu, and Daniel P. Palomar, "Regularized Robust Estimation of Mean and Covariance Matrix Under Heavy-Tailed Distributions," IEEE Trans. on Signal Processing, vol. 63, no. 12, pp. 3096-3109, June 2015.

See Also

fit_Tyler and fit_mvt

Examples

```
library(mvtnorm)       # to generate heavy-tailed data
library(fitHeavyTail)

X <- rmvt(n = 1000, df = 6)  # generate Student's t data
fit_Cauchy(X)
```

`fit_mvst` *Estimate parameters of a multivariate (generalized hyperbolic) skewed t distribution to fit data*

Description

Estimate parameters of a multivariate (generalized hyperbolic) skewed Student's t distribution to fit data, namely, the location vector, the scatter matrix, the skewness vector, and the degrees of freedom. The estimation is based on the maximum likelihood estimation (MLE) and the algorithm is obtained from the expectation-maximization (EM) method.

Usage

```
fit_mvst(
  X,
  nu = NULL,
  gamma = NULL,
  initial = NULL,
  max_iter = 500,
  ptol = 0.001,
  ftol = Inf,
  PXEM = TRUE,
  return_iterates = FALSE,
  verbose = FALSE
)
```

Arguments

`X` Data matrix containing the multivariate time series (each column is one time series).

`nu` Degrees of freedom of the skewed t distribution (otherwise it will be iteratively estimated).

`gamma` Skewness vector of the skewed t distribution (otherwise it will be iteratively estimated).

`initial` List of initial values of the parameters for the iterative estimation method. Possible elements include:

- `nu`: default is 4,
- `mu`: default is the data sample mean,

- gamma: default is the sample skewness vector,
- scatter: default follows from the scaled sample covariance matrix,

max_iter
: Integer indicating the maximum number of iterations for the iterative estimation method (default is 500).

ptol
: Positive number indicating the relative tolerance for the change of the variables to determine convergence of the iterative method (default is 1e-3).

ftol
: Positive number indicating the relative tolerance for the change of the log-likelihood value to determine convergence of the iterative method (default is Inf, so it is not active). Note that using this argument might have a computational cost as a convergence criterion due to the computation of the log-likelihood (especially when X is high-dimensional).

PXEM
: Logical value indicating whether to use the parameter expansion (PX) EM method to accelerating the convergence.

return_iterates
: Logical value indicating whether to record the values of the parameters (and possibly the log-likelihood if ftol < Inf) at each iteration (default is FALSE).

verbose
: Logical value indicating whether to allow the function to print messages (default is FALSE).

Details

This function estimates the parameters of a (generalized hyperbolic) multivariate Student's t distribution (mu, scatter, gamma and nu) to fit the data via the expectation-maximization (EM) algorithm.

Value

A list containing (possibly) the following elements:

mu
: Location vector estimate (not the mean).

gamma
: Skewness vector estimate.

scatter
: Scatter matrix estimate.

nu
: Degrees of freedom estimate.

mean
: Mean vector estimate:

```
mean = mu + nu/(nu-2) * gamma
```

cov
: Covariance matrix estimate:

```
cov = nu/(nu-2) * scatter + 2*nu^2 / (nu-2)^2 / (nu-4) * gamma*gamma'
```

converged
: Boolean denoting whether the algorithm has converged (TRUE) or the maximum number of iterations max_iter has been reached (FALSE).

num_iterations
: Number of iterations executed.

cpu_time
: Elapsed overall CPU time.

log_likelihood_vs_iterations
: Value of log-likelihood over the iterations (if ftol < Inf).

iterates_record
: Iterates of the parameters (mu, scatter, nu, and possibly log_likelihood (if ftol < Inf)) along the iterations (if return_iterates = TRUE).

cpu_time_at_iter
: Elapsed CPU time at each iteration (if return_iterates = TRUE).

Author(s)

Rui Zhou, Xiwen Wang, and Daniel P. Palomar

References

Aas Kjersti and Ingrid Hobæk Haff. "The generalized hyperbolic skew Student's t-distribution," Journal of financial econometrics, pp. 275-309, 2006.

See Also

fit_mvt

Examples

```
library(mvtnorm)       # to generate heavy-tailed data
library(fitHeavyTail)

# parameter setting
N <- 5
T <- 200
nu <- 6
mu <- rnorm(N)
scatter <- diag(N)
gamma <- rnorm(N)   # skewness vector

# generate GH Skew t data
taus <- rgamma(n = T, shape = nu/2, rate = nu/2)
X <- matrix(data = mu, nrow = T, ncol = N, byrow = TRUE) +
     matrix(data = gamma, nrow = T, ncol = N, byrow = TRUE) / taus +
     rmvnorm(n = T, mean = rep(0, N), sigma = scatter) / sqrt(taus)

# fit skew t model
fit_mvst(X)

# setting lower limit for nu (e.g., to guarantee existence of co-skewness and co-kurtosis matrices)
options(nu_min = 8.01)
fit_mvst(X)
```

`fit_mvt` *Estimate parameters of a multivariate Student's t distribution to fit data*

Description

Estimate parameters of a multivariate Student's t distribution to fit data, namely, the mean vector, the covariance matrix, the scatter matrix, and the degrees of freedom. The data can contain missing values denoted by NAs. It can also consider a factor model structure on the covariance matrix. The estimation is based on the maximum likelihood estimation (MLE) and the algorithm is obtained from the expectation-maximization (EM) method.

Usage

```
fit_mvt(
  X,
  na_rm = TRUE,
  nu = c("kurtosis", "MLE-diag", "MLE-diag-resampled", "iterative"),
  nu_iterative_method = c("ECME-diag", "ECME", "ECM", "ECME-cov", "theta-0",
   "theta-1a", "theta-1b", "theta-2a", "theta-2b", "POP", "POP-sigma-corrected",
    "POP-sigma-corrected-true"),
  initial = NULL,
  optimize_mu = TRUE,
  weights = NULL,
  scale_minMSE = FALSE,
  factors = ncol(X),
  max_iter = 100,
  ptol = 0.001,
  ftol = Inf,
  return_iterates = FALSE,
  verbose = FALSE
)
```

Arguments

`X` Data matrix containing the multivariate time series (each column is one time series).

`na_rm` Logical value indicating whether to remove observations with some NAs (default). Otherwise, the NAs will be imputed at a higher computational cost.

`nu` Degrees of freedom of the t distribution. Either a number (>2) or a string indicating the method to compute it:

- `"kurtosis"`: based on the kurtosis obtained from the sampled moments (default method);
- `"MLE-diag"`: based on the MLE assuming a diagonal sample covariance;
- `"MLE-diag-resampled"`: method "MLE-diag" resampled for better stability;

- "iterative": iterative estimation with the rest of the parameters via the EM algorithm.

nu_iterative_method
: String indicating the method for iteratively estimating nu (in case nu = "iterative"):
  - "ECM": maximization of the Q function;
  - "ECME": maximization of the log-likelihood function;
  - "ECME-diag": maximization of the log-likelihood function assuming a diagonal scatter matrix (default method).

initial
: List of initial values of the parameters for the iterative estimation method (in case nu = "iterative"). Possible elements include:
  - mu: default is the data sample mean,
  - cov: default is the data sample covariance matrix,
  - scatter: default follows from the scaled sample covariance matrix,
  - nu: can take the same values as argument nu, default is 4,
  - B: default is the top eigenvectors of initial$cov multiplied by the sqrt of the eigenvalues,
  - psi: default is diag(initial$cov - initial$B %*% t(initial$B)).

optimize_mu
: Boolean indicating whether to optimize mu (default is TRUE).

weights
: Optional weights for each of the observations (the length should be equal to the number of rows of X).

scale_minMSE
: Logical value indicating whether to scale the scatter and covariance matrices to minimize the MSE estimation error by introducing bias (default is FALSE).

factors
: Integer indicating number of factors (default is ncol(X), so no factor model assumption).

max_iter
: Integer indicating the maximum number of iterations for the iterative estimation method (default is 100).

ptol
: Positive number indicating the relative tolerance for the change of the variables to determine convergence of the iterative method (default is 1e-3).

ftol
: Positive number indicating the relative tolerance for the change of the log-likelihood value to determine convergence of the iterative method (default is Inf, so it is not active). Note that using this argument might have a computational cost as a convergence criterion due to the computation of the log-likelihood (especially when X is high-dimensional).

return_iterates
: Logical value indicating whether to record the values of the parameters (and possibly the log-likelihood if ftol < Inf) at each iteration (default is FALSE).

verbose
: Logical value indicating whether to allow the function to print messages (default is FALSE).

Details

This function estimates the parameters of a multivariate Student's t distribution (mu, cov, scatter, and nu) to fit the data via the expectation-maximization (EM) algorithm. The data matrix X can contain missing values denoted by NAs. The estimation of nu if very flexible: it can be directly passed as an argument (without being estimated), it can be estimated with several one-shot methods (namely, "kurtosis", "MLE-diag", "MLE-diag-resampled"), and it can also be iteratively estimated with the other parameters via the EM algorithm.

Value

A list containing (possibly) the following elements:

- `mu` Mu vector estimate.
- `scatter` Scatter matrix estimate.
- `nu` Degrees of freedom estimate.
- `mean` Mean vector estimate:

  `mean = mu`
- `cov` Covariance matrix estimate:

  `cov = nu/(nu-2) * scatter`
- `converged` Boolean denoting whether the algorithm has converged (`TRUE`) or the maximum number of iterations `max_iter` has been reached (`FALSE`).
- `num_iterations` Number of iterations executed.
- `cpu_time` Elapsed CPU time.
- `B` Factor model loading matrix estimate according to `cov = (B %*% t(B) + diag(psi)` (only if factor model requested).
- `psi` Factor model idiosynchratic variances estimates according to `cov = (B %*% t(B) + diag(psi)` (only if factor model requested).
- `log_likelihood_vs_iterations`
  Value of log-likelihood over the iterations (if `ftol < Inf`).
- `iterates_record`
  Iterates of the parameters (`mu`, `scatter`, `nu`, and possibly `log_likelihood` (if `ftol < Inf`)) along the iterations (if `return_iterates = TRUE`).

Author(s)

Daniel P. Palomar and Rui Zhou

References

Chuanhai Liu and Donald B. Rubin, "ML estimation of the t-distribution using EM and its extensions, ECM and ECME," Statistica Sinica (5), pp. 19-39, 1995.

Chuanhai Liu, Donald B. Rubin, and Ying Nian Wu, "Parameter Expansion to Accelerate EM: The PX-EM Algorithm," Biometrika, Vol. 85, No. 4, pp. 755-770, Dec., 1998

Rui Zhou, Junyan Liu, Sandeep Kumar, and Daniel P. Palomar, "Robust factor analysis parameter estimation," Lecture Notes in Computer Science (LNCS), 2019. <https://arxiv.org/abs/1909.12530>

Esa Ollila, Daniel P. Palomar, and Frédéric Pascal, "Shrinking the Eigenvalues of M-estimators of Covariance Matrix," IEEE Trans. on Signal Processing, vol. 69, pp. 256-269, Jan. 2021.

See Also

fit_Tyler, fit_Cauchy, and fit_mvst

Examples

```
library(mvtnorm)       # to generate heavy-tailed data
library(fitHeavyTail)

X <- rmvt(n = 1000, df = 6)  # generate Student's t data
fit_mvt(X)

# setting lower limit for nu
options(nu_min = 4.01)
fit_mvt(X, nu = "iterative")
```

---

fit_Tyler — *Estimate parameters of a multivariate elliptical distribution to fit data via Tyler's method*

Description

Estimate parameters of a multivariate elliptical distribution, namely, the mean vector and the covariance matrix, to fit data. Any data sample with NAs will be simply dropped. The algorithm is based on Tyler's method, which normalizes the centered samples to get rid of the shape of the distribution tail. The data is first demeaned (with the geometric mean by default) and normalized. Then the estimation is based on the maximum likelihood estimation (MLE) and the algorithm is obtained from the majorization-minimization (MM) optimization framework. Since Tyler's method can only estimate the covariance matrix up to a scaling factor, a very effective method is employed to recover the scaling factor.

Usage

```
fit_Tyler(
  X,
  initial = NULL,
  max_iter = 100,
  ptol = 0.001,
  ftol = Inf,
  return_iterates = FALSE,
  verbose = FALSE
)
```

Arguments

X	Data matrix containing the multivariate time series (each column is one time series).

initial
: List of initial values of the parameters for the iterative estimation method. Possible elements include:
  - mu: default is the data sample mean,
  - cov: default is the data sample covariance matrix.

max_iter
: Integer indicating the maximum number of iterations for the iterative estimation method (default is 100).

ptol
: Positive number indicating the relative tolerance for the change of the variables to determine convergence of the iterative method (default is 1e-3).

ftol
: Positive number indicating the relative tolerance for the change of the log-likelihood value to determine convergence of the iterative method (default is Inf, so it is not active). Note that using this argument might have a computational cost as a convergence criterion due to the computation of the log-likelihood (especially when X is high-dimensional).

return_iterates
: Logical value indicating whether to record the values of the parameters (and possibly the log-likelihood if ftol < Inf) at each iteration (default is FALSE).

verbose
: Logical value indicating whether to allow the function to print messages (default is FALSE).

Value

A list containing possibly the following elements:

mu
: Mean vector estimate.

cov
: Covariance matrix estimate.

converged
: Boolean denoting whether the algorithm has converged (TRUE) or the maximum number of iterations max_iter has reached (FALSE).

num_iterations
: Number of iterations executed.

cpu_time
: Elapsed CPU time.

log_likelihood
: Value of log-likelihood after converge of the estimation algorithm (if ftol < Inf).

iterates_record
: Iterates of the parameters (mu, scatter, and possibly log_likelihood (if ftol < Inf)) along the iterations (if return_iterates = TRUE).

Author(s)

Daniel P. Palomar

References

Ying Sun, Prabhu Babu, and Daniel P. Palomar, "Regularized Tyler's Scatter Estimator: Existence, Uniqueness, and Algorithms," IEEE Trans. on Signal Processing, vol. 62, no. 19, pp. 5143-5156, Oct. 2014.

See Also

fit_Cauchy and fit_mvt

Examples

```
library(mvtnorm)       # to generate heavy-tailed data
library(fitHeavyTail)

X <- rmvt(n = 1000, df = 6)  # generate Student's t data
fit_Tyler(X)
```

Index