Package ‘stepmixr’

November 18, 2022

Note -*- Encoding: utf-8 -*-

Type Package

Title Interface to 'Python' Package 'stepmix'

Version 0.1.0

Date 2022-10-28

Author Charles-Édouard Giguère [aut, cre],
Sacha Morin [aut],
Robin Legault [aut]

Maintainer Charles-Édouard Giguère <ce.giguere@gmail.com>

Depends R (>= 4.0.0)

Imports reticulate (>= 1.8)

Description This is an interface for the 'python' package 'stepmix'. This is a package multi-step estimation of latent class models with measurement and structural components. The package can also be used to fit mixture models with various observed random variables. Largely based on Bakk & Kuha (2018) <doi:10.1007/s11336-017-9592-7>.

License GPL-2

Encoding UTF-8

LazyLoad TRUE

URL https://github.com/Labo-Lacourse/stepmixr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-11-18 10:00:02 UTC

R topics documented:

fit 2
install.stepmix 3
mixed_descriptor 4

predict.stepmix.stepmix.StepMix 5
savefit 6
stepmix 7

Index **11**

fit *Fit a mixture using the stepmix python package.*

Description

This function initializes the stepmix object in python and fit X and optionnally Y to the object.

Usage

```
fit(smx, X = NULL, Y = NULL)
```

Arguments

`smx`	An object created with the stepmix function.
`X`	The X matrix or data.frame for the measurement part of the model
`Y`	The Y matrix or data.frame for the structural part of the model

Details

This methods returns a pointer to a python object of type StepMix. It can be used within reticulate but not within R. To save this type of object, you need to use the savefit function.

Value

A pointer to a python object of type StepMix.

Author(s)

Charles-Édouard Giguère (stepmixr) Sacha Morin (stepmix) Robin Legault (stepmix)

References

Bolck, A., Croon, M., and Hagenaars, J. Estimating latent structure models with categorical variables: One-step versus three-step estimators. Political analysis, 12(1): 3-27, 2004.

Vermunt, J. K. Latent class modeling with covariates: Two improved three-step approaches. Political analysis, 18 (4):450-469, 2010.

Bakk, Z., Tekle, F. B., and Vermunt, J. K. Estimating the association between latent class membership and external variables using bias-adjusted three-step approaches. Sociological Methodology, 43(1):272-311, 2013.

Bakk, Z. and Kuha, J. Two-step estimation of models between latent classes and external variables. Psychometrika, 83(4):871-892, 2018

Examples

```
if (reticulate::py_module_available("stepmix")) {
  model1 <- stepmix(n_components = 3, n_steps = 2, measurement = "continuous")
  X <- iris[c(1:10, 51:60, 101:110), 1:4]
  fit1 <- fit(model1, X)
}
```

`install.stepmix` *Install stepmix python package into python via reticulate.*

Description

Install the stepmix python package in the python instance used by reticulate.

Usage

```
install.stepmix(envname, method, conda, pip, ...)
check_pystepmix_version()
```

Arguments

`envname` Name of the python environment. "r-reticulate" by default.

`method` installation method. See doc in reticulate

`conda` Path to a conda install. See doc in reticulate

`pip` Logical value to choose pip as the install method

`...` Not used in function

Details

This methods installs stepmix in the python instance or environment used by reticulate. It uses reticulate::py_install.

Value

It doesn't return anything.

Author(s)

Charles-Édouard Giguère (stepmixr) Sacha Morin (stepmix) Robin Legault (stepmix)

References

Bolck, A., Croon, M., and Hagenaars, J. Estimating latent structure models with categorical variables: One-step versus three-step estimators. Political analysis, 12(1): 3-27, 2004.

Vermunt, J. K. Latent class modeling with covariates: Two improved three-step approaches. Political analysis, 18 (4):450-469, 2010.

Bakk, Z., Tekle, F. B., and Vermunt, J. K. Estimating the association between latent class membership and external variables using bias-adjusted three-step approaches. Sociological Methodology, 43(1):272-311, 2013.

Bakk, Z. and Kuha, J. Two-step estimation of models between latent classes and external variables. Psychometrika, 83(4):871-892, 2018

`mixed_descriptor` *Utility function for mixture using mixed description.*

Description

This function creates a data.frame ordered by continuous, binary and categorical columns. It also creates a list used if the model uses mixed column types.

Usage

```
mixed_descriptor(data, continuous = NULL, binary = NULL, categorical = NULL)
```

Arguments

`data`	Data.frame with the mixed data
`continuous`	index or name of continuous column
`binary`	index or name of binary column
`categorical`	index or name of categorical column

Details

This methods returns a list of a data.frame sorted by continuous, binary and categorical columns. It contains also a descriptor that can be used in the measurement section.

Value

A list containing data and a descriptor.

Author(s)

Charles-Édouard Giguère (stepmixr) Sacha Morin (stepmix) Robin Legault (stepmix)

References

Bolck, A., Croon, M., and Hagenaars, J. Estimating latent structure models with categorical variables: One-step versus three-step estimators. Political analysis, 12(1): 3-27, 2004.

Vermunt, J. K. Latent class modeling with covariates: Two improved three-step approaches. Political analysis, 18 (4):450-469, 2010.

Bakk, Z., Tekle, F. B., and Vermunt, J. K. Estimating the association between latent class membership and external variables using bias-adjusted three-step approaches. Sociological Methodology, 43(1):272-311, 2013.

Bakk, Z. and Kuha, J. Two-step estimation of models between latent classes and external variables. Psychometrika, 83(4):871-892, 2018

Examples

```
md <- mixed_descriptor(iris, continuous = 1:4, categorical = 5)
```

`predict.stepmix.stepmix.StepMix` *Predict the membership using the fit of the stepmix python package.*

Description

Predict the membership of a mixture using a stepmix object in python using X and optionally Y to the object.

Usage

```
## S3 method for class 'stepmix.stepmix.StepMix'
predict(object, X = NULL, Y = NULL, ...)
```

Arguments

`object`	An object created with the fit function.
`X`	The X matrix or data.frame for the measurement part of the model
`Y`	The Y matrix or data.frame for the structural part of the model
`...`	not used in this function

Value

A vector containing the membership of the mixture.

Author(s)

Charles-Édouard Giguère (stepmixr) Sacha Morin (stepmix) Robin Legault (stepmix)

References

Bolck, A., Croon, M., and Hagenaars, J. Estimating latent structure models with categorical variables: One-step versus three-step estimators. Political analysis, 12(1): 3-27, 2004.

Vermunt, J. K. Latent class modeling with covariates: Two improved three-step approaches. Political analysis, 18 (4):450-469, 2010.

Bakk, Z., Tekle, F. B., and Vermunt, J. K. Estimating the association between latent class membership and external variables using bias-adjusted three-step approaches. Sociological Methodology, 43(1):272-311, 2013.

Bakk, Z. and Kuha, J. Two-step estimation of models between latent classes and external variables. Psychometrika, 83(4):871-892, 2018

Examples

```
if (reticulate::py_module_available("stepmix")) {
  require(stepmixr)
  model1 <- stepmix(n_components = 3, n_steps = 2, measurement = "continuous")
  X <- iris[c(1:10, 51:60, 101:110), 1:4]
  fit1 <- fit(model1, X)
  pr1 <- predict(fit1, X)
}
```

savefit *Save the fit of a mixture using the stepmix python package.*

Description

This function saves the stepmix fitted object in python using the pickle package.

Usage

```
savefit(fitx, f)
loadfit(f)
```

Arguments

fitx	An object created with the stepmix function.
f	String indicating the name of the file

Details

This methods allows to save/load the stepmix object in a binary file using the pickle package.

Value

A pointer to a python object of type StepMix.

Author(s)

Charles-Édouard Giguère (stepmixr) Sacha Morin (stepmix) Robin Legault (stepmix)

References

Bolck, A., Croon, M., and Hagenaars, J. Estimating latent structure models with categorical variables: One-step versus three-step estimators. Political analysis, 12(1): 3-27, 2004.

Vermunt, J. K. Latent class modeling with covariates: Two improved three-step approaches. Political analysis, 18 (4):450-469, 2010.

Bakk, Z., Tekle, F. B., and Vermunt, J. K. Estimating the association between latent class membership and external variables using bias-adjusted three-step approaches. Sociological Methodology, 43(1):272-311, 2013.

Bakk, Z. and Kuha, J. Two-step estimation of models between latent classes and external variables. Psychometrika, 83(4):871-892, 2018

Examples

```
if (reticulate::py_module_available("stepmix")) {
  model1 <- stepmix(n_components = 2, n_steps = 3)
  X <- data.frame(x1 = c(0,1,1,1,1,0,0,0,0,0,1,1,0),
                  x2 = c(0,1,1,0,0,1,1,0,0,0,1,0,1))
  fit1 <- fit(model1, X)
  savefit(fit1, "fit1.pickle")

  ### clean the directory.
  file.remove("fit1.pickle")
}
```

stepmix *R interface to stepmix in StepMix python.*

Description

This function creates a basic R list that will be used to initialize the stepmix object in python, in order to use the fit and predict function.

Usage

```
stepmix(n_components = 2, n_steps = 1,
        measurement = "bernoulli", structural = "bernoulli",
        assignment = "modal", correction = NULL,
        abs_tol = 1e-3, rel_tol = 1e-10, max_iter = 100,
        n_init = 1, init_params = "random", random_state = NULL,
        verbose = 0, verbose_interval = 10,
        measurement_params = NULL, structural_params = NULL)
```

Arguments

n_components The number of latent class. 2 by default.

n_steps 1, 2, or 3, 1 by default. Number of steps in the estimation. Must be one of : 1: run EM on both the measurement and structural models.

2: first run EM on the measurement model, then on the complete model, but keep the measurement parameters fixed for the second step. See Bakk, 2018.

3: first run EM on the measurement model, assign class probabilities, then fit the structural model via maximum likelihood. See the correction parameter for bias correction.

See Bakk & Kuha (2018) for more details.

measurement String describing the measurement model. See details for the different available model. The default model is "bernouilli"

structural String describing the structural model. See details for the different available model. The default model is "bernouilli"

assignment String indicating the type of class assignments for 3-step estimation, "modal" by default. Must be one of:

soft: keep class responsibilities (posterior probabilities) as is.

modal: assign 1 to the class with max probability, 0 otherwise (one-hot encoding).

correction Bias correction for 3-step estimation. Must be one of :

None: No correction. Run Naive 3-step.

BCH: Apply the empirical BCH correction from Vermunt, 2004.

ML: Apply the ML correction from Vermunt, 2010, Bakk et al., 2013.

abs_tol The convergence threshold. EM iterations will stop when the lower bound average gain is below this threshold. The default value is 1e-3.

rel_tol The convergence threshold. EM iterations will stop when the relative lower bound average gain is below this threshold.

max_iter The number of EM iterations to perform.

n_init The number of initializations to perform. The best results are kept.

init_params "kmeans", or "random", default="random". The method used to initialize the weights, the means and the precisions. Must be one of:

kmeans : responsibilities are initialized using kmeans.

random : responsibilities are initialized randomly.

random_state State instance or NULL, default=NULL. Controls the random seed given to the method chosen to initialize the parameters. Pass an int for reproducible output across multiple function calls.

verbose Default=0. Enable verbose output. If 1, will print detailed report of the model and the performance metrics after fitting.

verbose_interval

Default=10. Number of iteration done before the next print. TODO: Not currently implemented.

`measurement_params`
: Default=NULL, Additional params passed to the measurement model class. Particularly useful to specify optimization parameters for stepmix.emission.covariate.Covariate. Ignored if the measurement descriptor is a nested object (see stepmix.emission.nested.Nested).

`structural_params`
: Default=NULL, Additional params passed to the structural model class. Particularly useful to specify optimization parameters for stepmix.emission.covariate.Covariate. Ignored if the structural descriptor is a nested object (see stepmix.emission.nested.Nested).

Details

The options for both the measurement and structural part are describe here:

bernoulli: The observed data consists of n_features bernoulli (binary) random variables.

bernoulli_nan: the observed data consists of n_features bernoulli (binary) random variables. Supports missing values.

binary: alias for bernoulli.

binary_nan: alias for bernoulli_nan.

categorical: alias for multinoulli.

categorical_nan: alias for multinoulli_nan.

continuous: alias for gaussian diag.

continuous_nan: alias for gaussian_diag_nan. supports missing values.

covariate: covariate model where class probabilities are a multinomial logistic model of the features.

gaussian: alias for gaussian_unit.

gaussian_nan: alias for gaussian_unit. Supports missing values.

gaussian_unit: each gaussian component has unit variance. Only fit the mean.

gaussian_unit_nan: each gaussian component has unit variance. Only fit the mean. Supports missing values.

gaussian_spherical: each gaussian component has its own single variance.

gaussian_spherical_nan: each gaussian component has its own single variance. Supports missing values.

gaussian_tied: all gaussian components share the same general covariance matrix.

gaussian_diag: each gaussian component has its own diagonal covariance matrix.

gaussian_diag_nan: each gaussian component has its own diagonal covariance matrix. Supports missing values.

gaussian_full: each gaussian component has its own general covariance matrix.

multinoulli: the observed data consists of n_features multinoulli (categorical) random variables.

multinoulli_nan: the observed data consists of n_features multinoulli (categorical) random variables. Supports missing values.

Value

It returns a list of type stepmixr that contains the arguments of the object.

Author(s)

Charles-Édouard Giguère (stepmixr) Sacha Morin (stepmix) Robin Legault (stepmix)

References

Bolck, A., Croon, M., and Hagenaars, J. Estimating latent structure models with categorical variables: One-step versus three-step estimators. Political analysis, 12(1): 3-27, 2004.

Vermunt, J. K. Latent class modeling with covariates: Two improved three-step approaches. Political analysis, 18 (4):450-469, 2010.

Bakk, Z., Tekle, F. B., and Vermunt, J. K. Estimating the association between latent class membership and external variables using bias-adjusted three-step approaches. Sociological Methodology, 43(1):272-311, 2013.

Bakk, Z. and Kuha, J. Two-step estimation of models between latent classes and external variables. Psychometrika, 83(4):871-892, 2018

See Also

`fit`

Examples

```
model1 <- stepmix(n_components = 2, n_steps = 3)
```

Index

check_pystepmix_version
(install.stepmix), 3

fit, 2, *10*

install.stepmix, 3

loadfit *(*savefit*)*, 6

mixed_descriptor, 4

predict.stepmix.stepmix.StepMix, 5

savefit, 6
stepmix, 7