Package 'EMMIXcskew'

January 30, 2017

Type Package

Title Fitting Mixtures of CFUST Distributions

Version 0.9-3

Date 2016-03-11

Author S.X. Lee and G.J. McLachlan

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

Description Functions to fit finite mixture of multivariate canonical fundamental skew t (FM-CFUST) distributions, random sample generation, 2D and 3D contour plots

License GPL

LazyLoad yes

Suggests rgl, stats, grDevices, KernSmooth

Depends MASS, graphics

Archs i386, x64

R topics documented:

EMMIXcskew-package *Finite mixture of multivariate canonical fundamental skew t-distributions*

Description

Theis package implements an EM algorithm for fitting mixtures of multivariate canonical fundamental skew t (FM-CFUST) distributions. Functions for random sample generation, initial value generation, and visualisation (in 2D and 3D) are also provided.

Details

Package:	EMMIXcskew
Type:	Package
Version:	0.9-1
Date:	2015-09-07
Licence:	GPL
LazyLoad:	yes

Author(s)

S.X. Lee, G.J. McLachlan

References

Lee S.X. and McLachlan, G.J. (2015) Finite mixtures of canonical fundamental skew t-distributions: the unification of the restricted and unrestricted skew t-mixture models. Statistics and Computing. doi:10.1007/s11222-015-9545-x

See Also

`fmcfust`, `dfmcfust`, `rfmcfust`, `fmcfust.contour.3d`

dfmcfust *Multivariate canonical fundamental skew t-distribution*

Description

The probability density function for the multivariate canonical fundamental skew t (CFUST) distribution and finite mixture of CFUST distributions

Usage

```
dfmcfust(dat, mu=NULL, sigma=NULL, delta=NULL, dof=NULL, pro=NULL, known=NULL)
dcfust(dat, mu=NULL, sigma=NULL, delta=NULL, dof=1, known=NULL)
```

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 dcfust, this is a numeric vector of length p representing the location parameter; for dfmcfust, this is list of g numeric matrices each having p rows and 1 column containing the location parameter for each component.
sigma	for dcfust, this is a numeric positive definite matrix with dimension (p,p) representing the scale parameter; for dfmcfust, this is list of g numeric matrices containing the scale parameter for each component.
delta	for dcfust, this is a numeric matrix of size p by q representing the skewness matrix; for rfmcfust, this is list of g numeric matrices each having p rows and q column containing the skewness parameter for each component.
dof	for dcfust, this is a positive integer specifying the degrees of freedom; for dfmcfust, this is numeric vector of length g representing the degrees of freedom for each component.
pro	the mixing proportions; fordcfust, this is equal to 1; for dfmcfust, 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

The function dcfust computes the density value of a specified multivariate canonical fundamental skew t (CFUST) 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 dfmcfust computes the density value for a specified mixture of MST distribution. Note that dfmcfust expects at least dof is specified. Other missing parameters will take the default value described above. When g=1, dfmcfust passes the call to dcfust. Model parameters can be passed to dcfust and dfmcfust 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

dcfust and dfmcfust returns a numeric vector of density values

References

Lee S.X. and McLachlan, G.J. (2015) Finite mixtures of canonical fundamental skew t-distributions: the unification of the restricted and unrestricted skew t-mixture models. Statistics and Computing. doi:10.1007/s11222-015-9545-x

See Also

rcfust, rfmcfust

Examples

```
dcfust(c(1,2), mu=c(1,5), sigma=diag(2), delta=matrix(c(-3,1,1,1),2,2), dof=4)
```

`fmcfust` *Fitting Finite Mixtures of Multivariate Canonical Fundamental Skew t-Distributions*

Description

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

Usage

```
fmcfust(g=1, dat, q, initial=NULL, known=NULL, clust=NULL, itmax=100, eps=1e-6,
    nkmeans=20, verbose=T, method=c("moments","transformation","EMMIXskew","EMMIXuskew"),
    convergence=c("Aitken","likelihood","parameters"))
## S3 method for class fmcfust
summary(object, ...)
## S3 method for class fmcfust
print(x, ...)
```

Arguments

`object, x`	an object class of class `"fmcfust"`, 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.
`q`	a scalar specifying how many number of columns the skewness matrix `delta` has.
`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`.
`verbose`	(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.
`method`	(optional) a string indicating which method to use to generate initial values. See `init.cfust`.
`convergence`	(optional) a string indicating which convergence criterion to use to terminate the iterations. The default `"Aitken"` uses Aitken accelation, whereas `"likelihood"` uses the relative difference in log likleihood value, and `"parameters"` checks the changes in parameter estimates.

...	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 `dfmcfust` for the structure of each of these elements). If `init=FALSE` (default), the program uses an automatic approach based on moments estimate and 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. Some strategies are implemented in `init.cfust`.

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

Lee S.X. and McLachlan, G.J. (2015) Finite mixtures of canonical fundamental skew t-distributions: the unification of the restricted and unrestricted skew t-mixture models. Statistics and Computing. doi:10.1007/s11222-015-9545-x

See Also

`init.cfust`, `rfmcfust`, `dfmcfust`, `fmcfust.contour.2d`

Examples

```
#a short demo using geyser data
library(MASS)
Fit <- fmcfust(3, geyser)
summary(Fit)
print(Fit)
```

`fmcfust.contour.2d` *2D and 3D Visualisation of Fitted Contours*

Description

Create 2D or 3D contour plot.

Usage

```
fmcfust.contour.2d(dat, model, grid=50, drawpoints=TRUE, clusters=NULL,
    nlevels=10, map=c("scatter", "heat", "cluster"),
    component=NULL, xlim, ylim, xlab, ylab, main, pcol=NULL, ccol=NULL, ...)
fmcfust.contour.3d(dat, model, grid=20, drawpoints=TRUE, levels=0.9,
    clusters=NULL, xlim, ylim, zlim, xlab, ylab, zlab, main, component=NULL,
    pcol=NULL, ccol=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 `fmcfust.contour.2d` or 3 columns for `fmcfust.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 `fmcfust`, containing `mu`, `sigma`, `delta`, `dof`, `pro` and `clusters`; see `fmcfust` 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`.
`nlevels`	a positive integer specifying the number of contour levels
`levels`	either a positive integer or a numeric vector specifying the contour levels to be drawn for each component
`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
`xlab, ylab, zlab`	labels for x-, y- and z- axis
`main`	title of the plot
`pcol`	the color(s) to be used for plotted points
`ccol`	the color(s) to be used for plotted contours
`...`	additional arguments to `plot.default`

Details

fmcfust.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 datset, 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

fmcfust, contour

Examples

```
#2D plots
data(iris)
iris.versicolor <- iris[iris$Species=="versicolor",2:3]
Fit.versicolor <- fmcfust(1, iris.versicolor)
fmcfust.contour.2d(iris.versicolor, Fit.versicolor, drawpoints=FALSE, main="versicolor")


#3D plot
## Not run:
obj <- list()
obj$mu <- list(matrix(c(0,0,0),3), matrix(c(5,5,5),3))
obj$sigma <- list(matrix(c(5,2,1,2,5,1,1,1,1),3,3), 2*diag(3))
obj$delta <- list(matrix(c(1,0,0,1,0,0,1,0,0),3,3), matrix(c(5,0,0,0,10,0,0,0,15),3,3))
obj$dof <- c(3,3)
obj$pro <- c(0.2, 0.8)
fmcfust.contour.3d(model=obj, level=0.98, drawpoints=TRUE, xlab="X", ylab="Y", zlab="Z")

## End(Not run)
```

---

fmmst *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

```
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.
...	not used.

Details

The arguments init and known, if specified, is a list structure containing at least one of mu, sigma, delta, dof, pro. 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.

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

Lee S, McLachlan G (2011). On the fitting of mixtures of multivariate skew t-distributions via the EM algorithm. arXiv:1109.4706 [stat.ME]

Lee, S. and McLachlan, G.J. (2014) Finite mixtures of multivariate skew t-distributions: some recent and new results. Statistics and Computing, 24, 181-202.

Lee, S. and McLachlan, G.J. (2013) EMMIXuskew: An R package for fitting mixtures of multivariate skew t-distributions via the EM algorithm. Journal of Statistical Software, 55(12), 1-22. URL http://www.jstatsoft.org/v55/i12/.

See Also

fmcfust

Examples

```
#a short demo using geyser data
library(MASS)
Fit <- fmmst(3, geyser)
summary(Fit)
print(Fit)
```

---

init.cfust *Initialization for Fitting Finite Mixtures of Canonical Fundamental Skew t-Distributions*

Description

Computes different sets of initial values for finite mixtures of canonical fundamental skew t (FM-CFUST) model based on an initial clustering, transformation approiach, moment-based approach, or nested-model appraoch.

Usage

```
init.cfust(g, dat, q=p, initial=NULL, known=NULL, clust=NULL, nkmeans=20,
    method=c("moments","transformation","EMMIXskew","EMMIXuskew"))
init.fmcfust(g, dat, q=p, initial=NULL, known=NULL, clust=NULL, nkmeans=20,
    method=c("moments","transformation","EMMIXskew","EMMIXuskew"))
```

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.
q	a scalar specifying how many number of columns the skewness matrix delta has.
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.
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.
method	(optional) a string indicating which method to use to generate initial values. See 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), `init.cfust()` 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 `dfmcfust` 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 object 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 matrix 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

Lee S.X. and McLachlan, G.J. (2015) Finite mixtures of canonical fundamental skew t-distributions: the unification of the restricted and unrestricted skew t-mixture models. Statistics and Computing. doi:10.1007/s11222-015-9545-x

See Also

rfmcfust, dfmcfust, fmcfust.contour.2d

Examples

```
#a short demo using geyser data
library(MASS)
data(geyser)
initial.transformation <- init.cfust(3, geyser, method="transformation")
initial.transformation$loglik
```

---

rfmcfust *Simulation of Mixture Data*

Description

Generate random sample from a specified mixture of multivariate canonical fundamental skew t distribution

Usage

```
rfmcfust(g, n, mu, sigma, delta, dof=rep(10,g), pro=rep(1/g,g), known=NULL)
rcfust(n=1, mu = NULL, sigma=NULL, delta=NULL, dof=1, known=NULL)
```

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 rcfust, this is a numeric vector of length p representing the location parameter; for rfmcfust, this is list of g numeric matrices each having p rows and 1 column containing the location parameter for each component.
sigma	for rcfust, this is a numeric positive definite matrix with dimension (p,p) representing the scale parameter; for rfmcfust, this is list of g numeric matrices containing the scale parameter for each component.
delta	for rcfust, this is a numeric matrix of size p by q representing the skewness matrix; for rfmcfust, this is list of g numeric matrices each having p rows and q column containing the skewness parameter for each component.
dof	for rcfust, this is a positive integer specifying the degrees of freedom; for rfmcfust, this is numeric vector of length g representing the degrees of freedom for each component.
pro	the mixing proportions; for rcfust, this is equal to 1; for rfmcfust, 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

`rcfust` generates a sample `n` multivariate CFUST observations. `rfmcfust` generates a mixture of CFUST observation. Note that model parameters can be passed to `rcfust` and `rfmcfust` 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

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

References

Lee S.X. and McLachlan, G.J. (2015) Finite mixtures of canonical fundamental skew t-distributions: the unification of the restricted and unrestricted skew t-mixture models. Statistics and Computing. doi:10.1007/s11222-015-9545-x

See Also

`dcfust`, `dfmcfust`

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.
rcfust(10,c(1,2),diag(2),matrix(c(2,1,1,2),2,2),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$delta <- list(matrix(c(3,0,2,1.5),2,2), matrix(c(5,0,0,10),2,2), matrix(c(2,0,5,0),2,2))
obj$dof <- c(1, 2, 3)
obj$pro <- c(0.25, 0.25, 0.5)
rfmcfust(3, 100, known=obj)
```

Index