Package ‘CondMVT’

June 28, 2022

Type Package

Title Conditional Multivariate t Distribution, Expectation Maximization Algorithm, and Its Stochastic Variants

Version 0.1.0

Maintainer Paul Kinyanjui <kinyanjui.access@gmail.com>

Description Computes conditional multivariate t probabilities, random deviates, and densities. It can also be used to create missing values at random in a dataset, resulting in a missing at random (MAR) mechanism. Inbuilt in the package are the Expectation-Maximization (EM), Monte Carlo EM, and Stochastic EM algorithms for imputation of missing values in datasets assuming the multivariate t distribution. See Kinyanjui, Tamba, Orawo, and Okenye (2020)<doi:10.3233/mas-200493>, and Kinyanjui, Tamba, and Okenye(2021)<http://www.ceser.in/ceserp/index.php/ijamas/article/view/6726/0> for more details.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.0

Imports stats, mvtnorm

NeedsCompilation no

Author Paul Kinyanjui [aut, cre],
Cox Tamba [aut],
Justin Okenye [aut],
Luke Orawo [ctb]

Repository CRAN

Date/Publication 2022-06-28 07:20:08 UTC

R topics documented:

`CondMVT` *Conditional Location Vector, Scatter Matrix, and Degrees of Freedom of Multivariate t Distribution*

Description

These functions provide the conditional location vector, scatter matrix, and degrees of freedom of [Y given X], where Z = (X,Y) is the fully-joint multivariate t distribution with location vector equal to mean, scatter matrix sigma, and degrees of freedom df. For more details on the computation of the parameters and their respective formulae, see Roth (2013).

Usage

```
CondMVT(mean, sigma, df, dependent.ind, given.ind, X.given,
  check.sigma = TRUE)
```

Arguments

`mean`	location vector, which must be specified.
`sigma`	a symmetric, positive-definte matrix of dimension n x n, which must be specified.
`df`	degrees of freedom, which must be specified.
`dependent.ind`	a vector of integers denoting the indices of dependent variable Y.
`given.ind`	a vector of integers denoting the indices of conditoning variable X.
`X.given`	a vector of reals denoting the conditioning value of X. When both given.ind and X.given are missing, the distribution of Y becomes Z[dependent.ind]
`check.sigma`	logical; if TRUE, the scatter matrix is checked for appropriateness (symmetry, positive-definiteness). This could be set to FALSE if the user knows it is appropriate.

Value

Returns the conditional location vector (`condMean`), conditional scatter matrix (`condVar`), and the conditional degrees of freedom (`cond_df`) for the multvariate t distribution.

References

Roth, M. (2013). On the multivariate t-distribution, Tech Rep.

Examples

```
# 10-dimensional multivariate normal distribution
n <- 10
df=3
A <- matrix(rt(n^2,df), n, n)
A <- tcrossprod(A,A) #A %*% t(A)
CondMVT(mean=rep(1,n), sigma=A, df=df, dependent=c(2,3,5), given=c(1,4,7,9),X.given=c(1,1,0,-1))
CondMVT(mean=rep(1,n), sigma=A, df=df, dep=3, given=c(1,4,7,9), X=c(1,1,0,-1))
```

---

dcmvt *Conditional Multivariate t Density and Random Deviates*

Description

This function provides the density function for the conditional multivariate t distribution, [Y given X], where Z = (X,Y) is the fully-joint multivariate t distribution with location vector (or mode) equal to mean and covariance matrix sigma.

Usage

```
dcmvt(x, mean, sigma,df, dependent.ind, given.ind, X.given, check.sigma=TRUE, log = FALSE)
```

Arguments

`x`	vector or matrix of quantiles of Y. If x is a matrix, each row is taken to be a quantile.
`mean`	location vector, which must be specified.
`sigma`	a symmetric, positive-definte matrix of dimension n x n, which must be specified.
`df`	degrees of freedom, which must be specified.
`dependent.ind`	a vector of integers denoting the indices of dependent variable Y.
`given.ind`	a vector of integers denoting the indices of conditoning variable X.
`X.given`	a vector of reals denoting the conditioning value of X. When both given.ind and X.given are missing, the distribution of Y becomes Z[dependent.ind]
`check.sigma`	logical; if TRUE, the scatter matrix is checked for appropriateness (symmetry, positive-definiteness). This could be set to FALSE if the user knows it is appropriate.
`log`	logical; if TRUE, densities d are given as log(d).

Value

numeric

References

Genz, A. and Bretz, F. (2009), Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics, Vol. 195. Springer-Verlag, Heidelberg.

S. Kotz and S. Nadarajah (2004), Multivariate t Distributions and Their Applications. Cambridge University Press. Cambridge.

Examples

```
# 10-dimensional multivariate t distribution
n <- 10
df=3
A <- matrix(rt(n^2,df), n, n)
A <- tcrossprod(A,A) #A %*% t(A)

# density of Z[c(2,5)] given Z[c(1,4,7,9)]=c(1,1,0,-1)
dcmvt(x=c(1.2,-1), mean=rep(1,n), sigma=A,dependent.ind=c(2,5),df=df, given.ind=c(1,4,7,9),
X.given=c(1,1,0,-1))

dcmvt(x=-1, mean=rep(1,n), sigma=A,df=df, dep=3, given=c(1,4,7,9,10), X=c(1,1,0,0,-1))
dcmvt(x=c(1.2,-1), mean=rep(1,n), sigma=A,df=df, dep=c(2,5))

# gives an error since `x' and `dep' are incompatibe
#dcmvt(x=-1, mean=rep(1,n), sigma=A,df=df, dep=c(2,3),
#      given=c(1,4,7,9,10), X=c(1,1,0,0,-1))

rcmvt(n=10, mean=rep(1,n), sigma=A,df=df, dep=c(2,5),
         given=c(1,4,7,9,10), X=c(1,1,0,0,-1),type="shifted",
         method="eigen")

rcmvt(n=10, mean=rep(1,n), sigma=A,df=df, dep=3,
         given=c(1,4,7,9,10), X=c(1,1,0,0,-1),type="Kshirsagar",
         method="chol")
```

EM_msteps — *Data Imputation Using EM (Multiple Iterations; Degrees of Freedom Known)*

Description

The sub-package contains subroutines for imputation of missing values as well as parameter estimation (for the location vector and the scatter matrix) in multivariate t distribution using the Expectation Maximization (EM) algorithm when the degrees of freedom are known. EM algorithm iteratively imputes the missing values and computes the estimates for the multivariate t parameters in two steps (E-step and M-step) as explained in Kinyanjui et al. (2021). For a single iteration, the function EM_onestep is run. For multiple iterations, the function EM_msteps is run. The function LIKE (specifying the likelihood) facilitates the setting of tolerance level for convergence of the algorithm (that is $L(\theta^{t+1}) - L(\theta^t) \leq \delta$, where δ is a set tolerance level and t denotes the number of iterations).

Usage

```
EM_msteps(Y,mu,Sigma,df,K,error)
```

Arguments

Y	the multivariate t dataset
mu	the location vector, which must be specified. In cases where it is unknown, starting values are provided.
Sigma	scatter matrix, which must be specified. In cases where it is unknown, starting values are provided.
df	degrees of freedom, which must be specified.
K	the number of iterations, which must be specified.
error	tolerance level for convergence of the EM algorithm.

Value

Completed dataset (with imputed values), updated location vector, and scatter matrix. All outputs are numeric.

References

Kinyanjui, P.K., Tamba, C.L., & Okenye, J.O. (2021). Missing Data Imputation in a t -Distribution with Known Degrees of Freedom Via Expectation Maximization Algorithm and Its Stochastic Variants. International Journal of Applied Mathematics and Statistics.

Examples

```
# 3-dimensional multivariate t distribution
n <- 10
p=3
df=3
mu=c(1:3)
A <- matrix(rt(p^2,df), p, p)
A <- tcrossprod(A,A) #A %*% t(A)

Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
Y8= MISS(TT,20) #The newly created incomplete dataset.
Y8
```

```
#Initializing Values
mu_stat=c(0.5,1,2)
Sigma_stat=matrix(c(0.33,0.31,0.3,0.31,0.335,0.295,0.3,0.295,0.32),3,3)

#Imputing Missing Values and Updating Parameter Estimates
#Single Iteration (EM)
EM1=EM_onestep(Y=Y8,mu=mu_stat,Sigma=Sigma_stat,df=df)

#Multiple Iterations (EM)
EM=EM_msteps(Y=Y8,mu=mu_stat,Sigma=Sigma_stat,df=3,K=1000,error=0.00001)

#Results for Newly Completed Dataset (EM)
EM$IMP   #Newly completed Dataset (with imputed values)
EM$mu    #updated location vector
EM$Sigma #updated scatter matrix
EM$K1 #number of iterations the algorithm takes to converge
```

EM_onestep *Data Imputation Using EM (Single Iteration; Degrees of Freedom Known)*

Description

The sub-package contains subroutines for imputation of missing values as well as parameter estimation (for the location vector and the scatter matrix) in multivariate t distribution using the Expectation Maximization (EM) algorithm when the degrees of freedom are known. EM algorithm imputes the missing values and computes the estimates for the multivariate t parameters in two steps (E-step and M-step) as explained in Kinyanjui et al. (2021). For a single iteration, the function EM_onestep is run.Arbitrary starting values are supplied to initiate the algorithm.

Usage

```
EM_onestep(Y,mu,Sigma,df)
```

Arguments

Y	the multivariate t dataset
mu	the location vector, which must be specified. In cases where it is unknown, starting values are provided.
Sigma	scatter matrix, which must be specified. In cases where it is unknown, starting values are provided.
df	degrees of freedom, which must be specified. algorithm.

Value

Completed dataset (with imputed values), updated location vector, and scatter matrix. All outputs are numeric.

References

Kinyanjui, P.K., Tamba, C.L., & Okenye, J.O. (2021). Missing Data Imputation in a t -Distribution with Known Degrees of Freedom Via Expectation Maximization Algorithm and Its Stochastic Variants. International Journal of Applied Mathematics and Statistics.

Examples

```
# 3-dimensional multivariate t distribution
n <- 10
p=3
df=3
mu=c(1:3)
A <- matrix(rt(p^2,df), p, p)
A <- tcrossprod(A,A) #A %*% t(A)

Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
Y8= MISS(TT,20) #The newly created incomplete dataset.
Y8

#Initializing Values
mu_stat=c(0.5,1,2)
Sigma_stat=matrix(c(0.33,0.31,0.3,0.31,0.335,0.295,0.3,0.295,0.32),3,3)

#Imputing Missing Values and Updating Parameter Estimates
#Single Iteration (EM)
EM1=EM_onestep(Y=Y8,mu=mu_stat,Sigma=Sigma_stat,df=df)

#Results for Newly Completed Dataset (EM)
EM1$Y2    #Newly completed Dataset (with imputed values)
EM1$mu   #updated location vector
EM1$Sigma #updated scatter matrix
```

EM_Umsteps — *Data Imputation Using EM (Multiple Iterations, Degrees of Freedom Unknown)*

Description

This sub-package constitutes the subroutines for EM algorithm (for multiple iterations). It has 2 functions namely `LIKE` and `EM_Umsteps`. The function `EM_Umsteps` carries out missing data imputation as well as parameter estimation in multivariate t distribution in multiple iterations; assuming that the degrees of freedom are unknown. In addition to updating the location vector and the scatter matrix, therefore, the function also finds an estimate for the degrees of freedom. The bisection method is employed in the algorithm to iteratively update the degrees of freedom.The function `LIKE` (specifying the likelihood) facilitates the setting of tolerance level for convergence of the EM algorithm (that is $L(\theta^{t+1}) - L(\theta^t) \leq \delta$, where δ is a set tolerance level and t denotes the number of iterations).Details of how EM works in light of unknown degrees of freedom can be found in Kinyanjui et al. (2020) and Liu and Rubin (1995).

Usage

```
EM_Umsteps(Y,mu,Sigma,df,K,e,error)
```

Arguments

`Y`	the multivariate t dataset
`mu`	the location vector, which must be specified. In cases where it is unknown, starting values are provided.
`Sigma`	Scatter matrix, which must be specified. In cases where it is unknown, starting values are provided.
`df`	degrees of freedom, which must be specified.
`e`	tolerance level for convergence of the bisection method for estimation of df.
`error`	tolerance level for convergence of the EM algorithm.
`K`	the number of iterations, which must be specified.

Value

Completed dataset, updated location vector,scatter matrix, and degrees of freedom. All outputs are numeric.

References

Kinyanjui, P. K., Tamba, C. L., Orawo, L. A. O., & Okenye, J. O. (2020). Missing data imputation in multivariate t distribution with unknown degrees of freedom using expectation maximization algorithm and its stochastic variants. Model Assisted Statistics and Applications, 15(3), 263-272.

Liu, C. and Rubin, D. B. (1995). ML estimation of the t distribution using EM and its extensions, ECM and ECME. Statistica Sinica, 19-39.

Examples

```
# 3-dimensional multivariate t distribution
n <- 25
p=3
df=3
mu=c(10,20,30)
A=matrix(c(14,10,12,10,13,9,12,9,18), 3,3)
Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
```

```
Y8= MISS(TT,20) #The newly created incomplete dataset.
Y8

#Initializing Values
mu_stat=c(0.5,1,2)
Sigma_stat=matrix(c(0.33,0.31,0.3,0.31,0.335,0.295,0.3,0.295,0.32),3,3)
df_stat=6

#Imputing Missing Values and Updating Parameter Estimates
#Single Iteration (EM)
EMU1=EM_Uonestep (Y=Y8,mu=mu,Sigma= Sigma_stat,df= df_stat,e=0.00001)

#Multiple Iterations (EM)
EMU=EM_Umsteps(Y=Y8,mu=mu_stat,Sigma=Sigma_stat,df=df_stat,K=1000,e=0.00001,error=0.00001)

#Results for Newly Completed Dataset (EM)
EMU$IMP #Newly completed Dataset (with imputed values)
EMU$mu   #updated location vector
EMU$Sigma #updated scatter matrix
EMU$df     #Updated degrees of freedom.
EMU$K1 #number of iterations the algorithm takes to converge
```

EM_Uonestep   *Data Imputation Using EM (Single Iteration, Degrees of Freedom Unknown)*

Description

This sub-package constitutes the subroutines for EM algorithm (for a single iteration). It has 4 functions namely `fun1`, `dfun1`, `Bisec`, and `EM_Uonestep`. The function EM_Uonestep carries out missing data imputation as well as parameter estimation in multivariate t distribution in one iteration; assuming that the degrees of freedom are unknown. In addition to updating the location vector and the scatter matrix, therefore, the function also finds an estimate for the degrees of freedom. The bisection method is employed in the algorithm to iteratively update the degrees of freedom. In this respect, function `fun1` specifies the degrees of freedom equation to be solved. `dfun1` is its derivative. The two functions (`fun1` and `dfun1`) are then solved numerically using the bisection method as specified in the function `Bisec`.Details of how EM works in light of unknown degrees of freedom can be found in Kinyanjui et al. (2020) and Liu and Rubin (1995).

Usage

```
EM_Uonestep(Y,mu,Sigma,df,e)
```

Arguments

`Y`	the multivariate t dataset
`mu`	the location vector, which must be specified. In cases where it is unknown, starting values are provided.

Sigma	Scatter matrix, which must be specified. In cases where it is unknown, starting values are provided.
df	degrees of freedom, which must be specified.
e	tolerance level for convergence of the bisection method for estimation of df.

Value

Completed dataset, updated location vector,scatter matrix, and degrees of freedom. All outputs are numeric.

References

Kinyanjui, P. K., Tamba, C. L., Orawo, L. A. O., & Okenye, J. O. (2020). Missing data imputation in multivariate t distribution with unknown degrees of freedom using expectation maximization algorithm and its stochastic variants. Model Assisted Statistics and Applications, 15(3), 263-272.

Liu, C., & Rubin, D. B. (1995). ML estimation of the t distribution using EM and its extensions, ECM and ECME. Statistica Sinica, 19-39.

Examples

```
# 3-dimensional multivariate t distribution
n <- 25
p=3
df=3
mu=c(10,20,30)
A=matrix(c(14,10,12,10,13,9,12,9,18), 3,3)
Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
Y8= MISS(TT,20) #The newly created incomplete dataset.
Y8

#Initializing Values
mu_stat=c(0.5,1,2)
Sigma_stat=matrix(c(0.33,0.31,0.3,0.31,0.335,0.295,0.3,0.295,0.32),3,3)
df_stat=6

#Imputing Missing Values and Updating Parameter Estimates
#Single Iteration (EM)
EMU1=EM_Uonestep (Y=Y8,mu=mu,Sigma= Sigma_stat,df= df_stat,e=0.00001)

#Results for Newly Completed Dataset (EM)
EMU1$Y2    #Newly completed Dataset (with imputed values)
EMU1$mu    #updated location vector
EMU1$Sigma #updated scatter matrix
EMU1$df
```

MISS *Creating Missing Values at Random in Multivariate Datasets*

Description

This function randomly creates missing values in a multivariate dataset. The resultant missing data mechanism is missing at random (MAR). The percentage of missingness has to be specified. This percentage is computed as a proportion of the sample size. In addition, the function allows for more than one missing value in any given case. It is set such that in a p-variate dataset, for any i^{th} case, the maximum allowable number of missing values is $p - 1$. This helps avoid a situation where a case has no observed value.

Usage

```
MISS (TT, Percent)
```

Arguments

TT	n×p complete dataset.
Percent	the proportion of missing values, which must be specified.

Value

Data Y of size n×p with missing values (NA) created at random. The missing values are logical in nature.

Examples

```
# 3-dimensional multivariate t distribution
n <- 10
p=3
df=3
mu=c(1:3)
A <- matrix(rt(p^2,df), p, p)
A <- tcrossprod(A,A) #A %*% t(A)

Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
Y8= MISS(TT,20) #The newly created incomplete dataset.
Y8
```

pcmvt *Conditional Multivariate t Distribution*

Description

Computes the distribution function of the conditional multivariate t, [Y given X], where Z = (X,Y) is the fully-joint multivariate t distribution with mean equal to location vector, df equal to degrees of freedom and scatter matrix sigma. Computations are based on algorithms by Genz and Bretz.

Usage

```
pcmvt(lower = -Inf, upper = Inf, mean, sigma, df, dependent.ind, given.ind, X.given,
check.sigma = TRUE, algorithm = GenzBretz(), ...)
```

Arguments

`lower`	the vector of lower limits of length n.
`upper`	the vector of upper limits of length n.
`mean`	the mean vector of length n.
`sigma`	a symmetric, positive-definte matrix, of dimension n x n, which must be specified.
`df`	degrees of freedom, which must be specified.
`dependent.ind`	a vector of integers denoting the indices of the dependent variable Y.
`given.ind`	a vector of integers denoting the indices of the conditioning variable X.
`X.given`	a vector of reals denoting the conditioning value of X. When both given.ind and X.given are missing, the distribution of Y becomes Z[dependent.ind]
`check.sigma`	logical; if TRUE, the variance-covariance matrix is checked for appropriateness (symmetry, positive-definiteness). This could be set to FALSE if the user knows it is appropriate.
`algorithm`	an object of class GenzBretz, Miwa or TVPACK specifying both the algorithm to be used as well as the associated hyper parameters.
`...`	additional parameters (currently given to GenzBretz for backward compatibility issues).

Details

This program involves the computation of multivariate t probabilities with arbitrary correlation matrices.

Value

The evaluated distribution function is returned with attributes

`error`	estimated absolute error and
`msg`	Normal Completion

References

Genz, A. and Bretz, F. (1999), Numerical computation of multivariate t-probabilities with application to power calculation of multiple contrasts. Journal of Statistical Computation and Simulation, 63, 361–378.

Genz, A. and Bretz, F. (2002), Methods for the computation of multivariate t-probabilities. Journal of Computational and Graphical Statistics, 11, 950–971.

Genz, A. (2004), Numerical computation of rectangular bivariate and trivariate normal and t-probabilities, Statistics and Computing, 14, 251–260.

Genz, A. and Bretz, F. (2009), Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics, Vol. 195. Springer-Verlag, Heidelberg.

See Also

`dcmvt()`,`rcmvt()`,`pmvt()`,`GenzBretz()`

Examples

```
n <- 10
df=3
A <- matrix(rt(n^2,df), n, n)
A <- tcrossprod(A,A) #A %*% t(A)

pcmvt(lower=-Inf, upper=1, mean=rep(1,n), sigma=A, df=df,dependent.ind=3,
      given.ind=c(1,4,7,9,10), X.given=c(1,1,0,0,-1))


pcmvt(lower=-Inf, upper=c(1,2), mean=rep(1,n),
      sigma=A,df=df, dep=c(2,5), given=c(1,4,7,9,10),
      X=c(1,1,0,0,-1))

pcmvt(lower=-Inf, upper=c(1,2), mean=rep(1,n), sigma=A,df=df,
      dep=c(2,5))
```

---

rcmvt *Conditional Multivariate t Density and Random Deviates*

Description

This function provides the random number generator for the conditional multivariate t distribution, [Y given X], where Z = (X,Y) is the fully-joint multivariate t distribution with location vector equal to mean and scatter matrix sigma.

Usage

```
rcmvt(n, mean, sigma, df,dependent.ind, given.ind, X.given,
check.sigma = TRUE,type = c("Kshirsagar", "shifted"),
method = c("eigen", "svd", "chol"))
```

Arguments

n	number of random deviates.
mean	location vector, which must be specified.
sigma	a symmetric, positive-definte matrix of dimension n x n, which must be specified.
df	degrees of freedom, which must be specified
dependent.ind	a vector of integers denoting the indices of dependent variable Y.
given.ind	a vector of integers denoting the indices of conditoning variable X.
X.given	a vector of reals denoting the conditioning value of X. When both given.ind and X.given are missing, the distribution of Y becomes Z[dependent.ind]
check.sigma	logical; if `TRUE`, the scatter matrix is checked for appropriateness (symmetry, positive-definiteness). This could be set to `FALSE` if the user knows it is appropriate.
type	type of the noncentral multivariate t-distribution. `type = "Kshirsagar"` corresponds to formula (1.4) in Genz and Bretz (2009) (see also Chapter 5.1 in Kotz and Nadarajah (2004)). This is the noncentral t-distribution needed for calculating the power of multiple contrast tests under a normality assumption. `type = "shifted"` corresponds to the formula right before formula (1.4) in Genz and Bretz (2009) (see also formula (1.1) in Kotz and Nadarajah (2004)). It is a location shifted version of the central t-distribution. This noncentral multivariate t-distribution appears for example as the Bayesian posterior distribution for the regression coefficients in a linear regression. In the central case both types coincide.
method	string specifying the matrix decomposition used to determine the matrix root of `sigma`. Possible methods are eigenvalue decomposition (`"eigen"`, default), singular value decomposition (`"svd"`), and Cholesky decomposition (`"chol"`). The Cholesky is typically fastest, not by much though.

Value

A `'vector'` of length `n`, equal to the length of `'mean'`

Examples

```
# 10-dimensional multivariate t distribution
n <- 10
df=3
A <- matrix(rt(n^2,df), n, n)
A <- tcrossprod(A,A) #A %*% t(A)

# density of Z[c(2,5)] given Z[c(1,4,7,9)]=c(1,1,0,-1)
dcmvt(x=c(1.2,-1), mean=rep(1,n), sigma=A, df=df,
         dependent.ind=c(2,5), given.ind=c(1,4,7,9),
         X.given=c(1,1,0,-1))

dcmvt(x=-1, mean=rep(1,n), sigma=A,df=df, dep=3, given=c(1,4,7,9,10), X=c(1,1,0,0,-1))
```

```
dcmvt(x=c(1.2,-1), mean=rep(1,n), sigma=A,df=df, dep=c(2,5))

# gives an error since `x' and `dep' are incompatibe
#dcmvt(x=-1, mean=rep(1,n), sigma=A,df=df, dep=c(2,3),
#given=c(1,4,7,9,10), X=c(1,1,0,0,-1))

rcmvt(n=10, mean=rep(1,n), sigma=A,df=df, dep=c(2,5),
        given=c(1,4,7,9,10), X=c(1,1,0,0,-1),type="shifted",
        method="eigen")

rcmvt(n=10, mean=rep(1,n), sigma=A,df=df, dep=3,
        given=c(1,4,7,9,10), X=c(1,1,0,0,-1),type="Kshirsagar",
        method="chol")
```

SMCEM_msteps *Data Imputation Using SEM and MCEM (Multiple Iterations, Degrees of Freedom Known)*

Description

This sub-package contains the subroutines for iterative imputation of missing values as well as parameter estimation (for the location vector and the scatter matrix) in multivariate t distribution using Stochastic EM (SEM) and Monte Carlo EM (MCEM). In this case, the degrees of freedom for the distribution are known or fixed a priori. SEM is implemented when the analyst specifies a single draw in the E-step. In case we have multiple draws in the E-step, the algorithm changes to MCEM. In both algorithms, the function SMCEM_onestep is run when we are only interested in the imputed values and the parameter updates in a single iteration. The function SMCEM_msteps is run when we are interested in multiple iterations (this is usually the case). Essentially, the first iterations (for instance, 10 percent of all iterations) is usually burnt-in in order to ward off the effects of initial values. Details of how SEM and MCEM operate can be found in among others Kinyanjui et al. (2021), Nielsen (2000), Levine and Casella (2001) Jank (2005) and Karimi et al. (2019).

Usage

```
SMCEM_msteps(Y,mu,Sigma,df, nob,K)
```

Arguments

Y	the multivariate t dataset
mu	the location vector, which must be specified. In cases where it is unknown, starting values are provided.
Sigma	scatter matrix, which must be specified. In cases where it is unknown, starting values are provided.
df	degrees of freedom, which must be specified.
nob	number of draws in the E-step
K	the number of iterations, which must be specified.

Value

Completed dataset, updated location vector, and scatter matrix when employing the SEM and MCEM algorithms. All outputs are numeric.

References

Karimi, B., Lavielle, M., and Moulines, É. (2019). On the Convergence Properties of the Mini-Batch EM and MCEM Algorithms.

Kinyanjui, P.K., Tamba, C.L., & Okenye, J.O. (2021). Missing Data Imputation in a t -Distribution with Known Degrees of Freedom Via Expectation Maximization Algorithm and Its Stochastic Variants. International Journal of Applied Mathematics and Statistics.

Levine, R. A. and Casella, G. (2001). Implementations of the Monte Carlo EM algorithm. Journal of Computational and Graphical Statistics, 10(3), 422-439.

Nielsen, S.F. (2000). The stochastic EM algorithm: estimation and asymptotic results. Bernoulli, 6(3), 457-489.

Examples

```
# 3-dimensional multivariate t distribution
n <- 10
p=3
df=3
mu=c(1:3)
A <- matrix(rt(p^2,df), p, p)
A <- tcrossprod(A,A) #A %*% t(A)

Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
Y8= MISS(TT,20) #The newly created incomplete dataset.
Y8

#Initializing Values
mu_stat=c(0.5,1,2)
Sigma_stat=matrix(c(0.33,0.31,0.3,0.31,0.335,0.295,0.3,0.295,0.32),3,3)

#Imputing Missing Values and Updating Parameter Estimates
#Single Iteration (SEM)
SEM1=SMCEM_onestep(Y=Y8,mu= mu_stat,Sigma=Sigma_stat,df=df,nob=1)

#Single Iteration (MCEM)
MCEM1=SMCEM_onestep(Y=Y8,mu= mu_stat,Sigma=Sigma_stat,df=df,nob=100)

#Multiple Iterations (SEM)
SEM=SMCEM_msteps(Y=Y8,mu= mu_stat,Sigma= Sigma_stat,df=df,nob=1,K=500)

#Results for Newly Completed Dataset (Burning in first 50 iterations in SEM)
T_mu=rep(0,3)
```

```
T_Sigma=matrix(rep(0,3*3),nrow=3)
T_Data=matrix(rep(0,3*10), nrow =10)
for (l in 51:500){
  T_mu = T_mu + SEM$muchain[l,]
  T_Sigma = T_Sigma + SEM$SigmaChain[,,l]
  T_Data= T_Data+ SEM$YChain[,,l]
}
#updated location vector
round((T_mu/450),4)
#updated scatter matrix
round((T_Sigma/450),4)
#complete dataset as an average of (K-50) complete datasets for the various iterations.
T_Data1=  T_Data/450
T_Data1

#Multiple Iterations (MCEM)
MCEM=SMCEM_msteps(Y=Y8,mu=mu_stat,Sigma=Sigma_stat,df=df,nob=100,
K=500)

#Results for Newly Completed Dataset (Burning in first 50 iterations in MCEM)
T_mu=rep(0,3)
T_Sigma=matrix(rep(0,3*3),nrow=3)
T_Data=matrix(rep(0,3*10), nrow =10)
for (l in 51:500){
  T_mu = T_mu + MCEM$muchain[l,]
  T_Sigma = T_Sigma + MCEM$SigmaChain[,,l]
  T_Data= T_Data+ MCEM$YChain[,,l]
}
#updated location vector
round((T_mu/450),4)
#updated scatter matrix
round((T_Sigma/450),4)
#complete dataset as an average of (K-50) complete datasets for the various iterations.
T_Data1= T_Data/450
T_Data1
```

`SMCEM_onestep` *Data Imputation Using SEM and MCEM (Single Iteration, Degrees of Freedom Known)*

Description

This sub-package contains the subroutines for iterative imputation of missing values as well as parameter estimation (for the location vector and the scatter matrix) in multivariate t distribution using Stochastic EM (SEM) and Monte Carlo EM (MCEM). In this case, the degrees of freedom for the distribution are known or fixed a priori. SEM is implemented when the analyst specifies a single draw in the E-step. In case we have multiple draws in the E-step, the algorithm changes to MCEM. In both algorithms, the function SMCEM_onestep is run when we are only interested in the imputed values and the parameter updates in a single iteration.

Usage

```
SMCEM_onestep(Y,mu,Sigma,df,nob)
```

Arguments

Y	the multivariate t dataset
mu	the location vector, which must be specified. In cases where it is unknown, starting values are provided.
Sigma	scatter matrix, which must be specified. In cases where it is unknown, starting values are provided.
df	degrees of freedom, which must be specified.
nob	number of draws in the E-step

Value

Completed dataset, updated location vector, and scatter matrix when employing the SEM and MCEM algorithms. All outputs are numeric.

Examples

```
# 3-dimensional multivariate t distribution
n <- 10
p=3
df=3
mu=c(1:3)
A <- matrix(rt(p^2,df), p, p)
A <- tcrossprod(A,A) #A %*% t(A)

Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
Y8= MISS(TT,20) #The newly created incomplete dataset.

#Initializing Values
mu_stat=c(0.5,1,2)
Sigma_stat=matrix(c(0.33,0.31,0.3,0.31,0.335,0.295,0.3,0.295,0.32),3,3)

#Imputing Missing Values and Updating Parameter Estimates
#Single Iteration (SEM)
SEM1=SMCEM_onestep(Y=Y8,mu= mu_stat,Sigma=A,df=df,nob=1)

#Single Iteration (MCEM)
MCEM1=SMCEM_onestep(Y=Y8,mu= mu_stat,Sigma=A,df=df,nob=100)

#Results for Newly Completed Dataset (SEM)
SEM1$Y2    #Newly completed Dataset (with imputed values)
SEM1$mu    #updated location vector
SEM1$Sigma #updated scatter matrix
```

```
#Results for Newly Completed Dataset (MCEM)
MCEM1$Y2    #Newly completed Dataset (with imputed values)
MCEM1$mu    #updated location vector
MCEM1$Sigma #updated scatter matrix
```

SMCEM_Umsteps — *Data Imputation Using SEM and MCEM (Multiple Iterations; Degrees of Freedom Unknown)*

Description

This sub-package provides subroutines for implementation of SEM and MCEM techniques in imputing missing values as well as estimating multivariate t parameters when the degrees of freedom are unknown.The functions SMCEM_msteps constitute the SEM and MCEM algorithms for multiple-iterative data imputation and parameter estimation for multivariate t data with unknown degrees of freedom. The functions represent SEM when the number of draws in the E-step (denoted by nob) is 1 and MCEM when we have more than one draw in the E-step.More details on the implementation of SEM and MCEM techniques can be found in Kinyanjui et al. (2020).

Usage

```
SMCEM_Umsteps(Y,mu,Sigma,df,nob,K,e)
```

Arguments

Y	the multivariate t dataset
mu	the location vector, which must be specified. In cases where it is unknown, starting values are provided.
Sigma	scatter matrix, which must be specified. In cases where it is unknown, starting values are provided.
df	degrees of freedom, which must be specified.
nob	number of draws in the E-step
K	the number of iterations, which must be specified.
e	tolerance level for convergence of the bisection method for estimation of df.

Value

Completed dataset, updated location vector,scatter matrix, and degrees of freedom when employing the SEM and MCEM algorithms. All outputs are numeric.

References

Kinyanjui, P. K., Tamba, C. L., Orawo, L. A. O., & Okenye, J. O. (2020). Missing data imputation in multivariate t distribution with unknown degrees of freedom using expectation maximization algorithm and its stochastic variants. Model Assisted Statistics and Applications, 15(3), 263-272.

Examples

```
# 3-dimensional multivariate t distribution
n <- 25
p=3
df=3
mu=c(10,20,30)
A=matrix(c(14,10,12,10,13,9,12,9,18), 3,3)
Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
Y8= MISS(TT,20) #The newly created incomplete dataset.
Y8

#Initializing Values
mu_stat=c(0.5,1,2)
Sigma_stat=matrix(c(0.33,0.31,0.3,0.31,0.335,0.295,0.3,0.295,0.32),3,3)
df_stat=6

#Imputing Missing Values and Updating Parameter Estimates

#Single Iteration (SEM)
SEMU1=SMCEM_Uonestep(Y=Y8,mu=mu,Sigma=Sigma_stat,df= df_stat,nob=1,e=0.0001)

#Single Iteration (MCEM)
MCEMU1=SMCEM_Uonestep(Y=Y8,mu=mu,Sigma=Sigma_stat,df= df_stat,nob=50,e=0.0001)

#Multiple Iterations (SEM)
SEMU=SMCEM_Umsteps(Y=Y8,mu=mu_stat,Sigma=Sigma_stat,df=df_stat,nob=1,K=100,e=0.0001)

#Results for Newly Completed Dataset (Burning in first 10 iterations in SEM)
T_mu=rep(0,3)
T_Sigma=matrix(rep(0,3*3),nrow=3)
T_Data=matrix(rep(0,3*25), nrow =25)
T_df=rep()
for (l in 11:100){
 T_mu = T_mu + SEMU$muchain[l,]
  T_Sigma = T_Sigma + SEMU$SigmaChain[,,l]
 T_Data= T_Data+ SEMU$YChain[,,l]
}
#updated location vector
round((T_mu/90),4)

#updated scatter matrix
round((T_Sigma/90),4)

#updated degrees of freedom
udfs=mean(SEMU$dfchain[11:100])

#complete dataset as an average of (K-10) complete datasets for the various iterations.
T_Data1=  T_Data/90
```

```
#Multiple Iterations (MCEM)
MCEMU=SMCEM_Umsteps(Y=Y8,mu=mu_stat,Sigma=Sigma_stat,df=df_stat,nob=50,K=100,e=0.0001)

#Results for Newly Completed Dataset (Burning in first 10 iterations in MCEM)
T_mu=rep(0,3)
T_Sigma=matrix(rep(0,3*3),nrow=3)
T_Data=matrix(rep(0,3*25), nrow =25)
T_df=rep()
for (l in 11:100){
  T_mu = T_mu + MCEMU$muchain[l,]
  T_Sigma = T_Sigma + MCEMU$SigmaChain[,,l]
  T_Data= T_Data+ MCEMU$YChain[,,l]
}
#updated location vector
round((T_mu/90),4)

#updated scatter matrix
round((T_Sigma/90),4)

#updated degrees of freedom
udf=mean(MCEMU$dfchain[11:100])
udf

#complete dataset as an average of (K-10) complete datasets for the various iterations.
T_Data1=  T_Data/90
T_Data1
```

SMCEM_Uonestep   *Data Imputation Using SEM and MCEM (Single Iteration; Degrees of Freedom Unknown)*

Description

This sub-package provides subroutines for implementation of SEM and MCEM techniques in imputing missing values as well as estimating multivariate t parameters when the degrees of freedom are unknown. It has 4 functions namely fun1, dfun1, Bisec, and SMCEM_Uonestep. The functions fun1 and dfun1 in the sub-package constitute the equation for the degrees of freedom and its derivative respectively. The Bisec function contains the bisection method subroutines to facilitate the iterative estimation of the degrees of freedom using fun1 and dfun1. The function SMCEM_Uonestep constitute the SEM and MCEM algorithms for single-iteration data imputation and parameter estimation for multivariate t data with unknown degrees of freedom. The functions represent SEM when the number of draws in the E-step (denoted by nob) is 1 and MCEM when we have more than one draw in the E-step.Details of how SEM and MCEM impute missing values and estimate parameters in multivariate t context (unknown degrees of freedom) are explained by Kinyanjui et al. (2020).

Usage

```
SMCEM_Uonestep(Y,mu,Sigma,df,nob,e)
```

Arguments

Y	the multivariate t dataset
mu	the location vector, which must be specified. In cases where it is unknown, starting values are provided.
Sigma	scatter matrix, which must be specified. In cases where it is unknown, starting values are provided.
df	degrees of freedom, which must be specified.
nob	number of draws in the E-step
e	tolerance level for convergence of the bisection method for estimation of df.

Value

Completed dataset, updated location vector,scatter matrix, and degrees of freedom when employing the SEM and MCEM algorithms. All outputs are numeric.

References

Kinyanjui, P. K., Tamba, C. L., Orawo, L. A. O., & Okenye, J. O. (2020). Missing data imputation in multivariate t distribution with unknown degrees of freedom using expectation maximization algorithm and its stochastic variants. Model Assisted Statistics and Applications, 15(3), 263-272.

Examples

```
# 3-dimensional multivariate t distribution
n <- 25
p=3
df=3
mu=c(10,20,30)
A=matrix(c(14,10,12,10,13,9,12,9,18), 3,3)
Y7 <-mvtnorm::rmvt(n, delta=mu, sigma=A, df=df)
Y7
TT=Y7 #Complete Dataset

#Introduce MAR Data
Y8= MISS(TT,20) #The newly created incomplete dataset.
Y8

#Initializing Values
mu_stat=c(0.5,1,2)
Sigma_stat=matrix(c(0.33,0.31,0.3,0.31,0.335,0.295,0.3,0.295,0.32),3,3)
df_stat=6

#Imputing Missing Values and Updating Parameter Estimates

#Single Iteration (SEM)
SEMU1=SMCEM_Uonestep(Y=Y8,mu=mu,Sigma=Sigma_stat,df= df_stat,nob=1,e=0.00001)

#Single Iteration (MCEM)
MCEMU1=SMCEM_Uonestep(Y=Y8,mu=mu,Sigma=Sigma_stat,df= df_stat,nob=1000,e=0.00001)
```

```
#Results for Newly Completed Dataset (SEM)
SEMU1$Y2     #Newly completed Dataset (with imputed values)
SEMU1$mu    #updated location vector
SEMU1$Sigma  #updated scatter matrix

#Results for Newly Completed Dataset (MCEM)
MCEMU1$Y2     #Newly completed Dataset (with imputed values)
MCEMU1$mu     #updated location vector
MCEMU1$Sigma  #updated scatter matrix
MCEMU1$df     #updated degrees of freedom
```

Index