Package 'gmmsslm'

October 16, 2023

Type Package

Title Semi-Supervised Gaussian Mixture Model with a Missing-Data Mechanism

Version 1.1.5

Description The algorithm of semi-supervised learning is based on finite Gaussian mixture models and includes a mechanism for handling missing data. It aims to fit a g-class Gaussian mixture model using maximum likelihood. The algorithm treats the labels of unclassified features as missing data, building on the framework introduced by Rubin (1976) <doi:10.2307/2335739> for missing data analysis. By taking into account the dependencies in the missing pattern, the algorithm provides more information for determining the optimal classifier, as specified by Bayes' rule.

Depends R (>= 3.1.0), mvtnorm,stats,methods

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Ziyang Lyu [aut, cre],
Daniel Ahfock [aut],
Ryan Thompson [aut],
Geoffrey J. McLachlan [aut]

Maintainer Ziyang Lyu <ziyang.lyu@unsw.edu.au>

Repository CRAN

Date/Publication 2023-10-16 04:30:06 UTC

R topics documented:

`bayesclassifier` *Bayes' rule of allocation*

Description

Bayes' rule of allocation

Usage

```
bayesclassifier(dat, p, g, pi = NULL, mu = NULL, sigma = NULL, paralist = NULL)
```

Arguments

`dat`	An $n \times p$ matrix where each row represents an individual observation.
`p`	Dimension of observation vecor.
`g`	Number of multivariate normal classes.
`pi`	A g-dimensional vector for the initial values of the mixing proportions.
`mu`	A $p \times g$ matrix for the initial values of the location parameters.

sigma	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if sigma is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.
paralist	A list containing the required parameters (π, μ, Σ).

Details

Classifier specified by Bayes' rule

The classifier/Bayes rule of allocation $R(y_j; \theta)$ assigns an entity with observation y_j to class C_k(that is, $R(y_j; \theta) = k$) if $k = \arg\max_i \tau_i(y_j; \theta)$,

Value

clust	Class membership for the ith entity

Examples

```
n <- 150
pi <- c(0.25, 0.25, 0.25, 0.25)
sigma <- array(0, dim = c(3, 3, 4))
sigma[, , 1] <- diag(1, 3)
sigma[, , 2] <- diag(2, 3)
sigma[, , 3] <- diag(3, 3)
sigma[, , 4] <- diag(4, 3)
mu <- matrix(c(0.2, 0.3, 0.4, 0.2, 0.7, 0.6, 0.1, 0.7, 1.6, 0.2, 1.7, 0.6), 3, 4)
dat <- rmix(n = n, pi = pi, mu = mu, sigma = sigma)
params <- list(pi=pi,mu = mu, sigma = sigma)
clust <- bayesclassifier(dat=dat$Y,p=3,g=4,paralist=params)
```

---

bootstrap_gmmsslm *Bootstrap Analysis for gmmsslm*

Description

This file provides functions to perform bootstrap analysis on the results of the gmmsslm function.

This function performs non-parametric bootstrap to assess the variability of the gmmsslm function outputs.

Usage

```
bootstrap_gmmsslm(
  dat,
  zm,
  pi,
  mu,
  sigma,
  paralist,
```

```
  xi,
  type,
  iter.max = 500,
  eval.max = 500,
  rel.tol = 1e-15,
  sing.tol = 1e-15,
  B = 2000
)
```

Arguments

dat	A matrix where each row represents an individual observation.
zm	A matrix or data frame of labels corresponding to dat.
pi	A numeric vector representing the mixing proportions.
mu	A matrix representing the location parameters.
sigma	An array representing the covariance matrix or list of covariance matrices.
paralist	A list of parameters.
xi	A numeric value representing the coefficient for a logistic function of the Shannon entropy.
type	A character value indicating the type of Gaussian mixture model.
iter.max	An integer indicating the maximum number of iterations.
eval.max	An integer indicating the maximum number of evaluations.
rel.tol	A numeric value indicating the relative tolerance.
sing.tol	A numeric value indicating the singularity tolerance.
B	An integer indicating the number of bootstrap samples.

Value

A list containing mean and sd of bootstrap samples for pi, mu, sigma, and xi.

---

cov2vec *Transform a variance matrix into a vector*

Description

Transform a variance matrix into a vector i.e., Sigma=R^T*R

Usage

```
cov2vec(sigma)
```

Arguments

sigma	A $p \times p$ variance matrix

Details

The variance matrix is decomposed by computing the Choleski factorization of a real symmetric positive-definite square matrix. Then, storing the upper triangular factor of the Choleski decomposition into a vector.

Value

par A vector representing a variance matrix

`discriminant_beta` *Discriminant function*

Description

Discriminant function in the particular case of g=2 classes with an equal-covariance matrix

Usage

```
discriminant_beta(pi, mu, sigma)
```

Arguments

`pi`	A g-dimensional vector for the initial values of the mixing proportions.
`mu`	A $p \times g$ matrix for the initial values of the location parameters.
`sigma`	A $p \times p$ covariance matrix.

Details

Discriminant function in the particular case of g=2 classes with an equal-covariance matrix can be expressed

$$d(y_i, \beta) = \beta_0 + \beta_1 y_i,$$

where $\beta_0 = \log \frac{\pi_1}{\pi_2} - \frac{1}{2} \frac{\mu_1^2 - \mu_2^2}{\sigma^2}$ and $\beta_1 = \frac{\mu_1 - \mu_2}{\sigma^2}$.

Value

`beta0`	An intercept of discriminant function
`beta`	A coefficient of discriminant function

erate *Error rate of the Bayes rule for a g-class Gaussian mixture model*

Description

Error rate of the Bayes rule for a g-class Gaussian mixture model

Usage

```
erate(dat, p, g, pi = NULL, mu = NULL, sigma = NULL, paralist = NULL, clust)
```

Arguments

dat	An $n \times p$ matrix where each row represents an individual observation
p	Dimension of observation vecor.
g	Number of multivariate normal classes.
pi	A g-dimensional vector for the initial values of the mixing proportions.
mu	A $p \times g$ matrix for the initial values of the location parameters.
sigma	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if sigma is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.
paralist	A list containing the required parameters (π, μ, Σ).
clust	An n-dimensional vector of class partition.

Details

The error rate of the Bayes rule for a g-class Gaussian mixture model is given by

$$err(y;\theta) = 1 - \sum_{i=1}^{g} \pi_i Pr\{R(y;\theta) = i \mid Z \in C_i\}.$$

Here, we write

$$Pr\{R(y;\theta) \in C_i \mid Z \in C_i\} = \frac{\sum_{j=1}^{n} I_{C_i}(z_j) Q[z_j, R(y;\theta)]}{\sum_{j=1}^{n} I_{C_i}(z_j)},$$

where $Q[u, v] = 1$ if $u = v$ and $Q[u, v] = 0$ otherwise, and $I_{C_i}(z_j)$ is an indicator function for the ith class.

Value

errval	a value of error rate

Examples

```
n<-150
pi<-c(0.25,0.25,0.25,0.25)
sigma<-array(0,dim=c(3,3,4))
sigma[,,1]<-diag(1,3)
sigma[,,2]<-diag(2,3)
sigma[,,3]<-diag(3,3)
sigma[,,4]<-diag(4,3)
mu<-matrix(c(0.2,0.3,0.4,0.2,0.7,0.6,0.1,0.7,1.6,0.2,1.7,0.6),3,4)
dat<-rmix(n=n,pi=pi,mu=mu,sigma=sigma)
xi<-c(-0.5,1)
m<-rlabel(dat=dat$Y,pi=pi,mu=mu,sigma=sigma,xi=xi)
zm<-dat$clust
zm[m==1]<-NA
inits<-initialvalue(g=4,zm=zm,dat=dat$Y)

fit_pc<-gmmsslm(dat=dat$Y,zm=zm,pi=inits$pi,mu=inits$mu,sigma=inits$sigma,xi=xi,type='full')
parlist<-paraextract(fit_pc)
erate(dat=dat$Y,p=3,g=4,paralist=parlist,clust=dat$clust)
```

`errorrate` *Error rate of the Bayes rule for two-class Gaussian homoscedastic model*

Description

The optimal error rate of Bayes rule for two-class Gaussian homoscedastic model

Usage

```
errorrate(beta0, beta, pi, mu, sigma)
```

Arguments

`beta0`	An intercept parameter of the discriminant function coefficients.
`beta`	A $p \times 1$ vector for the slope parameter of the discriminant function.
`pi`	A g-dimensional vector for the initial values of the mixing proportions.
`mu`	A $p \times g$ matrix for the initial values of the location parameters.
`sigma`	A $p \times p$ covariance matrix.

Details

The optimal error rate of Bayes rule for two-class Gaussian homoscedastic model can be expressed as

$$err(\beta) = \pi_1 \phi\{-\frac{\beta_0 + \beta_1^T \mu_1}{(\beta_1^T \Sigma \beta_1)^{\frac{1}{2}}}\} + \pi_2 \phi\{\frac{\beta_0 + \beta_1^T \mu_2}{(\beta_1^T \Sigma \beta_1)^{\frac{1}{2}}}\}$$

where ϕ is a normal probability function with mean μ_i and covariance matrix Σ_i.

Value

errval	A vector of error rate.

`gastro_data` *Gastrointestinal dataset*

Description

The collected dataset is composed of 76 colonoscopic videos (recorded with both White Light (WL) and Narrow Band Imaging (NBI)), the histology (classification ground truth), and the endoscopist's opinion (including 4 experts and 3 beginners). There are $n=76$ observations, and each observation consists of 698 features extracted from colonoscopic videos on patients with gastrointestinal lesions.

References

http://www.depeca.uah.es/colonoscopy_dataset/

`get_clusterprobs` *Posterior probability*

Description

Get posterior probabilities of class membership

Usage

```
get_clusterprobs(
  dat,
  n,
  p,
  g,
  pi = NULL,
  mu = NULL,
  sigma = NULL,
  paralist = NULL
)
```

Arguments

dat	An $n \times p$ matrix where each row represents an individual observation
n	Number of observations.
p	Dimension of observation vecor.
g	Number of multivariate normal classes.
pi	A g-dimensional vector for the initial values of the mixing proportions.

mu	A $p \times g$ matrix for the initial values of the location parameters.
sigma	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if sigma is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.
paralist	A list containing the required parameters (π, μ, Σ).

Details

The posterior probability can be expressed as

$$\tau_i(y_j;\theta) = Prob\{z_{ij} = 1|y_j\} = \frac{\pi_i \phi(y_j;\mu_i,\Sigma_i)}{\sum_{h=1}^{g} \pi_h \phi(y_j;\mu_h,\Sigma_h)},$$

where ϕ is a normal probability function with mean μ_i and covariance matrix Σ_i, and z_{ij} is is a zero-one indicator variable denoting the class of origin.

Value

clusprobs	Posterior probabilities of class membership for the ith entity

Examples

```
n<-150
pi<-c(0.25,0.25,0.25,0.25)
sigma<-array(0,dim=c(3,3,4))
sigma[,,1]<-diag(1,3)
sigma[,,2]<-diag(2,3)
sigma[,,3]<-diag(3,3)
sigma[,,4]<-diag(4,3)
mu<-matrix(c(0.2,0.3,0.4,0.2,0.7,0.6,0.1,0.7,1.6,0.2,1.7,0.6),3,4)
dat<-rmix(n=n,pi=pi,mu=mu,sigma=sigma)
tau<-get_clusterprobs(dat=dat$Y,n=150,p=3,g=4,mu=mu,sigma=sigma,pi=pi)
```

---

get_entropy *Shannon entropy*

Description

Shannon entropy

Usage

```
get_entropy(dat, n, p, g, pi = NULL, mu = NULL, sigma = NULL, paralist = NULL)
```

Arguments

dat	An $n \times p$ matrix where each row represents an individual observation
n	Number of observations.
p	Dimension of observation vecor.
g	Number of multivariate normal classes.
pi	A g-dimensional vector for the initial values of the mixing proportions.
mu	A $p \times g$ matrix for the initial values of the location parameters.
sigma	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$.
paralist	A list containing the required parameters (π, μ, Σ). It is assumed to fit the model with a common covariance matrix if sigma is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.

Details

The concept of information entropy was introduced by *shannon1948mathematical*. The entropy of y_j is formally defined as

$$e_j(y_j;\theta) = -\sum_{i=1}^{g} \tau_i(y_j;\theta) \log \tau_i(y_j;\theta).$$

Value

clusprobs	The posterior probabilities of the i-th entity that belongs to the j-th group.

Examples

```
n<-150
pi<-c(0.25,0.25,0.25,0.25)
sigma<-array(0,dim=c(3,3,4))
sigma[,,1]<-diag(1,3)
sigma[,,2]<-diag(2,3)
sigma[,,3]<-diag(3,3)
sigma[,,4]<-diag(4,3)
mu<-matrix(c(0.2,0.3,0.4,0.2,0.7,0.6,0.1,0.7,1.6,0.2,1.7,0.6),3,4)
dat<-rmix(n=n,pi=pi,mu=mu,sigma=sigma)
en<-get_entropy(dat=dat$Y,n=150,p=3,g=4,mu=mu,sigma=sigma,pi=pi)
```

---

gmmsslm — *Fitting Gaussian mixture model to a complete classified dataset or an incomplete classified dataset with/without the missing-data mechanism.*

Description

Fitting Gaussian mixture model to a complete classified dataset or an incomplete classified dataset with/without the missing-data mechanism.

Usage

```
gmmsslm(
  dat,
  zm,
  pi = NULL,
  mu = NULL,
  sigma = NULL,
  paralist = NULL,
  xi = NULL,
  type,
  iter.max = 500,
  eval.max = 500,
  rel.tol = 1e-15,
  sing.tol = 1e-15
)
```

Arguments

dat	An $n \times p$ matrix where each row represents an individual observation
zm	An n-dimensional vector containing the class labels including the missing-label denoted as NA.
pi	A g-dimensional vector for the initial values of the mixing proportions.
mu	A $p \times g$ matrix for the initial values of the location parameters.
sigma	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if sigma is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.
paralist	A list containing the required parameters (π, μ, Σ).
xi	A 2-dimensional vector containing the initial values of the coefficients in the logistic function of the Shannon entropy.
type	Three types of Gaussian mixture models, 'ign' indicates fitting the model to a partially classified sample on the basis of the likelihood that ignores the missing label mechanism, 'full' indicates fitting the model to a partially classified sample on the basis of the full likelihood, taking into account the missing-label mechanism, and 'com' indicate fitting the model to a completed classified sample.
iter.max	Maximum number of iterations allowed. Defaults to 500
eval.max	Maximum number of evaluations of the objective function allowed. Defaults to 500
rel.tol	Relative tolerance. Defaults to 1e-15
sing.tol	Singular convergence tolerance; defaults to 1e-20.

Value

A gmmsslmFit object containing the following slots:

objective	Value of objective likelihood
convergence	Value of convergence
iteration	Number of iterations
obs	Input data matrix
n	Number of observations
p	Number of variables
g	Number of Gaussian components
type	Type of Gaussian mixture model
pi	Estimated vector of the mixing proportions
mu	Estimated matrix of the location parameters
sigma	Estimated covariance matrix or list of covariance matrices
xi	Estimated coefficient vector for a logistic function of the Shannon entropy

Examples

```
n<-150
pi<-c(0.25,0.25,0.25,0.25)
sigma<-array(0,dim=c(3,3,4))
sigma[,,1]<-diag(1,3)
sigma[,,2]<-diag(2,3)
sigma[,,3]<-diag(3,3)
sigma[,,4]<-diag(4,3)
mu<-matrix(c(0.2,0.3,0.4,0.2,0.7,0.6,0.1,0.7,1.6,0.2,1.7,0.6),3,4)
dat<-rmix(n=n,pi=pi,mu=mu,sigma=sigma)
xi<-c(-0.5,1)
m<-rlabel(dat=dat$Y,pi=pi,mu=mu,sigma=sigma,xi=xi)
zm<-dat$clust
zm[m==1]<-NA
inits<-initialvalue(g=4,zm=zm,dat=dat$Y)

fit_pc<-gmmsslm(dat=dat$Y,zm=zm,paralist=inits,xi=xi,type='full')
```

---

gmmsslmFit-class *gmmsslmFit Class*

Description

gmmsslmFit objects store the results of fitting Gaussian mixture models using the gmmsslm function.

An S4 class representing the result of fitting a Gaussian mixture model using gmmsslm()

Slots

`objective` A numeric value representing the objective likelihood.

`ncov` A numeric value representing the number of covariance matrices.

`convergence` A numeric value representing the convergence value.

`iteration` An integer value representing the number of iterations.

`obs` A matrix containing the input data.

`m` A logical vector representing label indicators.

`n` An integer value representing the number of observations.

`p` An integer value representing the number of variables.

`g` An integer value representing the number of Gaussian components.

`type` A character value representing the type of Gaussian mixture model.

`pi` A numeric vector representing the mixing proportions.

`mu` A matrix representing the location parameters.

`sigma` An array representing the covariance matrix or list of covariance matrices.

`xi` A numeric value representing the coefficient for a logistic function of the Shannon entropy.

See Also

gmmsslm

`initialvalue` *Initial values for ECM*

Description

Inittial values for claculating the estimates based on solely on the classified features.

Usage

```
initialvalue(dat, zm, g, ncov = 2)
```

Arguments

`dat`	An $n \times p$ matrix where each row represents an individual observation
`zm`	An n-dimensional vector containing the class labels including the missing-label denoted as NA.
`g`	Number of multivariate normal classes.
`ncov`	Options of structure of sigma matrix; the default value is 2; `ncov = 1` for a common covariance matrix; `ncov = 2` for the unequal covariance/scale matrices.

Value

pi	A g-dimensional initial vector of the mixing proportions.
mu	A initial $p \times g$ matrix of the location parameters.
sigma	A $p \times p$ covariance matrix if ncov=1, or a list of g covariance matrices with dimension $p \times p \times g$ if ncov=2.

Examples

```
n<-150
pi<-c(0.25,0.25,0.25,0.25)
sigma<-array(0,dim=c(3,3,4))
sigma[,,1]<-diag(1,3)
sigma[,,2]<-diag(2,3)
sigma[,,3]<-diag(3,3)
sigma[,,4]<-diag(4,3)
mu<-matrix(c(0.2,0.3,0.4,0.2,0.7,0.6,0.1,0.7,1.6,0.2,1.7,0.6),3,4)
dat<-rmix(n=n,pi=pi,mu=mu,sigma=sigma)
xi<-c(-0.5,1)
m<-rlabel(dat=dat$Y,pi=pi,mu=mu,sigma=sigma,xi=xi)
zm<-dat$clust
zm[m==1]<-NA
initlist<-initialvalue(g=4,zm=zm,dat=dat$Y,ncov=2)
```

---

list2par *Transfer a list into a vector*

Description

Transfer a list into a vector

Usage

```
list2par(p, g, pi, mu, sigma, xi = NULL, type = c("ign", "full", "com"))
```

Arguments

p	Dimension of observation vecor.
g	Number of multivariate normal classes.
pi	A g-dimensional vector for the initial values of the mixing proportions.
mu	A $p \times g$ matrix for the initial values of the location parameters.
sigma	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if sigma is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.

xi	A 2-dimensional vector containing the initial values of the coefficients in the logistic function of the Shannon entropy.
type	Three types to fit to the model, 'ign' indicates fitting the model on the basis of the likelihood that ignores the missing label mechanism, 'full' indicates that the model to be fitted on the basis of the full likelihood, taking into account the missing-label mechanism, and 'com' indicate that the model to be fitted to a completed classified sample.

Value

par	a vector including all list information

`loglk_full` *Full log-likelihood function*

Description

Full log-likelihood function with both terms of ignoring and missing

Usage

```
loglk_full(dat, zm, pi, mu, sigma, xi)
```

Arguments

dat	An $n \times p$ matrix where each row represents an individual observation
zm	An n-dimensional vector containing the class labels including the missing-label denoted as NA.
pi	A g-dimensional vector for the initial values of the mixing proportions.
mu	A $p \times g$ matrix for the initial values of the location parameters.
sigma	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if `sigma` is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.
xi	A 2-dimensional vector containing the initial values of the coefficients in the logistic function of the Shannon entropy.

Details

The full log-likelihood function can be expressed as

$$\log L_{PC}^{(full)}(\mathbf{\Psi}) = \log L_{PC}^{(ig)}(\theta) + \log L_{PC}^{(miss)}(\theta, \boldsymbol{\xi}),$$

where$\log L_{PC}^{(ig)}(\theta)$is the log likelihood function formed ignoring the missing in the label of the unclassified features, and $\log L_{PC}^{(miss)}(\theta, \boldsymbol{\xi})$ is the log likelihood function formed on the basis of the missing-label indicator.

Value

lk	Log-likelihood value

`loglk_ig` *Log likelihood for partially classified data with ingoring the missing mechanism*

Description

Log likelihood for partially classified data with ingoring the missing mechanism

Usage

```
loglk_ig(dat, zm, pi, mu, sigma)
```

Arguments

dat	An $n \times p$ matrix where each row represents an individual observation
zm	An n-dimensional vector containing the class labels including the missing-label denoted as NA.
pi	A g-dimensional vector for the initial values of the mixing proportions.
mu	A $p \times g$ matrix for the initial values of the location parameters.
sigma	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if sigma is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.

Details

The log-likelihood function for partially classified data with ingoring the missing mechanism can be expressed as

$$\log L_{PC}^{(ig)}(\theta) = \sum_{j=1}^{n} \left[(1 - m_j) \sum_{i=1}^{g} z_{ij} \left\{ \log \pi_i + \log f_i(y_j; \omega_i) \right\} + m_j \log \left\{ \sum_{i=1}^{g} \pi_i f_i(y_j; \omega_i) \right\} \right],$$

where m_j is a missing label indicator, z_{ij} is a zero-one indicator variable defining the known group of origin of each, and $f_i(y_j; \omega_i)$ is a probability density function with parameters ω_i.

Value

lk	Log-likelihood value.

`loglk_miss` *Log likelihood function formed on the basis of the missing-label indicator*

Description

Log likelihood for partially classified data based on the missing mechanism with the Shanon entropy

Usage

```
loglk_miss(dat, zm, pi, mu, sigma, xi)
```

Arguments

`dat`	An $n \times p$ matrix where each row represents an individual observation
`zm`	An n-dimensional vector containing the class labels including the missing-label denoted as NA.
`pi`	A g-dimensional vector for the initial values of the mixing proportions.
`mu`	A $p \times g$ matrix for the initial values of the location parameters.
`sigma`	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if `sigma` is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.
`xi`	A 2-dimensional vector containing the initial values of the coefficients in the logistic function of the Shannon entropy.

Details

The log-likelihood function formed on the basis of the missing-label indicator can be expressed by

$$\log L_{PC}^{(miss)}(\theta, \boldsymbol{\xi}) = \sum_{j=1}^{n} \left[(1 - m_j) \log \{1 - q(y_j; \theta, \boldsymbol{\xi})\} + m_j \log q(y_j; \theta, \boldsymbol{\xi}) \right],$$

where $q(y_j; \theta, \boldsymbol{\xi})$ is a logistic function of the Shannon entropy $e_j(y_j; \theta)$, and m_j is a missing label indicator.

Value

`lk`	loglikelihood value

logsumexp *log summation of exponential function*

Description

log summation of exponential variable vector.

Usage

```
logsumexp(x)
```

Arguments

x	A variable vector.

Value

val	log summation of exponential variable vector.

makelabelmatrix *Label matrix*

Description

Convert class indicator into a label maxtrix.

Usage

```
makelabelmatrix(clust)
```

Arguments

clust	An n-dimensional vector of class partition.

Value

Z	A matrix of class indicator.

Examples

```
cluster<-c(1,1,2,2,3,3)
label_maxtrix<-makelabelmatrix(cluster)
```

neg_objective_function *Negative objective function for gmmssl*

Description

Negative objective function for gmmssl

Usage

```
neg_objective_function(
  dat,
  zm,
  g,
  par,
  ncov = 2,
  type = c("ign", "full", "com")
)
```

Arguments

dat	An $n \times p$ matrix where each row represents an individual observation
zm	An n-dimensional vector of group partition including the missing-label, denoted as NA.
g	Number of multivariate Gaussian groups.
par	An informative vector including mu, pi,sigma and xi.
ncov	Options of structure of sigma matrix; the default value is 2; ncov = 1 for a common covariance matrix; ncov = 2 for the unequal covariance/scale matrices.
type	Three types to fit to the model, 'ign' indicates fitting the model on the basis of the likelihood that ignores the missing label mechanism, 'full' indicates that the model to be fitted on the basis of the full likelihood, taking into account the missing-label mechanism, and 'com' indicate that the model to be fitted to a completed classified sample.

Value

val	Value of negatvie objective function.

normalise_logprob *Normalize log-probability*

Description

Normalize log-probability.

Usage

```
normalise_logprob(x)
```

Arguments

x	A variable vector.

Value

val	A normalize log probability of variable vector.

par2list *Transfer a vector into a list*

Description

Transfer a vector into a list

Usage

```
par2list(par, g, p, ncov = 2, type = c("ign", "full", "com"))
```

Arguments

par	A vector with list information.
g	Number of multivariate normal classes.
p	Dimension of observation vecor.
ncov	Options of structure of sigma matrix; the default value is 2; ncov = 1 for a common covariance matrix that sigma is a $p \times p$ matrix. ncov = 2 for the unequal covariance/scale matrices that sigma represents a list of g matrices with dimension $p \times p \times g$.
type	Three types to fit to the model, 'ign' indicates fitting the model on the basis of the likelihood that ignores the missing label mechanism, 'full' indicates that the model to be fitted on the basis of the full likelihood, taking into account the missing-label mechanism, and 'com' indicate that the model to be fitted to a completed classified sample.

Value

`parlist`	Return a list including `mu`, `pi`, `sigma` and `xi`.

---

paraextract *Extract parameter list from gmmsslmFit objects*

Description

This function extracts the parameters from a gmmsslmFit object, including p, g, pi, mu, and sigma.

Usage

```
paraextract(object)
```

Arguments

`object`	A gmmsslmFit object.

---

plot_missingness *Plot Missingness Mechanism and Boxplot*

Description

This function plots the smoothed values of '-log(entropy)' against the missingness mechanism and a boxplot of entropy for labeled vs. unlabeled observations.

Usage

```
plot_missingness(
  dat,
  g,
  parlist,
  zm,
  bandwidth = 5,
  range.x = c(0, 5),
  ylim = NULL,
  kernel = "normal"
)
```

Arguments

dat	An $n \times p$ matrix where each row represents an individual observation
g	Number of multivariate normal classes.
parlist	A list containing the required parameters (π, μ, Σ).
zm	An n-dimensional vector containing the class labels including the missing-label denoted as NA.
bandwidth	Bandwidth for kernel smoothing. Default is 5.
range.x	Range for x values. Default is c(0, 5).
ylim	The y-axis limits in the form of c(ylim[1], ylim[2]). Default is NULL.
kernel	Kernel type for smoothing. Default is 'normal'.

Value

A plot.

---

predict *Predict unclassified label*

Description

This function predicts unclassified label from a gmmsslmFit object.

Usage

```
predict(object)
```

Arguments

object	A gmmsslmFit object.

---

pro2vec *Transfer a probability vector into a vector*

Description

Transfer a probability vector into an informative vector

Usage

```
pro2vec(pro)
```

Arguments

pro	An propability vector

Value

y An informative vector

`rlabel` *Generation of a missing-data indicator*

Description

Generate the missing label indicator

Usage

```
rlabel(dat, pi, mu, sigma, xi)
```

Arguments

`dat`	An $n \times p$ matrix where each row represents an individual observation.
`pi`	A g-dimensional vector for the initial values of the mixing proportions.
`mu`	A $p \times g$ matrix for the initial values of the location parameters.
`sigma`	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if `sigma` is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.
`xi`	A 2-dimensional coefficient vector for a logistic function of the Shannon entropy.

Value

`m`	A n-dimensional vector of missing label indicator. The element of outputs `m` represents its label indicator is missing if m equals 1, otherwise its label indicator is available if m equals to 0.

Examples

```
n<-150
pi<-c(0.25,0.25,0.25,0.25)
sigma<-array(0,dim=c(3,3,4))
sigma[,,1]<-diag(1,3)
sigma[,,2]<-diag(2,3)
sigma[,,3]<-diag(3,3)
sigma[,,4]<-diag(4,3)
mu<-matrix(c(0.2,0.3,0.4,0.2,0.7,0.6,0.1,0.7,1.6,0.2,1.7,0.6),3,4)
dat<-rmix(n=n,pi=pi,mu=mu,sigma=sigma)
xi<-c(-0.5,1)
m<-rlabel(dat=dat$Y,pi=pi,mu=mu,sigma=sigma,xi=xi)
```

`rmix` *Normal mixture model generator.*

Description

Generate random observations from the normal mixture distributions.

Usage

```
rmix(n, pi, mu, sigma)
```

Arguments

`n`	Number of observations.
`pi`	A g-dimensional vector for the initial values of the mixing proportions.
`mu`	A $p \times g$ matrix for the initial values of the location parameters.
`sigma`	A $p \times p$ covariance matrix,or a list of g covariance matrices with dimension $p \times p \times g$. It is assumed to fit the model with a common covariance matrix if `sigma` is a $p \times p$ covariance matrix; otherwise it is assumed to fit the model with unequal covariance matrices.

Value

`Y`	An $n \times p$ numeric matrix with samples drawn in rows.
`Z`	An $n \times g$ numeric matrix; each row represents zero-one indicator variables defining the known class of origin of each.
`clust`	An n-dimensional vector of class partition.

Examples

```
n<-150
pi<-c(0.25,0.25,0.25,0.25)
sigma<-array(0,dim=c(3,3,4))
sigma[,,1]<-diag(1,3)
sigma[,,2]<-diag(2,3)
sigma[,,3]<-diag(3,3)
sigma[,,4]<-diag(4,3)
mu<-matrix(c(0.2,0.3,0.4,0.2,0.7,0.6,0.1,0.7,1.6,0.2,1.7,0.6),3,4)
dat<-rmix(n=n,pi=pi,mu=mu,sigma=sigma)
```

summary *Summary method for gmmsslmFit objects*

Description

This function extracts summary information from a gmmsslmFit object, including objective value, ncov, convergence, iteration, and type.

Usage

```
summary(object)
```

Arguments

object	A gmmsslmFit object.

vec2cov *Transform a vector into a matrix*

Description

Transform a vector into a matrix i.e., Sigma=R^T*R

Usage

```
vec2cov(par)
```

Arguments

par	A vector representing a variance matrix

Details

The variance matrix is decomposed by computing the Choleski factorization of a real symmetric positive-definite square matrix. Then, storing the upper triangular factor of the Choleski decomposition into a vector.

Value

sigma A variance matrix

vec2pro *Transfer an informative vector to a probability vector*

Description

Transfer an informative vector to a probability vector

Usage

```
vec2pro(vec)
```

Arguments

vec	An informative vector

Value

pro A probability vector

Index