

Package ‘EMMIXskew’

February 13, 2018

Title The EM Algorithm and Skew Mixture Distribution

Description EM algorithm for Fitting Mixture of Multivariate Skew Normal and Skew t Distributions. An implementation of the algorithm described in Wang, Ng, and McLachlan (2009) <doi:10.1109/DICTA.2009.88>.

Imports lattice, mvtnorm, KernSmooth

Depends graphics, stats, grDevices

Version 1.0.3

Author Kui Wang, Angus Ng and Geoff McLachlan.

Maintainer Sharon Lee <s.lee11@uq.edu.au>

License GPL

LazyLoad yes

Date 2018-02-13

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-02-13 04:19:56 UTC

R topics documented:

bootstrap	2
conplot	4
ddmix	6
ddmsn	7
ddmst	8
ddmvn	9
ddmvt	10
EmSkew	11
EmSkew.contours	15
EmSkewfit	16
EmSkewMOD	19
error.rate	20
getcov	21
getICL	21

getSWR	23
initEmmix	24
inverse	26
mvt.dof	26
rdemmix	27

Index	30
--------------	-----------

bootstrap	<i>Bootstrap</i>
-----------	------------------

Description

The standard error analysis and the bootstrap analysis of $-2\log(\text{Lambda})$.

Usage

```
bootstrap(x,n,p,g,distr,ncov,popPAR,B=99,replace=TRUE,
itmax=1000,epsilon=1e-5)
bootstrap.noc(x,n,p,g1,g2,distr,ncov,B=99,replace=TRUE,
itmax=1000,epsilon=1e-5)
```

Arguments

n	The number of observations
p	The dimension of data
B	The number of simulated data or replacements to be tried
x	The dataset, an n by p numeric matrix, where n is number of observations and p the dimension of data.
g	The number of components of the mixture model
g1,g2	The range of the number of components of the mixture model
distr	A three letter string indicating the type of distribution to be fit. See Details.
ncov	A small integer indicating the type of covariance structure. See Details.
popPAR	A list with components pro, a numeric vector of the mixing proportion of each component; mu, a p by g matrix with each column as its corresponding mean; sigma, a three dimensional p by p by g array with its jth component matrix (p,p,j) as the covariance matrix for jth component of mixture models; dof, a vector of degrees of freedom for each component; delta, a p by g matrix with its columns corresponding to skew parameter vectors.
replace	A logical value indicating whether replacement to be used
itmax	A big integer specifying the maximum number of iterations to apply
epsilon	A small number used to stop the EM algorithm loop when the relative difference between log-likelihood at each iteration become sufficient small.

Details

The distribution type, `distr`, is one of the following values: "mvn" for a multivariate normal, "mvt" for a multivariate t-distribution, "msn" for a multivariate skew normal distribution and "mst" for a multivariate skew t-distribution.

The covariance matrix type, represented by the `ncov` parameter, may be any one of the following: `ncov=1` for a common variance, `ncov=2` for a common diagonal variance, `ncov=3` for a general variance, `ncov=4` for a diagonal variance, `ncov=5` for $\sigma(h) \cdot I(p)$ (diagonal covariance with same identical diagonal element values).

When `replace` is FALSE, parametric bootstrap is used; otherwise replacement method is used.

Value

`bootstrap` gives standard errors. `bootstrap.noc` returns a list with components `ret`, a B by (g2-g1) matrix of $-2\log(\text{Lambda})$, `v1k`, the loglikelihood for each g in the range of g1 to g2, and `pvalue`, the p-values of g vs g+1. The results of fitting mixture models are stored in current working directory, which can be used via command in R: `obj <- dget("ReturnOf_g_???.ret")`.

References

McLachlan G.J. and Krishnan T. (2008). The EM Algorithm and Extensions (2nd). New Jersey: Wiley.

McLachlan G.J. and Peel D. (2000). Finite Mixture Models. New York: Wiley.

See Also

[EmSkew,rdemmix](#)

Examples

```
n1=300;n2=300;n3=400;
nn <-c(n1,n2,n3)
n <- sum(nn)
p <- 2
g <- 3

sigma<-array(0,c(p,p,g))
for(h in 1:3) sigma[,,h]<-diag(p)

mu <- cbind(c(4,-4),c(3.5,4),c( 0, 0))

# for other distributions,
#delta <- cbind(c(3,3),c(1,5),c(-3,1))
#dof <- c(3,5,5)

distr="mvn"
ncov=3

#first we generate a data set
```

```

set.seed(111) #random seed is set
dat <- rdemmix(nn,p,g,distr,mu,sigma,dof=NULL,delta=NULL)

#start from initial partition
clust<- rep(1:g,nn)
obj <- EmSkewfit1(dat,g,clust,distr,ncov,itmax=1000,epsilon=1e-5)

# do bootstrap (stadard error analysis)

## Not run:
std <- bootstrap(dat,n,p,g,distr,ncov,obj,B=19,
replace=TRUE,itmax=1000,epsilon=1e-5)
print(std)

# do booststrap analysis of -2log(Lambda).

# alternatively data can be input as follow,
# dat <- read.table("mydata.txt",header=TRUE)
# p   <- ncol(dat)
# n   <- nrow(dat)

lad <- bootstrap.noc(dat,n,p,2,4,distr,ncov,B=19,
replace=FALSE,itmax=1000,epsilon=1e-5)
print(lad)

# return of g=2
obj2 <- dget("ReturnOf_g_2.ret")

# return of g=3
obj3 <- dget("ReturnOf_g_3.ret")

# return of g=4
obj4 <- dget("ReturnOf_g_4.ret")

#The posterior probability matrix for (g=3) is obtained by

tau <- obj3$tau

## End(Not run)

```

conplot

Functions of Contours

Description

These functions are called by `EmSkew.contours`, `EmSkew.filter` and `EmSkew.flow` to plot the contours of (skew) mixture density after fitting to the data.

Usage

```

conplot(x, y, pro, mu, sigma, dof, delta, distr, grid = 300,
        nrand = 6000, levels = seq(5, 95, by = 20), col = "white")
conplot2(x, y, pro, mu, sigma, dof, delta, distr, grid = 300,
        nrand = 6000, levels = seq(5, 95, by = 20))
conplot3(x, y, pro, mu, sigma, dof, delta, modpts, distr, grid = 300,
        nrand = 10000, levels = seq(5, 95, by = 20))
mypanel2(x, y, ...)
mypanel3(x, y, ...)
mypanel4(x, y, ...)
panel.density(x, col=1, ...)

```

Arguments

x	A vector of observations on variable x.
y	A vector of observations on variable y.
pro	A vector of mixing proportions in the (skew) mixture model.
mu	A matrix with each column corresponding to the mean or location vector of one mixture component.
sigma	An array of covariance matrices for each component of the mixture distribution.
dof	A vector of degrees of freedom when "distr"ibution is "mvt" or "mst".
delta	A matrix with each column as skew parameter vector of one component when "distr"ibution is "msn" or "mst".
distr	A three letter string indicating component distribution, "mvn"=normal distribution, "mvt"=t-distribution, "msn"=skew normal distribution, "mst"=skew t-distribution.
modpts	The mode points.
grid	An integer for the number of grid points in one direction.
nrand	A large integer for the number of random numbers being drawn.
levels	A vector of contour percentage levels for the plots. It should be in the range of 0 to 100.
col	The colour of contour lines.
...	other

Details

In most case, users do not call this function directly, instead they call the function `emmix.flow`.

See Also

[EmSkew.flow](#)

`ddmix`*Density Functions of Mixture Models*

Description

Calculate the density of multivariate mixture models at data points for each component

Usage

```
ddmix( dat, n, p, g, distr, mu, sigma, dof=NULL, delta=NULL)
```

Arguments

<code>dat</code>	The dataset
<code>n</code>	The total number of points
<code>p</code>	Dimension of data
<code>g</code>	The number of clusters
<code>distr</code>	A three letter string indicating the distribution; "mvn" for normal, "mvt" for t distribution, "msn" for skew normal, and "mst" for skew t distribution.
<code>mu</code>	A numeric mean matrix with each column corresponding to the mean
<code>sigma</code>	An array of dimension (p,p,g) with first two dimensions corresponding covariance matrix of each component
<code>dof</code>	A vector of degrees of freedom for each component
<code>delta</code>	A matrix with each column as skew parameter vector

Value

`ddmix` gives an `n` by `g` matrix of logarithm of density at each data point for each component.

References

McLachlan G.J. and Krishnan T. (2008). The EM Algorithm and Extensions (2nd). New Jersey: Wiley.

McLachlan G.J. and Peel D. (2000). Finite Mixture Models. New York: Wiley.

See Also

[ddmvn](#), [ddmvt](#), [ddmsn](#), [ddmst](#).

Examples

```

p=2
g=3

#mixing propotion of each component
pro  <- c(0.3,0.3,0.4)

#specify mean and covariance matrix for each component

sigma<-array(0,c(2,2,3))
for(h in 2:3) sigma[,,h]<-diag(2)
sigma[,,1]<-cbind( c(1,0),c(0,.1))

mu  <- cbind(c(4,-4),c(3.5,4),c( 0, 0))

#specify other parameters for "mvt","msn","mst"

delta <- cbind(c(3,3),c(1,5),c(-3,1))
dof   <- c(3,5,5)

#specify the distribution
distr <- "mst"

y <- c(1,2)

n=1

#then the density value at y for the mixture model is

ddmix(y, n, p, g, distr, mu, sigma, dof, delta)

```

Description

Density and random generation for Multivariate Skew Normal distributions with mean vector mean, covariance matrix cov, and skew parameter vector del.

Usage

```

ddmsn(dat,n, p, mean, cov, del)
rdmsn(  n, p, mean, cov, del)

```

Arguments

dat	An n by p numeric matrix, the dataset
n	An integer, the number of observations
p	An integer, the dimension of data
mean	A length of p vector, the mean
cov	A p by p matrix, the covariance
del	A length of p vector, the skew parameter

Value

ddmsn gives the density values; rdmsn generates the random numbers

See Also

[rdemmix](#), [ddmvn](#), [ddmvt](#), [ddmst](#), [rdmvn](#), [rdmvt](#), [rdmst](#).

Examples

```
n <- 100
p <- 2

mean <- rep(0,p)
cov <- diag(p)
del<- c(0,1)

set.seed(3214)

y <- rdmsn( n,p,mean,cov,del)

den <- ddmsn(y,n,p,mean,cov,del)
```

ddmst

The Multivariate Skew t-distribution

Description

Density and random generation for Multivariate Skew t-distributions with mean vector mean, covariance matrix cov, degrees of freedom nu, and skew parameter vector del.

Usage

```
ddmst(dat,n, p, mean, cov, nu, del)
rdmst( n, p, mean, cov, nu, del)
```


Arguments

dat	An n by p numeric matrix, the dataset
n	An integer, the number of observations
p	An integer, the dimension of data
mean	A length of p vector, the mean
cov	A p by p matrix, the covariance
nu	A positive number, the degrees of freedom
del	A length of p vector, the skew parameter

Value

ddmst gives the density values; rdmst generates the random numbers

See Also

[rdemmix](#), [ddmvn](#), [ddmvt](#), [ddmsn](#), [rdmvn](#), [rdmvt](#), [rdmsn](#).

Examples

```
n <- 100
p <- 2

mean <- rep(0,p)
cov <- diag(p)
nu <- 3
del <- c(0,1)

set.seed(3214)

y <- rdmst( n,p,mean,cov,nu,del)

den <- ddmst(y,n,p,mean,cov,nu,del)
```

Description

Density and random generation for Multivariate Normal distributions with mean vector mean, and covariance matrix cov.

Usage

```
ddmvn(dat,n, p, mean, cov)
rdmvn( n, p, mean, cov)
```

Arguments

dat	An n by p numeric matrix, the dataset
n	An integer, the number of observations
p	An integer, the dimension of data
mean	A length of p vector, the mean
cov	A p by p matrix, the covariance

Value

ddmvn gives the density values; rdmvn generates the random numbers

See Also

[rdemmix](#), [ddmvt](#), [ddmsn](#), [ddmst](#), [rdmvt](#), [rdmsn](#), [rdmst](#).

Examples

```
n <- 100
p <- 2

mean <- rep(0,p)
cov <- diag(p)

set.seed(3214)

y <- rdmvn( n,p,mean,cov)

den <- ddmvn(y,n,p,mean,cov)
```

ddmvt

The Multivariate t-Distribution

Description

Density and random generation for Multivariate t-distributions with mean vector mean, covariance matrix cov, and degrees of freedom nu.

Usage

```
ddmvt(dat,n, p, mean, cov, nu)
rdmvt( n, p, mean, cov, nu)
```

Arguments

<code>dat</code>	An n by p numeric matrix, the dataset
<code>n</code>	An integer, the number of observations
<code>p</code>	An integer, the dimension of data
<code>mean</code>	A length of p vector, the mean
<code>cov</code>	A p by p matrix, the covariance
<code>nu</code>	A positive number, the degrees of freedom

Value

`ddmvt` gives the density values; `rdmvt` generates the random numbers

See Also

[rdemmix](#), [ddmvn](#), [ddmsn](#), [ddmst](#), [rdmvn](#), [rdmsn](#), [rdmst](#).

Examples

```
n <- 100
p <- 2

mean <- rep(0,p)
cov <- diag(p)
nu <- 3

set.seed(3214)

x <- rdmvt( n,p,mean,cov,nu)

den <- ddmvt(x ,n,p,mean,cov,nu)
```

Description

As a main function, `EmSkew` fits the data into the specified multivariate mixture models via the EM Algorithm. Distributions (univariate and multivariate) available include Normal distribution, t-distribution, Skew Normal distribution, and Skew t-distribution.

Usage

```
EmSkew(dat, g, distr="mvn", ncov=3, clust=NULL, init=NULL, itmax=1000,
epsilon=1e-6, nkmeans=0, nrandom=10, nhclust=FALSE, debug=TRUE,
initloop=20)
```

Arguments

<code>dat</code>	The dataset, an n by p numeric matrix, where n is number of observations and p the dimension of data.
<code>g</code>	The number of components of the mixture model
<code>distr</code>	A three letter string indicating the type of distribution to be fitted, the default value is "mvn", the Normal distribution. See Details.
<code>ncov</code>	A small integer indicating the type of covariance structure; the default value is 3. See Details.
<code>clust</code>	A vector of integers specifying the initial partitions of the data; the default is NULL.
<code>init</code>	A list containing the initial parameters for the mixture model. See details. The default value is NULL.
<code>itmax</code>	A big integer specifying the maximum number of iterations to apply; the default value is 1000.
<code>epsilon</code>	A small number used to stop the EM algorithm loop when the relative difference between log-likelihood at each iteration become sufficient small; the default value is $1e-6$.
<code>nkmeans</code>	An integer to specify the number of KMEANS partitions to be used to find the best initial values; the default value is 0.
<code>nrandom</code>	An integer to specify the number of random partitions to be used to find the best initial values; the default value is 10.
<code>nhclust</code>	A logical value to specify whether or not to use hierarchical cluster methods; the default is FALSE. If TRUE, the Complete Linkage method will be used.
<code>debug</code>	A logical value, if it is TRUE, the output will be printed out; FALSE silent; the default value is TRUE.
<code>initloop</code>	A integer specifying the number of initial loops when searching the best initial partitions.

Details

The distribution type, determined by the `distr` parameter, which may take any one of the following values: "mvn" for a multivariate normal, "mvt" for a multivariate t-distribution, "msn" for a multivariate skew normal distribution and "mst" for a multivariate skew t-distribution.

The covariance matrix type, represented by the `ncov` parameter, may be any one of the following: `ncov=1` for a common variance, `ncov=2` for a common diagonal variance, `ncov=3` for a general variance, `ncov=4` for a diagonal variance, `ncov=5` for $\sigma(h) * I(p)$ (diagonal covariance with same identical diagonal element values).

The parameter `init` requires following elements: `pro`, a numeric vector of the mixing proportion of each component; `mu`, a p by g matrix with each column as its corresponding mean; `sigma`, a three dimensional p by p by g array with its j th component matrix (p,p,j) as the covariance matrix for j th component of mixture models; `dof`, a vector of degrees of freedom for each component; `delta`, a p by g matrix with its columns corresponding to skew parameter vectors.

Since we treat the list of `pro`, `mu`, `sigma`, `dof`, and `delta` as a common structure of parameters for our mixture models, we need to include all of them in the initial parameter list `init` by default

although in some cases it does not make sense, for example, `dof` and `delta` is not applicable to normal mixture model. But in most cases, the user only need give relevent paramters in the list.

When the parameter list `init` is given, the program ignores both initial partition `clust` and automatic partition methods such as `nkmeans`; only when both `init` and `clust` are not available, the program uses automatic approaches such as k-Means partition method to find the best inital values. All three automatic approaches are used to find the best initial partition and initial values if required.

The return values include all potential parameters `pro`,`mu`,`sigma`,`dof`,and `delta`, but user should not use or interpret irrelevant information arbitrarily. For example, `dof` and `delta` for Normal mixture models.

Value

<code>error</code>	Error code, 0 = normal exit; 1 = did not converge within <code>itmax</code> iterations; 2 = failed to get the initial values; 3 = singularity
<code>aic</code>	Akaike Information Criterion (AIC)
<code>bic</code>	Bayes Information Criterion (BIC)
<code>ICL</code>	Integrated Completed Likelihood Criterion (ICL)
<code>pro</code>	A vector of mixing proportions.
<code>mu</code>	A numeric matrix with each column corresponding to the mean.
<code>sigma</code>	An array of dimension (p,p,g) with first two dimension corresponding covariance matrix of each component.
<code>dof</code>	A vector of degrees of freedom for each component, see Details.
<code>delta</code>	A p by g matrix with each column corresponding to a skew parameter vector.
<code>clust</code>	A vector of final partition
<code>loglik</code>	The log likelihood at convergence
<code>lk</code>	A vector of log likelihood at each EM iteration
<code>tau</code>	An n by g matrix of posterior probability for each data point

References

- Biernacki C. Celeux G., and Govaert G. (2000). Assessing a Mixture Model for Clustering with the integrated Completed Likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22(7). 719-725.
- McLachlan G.J. and Krishnan T. (2008). *The EM Algorithm and Extensions* (2nd). New Jersay: Wiley.
- McLachlan G.J. and Peel D. (2000). *Finite Mixture Models*. New York: Wiley.

See Also

[initEmmix,rdemmix](#).

Examples

```

#define the dimension of dataset

n1=300;n2=300;n3=400;
nn<-c(n1,n2,n3)

p <- 2
ng <- 3

#define the parameters
sigma<-array(0,c(2,2,3))
for(h in 2:3) sigma[,,h]<-diag(2)
sigma[,,1]<-cbind( c(1,0.2),c(0.2,1))
mu      <- cbind(c(4,-4),c(3.5,4),c( 0, 0))

#and other parameters if required for "mvt","msn","mst"
delta  <- cbind(c(3,3),c(1,5),c(-3,1))
dof    <- c(3,5,5)

pro    <- c(0.3,0.3,0.4)

distr="mvn"
ncov=3

# generate a data set

set.seed(111) #random seed is reset

dat <- rdemmix(nn,p,ng,distr,mu,sigma)

# the following code can be used to get singular data (remarked off)
# dat[1:300,2]<--4
# dat[300+1:300,1]<-2
## dat[601:1000,1]<-0
## dat[601:1000,2]<-0

#fit the data using KMEANS to get the initial partitions (10 trials)
obj <- EmSkew(dat,ng,distr,ncov,itmax=1000,epsilon=1e-5,nkmeans=10)

# alternatively, if we define initial values like
initobj<-list()

initobj$pro <- pro
initobj$mu  <- mu
initobj$sigma<- sigma

```

```

initobj$dof <- dof
initobj$delta<- delta

# then we can fit the data from initial values
obj <- EmSkew(dat,ng,distr,ncov,init=initobj,itmax=1000,epsilon=1e-5)

# finally, if we know initial partition such as
clust      <- rep(1:ng,nn)

# then we can fit the data from given initial partition
obj <- EmSkew(dat,ng,distr,ncov,clust=clust,itmax=1000,epsilon=1e-5)

# plot the 2D contours

colnames(dat)<- paste("x",1:p,sep='')

# dev.new()
EmSkew.flow(dat,obj)

```

EmSkew.contours

Scatter plot and Contours

Description

Contour of fitted mixture density can be an important indicator of goodness-of-fit. Together with the heatmaps, this function provides a general idea of how well the mixture model fits to the data.

Usage

```

EmSkew.flow(S, obj = NULL, distr="",diag.panel=TRUE,upper.panel="type2",
lower.panel = "type3", levels=seq(5, 95,by=20),attop=FALSE,clust=NULL,
title="",path="",plot=TRUE)
EmSkew.contours(S, obj = NULL, clust = NULL,distr="",diag.panel=TRUE,
upper.panel="type2",lower.panel="type3",levels=seq(5,95,by=20),
plot=TRUE,title="",path='',attop =FALSE)

EmSkew.filter(S, g=1,distr="mst",diag.panel=TRUE,upper.panel="type2",
lower.panel = "type3", levels = 90, attop = FALSE,
title="",path="",plot=TRUE)

```

Arguments

g The number of components to filter the data at FSC and SSC channels.

S A dataframe.

obj	A list including the parameters of a (skew) mixture model.
distr	A three letter string specifying the distribution.
diag.panel	A logical value, plot density.
upper.panel	A string for the panel to be used in upper triangle.
lower.panel	A string for the panel to be used in lower triangle.
levels	A vector of contour percentage levels for the plots. It should be in the range of 0 to 100.
atop	A logical value indicating the direction of diagonal panels.
clust	A vector with the cluster labels for each observation of the sample.
title	The png file name.
path	The path to the folder where plots are stored.
plot	A logical variable, whether plot it in the windows.

Details

R package `geneplotter` is required.

Note

The input list 'obj' must be assigned a component "distr".

See Also

[conplot](#)

EmSkewfit

Fit the Multivariate Skew Mixture Models

Description

The engines to fit the data into mixture models using initial partition or initial values. `set`.

Usage

```
EmSkewfit1(dat, g, clust,          distr, ncov, itmax, epsilon,initloop=20)
EmSkewfit2(dat, g,              init, distr, ncov, itmax, epsilon)
```


Arguments

<code>dat</code>	The dataset, an n by p numeric matrix, where n is number of observations and p the dimension of data.
<code>g</code>	The number of components of the mixture model
<code>distr</code>	A three letter string indicating the type of distribution to be fit. See Details.
<code>ncov</code>	A small integer indicating the type of covariance structure. See Details.
<code>clust</code>	A vector of integers specifying the initial partitions of the data
<code>init</code>	A list containing the initial parameters for the mixture model. See details.
<code>itmax</code>	A big integer specifying the maximum number of iterations to apply
<code>epsilon</code>	A small number used to stop the EM algorithm loop when the relative difference between log-likelihood at each iteration become sufficient small.
<code>initloop</code>	A integer specifying the number of initial loops

Details

The distribution type, determined by the `distr` parameter, which may take any one of the following values: "mvn" for a multivariate normal, "mvt" for a multivariate t-distribution, "msn" for a multivariate skew normal distribution and "mst" for a multivariate skew t-distribution.

The covariance matrix type, represented by the `ncov` parameter, may be any one of the following: `ncov=1` for a common variance, `ncov=2` for a common diagonal variance, `ncov=3` for a general variance, `ncov=4` for a diagonal variance, `ncov=5` for $\sigma(h) * I(p)$ (diagonal covariance with same identical diagonal element values).

The parameter `init` is a list with elements: `pro`, a numeric vector of the mixing proportion of each component; `mu`, a p by g matrix with each column as its corresponding mean; `sigma`, a three dimensional p by p by g array with its j th component matrix (p, p, j) as the covariance matrix for j th component of mixture models; `dof`, a vector of degrees of freedom for each component; `delta`, a p by g matrix with its columns corresponding to skew parameter vectors.

Value

<code>error</code>	Error code, 0 = normal exit; 1 = did not converge within <code>itmax</code> iterations; 2 = failed to get the initial values; 3 = singularity
<code>aic</code>	Akaike Information Criterion (AIC)
<code>bic</code>	Bayes Information Criterion (BIC)
<code>pro</code>	A vector of mixing proportions, see Details.
<code>mu</code>	A numeric matrix with each column corresponding to the mean, see Details.
<code>sigma</code>	An array of dimension (p, p, g) with first two dimension corresponding covariance matrix of each component, see Details.
<code>dof</code>	A vector of degrees of freedom for each component, see Details.
<code>delta</code>	A p by g matrix with each column corresponding to a skew parameter vector, see Details.
<code>clust</code>	A vector of final partition
<code>loglik</code>	The loglikelihood at convergence
<code>lk</code>	A vector of loglikelihood at each EM iteration
<code>tau</code>	An n by g matrix of posterior probability for each data point

References

McLachlan G.J. and Krishnan T. (2008). The EM Algorithm and Extensions (2nd). New Jersey: Wiley.

McLachlan G.J. and Peel D. (2000). Finite Mixture Models. New York: Wiley.

See Also

[init.mix](#), [initEmmix](#), [EmSkew](#), [rdemmix](#), [rdemmix2](#), [rdmvn](#), [rdmvt](#), [rdmsn](#), [rdmst](#).

Examples

```
n1=300;n2=300;n3=400;
nn <-c(n1,n2,n3)
n=1000
p=2
ng=3

sigma<-array(0,c(2,2,3))
for(h in 2:3) sigma[,,h]<-diag(2)
sigma[,,1]<-cbind( c(1,0),c(0,1))
mu <- cbind(c(4,-4),c(3.5,4),c( 0, 0))

# for other distributions,
#delta <- cbind(c(3,3),c(1,5),c(-3,1))
#dof <- c(3,5,5)

pro <- c(0.3,0.3,0.4)

distr="mvn"
ncov=3

#first we generate a data set
set.seed(111) #random seed is set
dat <- rdemmix(nn,p,ng,distr,mu,sigma,dof=NULL,delta=NULL)

#start from initial partition
clust<- rep(1:ng,nn)
obj1 <- EmSkewfit1(dat, ng, clust, distr, ncov, itmax=1000, epsilon=1e-4)

#start from initial values
#alternatively, if we define initial values like

init<-list()

init$pro<-pro
```

```

init$mu<-mu
init$sigma<-sigma

# for other distributions,
#delta <- cbind(c(3,3),c(1,5),c(-3,1))
#dof   <- c(3,5,5)
#init$dof<-dof
#init$delta<-delta

obj2 <-EmSkewfit2(dat, ng, init, distr, ncov,itmax=1000, epsilon=1e-4)

```

EmSkewMOD

Calculate modes

Description

Calculate the mode points for each component of skew mixture models.

Usage

```
EmSkewMOD(p,g,distr,mu,sigma,dof,delta,nrand=10000)
```

Arguments

p	The dimension of the data
g	The number of components to be fit
distr	A three letter string of distribution id
mu	A numeric matrix with each column corresponding to the mean
sigma	An array of dimension (p,p,g) with first two dimensions corresponding covariance matrix of each component
dof	A vector of degrees of freedom for each component
delta	A matrix with each column as skew parameter vector
nrand	The number of random numbers.

Value

A p by g matrix of mode points for each component of the skew mixture model.

Examples

```

p=2
g=3

sigma<-array(0,c(2,2,3))
for(h in 2:3) sigma[,,h]<-diag(2)

```

```

sigma[,1]<-cbind( c(1,0),c(0,1))
mu <- cbind(c(4,-4),c(3.5,4),c( 0, 0))

delta <- cbind(c(3,3),c(1,5),c(-3,1))
dof <- c(3,5,5)

distr="mst"

EmSkewMOD(p,g,distr,mu,sigma,dof,delta,nrand=10000)

```

error.rate

Error Rate

Description

Calculate the Error Rate of a partition

Usage

```

error.rate(clust1,clust2)
rand.index(LabelA,LabelB)

```

Arguments

clust1	An integer vector of cluster label 1
clust2	An integer vector of cluster label 2
LabelA	An integer vector of the true membership labels
LabelB	An integer vector of the predicted labels

Details

clust1 and clust 2 must match, i.e, same number of clusters

Value

error.rate gives Error Rate

Examples

```

clu1<-c(1,2,3,1,1,2,2,3,3)
clu2<-c(2,2,2,1,1,1,3,3,3)
error.rate(clu1, clu2)
rand.index(clu1, clu2)

```

<code>getcov</code>	<i>Covariance</i>
---------------------	-------------------

Description

Recalculate covariance for a given structure.

Usage

```
getcov(msigma, sumtau, n, p, g, ncov)
```

Arguments

<code>msigma</code>	An array for the covariance matrices.
<code>sumtau</code>	A vector for the expected number of observations from each component.
<code>n</code>	An integer for the sample size.
<code>p</code>	An integer for the dimension.
<code>g</code>	An integer for the number of components.
<code>ncov</code>	An integer for type of covariance.

Value

The covariance array.

References

McLachlan G.J. and Krishnan T. (2008). *The EM Algorithm and Extensions* (2nd). Hoboken, New Jersey: Wiley.

McLachlan G.J. and Peel D. (2000). *Finite Mixture Models*. New York: Wiley.

<code>getICL</code>	<i>The ICL criterion</i>
---------------------	--------------------------

Description

Calculate the Integrated Completed Likelihood(ICL) criterion

Usage

```
getICL(x, n, p, g, distr, ncov, pro, mu, sigma, dof, delta, clust)
```

Arguments

x	An n by p data matrix
n	The total number of points
p	Dimension of data
g	the number of components of the mixture model
distr	A three letter string indicating the type of distribution to be fit.
ncov	A small integer indicating the type of covariance structure.
pro	A vector of mixing proportions
mu	A numeric matrix with each column corresponding to the mean
sigma	An array of dimension (p,p,g) with first two dimension corresponding covariance matrix of each component
dof	A vector of degrees of freedom for each component
delta	A p by g matrix with each column corresponding to a skew parameter vector
clust	A vector of partition

Value

ICL	ICL value
-----	-----------

References

Biernacki C. Celeux G., and Govaert G. (2000). Assessing a Mixture Model for Clustering with the integrated Completed Likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 22(7). 719-725.

Examples

```
n1=300;n2=300;n3=400;
nn <-c(n1,n2,n3)
n=1000
p=2
ng=3

sigma<-array(0,c(2,2,3))
for(h in 2:3) sigma[,,h]<-diag(2)
sigma[,,1]<-cbind( c(1,0),c(0,1))
mu <- cbind(c(4,-4),c(3.5,4),c( 0, 0))

pro <- c(0.3,0.3,0.4)

distr="mvn"
ncov=3

#first we generate a data set
set.seed(111) #random seed is set
dat <- rdemmix(nn,p,ng,distr,mu,sigma,dof=NULL,delta=NULL)
```

```

#start from initial partition
clust<- rep(1:ng,nn)
obj <- EmSkewfit1(dat, ng, clust, distr, ncov, itmax=1000,epsilon=1e-4)

getICL(dat,n,p,ng, distr,ncov,obj$pro,obj$mu,obj$sigma,obj$dof,
obj$delta,obj$clust)

```

getSWR

Calculate modes

Description

Calculate the distances.

Usage

```

getSWR(dat,g,sigma,clust, tau)
intradist(dat,g,sigma, clust, tau)
interdist(dat,g,sigma, clust, tau)
mahalonobis(p, g, mu, sigma)

```

Arguments

dat	The dataset, an n by p numeric matrix, where n is number of observations and p the dimension of data.
p	The dimension of the data
g	The number of components to be fit
mu	A numeric matrix with each column corresponding to the mean
sigma	An array of dimension (p,p,g) with first two dimensions corresponding covariance matrix of each component
clust	A vector of integers specifying the initial partitions of the data; the default is NULL.
tau	An n by g matrix of posterior probability for each data point

 initEmmix

Initialize Emmix Parameters

Description

Obtains initial parameter set for use in the EM algorithm. Grouping of the data occurs through one of three possible clustering methods: k-means, random start, and hierarchical clustering.

Usage

```
initEmmix(dat, g, clust, distr, ncov,maxloop=20)
init.mix( dat, g, distr, ncov, nkmeans, nrandom, nhclust,maxloop=20)
```

Arguments

dat	The dataset, an n by p numeric matrix, where n is number of observations and p the dimension of data.
g	The number of components of the mixture model
distr	A three letter string indicating the type of distribution to be fit. See Details.
ncov	A small integer indicating the type of covariance structure. See Details.
clust	An initial partition of the data
nkmeans	An integer to specify the number of KMEANS partitions to be used to find the best initial values
nrandom	An integer to specify the number of random partitions to be used to find the best initial values
nhclust	A logical value to specify whether or not to use hierarchical cluster methods. If TRUE, the Complete Linkage method will be used.
maxloop	An integer to specify how many iterations to be tried to find the initial values,the default value is 10.

Details

The distribution type, determined by the `distr` parameter, which may take any one of the following values: "mvn" for a multivariate normal, "mvt" for a multivariate t-distribution, "msn" for a multivariate skew normal distribution and "mst" for a multivariate skew t-distribution.

The covariance matrix type, represented by the `ncov` parameter, may be any one of the following: `ncov=1` for a common variance, `ncov=2` for a common diagonal variance, `ncov=3` for a general variance, `ncov=4` for a diagonal variance, `ncov=5` for $\sigma(h) \cdot I(p)$ (diagonal covariance with same identical diagonal element values).

The return values include following components: `pro`, a numeric vector of the mixing proportion of each component; `mu`, a p by g matrix with each column as its corresponding mean; `sigma`, a three dimensional p by p by g array with its jth component matrix (p,p,j) as the covariance matrix for jth component of mixture models; `dof`, a vector of degrees of freedom for each component; `delta`, a p by g matrix with its columns corresponding to skew parameter vectors.

When the dataset is huge, it becomes time-consuming to use a large maxloop to try every initial partition. The default is 10. During the procedure to find the best initial clustering and initial values, for t-distribution and skew t-distribution, we don't estimate the degrees of freedom dof, instead they are fixed at 4 for each component.

Value

pro	A vector of mixing proportions, see Details.
mu	A numeric matrix with each column corresponding to the mean, see Details.
sigma	An array of dimension (p,p,g) with first two dimension corresponding covariance matrix of each component, see Details.
dof	A vector of degrees of freedom for each component, see Details.
delta	A p by g matrix with each column corresponding to a skew parameter vector, see Details.

References

- McLachlan G.J. and Krishnan T. (2008). The EM Algorithm and Extensions (2nd). New Jersey: Wiley.
- McLachlan G.J. and Peel D. (2000). Finite Mixture Models. New York: Wiley.

See Also

[EmSkew](#)

Examples

```
sigma<-array(0,c(2,2,3))
for(h in 2:3) sigma[,,h]<-diag(2)
sigma[,,1]<-cbind( c(1,0.2),c(0.2,1))
mu <- cbind(c(4,-4),c(3.5,4),c( 0, 0))
delta <- cbind(c(3,3),c(1,5),c(-3,1))
dof <- c(3,5,5)
pro <- c(0.3,0.3,0.4)
n1=300;n2=300;n3=400;
nn<-c(n1,n2,n3)
n=1000
p=2
ng=3
distr="mvn"
ncov=3
#first we generate a data set
set.seed(111) #random seed is set
dat <- rdemmix(nn,p,ng,distr,mu,sigma,dof,delta)
clust<- rep(1:ng,nn)
initobj1 <- initEmmix(dat,ng,clust,distr, ncov)
initobj2 <- init.mix( dat,ng,distr,ncov,nkmeans=10,nrandom=0,nhclust=FALSE)
```

inverse	<i>Inverse of a covariance matrix</i>
---------	---------------------------------------

Description

Calculate the inverse of a covariance matrix.

Usage

```
inverse(sigma, p)
```

Arguments

sigma	The covariance matrix.
p	The dimension of the matrix.

Value

The inverse of the covariance matrix.

Note

The covariance matrix may be singular. This is of use only for the clustering of the data.

Author(s)

Kui Wang

Examples

```
a<- matrix(c(1,0,0,0),ncol=2)
a
inverse(a,2)
```

mvt.dof	<i>Degrees of freedom</i>
---------	---------------------------

Description

Calculate the degrees of freedom

Usage

```
mvt.dof(sumtau, sumInv, lx = 2+1e-04, ux = 200)
```

Arguments

sumtau	A quantity.
sumlnv	A quantity.
lx	Lower limit.
ux	Upper limit.

Details

It is called by msmvt, jcamvt and jcamst. When there is no solution between lx and ux, ux will be returned.

Value

degrees of freedom.

Note

It is called by msmvt, jcamst and jcamvt.

rdemmix

Simulate Data Using Mixture Models

Description

Generate random number from specified mixture models, including univariate and multivariate Normal distribution, t-distribution, Skew Normal distribution, and Skew t-distribution.

Usage

```
rdemmix(nvect,p,g,distr, mu,sigma,dof=NULL,delta=NULL)
rdemmix2(n, p,g,distr,pro,mu,sigma,dof=NULL,delta=NULL)
rdemmix3(n, p,g,distr,pro,mu,sigma,dof=NULL,delta=NULL)
```

Arguments

nvect	A vector of how many points in each cluster,c(n1,n2,...,ng)
n	The total number of points
p	Dimension of data
g	The number of clusters
distr	A three letter string indicating the distribution type
pro	A vector of mixing proportions, see Details.
mu	A numeric matrix with each column corresponding to the mean, see Details.
sigma	An array of dimension (p,p,g) with first two dimension corresponding covariance matrix of each component, see Details.
dof	A vector of degrees of freedom for each component, see Details.
delta	A p by g matrix with each column corresponding to a skew parameter vector, see Details.

Details

The distribution type, determined by the `distr` parameter, which may take any one of the following values: "mvn" for a multivariate normal, "mvt" for a multivariate t-distribution, "msn" for a multivariate skew normal distribution and "mst" for a multivariate skew t-distribution. `pro`, a numeric vector of the mixing proportion of each component; `mu`, a p by g matrix with each column as its corresponding mean; `sigma`, a three dimensional p by p by g array with its j th component matrix (p,p,j) as the covariance matrix for j th component of mixture models; `dof`, a vector of degrees of freedom for each component; `delta`, a p by g matrix with its columns corresponding to skew parameter vectors.

Value

both `rdemmix` and `rdemmix2` return an n by p numeric matrix of generated data;
`rdemmix3` gives a list with components `data`, the generated data, and `cluster`, the clustering of data.

References

- McLachlan G.J. and Krishnan T. (2008). The EM Algorithm and Extensions (2nd). New Jersey: Wiley.
 McLachlan G.J. and Peel D. (2000). Finite Mixture Models. New York: Wiley.

See Also

[rdmvn](#), [rdmvt](#), [rdmsn](#), [rdmst](#).

Examples

```
#specify the dimension of data, and number of clusters
#the number of observations in each cluster
n1=300;n2=300;n3=400;
nn<-c(n1,n2,n3)

p=2
g=3

#specify the distribution
distr <- "mvn"

#specify mean and covariance matrix for each component

sigma<-array(0,c(2,2,3))
for(h in 2:3) sigma[,,h]<-diag(2)
sigma[,,1]<-cbind( c(1,-0.1),c(-0.1,1))
mu <- cbind(c(4,-4),c(3.5,4),c( 0, 0))

#reset the random seed
set.seed(111)
```

```
#generate the dataset
dat <- rdemmix(nn,p,g,distr, mu,sigma)

# alternatively one can use
pro <- c(0.3,0.3,0.4)
n=1000
set.seed(111)
dat <- rdemmix2(n,p,g,distr,pro,mu,sigma)
plot(dat)

# and

set.seed(111)
dobj <- rdemmix3(n,p,g,distr,pro,mu,sigma)
plot(dobj$data)

#other distributions such as "mvt","msn", and "mst".

#t-distributions

dof <- c(3,5,5)
dat <- rdemmix2(n,p,g,"mvt",pro,mu,sigma,dof)
plot(dat)

#Skew Normal distribution
delta <- cbind(c(3,3),c(1,5),c(-3,1))
dat <- rdemmix2(n,p,g,"msn",pro,mu,sigma,delta=delta)
plot(dat)

#Skew t-distribution
dat <- rdemmix2(n,p,g,"mst",pro,mu,sigma,dof,delta)
plot(dat)
```

Index

*Topic **cluster**

- bootstrap, 2
- ddmix, 6
- ddmsn, 7
- ddmst, 8
- ddmvt, 10
- EmSkew, 11
- EmSkewfit, 16
- EmSkewMOD, 19
- error.rate, 20
- getICL, 21
- getSWR, 23
- initEmmix, 24
- rдемmix, 27

*Topic **datasets**

- bootstrap, 2
- ddmsn, 7
- ddmst, 8
- ddmvt, 10
- EmSkew, 11
- EmSkewfit, 16
- EmSkewMOD, 19
- error.rate, 20
- getICL, 21
- getSWR, 23
- initEmmix, 24
- rдемmix, 27

bootstrap, 2

conplot, 4, 16

conplot2 (conplot), 4

conplot3 (conplot), 4

ddmix, 6

ddmsn, 6, 7, 9–11

ddmst, 6, 8, 8, 10, 11

ddmvt, 6, 8, 9, 9, 11

ddmvt, 6, 8–10, 10

EmSkew, 3, 11, 18, 25

EmSkew.contours, 15

EmSkew.filter (EmSkew.contours), 15

EmSkew.flow, 5

EmSkew.flow (EmSkew.contours), 15

EmSkewfit, 16

EmSkewfit1 (EmSkewfit), 16

EmSkewfit2 (EmSkewfit), 16

EmSkewMOD, 19

error.rate, 20

getcov, 21

getICL, 21

getSWR, 23

init.mix, 18

init.mix (initEmmix), 24

initEmmix, 13, 18, 24

interdist (getSWR), 23

intradist (getSWR), 23

inverse, 26

mahalonobis (getSWR), 23

mvt.dof, 26

mypanel2 (conplot), 4

mypanel3 (conplot), 4

mypanel4 (conplot), 4

panel.density (conplot), 4

rand.index (error.rate), 20

rдемmix, 3, 8–11, 13, 18, 27

rдемmix2, 18

rдемmix2 (rдемmix), 27

rдемmix3 (rдемmix), 27

rdmsn, 9–11, 18, 28

rdmsn (ddmsn), 7

rdmst, 8, 10, 11, 18, 28

rdmst (ddmst), 8

rdm_{vn}, [8](#), [9](#), [11](#), [18](#), [28](#)

rdm_{vn} (ddm_{vn}), [9](#)

rdm_{vt}, [8–10](#), [18](#), [28](#)

rdm_{vt} (ddm_{vt}), [10](#)