Package 'EMMIXmfa'

October 18, 2017

Type Package

Title Mixture models with component-wise factor analyzers

Version 1.3.9

Date 2017-10-18

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 covariancence / scale matrices. Maximum likelihood estimators of model parameters are obtained via the Expectation-Maximization algorithm.

Suggests EMMIX, mvtnorm, GGally, ggplot2

License GPL (>= 2)

NeedsCompilation no

R topics documented:

`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 of multivariate normals or from the family of multivariate *t*-distributions. Maximum likelihood estimators of model parameters are obtained using the Expectation-Maximization algorithm.

Details

Package:	EMMIXmfa
Type:	Package
Version:	1.3.9
Date:	2017-10-18
License:	GPL

Author(s)

Suren Rathnayake, Jangsun Baek, Geoffrey McLachlan

References

McLachlan GJ, Peel D, and Bean RW (2003). Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics & Data Analysis* **41**, 379–388.

McLachlan GJ, Bean RW, Ben-Tovim Jones L (2007). Extension of the mixture of factor analyzers model to incorporate the multivariate *t* distribution. *Computational Statistics & Data Analysis*, **51**, 5327–5338.

McLachlan GJ, Baek J, and Rathnayake SI (2011). Mixtures of factor analyzers for the analysis of high-dimensional data. In *Mixture Estimation and Applications*, KL Mengersen, CP Robert, and DM Titterington (Eds). Hoboken, New Jersey: Wiley, pp. 171–191.

See Also

`mfa`

Examples

```
## Not run:
set.seed(1)
Y <- scale(iris[, -5])
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 cls.

Details

Measures the agreement between two set of partitions. The upper bound 1 implies perfect agreement. Expected value is zero if the partitions are random.

Value

Scaler specifying how closely two partitions agree.

References

Hubert L, and Arabie P (1985). Comparing Partitions. *Journal of the Classification* **2**, 193–218.

See Also

err

Examples

```
## Not run:
set.seed(1984)
Y <- scale(iris[, -5])
model <- mcfa(Y, g = 3, q = 3, nkmeans = 1, nrandom = 0)
#
ari(model$clust, iris[, 5])
#
err(model$clust, iris[, 5])

## End(Not run)
```

---

err *Minimum Number of Mis-Allocations.*

Description

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

Usage

```
err(cls, hat_cls)
```

Arguments

cls	Vector of labels.
hat_cls	Vector of labels same length as 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 when the number of categories is large.

Value

Integer specifying the minimum number of mis-allocations.

Author(s)

Suren Rathnayake

See Also

ari

Examples

```
## Not run:
set.seed(1984)
Y <- scale(iris[, -5])
model <- mcfa(Y, g = 3, q = 3, nkmeans = 1, nrandom = 0)
#
ari(model$clust, iris[, 5])
#
err(model$clust, iris[, 5])

## End(Not run)
```

---

factor_scores *Computes Factor Scores.*

Description

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

Usage

```
factor_scores(Y, model, tau = NULL, clust= NULL, ...)
```

Arguments

model	Model of class "mcfa", "mctfa", "mfa", or "mtfa".
Y	Data matrix with variables in columns in the same order as used in model estimation.
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 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 $n \times q \times g$, where `n` is the number of sample, `q` is the number of factors and `g` is the number components.
`Fmat`	Means of the estimated conditional expected factors scores over estimated posterior distributions. Size $n \times q$.
`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 $n \times q$.

Author(s)

Geoffrey McLachlan, Suren Rathnayake, Jungsun Baek

References

McLachlan GJ, Baek J, and Rathnayake SI (2011). Mixtures of factor analyzers for the analysis of high-dimensional data. In *Mixture Estimation and Applications*, KL Mengersen, CP Robert, and DM Titterington (Eds). Hoboken, New Jersey: Wiley, pp. 171–191.

McLachlan GJ, and Peel D (2000). *Finite Mixture Models*. New York: Wiley.

Examples

```
# 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 <- factor_scores(Y, model)
# Visualizing new data in factor space
plot_factors(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 (MC*t*FA). Maximum Likelihood estimates of the model parameters are obtained using the Expectation–Maximization algorithm.

MCFA adds the following restrictions to,

$$\Sigma_i = A\,\Omega_i\,A^T + D \quad (i = 1, \ldots, g),$$

and

$$\mu_i = A\xi_i \quad (i = 1, \ldots, g)$$

where A is a $p \times q$ matrix, ξ_i is a q-dimensional vector, Ω_i is a $q \times q$ positive definite symmetric matrix, and D is a diagonal $p \times p$ matrix.

With this representation, the component distribution of Y_j is modeled as

$$Y_j = A\,U_{ij} + e_{ij}$$

with prob. $\pi_i (i = 1, \ldots, g)$ for $j = 1, \ldots, n$, where the (unobservable) factors $U_{i1}, \ldots, U_{in}$ are distributed independently $N(\xi_i, \Omega_i)$, independently of the e_{ij}, which are distributed independently $N(0, D)$, where D is a diagonal matrix, $(i = 1, \ldots, g)$.

Usage

```
mcfa(Y, g, q, ...)
mctfa(Y, g, q, ...)
## Default S3 method:
mcfa(Y, g, q, itmax = 500, nkmeans = 20, nrandom = 20,
  tol = 1.e-5, init_clust = NULL, init_para = NULL,
  init_method = 'rand-A', conv_measure = 'diff',
  warn_messages = TRUE, ...)
## Default S3 method:
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 = 'rand-A',
  conv_measure = 'diff', warn_messages = TRUE, ...)
## S3 method for class 'emmixmfa'
print(x, ...)
## S3 method for class 'emmixmfa'
summary(object, ...)
## S3 method for class 'emmixmfa'
plot(x, ...)
## S3 method for class 'emmixmfa'
predict(object, Y, ...)
```

Arguments

`Y`	A matrix or a data frame of which rows correspond to observations and columns to variables.
`x, object`	An object of class `mcfa` or `mctfa`.
`g`	Number of components.
`q`	Number of factors.
`itmax`	Maximum number of EM iterations.
`nkmeans`	The number of times the k-means algorithm to be used in partition the data into `g` groups. These groupings are then used in initializing the parameters for the EM algorithm.
`nrandom`	The number of random `g`-group partitions for the data to be used initializing the EM algorithm.
`tol`	The EM algorithm 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 matrix of partitions, columns must corresponds individual partitions of the data. Optional.
`init_para`	A list containing model parameters to be used as initial parameter estimates for the EM algorithm. Optional.
`init_method`	To determine how the initial parameter values are computed. See Details.
`conv_measure`	The default `'diff'` stops the EM iterations if $\|l^{(k+1)} - l^{(k)}\| <$ `tol` where $l^{(k)}$ is the log-likelihood at the kth EM iteration. If `'ratio'`, then the convergence of the EM steps is measured using the $\|(l^{(k+1)} - l^{(k)})/l^{(k+1)}\|$.
`df_init`	Initial values of the degree of freedom parameters for `mctfa`.
`df_update`	If `df_update = TRUE` (default), then the degree of freedom parameters 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`	If `warn_messages = TRUE` (default), the output would include error messages for instances, if any, where the model fitting function failed to provide estimates of parameters. Otherwise the messages will not be stored.
`...`	Not used.

Details

With the default `init_method = "rand-A"`, initialization of the parameters is done by using the procedure in Baek et al. (2010) where initial values for elements of A are drawn from the $N(0, 1)$ distribution. This method is appropriate when the columns of the data are on the same scale. The `init_method = "eigen-A"` takes the first q eigen vectors of Y as the loading matrix A.

Value

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

`g`	Number of mixture components.
`q`	Number of factors.

`pivec`	Mixing proportions of the components.
`A`	Loading matrix. Size $p \times q$.
`xi`	Matrix containing factor means for components in columns. Size $q \times g$.
`omega`	Array containing factor covariance matrices for components. Size $q \times q \times g$.
`D`	Error covariance matrix. Size $p \times p$.
`U`	Estimated conditional expected component scores of the unobservable factors given the data and the component membership. Size is Size $n \times q \times g$.
`Fmat`	Means of the estimated conditional expected factors scores over estimated posterior distributions. Size $n \times q$.
`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 $n \times q$.
`clust`	Cluster labels.
`tau`	Posterior probabilities.
`logL`	Log-likelihood of the model.
`BIC`	Bayesian Information Criteria.
`warn_msg`	Description of error messages, if any.

Author(s)

Suren Rathnayake, Jangsun Baek, Geoffrey McLachlan

References

Baek J, McLachlan GJ, and Flack LK (2010). Mixtures of factor analyzers with common factor loadings: applications to the clustering and visualisation of high-dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 2089–2097.

Baek J, and McLachlan GJ (2011). Mixtures of common *t*-factor analyzers for clustering highdimensional microarray data. *Bioinformatics* **27**, 1269–1276.

McLachlan GJ, Baek J, and Rathnayake SI (2011). Mixtures of factor analyzers for the analysis of high-dimensional data. In *Mixture Estimation and Applications*, KL Mengersen, CP Robert, and DM Titterington (Eds). Hoboken, New Jersey: Wiley, pp. 171–191.

Examples

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

plot(mcfa_fit)
```

mfa *Mixtures of Factor Analyzers.*

Description

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

Usage

```
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 number of times the k-means algorithm to be used in partition the data into g groups. These groupings are then used in initializing the parameters for the EM algorithm.
nrandom	The number of random g-group partitions for the data to be used initializing the EM algorithm.
tol	The EM algorithm 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 matrix of partitions, columns must corresponds individual partitions of the data. Optional.

init_para	A list containing model parameters to be used as initial parameter estimates for the EM algorithm. Optional.
conv_measure	The default 'diff' stops the EM iterations if $\|l^{(k+1)} - l^{(k)}\| <$ tol where $l^{(k)}$ is the log-likelihood at the kth EM iteration. If 'ratio', then the convergence of the EM steps is measured using the $\|(l^{(k+1)} - l^{(k)})/l^{(k+1)}\|$.
df_init	Initial values of the degree of freedom parameters for mctfa.
df_update	If df_update = TRUE (default), then the degree of freedom parameters 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	If warn_messages = TRUE (default), the output would include error messages for instances, if any, where the model fitting function failed to provide estimates of parameters. 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 *t*-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	Mixing proportions of the components.
mu	Matrix containing estimates of component means for each mixture component. Size $p \times g$.
B	Array containing component dependent loading matrices. Size $p \times q \times g$.
D	Estimates of error covariance matrices. If D_type = "common" was used then D is $p \times p$ matrix common to all components, if D_type = "unique", then D is a size $p \times p \times g$ 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.
clust	Vector of integers 1 to g indicating cluster allocations of the observations.
U	Estimated conditional expected component scores of the unobservable factors given the data and the component membership. Size is Size $n \times q \times g$.
Fmat	Means of the estimated conditional expected factors scores over estimated posterior distributions. Size $n \times q$.
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 $n \times q$.
ERRMSG	Error messages.

D_type	Whether common or unique error covariance is used, as specified in model fitting.
df_update	Whether DOF (v) was fixed or estimated, as specified in model fitting.

Author(s)

Suren Rathnayake, Geoffrey McLachlan

References

Ghahramani Z, and Hinton GE (1997). *The EM algorithm for mixture of factor analyzers*. Technical Report, CRG-TR-96-1, University of Toronto, Toronto.

McLachlan GJ, Peel D, and Bean RW (2003). Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics & Data Analysis* **41**, 379–388.

McLachlan GJ, Bean RW, Ben-Tovim Jones L (2007). Extension of the mixture of factor analyzers model to incorporate the multivariate *t* distribution. *Computational Statistics & Data Analysis*, **51**, 5327–5338.

McLachlan GJ, Baek J, and Rathnayake SI (2011). Mixtures of factor analyzers for the analysis of high-dimensional data. In *Mixture Estimation and Applications*, KL Mengersen, CP Robert, and DM Titterington (Eds). Hoboken, New Jersey: Wiley, pp. 171–191.

See Also

mcfa

Examples

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

---

plot_factors *Plot Function for Factor Scores.*

Description

Plot function for factor scores given factor score matrix of fitted model.

Usage

```
plot_factors(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". If type = "U", then the estimated conditional expected component scores of the unobservable factors given the data and the component membership are plotted. If type = "Fmat", then the means of the estimated conditional expected factors scores over estimated posterior distributions are plotted. If type = "UC", then an alternative estimate of "Fmat", where the posterior probabilities are replaced by component indicator vector, is plotted.

Author(s)

Geoffrey McLachlan, Suren Rathnayake, Jungsun Baek

References

McLachlan GJ, and Peel D (2000). *Finite Mixture Models*. New York: Wiley.

McLachlan GJ, Baek J, and Rathnayake SI (2011). Mixtures of factor analyzers for the analysis of high-dimensional data. In *Mixture Estimation and Applications*, KL Mengersen, CP Robert, and DM Titterington (Eds). Hoboken, New Jersey: Wiley, pp. 171–191.

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 plot_factors(model, tyep = "Fmat", clust = model$clust)
plot (model)

#can provide alternative groupings of samples via plot_factors
plot_factors (model, clust = iris[indSample, 5])

#same as plot_factors(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 <- factor_scores(Y, model)
plot_factors (fac_scores, type="Fmat", clust = clust)
plot_factors (fac_scores, type="Fmat", clust = iris[-c(indSample), 5])
```

`rmix` *Random Deviates from EMMIXmcfa Models*

Description

Random number generator for `EMMIXmcfa` 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 proprotion 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

https://people.smp.uq.edu.au/GeoffMcLachlan/mix_soft/EMMIX_R/

McLachlan GJ, and Peel D (2000). *Finite Mixture Models*. New York: Wiley.

McLachlan GJ, Baek J, and Rathnayake SI (2011). Mixtures of factor analyzers for the analysis of high-dimensional data. In *Mixture Estimation and Applications*, KL Mengersen, CP Robert, and DM Titterington (Eds). Hoboken, New Jersey: Wiley, pp. 171–191.

Examples

```
## Not run:
 set.seed(1)
 model <- mcfa(iris[, -5], g=3, q=2, nkmeans=1, nrandom=1)
 dat <- rmix(n = 10, model = model)

## End(Not run)
```

Index