Package ‘EMMIXmfa’

Type Package
Title Mixture models with component-wise factor analyzers
Version 1.3.3
Date 2017-09-17
Author Suren Rathnayake, Geoff McLachlan, Jungsun Baek
Maintainer Geoff McLachlan <g.mclachlan@uq.edu.au>
Description We provide functions to fit finite mixture of multivariate normal or t-distributions to data with various factor analytic structures adopted for the covariance / scale matrices. Maximum likelihood estimators of model parameters are obtained via the Expectation-Maximization algorithm.
License GPL (>= 2)
NeedsCompilation no

R topics documented:

EMMIXmfa-package .................................................. 1
ari ................................................................. 3
err ................................................................. 3
getscores ......................................................... 4
mcfa ............................................................... 6
mf ................................................................. 8
plotscores ......................................................... 10
rmix ............................................................... 12

Index

EMMIXmfa-package Mixture models with component-wise factor analyzers.

Description
Fits finite mixture models that adopt component-wise factor analyzers to multivariate data. Component distributions can either be from the family multivariate normal or multivariate t-distributions. Maximum likelihood estimators of model parameters are obtained using the Expectation-Maximization algorithm.

Details
EMMIXmfa-package

Package: EMMIXmfa
Type: Package
Version: 1.3.3
Date: 2017-09-19
License: GPL

Author(s)
Suren Rathnayake, Jangsun Baek, Geoffrey McLachlan

References


See Also
mcfa,mfa

Examples

```r
## Not run:
set.seed(1)
Y <- scale(iris[, -5])
mcfa_model <- mcfa(Y, g = 3, q = 3)
mctfa_model <- mctfa(Y, g = 3, q = 3)
mfa_model <- mfa(Y, g = 3, q = 3)
mtfa_model <- mtfa(Y, g = 3, q = 3)

## End(Not run)
```
ari

Computes adjusted Rand Index

Description

Computes adjusted Rand Index.

Usage

ari(cls, hat_cls)

Arguments

cls vector containing labels or classes.
hat_cls vector of labels same length as in cls.

Details

Measures the agreement between two set of partitions.

Value

ARI Scaler specifying how closely two set of partitions agree. The upper bound 1 implies perfect agreement.

References


See Also

er

Examples

c1 <- floor(runif(100, 1, 3))
c2 <- floor(runif(100, 1, 3))
ari(c1, c2)

err

minimum number of mis-allocations.

Description

Given two vectors each corresponding to a set of categories, this function provides the minimum number of mis-allocations by rotating categories.

Usage

er(c1, c2)
Arguments

cls         vector of labels.
hat_cls     vector of labels same length as in cls.

Details

Rotates the categories for all possible permutations, and returns the minimum number of mis-
allocations. The number of categories in each set of labels does not need to be the same. It make
take several minutes to compute for large number of categories.

Value

min_err     Integer specifying the minimum number of mis-allocations.

Author(s)

Suren Rathnayake

See Also

ari

Examples

c1 <- floor(runif(100, 1, 3))
c2 <- floor(runif(100, 1, 3))
er(c1, c2)

gscores     Computes factor scores.

Description

This function computes factor scores given a data set and a EMMIXmfa model.

Usage

gscores(Y, model, tau = NULL, clust= NULL, ...)

Arguments

model       Model of class \texttt{"mcfa"}, \texttt{"mctfa"}, \texttt{"mfa"}, or \texttt{"mtfa"}.
Y           Data matrix with variables in columns in the same order as used in model esti-
mation.
tau         Optional. Posterior probabilities of belonging to the components in the mixture
model. If not provided, they will be estimated.
clust       Optional. Indicators of belonging to the components. If not provided, will be
estimated using \texttt{tau}.
...         Not used.
**Details**

Factor scores can be used in visualization of the data in the factor space.

**Value**

- **U**: Estimated conditional expected component scores of the unobservable factors given the data and the component membership. Size is \(n\) by \(q\) by \(g\), where \(n\) is the number of sample, \(q\) is the number of factors and \(g\) is the number of components.

- **Fmat**: Means of the estimated conditional expected factors scores over estimated posterior distributions. Size \(n\) by \(q\).

- **UC**: Alternative estimate of Fmat where the posterior probabilities for each sample are replaced by component indicator vectors which contain one in the element corresponding to the highest posterior probability while others zero. Size \(n\) by \(q\).

**Author(s)**

Geoffrey McLachlan, Suren Rathnayake, Jungsun Baek

**References**


**Examples**

```r
# Fit a MCFA model to a subset
set.seed(1)
samp_size <- dim(iris)[1]
sel_subset <- sample(1 : samp_size, 75)
model <- mcfa(iris[sel_subset, -5], g=3, q=2, nkmeans=1, nrandom=0)

# plot the data points in the factor space
plot(model)

# Allocating new samples to the clusters
Y <- iris[-c(sel_subset), -5]
Y <- as.matrix(Y)
clust <- predict(model, Y)

factor_scores <- getscores(Y, model)
# Visualizing new data in factor space
plotscores(factor_scores, type="Fmat", clust=clust)
```
**mcfa**  
*Mixture of Common Factor Analyzers*

**Description**

Functions for fitting of Mixtures Common Factor Analyzers (MCFA) and Mixtures of Common t-Factor Analyzers (MCtFA) data. Maximum Likelihood estimates of the model parameters are obtained using the Expectation–Maximization algorithm.

**Usage**

```r
mcfa(Y, g, q, ...)  
mctfa(Y, g, q, ...)  
```

## Default S3 method:

```r
mcfa(Y, g, q, itmax = 500,  
nkmeans = 20, nrandom = 20, tol = 1.e-5,  
itinit_clust = NULL, init_para = NULL,  
itinit_method = 'eigenA', conv_measure = 'diff',  
warn_messages = FALSE, ...)  
```

## Default S3 method:

```r
mctfa(Y, g, q, itmax = 500,  
nkmeans = 20, nrandom = 20, tol = 1.e-5, df.init = rep(30, g),  
df.update = TRUE, init_clust = NULL,  
init_para = NULL, init_method = 'eigenA',  
conv_measure = 'diff', warn_messages = TRUE, ...)  
```

## S3 method for class 'emmixmfa'

```r
print(x, ...)  
```

## S3 method for class 'emmixmfa'

```r
summary(object, ...)  
```

## S3 method for class 'emmixmfa'

```r
plot(x, ...)  
```

## S3 method for class 'emmixmfa'

```r
predict(object, Y, ...)  
```

**Arguments**

- **Y**
  - data.frame or a matrix of data matrix (samples in rows)
- **x, object**
  - An object of class "mcfa", an object returned from mcfa function.
- **g**
  - number of components
- **q**
  - number of factors
- **itmax**
  - maximum number of EM iterations
- **nkmeans**
  - The k-means algorithm is used nkmeans-times to partition the data into g groups. The EM algorithm is run on initial parameter values obtained from each partition.
- **nrandom**
  - The number of random g-group partitions for the data to be used initializing the EM algorithm.
- **tol**
  - EM steps terminates if the measure of convergence falls below this value.
- **init_clust**
  - A vector or matrix consisting of partition of samples to be used in the EM algorithm. For a matrix of partitions, rows corresponds individual partitions.
init_para A list containing model parameters to be used as initial parameter estimates for the EM algorithm. Optional.

init_method determine how the initial parameter values are computed. See Details.

cov_measure If 'ratio', then the convergence of the EM steps are measured using the \(|L^{(k+1)} - L^{(k)}/L^{(k+1)}|\). The default 'diff' stops the EM iterations if \(|L^{(k+1)} - L^{(k)}| < tol\)

df.init Initial values of the degree of freedom parameters for mctfa.

df.update If df.update = TRUE (default), then the DOF values will be updated during the EM iterations. Otherwise, if df.update = FALSE, they will be fixed at the initial values specified in df.init.

warn_messages This aims to provide a brief explanation for the cases where the function failed to provide estimates of parameters. Default is warn_messages = TRUE, otherwise the messages will not be stored.

Details

The default is 'eigenA' which initialize the columns of the loading vector A as the first q eigen vectors of the Y^T Y. Use init_method = "randA" for the procedure used by Baek et al. (2010).

Value

- pivec component probabilities
- A factor loading matrix
- xi matrix of factor mean vectors
- omega array of factor covariance matrices
- D matrix of error covariance matrix
- U array of estimated u_ij vectors for all components
- UC matrix of estimated u_ij vectors according to cluster labels
- Fmat matrix of estimated factor score vectors
- clust cluster label determined by MCFA
- tau posterior probabilities
- logL log-likelihood
- BIC Bayesian Information Criteria
- ERRMSG Error Message

References


Examples

```r
mcfa_fit <- mcfa(iris[, -5], g = 3, q = 3, 
                  itmax = 250, nkmeans = 5, nrandom = 5, tol = 1.e-5)

plot(mcfa_fit)
```

Description

Functions for fitting Mixtures of Factor Analyzers (MFA) and Mixtures of t-Factor Analyzers (MtFA) to data. Maximum Likelihood estimates of the model parameters are obtained using the Alternating Expectation Conditional Maximization (AECM) algorithm.

Usage

```r
mfa(Y, g, q, ...) 
mtfa(Y, g, q, ...) 
## Default S3 method: 
mfa(Y, g, q, itmax = 500, 
    nkmeans = 20, nrandom = 20, 
    tol = 1.e-5, sigma_type = 'common', 
    D_type = 'common', init_clust = NULL, 
    init_para = NULL, conv_measure = 'diff', 
    warn_messages = TRUE, ...) 
## Default S3 method: 
mtfa(Y, g, q, itmax = 500, 
    nkmeans = 20, nrandom = 20, 
    tol = 1.e-5, df.init = rep(30, g), 
    df.update = TRUE, sigma_type = 'common', 
    D_type = 'common', init_clust = NULL, 
    init_para = NULL, conv_measure = 'diff', 
    warn_messages = TRUE, ...)
```

Arguments

- **Y**: A matrix or a data frame of which rows correspond to observations and columns to variables.
- **g**: Number of components.
- **q**: Number of factors.
- **itmax**: Maximum number of EM iterations.
- **nkmeans**: The k-means algorithm is used nkmeans-times to partition the data into g groups. The EM algorithm is run on initial parameter values obtained from each partition.
- **nrandom**: The number of random g-group partitions for the data to be used initializing the EM algorithm.
- **tol**: EM steps terminates if the measure of convergence falls below this value.
sigma_type  This allows to specify whether the covariance matrices (for mfa) or the scale matrices (for mtfa) are constraint to be the same (default) for each component or not. The default is sigma_type = "common", otherwise use sigma_type = "unique".

D_type  To specify whether the diagonal error covariance matrix is common to all the components or not. If sigma_type = "unique", then D_type could either be "common" (the default) to each component, or "unique". If the sigma_type = "common", then D_type must also be "common".

init_clust  A vector or matrix consisting of partition of samples to be used in the EM algorithm. For a matrix of partitions, rows corresponds individual partitions.

init_para  A list containing model parameters to be used as initial parameter estimates for the EM algorithm. Optional.

conv_measure  Method to identify convergence. The default "diff" stops the EM iterations if the difference between the current and previous log-likelihood values falls below tol value. If conv_measure = "ratio" is used, then the relative difference in log-likelihood values are considered.

df.init  Initial values of the degree of freedom parameters for mtfa.

df.update  If df.update = TRUE (default), then the DOF values will be updated during the EM iterations. Otherwise, if df.update = FALSE, they will be fixed at the initial values specified in df.init.

warn_messages  This aims to provide a brief explanation for the cases where the function failed to provide estimates of parameters. Default is warn_messages=TRUE, otherwise the messages will not be stored.

...  Not used.

Details

Cluster a given data set using Mixtures of Factor Analyzers or approach or using Mixtures of StS-Factor Analyzers.

Value

Object of class c("emmixmfa", "mfa") or c("emmixmfa", "mtfa") containing the fitted model parameters is returned. Details of the components are as fellows:

g  Number of mixture components.
q  Number of factors.
pivec  Vector containing mixing proportions.
mu  Matrix containing estimates of component means for each mixture components. Size p by g.
B  Array containing component dependent loading matrices. Size p by q by g.
D  Estimates of error covariance matrices. If D_type = "common" was used then D is p by p matrix common to all components, if D_type = "unique", then D is a p by p by g array.
v  degrees of freedom for each component.
logL  Log-likelihood.
BIC  Bayesian Information Criterion.
tau  Matrix of posterior probabilities for the data used based on the fitted values. Matrix of size n by g.
Vector of integers 1 to g indicating cluster allocations of the observations.

Estimated factor scores. Matrix of size \( n \times q \times g \).

Estimated factor scores of the observations corresponding to clusters they were assigned. Matrix of size \( n \times q \).

Estimated conditional means of the factors over the estimated posterior distribution. Matrix of size \( n \times q \).

Error messages.

Whether common or unique error covariance is used, as specified in model fitting.

Whether DOF (\( v \)) was fixed or estimated, as specified in model fitting.

Suren Rathnayake, Geoffrey McLachlan


See Also

cmfa

Examples

```r
model <- mfa(iris[, -5], g=3, q=2, itmax=200, nkmeans=1, nrandom=5)
plot(model)
summary(model)
```

Description

Plots factors scores.

Usage

```r
plotscores(scores, type="Fmat",
           clust=if (exists('clust', where=scores)) scores$clust else NULL,
           limx=NULL, limy=NULL)
```
Arguments

scores  A list containing factor scores specified by Fmat, UC or U or a model of class "mcfa", "mctfa", "mfa", or "mtfa".
type    What type of factor scores are to be plotted. See Details.
clust   Indicators of belonging to components. If available, they will be portrayed in plots. If not provided, looks for clust in scores, and sets to NULL if still not available.
limx    Numeric vector. Values in limx will only be used in setting the x-axis range for 1-D and 2-D plots.
limy    Numeric vector. Values in limy will only be used in setting the y-axis range for 1-D and 2-D plots.

Details

The type should either be "U", "UC" or the default "Fmat". These stands for, "U" the estimated conditional expected component scores of the unobservable factors given the data and the component membership (size is n by q by g, where n is the number of sample, q number of factors and g number of components), "Fmat" is the means of the estimated conditional expected factors scores over estimated posterior distributions (size n by q), and "UC" is an alternative estimate of "Fmat" where the posterior probabilities are replaced by component indicator vector which contain one in the element corresponding to the highest posterior probability, while others zero.

Author(s)

Geoffrey McLachlan, Suren Rathnayake, Jungsun Baek

References


Examples

# Visualizing data used in model estimation
set.seed(1)
inds <- dim(iris)[1]
indSample <- sample(1 : inds, 75)
model <- mcfa(iris[indSample, -5], g = 3, q = 2, nkmeans = 1, nrandom = 0)
err(model$clust, iris[indSample, 5])
#same as plotscores(model, tyep = "Fmat", clust = model$clust)
plot(model)
#can provide alternative groupings of samples via plotscores
plotscores(model, clust = iris[indSample, 5])
#same as plotscores(model, tyep = "UC")
plot(model, type = "UC")

Y <- iris[-c(indSample), -5]
Y <- as.matrix(Y)
clust <- predict(model, Y)
err(clust, iris[-c(indSample), 5])
fac_scores <- getscores(Y, model)
plotscores(fac_scores, type="Fmat", clust = clust)
plotscores(fac_scores, type="Fmat", clust = iris[-c(indSample), 5])

rmix

Random Deviates from EMMIXmfa Models

Description
Random number generator for EMMIXmfa models.

Usage
rmix(n, model, ...)

Arguments
model Model of class "mcfa", "mctfa", "mfa", or "mtfa".
n Number of sample to generate.
... Not used.

Details
This function uses the rdemmix2 function in the EMMIX package to generate samples from the mixture components.
Algorithm works by first drawing a component based on the mixture proportion in the model, and then drawing a sample from the component distribution.

Value
dat Matrix with samples drawn in rows.

Author(s)
Geoffrey McLachlan, Suren Rathnayake

References

Examples
set.seed(1)
model <- mcfa(iris[, -5], g=3, q=2, nkmeans=1, nrandom=1)
dat <- rmix(n = 10, model = model)
Index

*Topic clustering
  EMMIXmfa-package, 1
*Topic cluster
  ari, 3
  err, 3
  getscores, 4
  mfa, 8
  plotscores, 10
  rmix, 12
*Topic models
  getscores, 4
  mfa, 8
  plotscores, 10
  rmix, 12
*Topic model
  EMMIXmfa-package, 1
*Topic multivariate
  EMMIXmfa-package, 1
  getscores, 4
  mfa, 8
  plotscores, 10
  rmix, 12
*Topic package
  EMMIXmfa-package, 1
  ari, 3, 4
  EMMIXmfa (EMMIXmfa-package), 1
  emmixmfa (EMMIXmfa-package), 1
  EMMIXmfa-package, 1
  emmixmfa-package (EMMIXmfa-package), 1
  err, 3, 3
  getscores, 4
  mfa, 6, 10
  mctfa (mfa), 6
  mfa, 8
  mtfa (mfa), 8
  plot.emmixmfa (mfa), 6
  plot.mfa (mfa), 8
  plot.mtfa (mfa), 8
  plotscores, 10
  predict.emmixmfa (mcfa), 6
  print.emmixmfa (mcfa), 6
  print.mfa (mfa), 8
  print.mtfa (mfa), 8
  rmix, 12
  summary.emmixmfa (mcfa), 6
  summary.mfa (mfa), 8
  summary.mtfa (mfa), 8