Package: autoMFA (via r-universe)

October 31, 2024

Title Algorithms for Automatically Fitting MFA Models

Version 1.1.0

Description Provides methods for fitting the Mixture of Factor Analyzers (MFA) model automatically. The MFA model is a mixture model where each sub-population is assumed to follow the Factor Analysis (FA) model. The FA model is a latent variable model which assumes that observations are normally distributed, but imposes constraints on their covariance matrix. The MFA model contains two hyperparameters; g (the number of components in the mixture) and q (the number of factors in each component Factor Analysis model). Usually, the Expectation-Maximisation algorithm would be used to fit the MFA model, but this requires g and q to be known. This package treats g and q as unknowns and provides several methods which infer these values with as little input from the user as possible. The available methods are a naïve search over both g and q, two different implementations of the AMFA algorithm (Wang and Lin, 2020) <doi = 10.1007/s11749-020-00702-6>, the AMoFA algorithm (Kaya and Salah, 2015) <doi = 10.48550/arXiv.1507.02801> and the VBMFA algorithm (Ghahramani and Beal, 2000) <url = https://mlg.eng.cam.ac.uk/zoubin/papers/nips99.pdf>.

Depends R (>= 3.5.0)

License GPL (>= 3)

Imports abind, MASS, Matrix, Rfast, expm, stats, utils, Rdpack, pracma, ggsci

RdMacros Rdpack

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.0

Repository https://john-c-davey.r-universe.dev

RemoteUrl https://github.com/john-c-davey/automfa

RemoteRef HEAD

RemoteSha 2ddcd05a5712d2a64b3f19f95fcd8c43b5c16a0a

Contents

`AMFA` *Automated Mixtures of Factor Analyzers*

Description

An implementation of AMFA algorithm from (Wang and Lin 2020). The number of factors, q, is estimated during the fitting process of each MFA model. The best value of g is chosen as the model with the minimum BIC of all candidate models in the range `gmin` <= g <= `gmax`.

Usage

```
AMFA(
  Y,
  gmin = 1,
  gmax = 10,
  eta = 0.005,
  itmax = 500,
  nkmeans = 5,
  nrandom = 5,
  tol = 1e-05,
  conv_measure = "diff",
  varimax = FALSE
)
```

Arguments

`Y`	An n by p data matrix, where n is the number of observations and p is the number of dimensions of the data.
`gmin`	The smallest number of components for which an MFA model will be fitted.
`gmax`	The largest number of components for which an MFA model will be fitted.

eta	The smallest possible entry in any of the error matrices *D_i* (Zhao and Yu 2008).
itmax	The maximum number of ECM iterations allowed for the estimation of each MFA model.
nkmeans	The number of times the *k*-means algorithm will be used to initialise models for each combination of *g* and *q*.
nrandom	The number of randomly initialised models that will be used for each combination of *g* and *q*.
tol	The ECM algorithm terminates if the measure of convergence falls below this value.
conv_measure	The convergence criterion of the ECM algorithm. The default `'diff'` stops the ECM iterations if |l^(k+1) - l^(k)| < `tol` where l^(k) is the log-likelihood at the *k*th ECM iteration. If `'ratio'`, then the convergence of the ECM iterations is measured using |(l^(k+1) - l^(k))/l^(k+1)|.
varimax	Boolean indicating whether the output factor loading matrices should be constrained using varimax rotation or not.

Value

A list containing the following elements:

- `model`: A list specifying the final MFA model. This contains:
  - `B`: A *p* by *p* by *q* array containing the factor loading matrices for each component.
  - `D`: A *p* by *p* by *g* array of error variance matrices.
  - `mu`: A *p* by *g* array containing the mean of each cluster.
  - `pivec`: A 1 by *g* vector containing the mixing proportions for each FA in the mixture.
  - `numFactors`: A 1 by *g* vector containing the number of factors for each FA.
- `clustering`: A list specifying the clustering produced by the final model. This contains:
  - `responsibilities`: A *n* by *g* matrix containing the probability that each point belongs to each FA in the mixture.
  - `allocations`: A *n* by 1 matrix containing which FA in the mixture each point is assigned to based on the responsibilities.
- `diagnostics`: A list containing various pieces of information related to the fitting process of the algorithm. This contains:
  - `bic`: The BIC of the final model.
  - `logL`: The log-likelihood of the final model.
  - `times`: A data frame containing the amount of time taken to fit each MFA model.
  - `totalTime`: The total time taken to fit the final model.

References

Wang W, Lin T (2020). "Automated learning of mixtures of factor analysis models with missing information." *TEST*. ISSN 1133-0686.

Zhao J, Yu PLH (2008). "Fast ML Estimation for the Mixture of Factor Analyzers via an ECM Algorithm." *IEEE Transactions on Neural Networks*, **19**(11), 1956-1961. ISSN 1045-9227.

Examples

```
RNGversion('4.0.3'); set.seed(3)
MFA.fit <- AMFA(testDataMFA,3,3, nkmeans = 3, nrandom = 3, itmax = 100)
```

AMFA.inc *Incremental Automated Mixtures of Factor Analyzers*

Description

An alternative implementation of AMFA algorithm (Wang and Lin 2020). The number of factors, q, is estimated during the fitting process of each MFA model. Instead of employing a grid search over g like the AMFA method, this method starts with a *1* component MFA model and splits components according to their multivariate kurtosis. This uses the same approach as amofa (Kaya and Salah 2015). Once a component has been selected for splitting, the new components are initialised in the same manner as vbmfa (Ghahramani and Beal 2000). It keeps trying to split components until all components have had numTries splits attempted with no decrease in BIC, after which the current model is returned.

Usage

```
AMFA.inc(
  Y,
  numTries = 2,
  eta = 0.005,
  itmax = 500,
  tol = 1e-05,
  conv_measure = "diff",
  nkmeans = 1,
  nrandom = 1,
  varimax = FALSE
)
```

Arguments

Y	An n by p data matrix, where n is the number of observations and p is the number of dimensions of the data.
numTries	The number of attempts that should be made to split each component.
eta	The smallest possible entry in any of the error matrices D_i (Zhao and Yu 2008).
itmax	The maximum number of ECM iterations allowed for the estimation of each MFA model.
tol	The ECM algorithm terminates if the measure of convergence falls below this value.
conv_measure	The convergence criterion of the ECM algorithm. The default 'diff' stops the ECM iterations if |l^(k+1) - l^(k)| < tol where l^(k) is the log-likelihood at the kth ECM iteration. If 'ratio', then the convergence of the ECM iterations is measured using |(l^(k+1) - l^(k))/l^(k+1)|.

nkmeans	The number of times the k-means algorithm will be used to initialise the (single component) starting models.
nrandom	The number of randomly initialised (single component) starting models.
varimax	Boolean indicating whether the output factor loading matrices should be constrained using varimax rotation or not.

Value

A list containing the following elements:

- model: A list specifying the final MFA model. This contains:
  - B: A p by p by q array containing the factor loading matrices for each component.
  - D: A p by p by g array of error variance matrices.
  - mu: A p by g array containing the mean of each cluster.
  - pivec: A 1 by g vector containing the mixing proportions for each FA in the mixture.
  - numFactors: A 1 by g vector containing the number of factors for each FA.
- clustering: A list specifying the clustering produced by the final model. This contains:
  - responsibilities: A n by g matrix containing the probability that each point belongs to each FA in the mixture.
  - allocations: A n by 1 matrix containing which FA in the mixture each point is assigned to based on the responsibilities.
- diagnostics: A list containing various pieces of information related to the fitting process of the algorithm. This contains:
  - bic: The BIC of the final model.
  - logL: The log-likelihood of the final model.
  - totalTime: The total time taken to fit the final model.

References

Wang W, Lin T (2020). "Automated learning of mixtures of factor analysis models with missing information." *TEST*. ISSN 1133-0686.

Kaya H, Salah AA (2015). "Adaptive Mixtures of Factor Analyzers." *arXiv preprint arXiv:1507.02801*.

Ghahramani Z, Beal MJ (2000). "Variational inference for Bayesian Mixtures of Factor Analysers." In *Advances in neural information processing systems*, 449–455.

Zhao J, Yu PLH (2008). "Fast ML Estimation for the Mixture of Factor Analyzers via an ECM Algorithm." *IEEE Transactions on Neural Networks*, **19**(11), 1956-1961. ISSN 1045-9227.

See Also

amofa vbmfa

Examples

```
RNGversion('4.0.3'); set.seed(3)
MFA.fit <- AMFA_inc(testDataMFA, itmax = 1, numTries = 0)
```

AMFA_inc *Incremental Automated Mixtures of Factor Analyzers*

Description

An alternative implementation of AMFA algorithm (Wang and Lin 2020). The number of factors, q, is estimated during the fitting process of each MFA model. Instead of employing a grid search over g like the AMFA method, this method starts with a *1* component MFA model and splits components according to their multivariate kurtosis. This uses the same approach as amofa (Kaya and Salah 2015). Once a component has been selected for splitting, the new components are initialised in the same manner as vbmfa (Ghahramani and Beal 2000). It keeps trying to split components until all components have had numTries splits attempted with no decrease in BIC, after which the current model is returned.

Usage

```
AMFA_inc(
  Y,
  numTries = 2,
  eta = 0.005,
  itmax = 500,
  tol = 1e-05,
  conv_measure = "diff",
  nkmeans = 1,
  nrandom = 1,
  varimax = FALSE
)
```

Arguments

Y	An n by p data matrix, where n is the number of observations and p is the number of dimensions of the data.
numTries	The number of attempts that should be made to split each component.
eta	The smallest possible entry in any of the error matrices D_i (Zhao and Yu 2008).
itmax	The maximum number of ECM iterations allowed for the estimation of each MFA model.
tol	The ECM algorithm terminates if the measure of convergence falls below this value.
conv_measure	The convergence criterion of the ECM algorithm. The default 'diff' stops the ECM iterations if |l^(k+1) - l^(k)| < tol where l^(k) is the log-likelihood at the kth ECM iteration. If 'ratio', then the convergence of the ECM iterations is measured using |(l^(k+1) - l^(k))/l^(k+1)|.
nkmeans	The number of times the k-means algorithm will be used to initialise the (single component) starting models.
nrandom	The number of randomly initialised (single component) starting models.

`varimax` Boolean indicating whether the output factor loading matrices should be constrained using varimax rotation or not.

Value

A list containing the following elements:

- `model`: A list specifying the final MFA model. This contains:
  - `B`: A p by p by q array containing the factor loading matrices for each component.
  - `D`: A p by p by g array of error variance matrices.
  - `mu`: A p by g array containing the mean of each cluster.
  - `pivec`: A 1 by g vector containing the mixing proportions for each FA in the mixture.
  - `numFactors`: A 1 by g vector containing the number of factors for each FA.
- `clustering`: A list specifying the clustering produced by the final model. This contains:
  - `responsibilities`: A n by g matrix containing the probability that each point belongs to each FA in the mixture.
  - `allocations`: A n by 1 matrix containing which FA in the mixture each point is assigned to based on the responsibilities.
- `diagnostics`: A list containing various pieces of information related to the fitting process of the algorithm. This contains:
  - `bic`: The BIC of the final model.
  - `logL`: The log-likelihood of the final model.
  - `totalTime`: The total time taken to fit the final model.

References

Wang W, Lin T (2020). "Automated learning of mixtures of factor analysis models with missing information." *TEST*. ISSN 1133-0686.

Kaya H, Salah AA (2015). "Adaptive Mixtures of Factor Analyzers." *arXiv preprint arXiv:1507.02801*.

Ghahramani Z, Beal MJ (2000). "Variational inference for Bayesian Mixtures of Factor Analysers." In *Advances in neural information processing systems*, 449–455.

Zhao J, Yu PLH (2008). "Fast ML Estimation for the Mixture of Factor Analyzers via an ECM Algorithm." *IEEE Transactions on Neural Networks*, **19**(11), 1956-1961. ISSN 1045-9227.

See Also

`amofa` `vbmfa`

Examples

```
RNGversion('4.0.3'); set.seed(3)
MFA.fit <- AMFA_inc(testDataMFA, itmax = 1, numTries = 0)
```

amofa *Adaptive Mixture of Factor Analyzers (AMoFA)*

Description

An implementation of the Adaptive Mixture of Factor Analyzers (AMoFA) algorithm from (Kaya and Salah 2015). This code is a R port of the MATLAB code which was included with that paper.

Usage

```
amofa(Y, itmax = 100, verbose = FALSE, varimax = FALSE)
```

Arguments

Y	An n by p data matrix, where n is the number of observations and p is the number of dimensions of the data.
itmax	The maximum number of EM iterations allowed for the estimation of each MFA model.
verbose	Boolean indicating whether or not to print more verbose output, including the number of EM-iterations used and the total running time. Default is FALSE.
varimax	Boolean indicating whether the output factor loading matrices should be constrained using varimax rotation or not.

Value

A list containing the following elements:

- model: A list specifying the final MFA model. This contains:
  - B: A list containing the factor loading matrices for each component.
  - D: A p by p by g array of error variance matrices.
  - mu: A p by g array containing the mean of each cluster.
  - pivec: A 1 by g vector containing the mixing proportions for each FA in the mixture.
  - numFactors: A *1* by g vector containing the number of factors for each FA.
- clustering: A list specifying the clustering produced by the final model. This contains:
  - responsibilities: A n by g matrix containing the probability that each point belongs to each FA in the mixture.
  - allocations: A n by 1 matrix containing which FA in the mixture each point is assigned to based on the responsibilities.
- diagnostics: A list containing various pieces of information related to the fitting process of the algorithm. This contains:
  - bic: The BIC of the final model.
  - logL: The log-likelihood of the final model.
  - totalEM: The total number of EM iterations used.
  - progress: A matrix containing information about the decisions made by the algorithm.
  - times: The time taken for each loop in the algorithm.
  - totalTime: The total time taken to fit the final model.

References

Kaya H, Salah AA (2015). "Adaptive Mixtures of Factor Analyzers." *arXiv preprint arXiv:1507.02801*.

Examples

```
RNGversion('4.0.3'); set.seed(3)
MFA.fit <- amofa(testDataMFA)
```

`MFA_ECM` *ECM-Based MFA Estimation*

Description

An implementation of an ECM algorithm for the MFA model which does not condition on the factors being known (Zhao and Yu 2008). Performs a grid search from `gmin` to `gmax`, and `qmin` to `qmax`, respectively. The best combination of g and q is chosen to be the model with the minimum BIC.

Usage

```
MFA_ECM(
  Y,
  gmin = 1,
  gmax = 10,
  qmin = 1,
  qmax = NULL,
  eta = 0.005,
  itmax = 500,
  nkmeans = 5,
  nrandom = 5,
  tol = 1e-05,
  conv_measure = "diff",
  varimax = FALSE
)
```

Arguments

`Y`	An n by p data matrix, where n is the number of observations and p is the number of dimensions of the data.
`gmin`	The smallest number of components for which an MFA model will be fitted.
`gmax`	The largest number of components for which an MFA model will be fitted.
`qmin`	The smallest number of factors with which an MFA model will be fitted.
`qmax`	The largest number of factors with which an MFA model will be fitted. Must obey the Ledermann bound.
`eta`	The smallest possible entry in any of the error matrices D_i (Zhao and Yu 2008).

itmax	The maximum number of ECM iterations allowed for the estimation of each MFA model.
nkmeans	The number of times the *k*-means algorithm will be used to initialise models for each combination of *g* and *q*.
nrandom	The number of randomly initialised models that will be used for each combination of g and q.
tol	The ECM algorithm terminates if the measure of convergence falls below this value.
conv_measure	The convergence criterion of the ECM algorithm. The default 'diff' stops the ECM iterations if |l^(k+1) - l^(k)| < tol where l^(k) is the log-likelihood at the *k*th ECM iteration. If 'ratio', then the convergence of the ECM iterations is measured using |(l^(k+1) - l^(k))/l^(k+1)|.
varimax	Boolean indicating whether the output factor loading matrices should be constrained using varimax rotation or not.

Value

A list containing the following elements:

- model: A list specifying the final MFA model. This contains:
  - B: A *p* by *p* by *q* array containing the factor loading matrices for each component.
  - D: A *p* by *p* by *g* array of error variance matrices.
  - mu: A *p* by *g* array containing the mean of each cluster.
  - pivec: A 1 by *g* vector containing the mixing proportions for each FA in the mixture.
  - numFactors: A 1 by *g* vector containing the number of factors for each FA.
- clustering: A list specifying the clustering produced by the final model. This contains:
  - responsibilities: A *n* by *g* matrix containing the probability that each point belongs to each FA in the mixture.
  - allocations: A *n* by 1 matrix containing which FA in the mixture each point is assigned to based on the responsibilities.
- diagnostics: A list containing various pieces of information related to the fitting process of the algorithm. This contains:
  - bic: The BIC of the final model.
  - logL: The log-likelihood of the final model.
  - times: A data frame containing the amount of time taken to fit each MFA model.
  - totalTime: The total time taken to fit the final model.

References

Zhao J, Yu PLH (2008). "Fast ML Estimation for the Mixture of Factor Analyzers via an ECM Algorithm." *IEEE Transactions on Neural Networks*, **19**(11), 1956-1961. ISSN 1045-9227.

Examples

```
RNGversion('4.0.3'); set.seed(3)
MFA.fit <- MFA_ECM(testDataMFA,3,3)
```

preprocess *Preprocess*

Description

Performs the pre-processing of a data matrix such that it is ready to be used by vbmfa.

Usage

```
preprocess(Y, ppp, shrinkQ)
```

Arguments

Y	An n by p data matrix which is to be scaled.
ppp	An optional p by 2 matrix where the columns represent the sample mean and sample standard deviation of the pth dimension of Y.
shrinkQ	If 1, the data is shrunk according to ppp. If 0, the data is expanded to invert a prior shrinking by ppp.

Value

A list containing

- Yout: A processed data matrix of observations.
- ppp: The shrinkage which as applied in the processing.

References

Ghahramani Z, Beal MJ (2000). "Variational inference for Bayesian Mixtures of Factor Analysers." In *Advances in neural information processing systems*, 449–455.

See Also

vbmfa for fitting models after using preprocess.

Examples

```
Yout <- preprocess(testDataMFA);
```

testDataMFA *Test dataset for the MFA model*

Description

A 720 x 3 test dataset generated from a MFA model with 3 components, 1 factor for each component. Uneven point distribution with large separation between clusters relative to the component variance matrices.

Usage

```
testDataMFA
```

Format

Data matrix with 720 observations of 3 variables. Generated using an MFA model with the following parameters:

- `pivec` Mixing proportion vector (0.5722, 0.3333, 0.0944) which corresponds to component sizes of 412, 240 and 68.
- `mu` Mean vectors (3;0;0), (0;3;0) and (0,0,3) respectively.
- `B` Loading matrices (0.8827434; -0.5617922; 0.0277005), (0.03121194; 0.14964642; 0.01180723) and (0.1306169; 0.7450665; 0.4357088) respectively.
- `D` Error variance matrices of diag(0.1) for all components.

Examples

```
pairs(testDataMFA)
```

vbmfa *Variational Bayesian Mixture of Factor Analyzers (VB-MoFA)*

Description

An implementation of the Variational Bayesian Mixutre of Factor Analysers (Ghahramani and Beal 2000). This code is an R port of the MATLAB code which was written by M.J.Beal and released alongside their paper.

Usage

```
vbmfa(Y, qmax = NULL, numTries = 3, verbose = FALSE, varimax = FALSE)
```

Arguments

Y	An n by p (normalised) data matrix (i.e. the result of a call to the function preprocess), where n is the number of observations and p is the number of dimensions of the data.
qmax	Maximum factor dimensionality (default p-1).
numTries	The maximum number of times the algorithm will attempt to split each component.
verbose	Whether or not verbose output should be printed during the model fitting process (defaults to false).
varimax	Boolean indicating whether the output factor loading matrices should be constrained using varimax rotation or not.

Value

A list containing the following elements:

- model: A list specifying the final MFA model. This contains:
  - B: A p by p by q array containing the factor loading matrices for each component.
  - D: A p by p by g array of error variance matrices.
  - mu: A p by g array containing the mean of each cluster.
  - pivec: A 1 by g vector containing the mixing proportions for each FA in the mixture.
  - numFactors: A 1 by g vector containing the number of factors for each FA.
- clustering: A list specifying the clustering produced by the final model. This contains:
  - responsibilities: A n by g matrix containing the probability that each point belongs to each FA in the mixture.
  - allocations: A n by 1 matrix containing which FA in the mixture each point is assigned to based on the responsibilities.
- diagnostics: A list containing various pieces of information related to the fitting process of the algorithm. This contains:
  - bic: The BIC of the final model.
  - logL: The log-likelihood of the final model.
  - Fhist:The values of F at each iteration of the algorithm. F is defined in (Ghahramani and Beal 2000).
  - times: The time taken for each loop in the algorithm.
  - totalTime: The total time taken to fit the final model.

References

Ghahramani Z, Beal MJ (2000). "Variational inference for Bayesian Mixtures of Factor Analysers." In *Advances in neural information processing systems*, 449–455.

See Also

preprocess for centering and scaling data prior to using vbmfa.

Examples

```
RNGversion('4.0.3'); set.seed(3)
Yout <- preprocess(testDataMFA)
MFA.fit <- vbmfa(Yout$Yout, numTries = 2)
```

Index