Package ‘EMMIXuskew’

February 19, 2015

Type Package
Title Fitting Unrestricted Multivariate Skew t Mixture Models
Version 0.11-6
Date 2014-07-31
Author S.X. Lee and G.J. McLachlan
Maintainer Sharon X. Lee <s.lee11@uq.edu.au>
Description Functions to fit finite mixture of unrestricted multivariate skew t (FM-uMST) model, random sample generation, discriminant analysis, 2D and 3D contour plots
License GPL
LazyLoad yes
Suggests rgl, stats, grDevices, KernSmooth
Depends MASS, graphics
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-08-05 07:55:01

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMMIXuskew-package</td>
<td>2</td>
</tr>
<tr>
<td>ais</td>
<td>3</td>
</tr>
<tr>
<td>delta.test</td>
<td>4</td>
</tr>
<tr>
<td>dfmmt</td>
<td>5</td>
</tr>
<tr>
<td>DLBCL</td>
<td>7</td>
</tr>
<tr>
<td>dmt</td>
<td>8</td>
</tr>
<tr>
<td>fmmst</td>
<td>10</td>
</tr>
<tr>
<td>fmmst.contour.2d</td>
<td>12</td>
</tr>
<tr>
<td>fmmst.init</td>
<td>14</td>
</tr>
<tr>
<td>fmmstDA</td>
<td>16</td>
</tr>
<tr>
<td>fmmt</td>
<td>17</td>
</tr>
<tr>
<td>Lympho</td>
<td>19</td>
</tr>
<tr>
<td>rfmmst</td>
<td>20</td>
</tr>
</tbody>
</table>

Index 22
Description

The package implements an EM algorithm for fitting mixtures of unrestricted multivariate skew t (FM-uMST) distributions. Functions for random sample generation, discriminant analysis, and visualisation (in 2D and 3D) is also provided.

Details

- Package: EMMIXuskew
- Type: Package
- Version: 0.11-6
- Date: 2014-07-31
- Licence: GPL
- LazyLoad: yes

Author(s)

S.X. Lee, G.J. McLachlan

References


See Also

fmmst, dfmmst, rfmmst, fmmst.contour.3d
Description

Data on 102 male and 100 female athletes collected at the Australian Institute of Sport.

Usage

data(ais)

Format

A data frame with 202 observations (rows) on the following 14 variables (columns).

- Sex  0 = male or 1 = female
- Ht  Height in cm
- Wt  Weight in kg
- LBM Lean body mass
- RCC Red cell count
- WCC White cell count
- Hc  Hematocrit
- Pfr  Plasma ferritin concentration
- BMI Body mass index = weight / (height^2)
- SSF sum of skin folds
- Bfat Percent body fat

Label case labels:

- f-b_ball f-field f-gym f-netball f-row f-swim f-t_400m f-t_sprnt
- f-tennis m-b_ball m-field m-row m-swim m-t_400m m-t_sprnt m-tennis m-w_polo

Sport: f-b_ball f-field f-gym f-netball f-row f-swim f-t_400m f-t_sprnt f-tennis m-b_ball m-field m-row m-swim m-t_400m m-t_sprnt m-tennis m-w_polo

Source

Richard Telford and Ross Cunningham, Australian National University.

References


Examples

data(ais)
pairs(ais[2:12], main = "AIS Data", pch = 21,
    bg = c("red", "blue")[[unclass(factor(ais$Sex))]], upper.panel=NULL)
legend(0.8, 0.8, legend=c("male", "female"), pt.bg = c("red", "blue"), pch=21)
delta.test  

Testing for the significance of the skewness parameter in a FM-MST model

Description

Perform a likelihood ratio for the significance of the skewness parameter delta in a multivariate skew t-mixture model.

Usage

delta.test(stmodel=NULL, tmodel=NULL, stloglik, tloglik, r)

Arguments

stmodel  
a list containing the parameters of the FM-MST model, including mu, sigma, delta, dof and pro. This is usually an output from a fmmst run.

tmodel  
a list containing the parameters of the FM-MT model, including mu, sigma, dof and pro. This is usually an output from a fmmt run.

stloglik  
a scalar specifying the log likelihood value of the skew t-mixture model

tloglik  
a scalar specifying the log likelihood value of the t-mixture model

r  
a scalar specifying the difference in the number of parameters between FM-MST and FM-MT model

Details

A likelihood ratio test for hypotheses:

H0: delta = 0 (for all components in the mixture model)

H1: delta different from 0 (for at least one component in the mixture)

The test statistics is LR = -2 (L1 - L2), which follows a chi-squared distribution with r degrees of freedom under H0. r is the difference between the number of parameters in H0 and H1. See references for further details.

Value

returns the P-value of the test

References


See Also

fmmt, fmmst
**dfmmst**

**Examples**

delta.test(stloglik=-1343.541, tloglik=-1353.842, r=4)

---

**dfmmst**  
*Multivariate skew t distribution*

**Description**

The probability density function for the unrestricted multivariate skew t (MST) distribution and finite mixture of MSN and MST distributions

**Usage**

dfmmst(dat, mu = NULL, sigma = NULL, delta = NULL, dof = NULL, pro = NULL, known = NULL, tmethod=1)  
dmst(dat, mu = NULL, sigma = NULL, delta = NULL, dof = 1, known = NULL, tmethod=1)

**Arguments**

dat the data matrix giving the coordinates of the point(s) where the density is evaluated. This is either a vector of length p or a matrix with p columns.

mu for *dmst*, this is a numeric vector of length p representing the location parameter; for *dfmmst*, this is list of g numeric matrices each having p rows and 1 column containing the location parameter for each component.

sigma for *dmst*, this is a numeric positive definite matrix with dimension (p,p) representing the scale parameter; for *dfmmst*, this is list of g numeric matrices containing the scale parameter for each component.

delta for *dmst*, this is a numeric vector of length p representing the skewness parameter; for *dfmmst*, this is list of g numeric matrices each having p rows and 1 column containing the skewness parameter for each component.

dof for *dmst*, this is a positive integer specifying the degrees of freedom; for *dfmmst*, this is numeric vector of length g representing the degrees of freedom for each component.

pro the mixing proportions; for *dmst*, this is equal to 1; for *dfmmst*, this is vector of length of g specifying the mixing proportions for each component.

known a list containing the parameters of the model. If specified, it overwrites the values of mu, sigma, delta, dof and pro.

tmethod (optional) an integer indicating which method to use when computing t distribution function values. See *pmt* for details.
Details

The function `dmst` computes the density value of a specified unrestricted multivariate skew t (MST) distribution. If any model parameters are not specified, their default values are used: `mu` and `delta` are zero vectors, `sigma` is the identity matrix, and `dof` is 1.

The function `dfmmst` computes the density value for a specified mixture of MST distribution. Note that `dfmmst` expects at least `dof` is specified. Other missing parameters will take the default value described above. When `g=1`, `dfmmst` passes the call to `dmst`. Model parameters can be passed to `dmst` and `dfmmst` through the argument `known` or listed as individual arguments. If both methods of input were used, the parameters specified in `known` will be used.

Value

`dmst` and `dfmmst` returns a numeric vector of density values.

References


See Also

`rmst`, `rfmmst`

Examples

dmst(c(1,2), mu=c(1,5), sigma=diag(2), delta=c(-3,1), dof=4)
obj <- list()
obj$mu <- list(c(17,19), c(5,22), c(6,10))
obj$sigma <- list(diag(2), matrix(c(2,0,0,1),2), matrix(c(3,7,7,24),2))
obj$delta <- list(c(3,1.5), c(5,10), c(2,0))
obj$dof <- c(1,2,3)
obj$pro <- c(0.25, 0.25, 0.5)
dfmmst(matrix(c(1,2,5,6,2,4),3,2), obj$mu, obj$sigma, obj$delta, obj$dof, obj$pro)
dfmmst(c(1,2), known=obj)
A Diffuse Large B-cell Lymphoma (DLBCL) data

Description
A sample from the Diffuse Large B-cell Lymphoma (DLBCL) dataset from Aghaeepour et al. (2013). The original data contain measurements from biopsies of 30 DLBCL patients. Each sample was stained with three antibodies, CD3, CD5, and CD19. This is a subset from one patient.

Usage
data(DLBCL)

Format
DLBCL is a data frame with over 8000 observations (rows) on the following 3 markers (rows).

CD3 marker 1
CD5 marker 2
CD19 marker 3

Details
DLBCL is a data frame as described above. true.clusters is a set of cluster labels given by manual gating.

Source
The raw data is available from the FlowRepository database. https://flowrepository.org/id/FR-FCM-ZZY

References


Examples

data(DLBCL)
## Not run:
RNGversion("3.0.2"); set.seed(240)
Fit <- fmmst(4, DLBCL, nkmeans=1)
fmmst.contour.3d(DLBCL, model = Fit, level = 0.985, drawpoints = FALSE,
                 xlab="CD3", ylab="CDS", zlab="CD19", component=1:4)
## End(Not run)

dmt

Multivariate t distribution

Description

The probability density function and distribution function for the multivariate Student t distribution
and mixtures of multivariate t distribution

Usage

dmt(dat, mu, sigma, dof = Inf, log = FALSE)
pmt(dat, mu=rep(0,length(dat)), sigma=diag(length(dat)), dof=Inf, method=1, ...)
dfmmt(dat, mu = NULL, sigma = NULL, dof = NULL, pro = NULL, known = NULL)

Arguments

dat       for dmt, this is the data matrix giving the coordinates of the point(s) where the
density is evaluated. for pmt, this is either a vector of length p. Currently, only
p up to 20 dimensions is supported.
mu        a numeric vector of length p representing the location parameter;
sigma     a numeric positive definite matrix with dimension (p,p) representing the scale
          parameter
dof       a positive real number specifying the degrees of freedom. If tmethod=1, dof
          will be rounded to the nearest integer.
pro        the mixing proportions; for dmt, this is equal to 1; for dfmmt, this is vector of
          length of g specifying the mixing proportions for each component.
log        a logical value; if TRUE, the logarithm of the density is computed
...        parameters passed to sadmv, among maxpts, absrel, releps
known      a list containing the parameters of the model. If specified, it overwrites the
            values of mu, sigma, dof and pro.
method     the method to use for computation of t distribution function. See description.
**Details**

There are three options in `pmt` for computing multivariate t distribution function values. `method=1` uses requires `dof` to be an integer. This provide interfaces to the Fortran-77 routines by Alan Genz. This is the fastest method of the three options available. `method=2` uses linear interpolation technique to calculate t distribution function values for a positive real `dof`. This method requires double the time of method 1. `method=3` uses a method described in Genz and Bretz (2002). This is the more accurate method for a non-integer `dof`, but more computationally intensive than the other two methods.

**Value**

The function `dmt` computes the density value of a specified multivariate t distribution. `pmt` computes the distribution value for a SINGLE point. `dfmmmt` returns a numeric vector of mixture density values.

**References**


**See Also**

`dmst, dfmmst`

**Examples**

```r
x <- seq(-2,4,length=21)
y <- 2*x+10
z <- x+cos(y)
mu <- c(1,12,2)
sigma <- matrix(c(1,2,0,2,5,0.5,0,0.5,3, 3), 3, 3)
dof <- 4
f <- dmt( cbind(x,y,z), mu, sigma,dof)
## Not run:
p1 <- pmt(c(2,11,3), mu, sigma, dof)
p2 <- pmt(c(2,11,3), mu, sigma, dof, maxpts=10000, abseps=1e-8)

## End(Not run)
obj <- list()
obj$mu <- list(c(17,19), c(5,22), c(6,10))
obj$sigma <- list(diag(2), matrix(c(2,0,0,1),2), matrix(c(3,7,7,24),2))
obj$dof <- c(1, 2, 3)
obj$pro <- c(0.25, 0.25, 0.5)
dfmmmt(matrix(c(1,2,5,6,2,4),3,2), obj$mu, obj$sigma, obj$dof, obj$pro)
dfmmmt(c(1,2), known=obj)
```
Fitting Finite Mixtures of Unrestricted Multivariate Skew t Distributions

Description

Computes maximum likelihood estimators (MLE) for finite mixtures of unrestricted multivariate skew t (FM-MST) model via the EM algorithm.

Usage

```r
fmmst(g = 1, dat, initial = NULL, known = NULL, itmax = 100,
      eps = 1e-03, clust=NULL, nkmeans=20, print = T, tmethod=1)
```

## S3 method for class 'fmmst'
summary(object, ...)
## S3 method for class 'fmmst'
print(x, ...)

Arguments

- **object, x**: an object class of class "fmmst", i.e. a fitted model.
- **g**: a scalar specifying the number of components in the mixture model.
- **dat**: the data matrix giving the coordinates of the point(s) where the density is evaluated. This is either a vector of length p or a matrix with p columns.
- **initial**: (optional) a list containing the initial parameters of the mixture model. See the 'Details' section. The default is NULL.
- **known**: (optional) a list containing parameters of the mixture model that are known and not required to be estimated. See the 'Details' section. The default is NULL.
- **itmax**: (optional) a positive integer specifying the maximum number of EM iterations to perform. The default is 100.
- **eps**: (optional) a numeric value used to control the termination criteria for the EM loops. It is the maximum tolerance for the absolute difference between the log-likelihood value and the asymptotic log likelihood value. The default is 1e-6.
- **clust**: (optional) a numeric value of length nrow(dat) containing the initial labels for each data point in dat. The default is NULL, indicating no initial clustering is known.
- **nkmeans**: (optional) a numeric value indicating how many k-means trials to be used when searching for initial values. The default is 20.
- **print**: (optional) a logical value. If TRUE, output for each iteration will be printed out. if FALSE, no output is printed. The default is TRUE. See the 'Details' section.
- **tmethod**: (optional) an integer indicating which method to use when computing t distribution function values. See `pmt` for details.
- **...**: not used.
Details

The arguments `init` and `known`, if specified, is a list structure containing at least one of `mu`, `sigma`, `delta`, `dof`, `pro` (See `dfmmst` for the structure of each of these elements). If `init=FALSE` (default), the program uses an automatic approach based on k-means clustering to generate an initial value for the model parameters. Note that this may not provide the best results.

As the EM algorithm is sensitive to the starting value, it is highly recommended to apply a wide range different initializations. A simple strategy is implemented in `fmmst.init`.

Value

- `mu` a list of `g` numeric matrices containing the location parameter for each component.
- `sigma` a list of `g` numeric matrices containing the scale parameter for each component.
- `delta` a list of `g` numeric matrices containing the skewness parameter for each component.
- `dof` a numeric vector of length `g` representing the degrees of freedom for each component.
- `pro` a vector of length of `g` specifying the mixing proportions for each component.
- `tau` an `g` by `n` matrix of posterior probability of component membership.
- `clusters` a vector of length `n` of final partition.
- `loglik` the final log likelihood value.
- `lk` a vector of log likelihood values at each EM iteration.
- `iter` number of iterations performed.
- `eps` the final absolute difference between the log likelihood value and the asymptotic log likelihood value.
- `aic, bic` Akaike Information Criterion (AIC), Bayes Information Criterion (BIC)

References


See Also

`fmmst.init`, `rfmmst`, `dfmmst`, `fmmst.contour.2d`
Examples

# a short demo using AIS data
data(ais)
Fit <- fmmst(2, ais[,c(2,12)], itmax=5)
summary(Fit)
print(Fit)

fmmst.contour.2d 2D and 3D Visualisation of Fitted Contours

Description

Create 2D or 3D contour plot.

Usage

fmmst.contour.2d(dat, model, grid = 50, drawpoints = TRUE, clusters=NULL, levels = 10,
map = c("scatter", "heat", "cluster"), component = NULL,
xlim, ylim, xlab, ylab, main, tmethod=1, ...)
fmmst.contour.3d(dat, model, grid=20, drawpoints=TRUE, levels=0.9,
clusters=NULL, xlim, ylim, zlim, xlab, ylab, zlab, main, component=NULL, ...)

Arguments

dat the data matrix giving the coordinates of the point(s) where the density is evaluated. This must be a matrix with at least 2 columns for fmmst.contour.2d or 3 columns for fmmst.contour.3d. If dat is not provided, then xlim, ylim and zlim must be provided, and drawpoints must be set to FALSE.

model a list containing the parameters of the model and also a vector of cluster labels for dat. This is typically an output from fmmst, containing mu, sigma, delta, dof, pro and clusters; see fmmst for structure of model.

grid a positive integer specifying the grid size used to calculate the density map.
drawpoints logical. Points are plotted if TRUE.
clusters a vector of cluster labels to be applied when colouring the points. This only applies when drawpoints is TRUE.
levels either a positive integer specifying the number of contour levels to draw or a numeric vector of contour levels to be drawn
map character string specifying how to plot the points if drawpoints=TRUE. Possible values are "scatter" (default), "heat" and "cluster". See the 'Details' section.
component the index of the components to be plotted. See the 'Details' section.
xlim, ylim, zlim x-, y- and z- limits for the plot
fmmst.contour.2d

xlab, ylab, zlab
labels for x-, y- and z- axis
main
  title of the plot
tmethod
  (optional) an integer indicating which method to use when computing t distribution function values. See pmt for details.
...additional arguments to plot.default

Details
fmmst.contour.2d draw contour plots for bivariate densities. The argument dat must be provided and must contain at least 2 columns. Note that only the first two columns of dat will be used if dat have more than 2 columns. For bivariate dataset, the data points can be drawn as a scatter plot by specifying map="scatter" (default), or as an intensity plot (map="heat"). Alternatively, a cluster map can be drawn instead (map="cluster"). Note that if an intensity plot is used, the data points will not be drawn, that is, drawpoints will be set to FALSE.

The argument component specifies which individual component is drawn. When component=FALSE, the mixture contour is drawn. If specified, component is a integer vector of the index of the components to be drawn. It can only take values between 1 an g inclusive. For example, component=c(1,3) will draw the first and third component contours.

If the argument model contains the cluster labels (model$clusters), the data point will be coloured according to their cluster.

See Also
fmmst, contour

Examples

#2D plots
obj <- list()
obj$mu <- list(c(17, 19), c(5, 22), c(6, 10))
obj$sigma <- list(diag(2), matrix(c(2, 0, 0, 1), 2, matrix(c(3, 7, 7, 24), 2))
obj$delta <- list(c(3, 1.5), c(5, 10), c(2, 0))
obj$dof <- c(1, 2, 3)
obj$pro <- c(0.25, 0.25, 0.5)
mySample <- rfmmst(3, 500, known=obj)
obj$clusters <- mySample[,3]

par(mfrow=c(2,2))
fmmst.contour.2d(mySample, model=obj, clusters=obj$clusters)
fmmst.contour.2d(mySample[,1:2], model=obj, clusters=obj$clusters, map="heat")
fmmst.contour.2d(mySample[,1:2], model=obj, clusters=obj$clusters, map="cluster")
fmmst.contour.2d(mySample[,1:2], model=obj, clusters=obj$clusters, component=1)

#3D plot
## Not run:
obj <- list()
obj$mu <- list(c(420, 360, 425), c(160, 570, 200), c(320, 540, 260), c(530, 80, 450))
obj$sigma <-
fmmst.init

**Initialization for Fitting Finite Mixtures of Unrestricted Multivariate Skew t Distributions**

**Description**

Computes different sets of initial values for finite mixtures of unrestricted multivariate skew t (FM-uMST) model based on an initial clustering.

**Usage**

```r
def fmmst.init(g, dat, known=NULL, clust=NULL, nkmeans=20, tmeth=1)

Arguments

- `g`: a scalar specifying the number of components in the mixture model
- `dat`: the data matrix giving the coordinates of the point(s) where the density is evaluated. This is either a vector of length `p` or a matrix with `p` columns.
- `known`: (optional) a list containing parameters of the mixture model that are known and not required to be estimated. See the 'Details' section. The default is NULL.
- `clust`: (optional) a numeric value of length `nrow(dat)` containing the initial labels for each data point in `dat`. The default is NULL, indicating no initial clustering is known.
- `nkmeans`: (optional) a numeric value indicating how many k-means trials to be used when searching for initial values. The default is 20.
- `tmeth`: (optional) an integer indicating which method to use when computing t distribution function values. See `pmt` for details.
```
Details

As the EM algorithm is sensitive to the starting value, it is highly recommended to apply a wide range different initializations. To obtain different sets of starting values using the strategy described in Section 5.1.3 of Lee and McLachlan (2014), `fmmst.init()` can be used, which will return a list of objects with the same structure as `initial`. An example is given in the examples section below.

The argument `known`, if specified, is a list structure containing at least one of `mu`, `sigma`, `delta`, `dof`, `pro` (See `dfmmst` for the structure of each of these elements). Note that although not all parameters need to be provided in `known`, the parameters that are provided must be fully specified. They cannot be partially specified, e.g. only some elements or some components are specified.

Value

a list of initializations for `fmmst`, each containing the following parameters:

- `mu` a list of g numeric matrices containing the location parameter for each component.
- `sigma` a list of g numeric matrices containing the scale parameter for each component.
- `delta` a list of g numeric matrices containing the skewness parameter for each component.
- `dof` a numeric vector of length g representing the degrees of freedom for each component.
- `pro` a vector of length of g specifying the mixing proportions for each component.
- `tau` an g by n matrix of initial probability of component membership.
- `clusters` a vector of length n of initial partition.
- `loglik` the initial log likelihood value.

References


See Also

`rfmmst`, `dfmmst`, `fmmst.contour.2d`

Examples

# a short demo using AIS data
data(ais)
Fit.init <- fmmst.init(2, ais[,c(2,12)])
#the number of available initializations
length(Fit.init)

#getting the first set of available initialization
Fit.init[[1]]
## Not run:
Fit1 <- fmmst(2, ais[,c(2,12)], initial=Fit.init[[1]])
Fit2 <- fmmst(2, ais[,c(2,12)], initial=Fit.init[[2]])

## End(Not run)

---

**fmmstDA**  
*Discriminant analysis using Multivariate Skew t Mixture Models*

**Description**

performs discriminant analysis (DA) for a specified multivariate skew t mixture distribution.

**Usage**

```r
fmmstDA(g, dat, model, tmethod=1)
```

**Arguments**

- `g`  
a scalar specifying the number of components in the mixture model
- `dat`  
the data matrix giving the coordinates of the point(s) to be classified.
- `model`  
a list containing the parameters of the model, including mu, sigma, delta, dof (for fmmstDA only) and pro.
- `tmethod`  
(optional) an integer indicating which method to use when computing t distribution function values. See `pmt` for details.

**Details**

For the structure of the elements of `model`, see `dfmmst`.

**Value**

`fmmstDA` returns a vector of length `nrow(dat)` of the cluster labels

**References**


**See Also**

dfmmst, fmmst
Examples

```r
obj <- list()
obj$mu <- list(c(17,19), c(5,22), c(6,10))
obj$sigma <- list(diag(2), matrix(c(2,0,0,1),2), matrix(c(3,7,7,24),2))
obj$delta <- list(c(3,1.5), c(5,10), c(2,0))
obj$dof <- c(1, 2, 3)
obj$pro <- c(0.25, 0.25, 0.5)
X2 <- rfmmst(3, 50, known=obj)
fmmstDA(3, X2[,1:2], obj)
```

Description

Computes maximum likelihood estimators (MLE) for finite mixtures of multivariate t (FM-MT) model via the EM algorithm.

Usage

```r
fmmt(g = 1, dat, initial = NULL, known = NULL, itmax = 100, eps = 1e-03, nkmeans=20, print = T)
```

Arguments

- `object`, `x`: an object class of class "fmmt", i.e. a fitted model.
- `g`: a scalar specifying the number of components in the mixture model.
- `dat`: the data matrix giving the coordinates of the point(s) where the density is evaluated. This is either a vector of length `p` or a matrix with `p` columns.
- `initial` (optional): a list containing the initial parameters of the mixture model. See the 'Details' section. The default is NULL.
- `known` (optional): a list containing parameters of the mixture model that are known and not required to be estimated. See the 'Details' section. The default is NULL.
- `itmax` (optional): a positive integer specifying the maximum number of EM iterations to perform. The default is 100.
- `eps` (optional): a numeric value used to control the termination criteria for the EM loops. It is the maximum tolerance for the absolute difference between the log-likelihood value and the asymptotic log likelihood value. The default is 1e-6.
- `nkmeans` (optional): a numeric value indicating how many k-means trials to be used when searching for initial values. The default is 20.
- `print` (optional): a logical value. If TRUE, output for each iteration will be printed out. If FALSE, no output is printed. The default is TRUE. See the 'Details' section.
- `...`: not used.
Details

The arguments init and known, if specified, is a list structure containing at least one of mu, sigma, delta, dof, pro (See dfmmt for the structure of each of these elements). If init=FALSE (default), the program uses an automatic approach based on k-means clustering to generate an initial value for the model parameters.

Value

mu         a list of g numeric matrices containing the location parameter for each component.
sigma      a list of g numeric matrices containing the scale parameter for each component.
dof        a numeric vector of length g representing the degrees of freedom for each component.
pro        a vector of length of g specifying the mixing proportions for each component.
tau        an g by n matrix of posterior probability of component membership.
clusters   a vector of length n of final partition.
loglik     the final log likelihood value.
lk         a vector of log likelihood values at each EM iteration.
iter       number of iterations performed.
eps        the final absolute difference between the log likelihood value and the asymptotic log likelihood value.
aic, bic   Akaike Information Criterion (AIC), Bayes Information Criterion (BIC)

References


See Also

rfmmt, dfmmt, fmmst.contour.2d

Examples

# a short demo using AIS data
data(ais)
Fit <- fmmt(2, ais[,c(2,12)], itmax=10)
summary(Fit)
print(Fit)
Description

A subset of the T-cell phosphorylation dataset. The original data contain measurements of blood samples stained with four antibodies, CD4, CD45RA, SLP76 and ZAP70. Measurements from each subject were taken before and after anti-CD3 stimulation. This is a subset of the pre-stimulation data from one subject.

Usage

data(Lympho)

Format

A data frame with 33399 observations (rows) on the following 2 variables (columns).

SLP76 marker 1
ZAP70 marker 2

Source


References


Examples

data(Lympho)
plot(Lympho, main="Lymphoma dataset")
smoothScatter(Lympho, nrpoints=Inf)
Simulation of Mixture Data

Description
Generate random sample from a specified mixture of unrestricted multivariate skew t distribution

Usage
rfmmst(g, n, mu, sigma, delta, dof = rep(10, g),
pro = rep(1/g, g), known = NULL)
rmst(n, mu, sigma, delta, dof = 1, known)

Arguments

- **g**: a scalar specifying the number of components in the mixture model
- **n**: either a positive integer specifying the total number of points to be generated or
  a vector (of length g) of positive integers specifying the number of points to be
  generated in each component.
- **mu**: for `rmst`, this is a numeric vector of length p representing the location parameter; for
  `rfmmst`, this is list of g numeric matrices each having p rows and 1 column
  containing the location parameter for each component.
- **sigma**: for `rmst`, this is a numeric positive definite matrix with dimension (p,p) rep-
  resenting the scale parameter; for `rfmmst`, this is list of g numeric matrices
  containing the scale parameter for each component.
- **delta**: for `rmst`, this is a numeric vector of length p representing the skewness param-
  eter; for `rfmmst`, this is list of g numeric matrices each having p rows and 1
  column containing the skewness parameter for each component.
- **dof**: for `rmst`, this is a positive integer specifying the degrees of freedom; for `rfmmst`,
  this is numeric vector of length g representing the degrees of freedom for each
  component.
- **pro**: the mixing proportions; for `rmst`, this is equal to 1; for `rfmmst`, this is vector of
  length of g specifying the mixing proportions for each component.
- **known**: a list containing the parameters of the model. If specified, it overwrites the
  values of mu, sigma, delta, dof and pro.

Details
`rmst` generates a sample n multivariate skew t (MST) variables. `rfmmst` generates a mixture of
MST samples. Note that model parameters can be passed to `rmst` and `rfmmst` through the argument
known or listed as individual arguments. If both methods of input were used, the parameters
specified in known will be used.
rfmmst

Value

rmst returns an n by p numeric matrix of generated data. rfmmst returns an n by p+1 numeric matrix of generated data. The first p gives the coordinates of the generated data. The last column specifies which component each data point is generated from.

References


See Also
dmst, dfmmst

Examples

```R
# Should be DIRECTLY executable !! ----
#--//-- Define data, use random,
#--or do help(data=index) for the standard data sets.
rfmmst(1,500, c(1,2), diag(2), c(-1,1), 4, 1)
obj <- list()
obj$mu <- list(c(17,19), c(5,22), c(6,10))
obj$sigma <- list(diag(2), matrix(c(2,0,0,1),2), matrix(c(3,7,7,24),2))
obj$delta <- list(c(3,1.5), c(5,10), c(2,0))
obj$dof <- c(1, 2, 3)
obj$pro <- c(0.25, 0.25, 0.5)
rfmmst(3, 250, obj$mu, obj$sigma, obj$delta, obj$dof, obj$pro)
rfmmst(3, 500, known=obj)
```
Index

*Topic 3d
  fmmst.contour.2d, 12
*Topic EM algorithm
  fmmst, 10
  fmmst.init, 14
  fmmt, 17
*Topic contour
  fmmst.contour.2d, 12
*Topic datasets
  ais, 3
  DLBCL, 7
  Lympho, 19
*Topic maximum likelihood estimation
  fmmst, 10
  fmmst.init, 14
*Topic maximum likleihood estimation
  fmmt, 17
*Topic mixture density
  delta.test, 4
  dfmmst, 5
  dmt, 8
*Topic multivariate distribution
  delta.test, 4
  dfmmst, 5
  dmt, 8
  rfmmst, 20
*Topic multivariate skew t
  delta.test, 4
  dfmmst, 5
  dmt, 8
  fmmst, 10
  fmmst.init, 14
  fmmt, 17
  rfmmst, 20
*Topic package
  EMMIXuskew-package, 2
*Topic random number
  rfmmst, 20
  ais, 3
  contour, 13
  delta.test, 4
  dfmmst, 2, 5, 9, 11, 15, 16, 18, 21
  dfmmt (dmt), 8
  DLBCL, 7
  dmst, 9, 21
  dmst (dfmmst), 5
  dmt, 8
  EMMIXuskew (EMMIXuskew-package), 2
  EMMIXuskew-package, 2
  fmmst, 2, 4, 10, 12, 13, 16
  fmmst.contour.2d, 11, 12, 15, 18
  fmmst.contour.3d, 2
  fmmt (dfmmst), 12
  fmmst.init, 11, 14
  fmmstda, 16
  fmmt, 4, 17
  Lympho, 19
  plot.default, 13
  pmst, 5, 10, 13, 14, 16
  pmst (dmt), 8
  print.fmmst (fmmst), 10
  print.fmmt (fmmt), 17
  rfmmst, 2, 6, 11, 15, 18, 20
  rmst, 6
  rmst (rfmmst), 20
  summary.fmmst (fmmst), 10
  summary.fmmt (fmmt), 17
  true.clusters (DLBCL), 7