Package 'mmeln'

May 9, 2026

Type Package

Title Estimation of Multinormal Mixture Distribution

Version 1.5

Date 2023-09-11

Author Charles-Edouard Giguere

Maintainer Charles-Edouard Giguere <ce.giguere@gmail.com>

Description Fit multivariate mixture of normal distribution using covariance structure.

License GPL-3

LazyLoad yes

Repository CRAN

Encoding UTF-8

NeedsCompilation no

Date/Publication 2023-09-13 03:00:02 UTC

Contents

mmeln-package 2
dmnorm 3
estim 4
exY 6
mmeln 6
plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln 8
post.mmeln,entropy.mmeln 10

Index **12**

`mmeln-package` *Estimation of Multinormal Mixture Distribution*

Description

Fit multivariate mixture of normal distribution using covariance structure.

Details

The DESCRIPTION file:

Package:	mmeln
Type:	Package
Title:	Estimation of Multinormal Mixture Distribution
Version:	1.5
Date:	2023-09-11
Author:	Charles-Edouard Giguere
Maintainer:	Charles-Edouard Giguere <ce.giguere@gmail.com>
Description:	Fit multivariate mixture of normal distribution using covariance structure.
License:	GPL-3
LazyLoad:	yes
Repository:	CRAN
Encoding:	UTF-8
NeedsCompilation:	no

Index of help topics:

```
dmnorm                  Multivariate Normal Density Function
estim                   Maximum Likelihood estimation of the model
                        parameters
exY                     A two mixture example
mmeln                   mmeln : mixture of multivariate normal
mmeln-package           Estimation of Multinormal Mixture Distribution
plot.mmeln              Utility methods for objects of class mmeln
post.mmeln              Posterior probabilities, entropy for mmeln
                        object
```

This package allows to fit multivariate normal mixture. Predictors of group membership can also be allowed in the model.

Author(s)

Charles-Edouard Giguere

Maintainer: Charles-Edouard Giguere <ce.giguere@gmail.com>

See Also

mmeln,estim.mmeln,anova.mmeln

Examples

```
### load an example.
data(exY)

### estimation of the parameters of the mixture.

temps <- factor(1:3)
mmeln1 <- mmeln(Y, G = 2, form.loc = ~temps-1, form.mel = ~1, cov = "CS")
mix1 <- estim(mmeln1, mu = list(rep(1,3), rep(2,3)), tau = c(0),
              sigma = list(c(1,.6), c(1,.6)), iterlim = 100,tol = 1e-6)
mix1
anova(mix1)
plot(mix1,main="Mixture of multivariate normal")
```

---

dmnorm *Multivariate Normal Density Function*

Description

Function to estimate Multivariate Normal Density Function

Usage

```
dmnorm(X, Mu, Sigma)
```

Arguments

X	A matrix or a vector (if you have only one multivariate observation) containing the data. This matrix may contain missing data.
Mu	A mean vector or a matrix where the number of column is p. If Mu is a matrix and X a vector, the density is evaluated for each value of Mu specified in the matrix Mu
Sigma	The covariance matrix. This matrix must be symmetric positive definite(all eigen values are positive. see eigen)

Details

This methods compute the value of the density function for a given data and a given set of parameters. It works like the R command dnorm in the stats package. Although this methods can be used directly it is not intended this way. If you want to estimate density of multivariate normal distribution, the library mvtnorm is more appropriate

Value

This command return a vector of density.

Note

This function can be used as a standalone but is implemented here for use within the mmeln package

Author(s)

Charles-Édouard Giguère

References

M.S. Srivastava (2002), Methods of Multivariate Statistics, WILEY

See Also

mmeln,eigen

Examples

```
dmnorm(1:3,1:3,diag(3))
```

`estim` *Maximum Likelihood estimation of the model parameters*

Description

Compute the MLE of the model parameters using the E-M (Expectation-Maximization) algorithm

Usage

```
## S3 method for class 'mmeln'
estim(X,...,mu=NULL,tau=NULL,sigma=NULL,random.start=FALSE,iterlim=500,tol=1e-8)
```

Arguments

`X`	An object of type mmeln containing the design of the model, see mmeln
`...`	For the moments no other arguments can be added
`mu`	A list of length X$G containing the starting value for the location parameters
`tau`	The starting value for the mixture parameters
`sigma`	A list of length X$G containing the starting value for the covariances parameters
`random.start`	A True/False value indicating if the starting parameters should be given at random. If true the starting values are not needed.
`iterlim`	The maximum number of iterations allowed
`tol`	Tolerance, degree of precision required to stop the iterative process

Details

Methods estim.mmeln... are used by the estim function but are of no use outside this method.

Value

Retourne un objet de type "mmeln" & "mmelnSOL" les arguments suivants :

`obj$Y`	The data matrix
`obj$G`	The number of groups
`obj$p`	Number of column in Y
`obj$N`	Number of row in Y
`obj$Xg`	The list of location design matrices
`obj$pl`	The number of location parameters
`obj$Z`	Mixture design matrix
`obj$pm`	The number of mixture parameters
`obj$cov`	Covariance type
`obj$equalcov`	logical value indicating if covariance is equal across group
`obj$pc`	The number of covariance parameters

Author(s)

Charles-Édouard Giguère

References

McLachlan, G. & Peel, D. (2000), Finite mixture models,Wiley

Flury, B. D. (1997), A first course in multivariate statistics, Springer

Pinheiro J. C. and Bates D. M. (2000), Mixed-Effects Models in S and S-PLUS, Springer

Srivastava, M.S. (2002), Methods of Multivariate Statistics, WILEY

Lindstrom M. J. and Bates D. M. (1988), Newton-Raphson and EM Algorithms for Linear Mixed-Effects Models for Repeated-MeasuresData, Journal of the American Statistical Association,American Statistical Association,V. 83,I. 404, P. 1014-1022

See Also

mmeln.package

Examples

```
data(exY)
### estimation of the parameters of the mixture
temps=0:2
mmeln1=mmeln(Y, G = 3, form.loc = list(~temps, ~temps + I(temps^2),
                       ~temps + I(temps^2)), form.mel = ~SEXE, cov = "CS")
mmelnSOL1=estim(mmeln1,mu = list(c(1,1), c(2,0,0), c(3,0,0)),
    tau = c(0,0,0,0), sigma = list(c(1,0), c(1,0), c(1,0)))
```

exY *A two mixture example*

Description

A simulated dataset used for example

Format

Two variables are available:

SEXE A variable identifying sex of participants.

Y A three column matrix containing the data.

Details

Half of the row follow the distribution N[(2,3,4)',matrix(c(1,.6,.5,.6,1,.3,.5,.3,1),3,3))], the other half follow the distribution N[(-1,5,-2)',matrix(c(1,.6,.5,.6,1,.3,.5,.3,1),3,3))]

mmeln *mmeln : mixture of multivariate normal*

Description

constructor for objects of class mmeln: mixture of multivariate normal

Usage

```
mmeln(Y,G=2,p=dim(Y)[2],form.loc=NULL,X=NULL,
form.mel=NULL,Z=NULL,cov="IND",equalcov=FALSE,param=NULL)
```

Arguments

Y	A matrix containing the data used for estimation. This matrix may contains NA but it needs at least one observation per row.It's assumed that the missing mechanism is not related to the data under study (MAR: Missing At Random).
G	The number of groups in the mixture
p	Doesn't need to be specified. It's the dimension of the multivariate data (number of column in Y)
form.loc, X	Location design of the model. By default, the mean model is used where we estimate p mean in each group. Only one of these two parameters must be specified depending if the model is specified through a formula (See R documentation) or a design matrix. If you want to specify a different design for each group you must pass the arguments as a list. See examples below for further details. If a formula is used it must use variable of length p representing the design across time, for example : ~temps where temps=factor(1:4). If a design matrix is used, it must be of dimension p*k where k<=p

`form.mel, Z` Mixture design of the model. Only one of these two parameters must be specified. The design is constant across groups. This is equivalent to multinomial regression

`cov` Covariance type (for now only the CS structure is implemented). Enter either the type of covariance as a string or as numeric corresponding to the position in the following choices : 1)UN (general unstructured covariance),2 CS (Compound Symmetry with constant variance) ,3) UCS (Compound Symmetry with unconstant variance) ,4) AR1 (Auto-regressive of order 1 with constant variance), 5) UAR1 (Auto-regressive of order 1 with unconstant variance),6) IND: (diagonal structure with constant variance), 7) UIND (diagonal structure with unconstant variance)

`equalcov` Logical value T/F indicating if the variance is equal across groups. Default to FALSE.

`param` list of list of parameters. Usually not specified. The parameters should be estimated through the estim.mmeln function. param will look like this list(mu=list(mu1,mu2,...,mug) ,tau=c(tau1,...,tauk),sigma=list(sigma1,sigma2,...,sigmag)) where mui is the vector of location parameter in the group i and sigmai is the vector of location parameter in the group i for which the length must equal the number of column in the design matrix. Also sigmai is the vector of covariance parameters in the group i. Each covariance is parameterized in a vector containing first the distinct value of standard deviation and then the distinct value of correlation from top to bottom and left to right.

Details

This object describes the way the mixture is design and permits a lot of different modelisation of the data. Many specific methods are associated with this class of objects: print, anova, logLik, post. Once a solution is find through the estim.mmeln function, the object is promoted to an object of class mmelnSOL but inherits of all the attributes and function of the mmeln class but gains is own print method. The attributes in a mmeln object should be accessed via adequate function inside the mmeln library except if handle by an advanced user.

Value

Retourne un objet de type "mmeln" ayant les arguments suivants :

`obj$Y` The data matrix

`obj$Yl` A list of length N containing the data in each row without the NA value.

`obj$Yv` A list of length N indicating the column where there is valid data

`obj$G` The number of groups

`obj$p` Number of column in Y

`obj$pi` A vector where pi[i] is the number of observation in row i

`obj$N` Number of row in Y

`obj$M` Number of total observations sum_i=1^N(pi)

`obj$Xg` The list of location design matrices

`obj$pl` The number of location parameters

obj$Z	Mixture design matrix
obj$pm	The number of mixture parameters
obj$cov	Covariance type
obj$equalcov	logical value indicating if covariance is equal across group
obj$pc	The number of covariance parameters

Author(s)

Charles-Édouard Giguère

References

McLachlan, G. & Peel, D. (2000), Finite mixture models,Wiley

Bernard D. Flury (1997), A first course in multivariate statistics, Springer

Pinheiro José C. & Bates Douglas M. (2000), Mixed-Effects Models in S and S-PLUS, Springer

M.S. Srivastava (2002), Methods of Multivariate Statistics, WILEY

See Also

estim.mmeln

Examples

```
data(exY)
### estimation of the parameters of the mixture
temps <- 0:2
mmeln1 <- mmeln(Y, G = 3,
                form.loc = list(~temps, ~temps + I(temps^2), ~temps + I(temps^2)),
                form.mel = ~SEXE, cov = "CS")
```

---

plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln

Utility methods for objects of class mmeln

Description

Methods to plot, compare and assessed the log(Likelihood) of objects of class mmeln. The method cov.tsf which convert a vector of covariance parameter into a covariance matrix and multnm which performs an estimation of multinomial model are internal methods that should not be used unless by experimented user

Usage

```
## S3 method for class 'mmeln'
plot(x,...,main="",xlab="Temps",ylab="Y",col=1:x$G,leg=TRUE)
## S3 method for class 'mmeln'
logLik(object,...,param=NULL)
## S3 method for class 'mmeln'
anova(object, ..., test = TRUE)
## S3 method for class 'mmelnSOL'
print(x,...,se.estim="MLR")
cov.tsf(param,type,p)
```

Arguments

x	An object of type mmeln or mmelnSOL (mmelnSOL required for the command print)
object	An object of type mmeln
main	Title of the graphic
xlab	Label of the X axis
ylab	label of the Y axis
col	Colour of the lines plotted in each group
leg	Logical value indicating if the legend is plotted or not
...	other object of type mmeln to compare (use is only valid in the anova command)
test	logical value indicating if the likelihood ratio test is required. Valid only when two objects are entered
param	For the function logLik a list of parameters like defined in mmeln, by default it is taken from the mmeln object. For the cov.tsf function it is vector containing the distinct value of the covariance as defined in the mmeln function
type	Type of covariance as defined in mmeln
p	Rank of covariance matrix
se.estim	Type of estimator. The default is MLR based on the information matrix define as Ir^(-1)=I^(-1)IeI^(-1). The other choices are the Observational information matrix "ML" and the Empirical information matrix based on the cross product of the gradient of the logLikehood "ML.E"

Details

The function plot draws X$G lines showing the expected value. The function logLik gives the log(Likelihood) of a model. The function anova compares mmeln models and gives the total number of parameters, the log(Likelihood), the AIC (Akaike information criterion), the BIC (Bayesian information criterion based on the number of observation) and the BIC2 (BIC based on the number of subjects). Optionally, the Likelihood ratio test is performed. The function print is used for solution given by the estim.mmeln function. The print method gives the number of iterations required for convergence and the statistics for the location, mixture and covariance parameters.

Author(s)

Charles-Édouard Giguère

References

McLachlan, G. & Peel, D. (2000), Finite mixture models,Wiley

Bernard D. Flury (1997), A first course in multivariate statistics, Springer

Pinheiro José C. & Bates Douglas M. (2000), Mixed-Effects Models in S and S-PLUS, Springer

M.S. Srivastava (2002), Methods of Multivariate Statistics, WILEY

See Also

mmeln

Examples

```
#### load an example.
data(exY)

### estimation of the parameters of the mixture
temps=1:3
mmeln1=mmeln(Y,G=2,form.loc=~factor(temps)-1,form.mel=~1,cov="CS")
mmeln2=mmeln(Y,G=2,form.loc=list(~temps,~I((temps-2)^2)),form.mel=~1,cov="CS")

mix1=estim(mmeln1,mu=list(rep(1,3),rep(2,3)),tau=c(0)
          ,sigma=list(c(1,.4),c(1,.4)),iterlim=100,tol=1e-6)

mix2=estim(mmeln2,mu=list(c(2,1),c(5,-1)),tau=c(0)
          ,sigma=list(c(1,.4),c(1,.4)),iterlim=100,tol=1e-6)


mix1
mix2

anova(mix1,mix2)
plot(mix1,main="Mixture of multivariate normal")
plot(mix2,main="Mixture of multivariate normal")
```

`post.mmeln,entropy.mmeln` *Posterior probabilities, entropy for mmeln object*

Description

Compute the posterior probabilities of membership in each group of the mixture

Usage

```
## S3 method for class 'mmeln'
post(X,...,mu=X$param$mu,tau=X$param$tau,sigma=X$param$sigma)
## S3 method for class 'mmeln'
entropy(X,...)
```

Arguments

X	An object of type mmeln containing the design of the model.
...	These parameters are useless
mu	Location parameters. By default, those are taken from X
tau	Mixture parameters. By default, those are taken from X
sigma	Covariance parameters. By default, those are taken from X

Details

This procedure returns the posterior probabilities of membership in each groups or the entropy of the model. They were computed as described in McLachlan and Peel (2000). If the parameters X$param is not null no further parameters are necessary, otherwise you have to give a value for mu, tau, sigma (this is mainly used inside the estim.mmeln function)

Value

Returns a matrix P with X$N row and X$G column where P[i,j] is the posterior probabilities of subject i being in the group j or the value of entropy.

Author(s)

Charles-Édouard Giguère

References

McLachlan, G. & Peel, D. (2000), Finite mixture models,Wiley

See Also

estim.mmeln

Examples

```
#### load an example.
data(exY)

### estimation of the parameters of the mixture
temps <- factor(1:3)
mmeln1 <- mmeln(Y, G = 2, form.loc = ~temps - 1, form.mel = ~1, cov = "CS")
mix1 <- estim(mmeln1, mu = list(rep(1,3),rep(2,3)), tau = c(0),
              sigma = list(c(1, .4), c(1, .4)), iterlim = 100, tol = 1e-6)
post(mix1)
entropy(mix1)
```

Index

∗ **datasets**
exY, 6
∗ **density**
dmnorm, 3
∗ **mixture**
estim, 4
mmeln, 6
plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln, 8
post.mmeln,entropy.mmeln, 10
∗ **mmeln**
estim, 4
mmeln, 6
plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln, 8
post.mmeln,entropy.mmeln, 10
∗ **multivariate normal**
dmnorm, 3
∗ **multivariate**
estim, 4
mmeln, 6
plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln, 8
post.mmeln,entropy.mmeln, 10
∗ **normal**
estim, 4
mmeln, 6
plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln, 8
post.mmeln,entropy.mmeln, 10
∗ **package**
mmeln-package, 2

anova.mmeln *(plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln)*, 8

cov.tsf *(plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln)*, 8

covNA.wt *(estim)*, 4

dmnorm, 3
dnorm, *3*

eigen, *3*
entropy *(post.mmeln,entropy.mmeln)*, 10
estim, 4
estim.mmeln, *7–9*, *11*
estimloc.disp.CS1 *(estim)*, 4
estimloc.disp.IND1 *(estim)*, 4
estimmmelnCS1 *(estim)*, 4
estimmmelnIND1 *(estim)*, 4
exY, 6

I.CS1 *(estim)*, 4
I.IND1 *(estim)*, 4
IE.CS1 *(estim)*, 4
IE.IND1 *(estim)*, 4

logit *(estim)*, 4
logLik.mmeln *(plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln)*, 8

mmeln, *4*, 6, *9*, *10*
mmeln-package, 2
mmeln.package, *5*
mmelnpackage *(mmeln-package)*, 2
multnm *(plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln)*, 8

pfQ.intermediate.CS1 *(estim)*, 4
plot.mmeln *(plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln)*, 8
plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln, 8
post.mmeln *(post.mmeln,entropy.mmeln)*, 10
post.mmeln,entropy.mmeln, 10

print.mmelnSOL
*(*plot.mmeln,logLik.mmeln,anova.mmeln,print.mmeln*)*, 8

SEXE *(*exY*)*, 6

Xinv *(*estim*)*, 4

Y *(*exY*)*, 6