Package ‘EMMIX’

November 8, 2013

Title  The EM Algorithm and Mixture Models

Description  Fit multivariate mixture models via the EM Algorithm. Multivariate distributions include Normal distribution, t-distribution, Skew Normal distribution and and Skew t-distribution. The emmix is an updated version of EMMIX with new features such as clustering the degenerated data and fitting skew mixture models.

Depends  mvtnorm

Version  1.0.1

Author  Geoff McLachlan, Kui Wang, Angus Ng and David Peel. EMMIX was originally written in Fortran by David Peel(1999).

Maintainer  Geoff McLachlan <g.mclachlan@uq.edu.au>

License  file LICENCE

LazyLoad  yes

Archs  i386, x64

R topics documented:

  bootstrap ................................................................. 2
  ddmix ................................................................. 4
  ddmvn ................................................................. 5
  ddmvt ................................................................. 6
  EMMIX ................................................................. 7
  EMMIX2 ............................................................... 10
  emmixfit ............................................................. 13
  emmixfit3 ............................................................ 15
  error.rate ............................................................ 17
  getICL ................................................................. 18
  initEmmix ............................................................. 19
  rdemmix ............................................................... 21

Index  24
**Description**

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

**Usage**

```r
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)*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 -2log(Lambda), `vlk`, 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


See Also

`EMMIX`, `rdemmix`

Examples

```r
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(1:3,nn)
obj <- emmixfit1(dat,g,clust,distr,ncov,itmmax=1000,epsilon=1e-5)

# do bootstrap (standard error analysis)

## Not run:
std <- bootstrap(dat,n,p,g,distr,ncov,itmmax=1000,epsilon=1e-5)
print(std)
# do bootstrap 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,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)

----

ddmix  

Density Functions of Mixture Models

Description

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

Usage

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

Arguments

- **dat**: The dataset
- **n**: The total number of points
- **p**: Dimension of data
- **g**: The number of clusters
- **distr**: A three letter string indicating the distribution; "mvn" for normal, "mvt" for t distribution, "msn" for skew normal, and "mst" for skew t distribution.
- **mu**: A numeric mean 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
Value

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

References


See Also
ddmvn, ddmvt.

Examples

```r
p = 2
n = 3

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

# specify mean and covariance matrix for each component
sigma <- array(c(2, 3), 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"
dof <- c(3, 5, 5)

distr <- "mvt"

# specify the distribution
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)
```

---

ddmvn

**The Multivariate Normal Distribution**

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`, `rdmvt`.

**Examples**

```r
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)
```
EMMIX

The EM Algorithm and Skew Mixture Models

Description

As a main function, EMMIX 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

EMMIX(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)
EMMIX

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 fitted, the default
value is "mvn", the Normal distribution. See Details.

cmp
A small integer indicating the type of covariance structure; the default value is
3. See Details.

clust
A vector of integers specifying the initial partitions of the data; the default is
NULL.

init
A list containing the initial parameters for the mixture model. See details. The
default value is NULL.

itmax
A big integer specifying the maximum number of iterations to apply; the default
value is 1000.

epsilon
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.

nkmeans
An integer to specify the number of KMEANS partitions to be used to find the
best initial values; the default value is 0.
nrandom
An integer to specify the number of random partitions to be used to find the best
initial values; the default value is 10.
nhclust
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.

deg
A logical value, if it is TRUE, the output will be printed out; FALSE silent; the
default value is TRUE.

initloop
A integer specifying the number of initial loops when searching the best intial
partitions.

Details

The distribution type, determined by the distr parameter, which may take any one of the follow-
ing 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 ncmp parameter, may be any one of the following:
ncmp=1 for a common variance, ncmp=2 for a common diagonal variance, ncmp=3 for a general
variance, ncmp=4 for a diagonal variance, ncmp=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 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.

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 parameters in the list.

When the parameter list init is given, the program ignores both initial partition clust and auto-
matic 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 initial 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**

- **error**: Error code, 0 = normal exit; 1 = did not converge within \( \text{itmax} \) iterations; 2 = failed to get the initial values; 3 = singularity
- **aic**: Akaike Information Criterion (AIC)
- **bic**: Bayes Information Criterion (BIC)
- **icl**: Integrated Completed Likelihood Criterion (ICL)
- **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, see Details.
- **delta**: A \(p\) by \(g\) matrix with each column corresponding to a skew parameter vector.
- **clust**: A vector of final partition
- **loglik**: The log likelihood at convergence
- **lk**: A vector of log likelihood at each EM iteration
- **tau**: An \(n\) by \(g\) matrix of posterior probability for each data point

**References**


**See Also**

- `initEmmix`, `rdemmix`.

**Examples**

```r
# 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)
```
```r
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)
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[1:3,2]<-4
# dat[3+1:3,1]<-2
## dat[6,1:1,2]<-0

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

# 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 <- EMMIX(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 <- EMMIX(dat,ng,distr,ncov,clust=clust,itmax=1000,epsilon=1e-5)
```

---

EMMIX2

**The EM Algorithm of the Mixture Models (Restricted)**
EMMIX2

Description
EMMIX2 fits the data into the Normal multivariate mixture models via the EM Algorithm, with constraints on either 1) to fix the mean values for components and then estimate the variances within components; or 2) to fix mean values for components, and specify equal variances for some components, but allow unequal variances for others.

Usage
EMMIX2(dat,g,init,nvcov=0,neq=0,itmax=1000,epsilon=1e-6, debug=TRUE)

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.
init A list containing the initial parameters for the mixture model. See details.
nvcov A vector of integer indicating which components have equal covariance matrices; the default value is 0, which means null of them. See Details.
neq An integer of how many components are of the equal covariance matrices; the default value is 0, which means null of them. See Details.
itmax A big integer specifying the maximum number of iterations to apply; the default value is 1000.
epsilon 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.
debug A logical value, if it is TRUE, the output will be printed out; the default value is TRUE.

Details
The means of the mixture have to be predefined based on prior information somehow and assigned to the mu of the list init. The format must be p by g matrix with each column corresponding to the mean of that component.
nvcov has to be matched with neq, for example, if we want the first and third components have equal covariance matrix, then just specify nvcov=c(1,3) and neq=2.
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 component matrix sigma[,]j as the covariance matrix for jth component of mixture models.

Value
error Error code, 0 = normal exit; 1 = did not converge within itmax iterations; 2 = failed to get the initial values; 3 = singularity.
aic Akaike Information Criterion (AIC).
bic Bayes Information Criterion (BIC).
pro A vector of mixing proportions.
mu A numeric matrix with each column corresponding to the mean.
EMMIX2

sigma  An array of dimension (p,p,g) with first two dimension corresponding covariance matrix of each component.
clust  A vector of final partition.
loglik  The log likelihood at convergence.
1k  A vector of log likelihood at each EM iteration.
tau  An n by g matrix of posterior probability at each data point.

References

See Also
emmixfit3, EMMIX.

Examples

```r
#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))

pro  <- c(0.3,0.3,0.4)
distr="mvn"

# generate a data set
set.seed(111) # random seed is reset
dat <- rdemmix(nn,p,ng,distr,mu,sigma)

init<-list()
init$pro<-pro
init$mu<-mu
init$sigma<-sigma

# for restriction 1, we have
ret1 <- EMMIX2(dat,ng,init,nvcov=0,neq=0,itmax=1000,epsilon=1e-6,debug=TRUE)
```
# for restriction 2, we have
ret2 <- EMMIX2(dat,ng,init,nvcov=c(1,2),neq=2,itmax=1000,epsilon=1e-6,debug=TRUE)

## emmixfit

### Fit the Multivariate Skew Mixture Models

**Description**

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

**Usage**

```r
emmixfit1(dat, g, clust, distr, ncov, itmax, epsilon,initloop=2)
emmixfit2(dat, g, init, distr, ncov, itmax, epsilon)
```

**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` A vector of integers specifying the initial partitions of the data
- `init` A list containing the initial parameters for the mixture model. See details.
- `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.
- `initloop` 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 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.
Value

- **error**
  - Error code, 0 = normal exit; 1 = did not converge within `itmax` iterations; 2 = failed to get the initial values; 3 = singularity
- **aic**
  - Akaike Information Criterion (AIC)
- **bic**
  - Bayes Information Criterion (BIC)
- **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.
- **clust**
  - A vector of final partition
- **loglik**
  - The loglikelihood at convergence
- **lk**
  - A vector of loglikelihood at each EM iteration
- **tau**
  - An n by g matrix of posterior probability for each data point

References


See Also

`init.mix`, `initEmmix`, `EMMIX`, `rdemmix`, `rdemmix2`, `rdmvn`, `rdmvt`.

Examples

```r
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)

pro <- c(0.3, 0.3, 0.4)
```
### emmixfit3

**Fit the Multivariate Mixture Models (Restricted)**

**Description**

emmixfit3 fits the data into the Normal multivariate mixture models via the EM Algorithm, with constraints on either 1) to fix the mean values for components and then estimate the variances within components; or 2) to fix mean values for components, and specify equal variances for some components, but allow unequal variances for others.

**Usage**

```r
emmixfit3(dat, g, init, nvcov=0, neq=0, itmax, epsilon)
```

**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.
- **init**: A list containing the initial parameters for the mixture model. See details.
- **nvcov**: A vector of integer indicating which components have equal covariance matrices; the default value is 0, which means null of them. See Details.
neq An integer of how many components are of the equal covariance matrices; the default value is 0, which means null of them. See Details.

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 means of the mixture have to be predefined based on prior information somehow and asigned to the \( \mu \) of the list \( \text{init} \). The format must be \( p \) by \( g \) matrix with each column corresponding to the mean of that component.

\( nvcov \) has to be matched with \( \text{neq} \), for example, if we want the first and third components have equal covariance matrix, then just specify \( nvcov=c(1,3) \) and \( \text{neq}=2 \).

The parameter \( \text{init} \) requires following elements: \( \text{pro} \), 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 component matrix \( \sigma[\cdot,j] \) as the covariance matrix for \( j \)th component of mixture models.

Value

\( \text{error} \) Error code, 0 = normal exit; 1 = did not converge within \( \text{itmax} \) iterations; 2 = failed to get the initial values; 3 = singularity.

\( \text{aic} \) Akaike Information Criterion (AIC).

\( \text{bic} \) Bayes Information Criterion (BIC).

\( \text{pro} \) 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.

\( \text{clust} \) A vector of final partition.

\( \loglik \) The loglikelihood at convergence.

\( \text{lk} \) A vector of loglikelihood at each EM iteration.

\( \tau \) An \( n \) by \( g \) matrix of posterior probability at each data point.

References


See Also

\( \text{EMMIX2,EMMIX} \)
error.rate

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[,]<-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"

#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 values
init<-list()
init$pro<-pro
init$mu<-mu
init$sigma<-sigma

#for restriction 1, we have
ret1<-emmixfit3(dat,ng,init,nvcov=0,neq=0,itmax=1000,epsilon=1e-6)

#for restriction 2, we have
ret2<-emmixfit3(dat,ng,init,nvcov=c(1,2),neq=2,itmax=1000,epsilon=1e-6)

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)

getICL

The ICL criterion

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
**initEmmix**

**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)*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 each column corresponding to a skew parameter vector.

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.
**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**

```r
demmix(nvect,p,g,distr, mu,sigma,dof=NULL, delta=NULL)
demmix2(n, p,g,distr,pro,mu,sigma,dof=NULL, delta=NULL)
demmix3(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 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.

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


See Also

rdmvn, rdmvt.

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

distr <- "mvn"
#specify mean and covariance matrix for each component

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

#reset the random seed
\[
\text{set.seed}(111)
\]
#generate the dataset
\[
\text{dat} \leftarrow \text{rdemmix}(\text{nn},\text{p},\text{g},\text{distr}, \text{mu},\text{sigma})
\]

# alternatively one can use
\[
\text{pro} \leftarrow \text{c}(0.3,0.3,0.4)
\]
\[
n=1000
\]
\[
\text{set.seed}(111)
\]
\[
\text{dat} \leftarrow \text{rdemmix2}(n,\text{p},\text{g},\text{distr},\text{pro},\text{mu},\text{sigma})
\]
\[
\text{plot(dat)}
\]

# and
\[
\text{set.seed}(111)
\]
\[
\text{dobj} \leftarrow \text{rdemmix3}(n,\text{p},\text{g},\text{distr},\text{pro},\text{mu},\text{sigma})
\]
\[
\text{plot(dobj$data)}
\]

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

#t-distributions
\[
\text{dof} \leftarrow \text{c}(3,5,5)
\]
\[
\text{dat} \leftarrow \text{rdemmix2}(n,\text{p},\text{g},"mvtn",\text{pro},\text{mu},\text{sigma},\text{dof})
\]
\[
\text{plot(dat)}
\]

#Skew Normal distribution
\[
\text{delta} \leftarrow \text{cbind}(\text{c}(3,3),\text{c}(1,5),\text{c}(-3,1))
\]
\[
\text{dat} \leftarrow \text{rdemmix2}(n,\text{p},\text{g},"msn",\text{pro},\text{mu},\text{sigma},\text{delta}=\text{delta})
\]
\[
\text{plot(dat)}
\]

#Skew t-distribution
\[
\text{dat} \leftarrow \text{rdemmix2}(n,\text{p},\text{g},"mst",\text{pro},\text{mu},\text{sigma},\text{dof},\text{delta})
\]
\[
\text{plot(dat)}
\]
Index

*Topic* **cluster**
- bootstrap, 2
- ddmix, 4
- ddmvn, 5
- ddmvt, 6
- EMMIX, 7
- EMMIX2, 10
- emmixfit, 13
- emmixfit2, 15
- error.rate, 17
- getICL, 18
- initEmmix, 19
- rdemmix, 21

*Topic* **datasets**
- bootstrap, 2
- ddmix, 4
- ddmvn, 5, 6
- ddmvt, 6
- EMMIX, 7
- EMMIX2, 10
- emmixfit, 13
- emmixfit2, 15
- error.rate, 17
- getICL, 18
- initEmmix, 19
- rdemmix, 21

bootstrap, 2
- ddmix, 4
- ddmvn, 5, 5, 7
- ddmvt, 5, 6, 6

EMMIX, 3, 7, 12, 14, 16, 21
EMMIX2, 10, 16
emmixfit, 13
emmixfit1 (emmixfit), 13
emmixfit2 (emmixfit), 13
emmixfit3, 12, 15
error.rate, 17
getICL, 18
init.mix, 14
init.mix (initEmmix), 19
initEmmix, 9, 14, 19

rand.index (error.rate), 17
rdemmix, 3, 6, 7, 9, 14, 21
rdemmix2, 14
rdemmix2 (rdemmix), 21
rdemmix3 (rdemmix), 21
rdmvn, 7, 14, 22
rdmvn (ddmvn), 5
rdmvt, 6, 14, 22
rdmvt (ddmvt), 6