Package 'sBIC'

May 9, 2026

Type Package

Title Computing the Singular BIC for Multiple Models

Version 0.2.0

Date 2016-09-20

Description Computes the sBIC for various singular model collections including: binomial mixtures, factor analysis models, Gaussian mixtures, latent forests, latent class analyses, and reduced rank regressions.

License GPL (>= 3)

URL https://github.com/Lucaweihs/sBIC

BugReports https://github.com/Lucaweihs/sBIC/issues

Imports poLCA, R.oo (>= 1.20.0), R.methodsS3, mclust, igraph (>= 1.0.1), Rcpp (>= 0.12.3), combinat, flexmix, hash

LinkingTo Rcpp

Suggests testthat, mvtnorm, knitr, MASS

VignetteBuilder knitr

RoxygenNote 5.0.1

Collate 'ModelPoset.R' 'MixtureModels.R' 'BinomialMixtures.R' 'FactorAnalyses.R' 'GaussianMixtures.R' 'LCAs.R' 'LatentForests-Helpers.R' 'LatentForests.R' 'RcppExports.R' 'ReducedRankRegressions.R' 'sBIC-Function.R' 'sbic.R'

NeedsCompilation yes

Author Luca Weihs [aut, cre], Martyn Plummer [ctb]

Maintainer Luca Weihs <lucaw@uw.edu>

Repository CRAN

Date/Publication 2016-10-01 14:31:14

Contents

`sBIC-package` *sBIC package documentation.*

Description

Computes the sBIC for various model collections including:

- Binomial mixtures
- Factor analyses
- Gaussian mixtures
- Latent forests
- Latent class analyses
- Reduced rank regressions

The primary functionality of this package can be accessed through the sBIC function.

`BinomialMixtures` *Construct a poset of binomial mixture models.*

Description

Creates an object representing a collection of binomial mixture models. There is one model for each fixed number of components from 1 to some specified maximum. In particular each model is identified by a single number specifiying the number of components in the model. Models are naturally ordered by inclusion so that, for example, a model with 2 components comes before a model with 3 or more components.

Usage

```
BinomialMixtures(maxNumComponents = 1, phi = "default")
```

Arguments

`maxNumComponents`	the maximum number of components allowed in a model, will create a hierarchy of all models with less than or equal to this number.
`phi`	parameter controlling the strength of the sBIC penalty.

Value

An object representing the collection.

emMain *EM-algorithm for latent forests.*

Description

Uses the EM-algorithm (with multiple random restarts) to compute an approximate maximum likelihood estimate for a given latent forest model.

Usage

```
emMain(this, model, starts, maxIter, tol)

## S3 method for class 'LatentForests'
emMain(this, model, starts=5, maxIter=1000, tol=1e-4)
```

Arguments

this	the LatentForests object.
model	the model for which to compute the approximate MLE.
starts	the number of random restarts.
maxIter	the maximum number of iterations to complete in the algorithm.
tol	the tolerance to use a convergence criterion.

emSteps *One EM-iteration.*

Description

A private method that performs a single iteration of the EM-algorithm, this is a helper function for emMain method.

Usage

```
emSteps(this, support, S)

## S3 method for class 'LatentForests'
emSteps(this, support, S)
```

Arguments

this	the LatentForests object.
support	the support of the model.
S	the current covariance matrix.

FactorAnalyses *Construct a poset of factor analysis models.*

Description

Creates an object representing a collection of factor analysis models. There is one model for each fixed number of factors from 0 to some specified maximum number of factors. Each model is identified by a single number, in particular the model with n factors is associated with the model numbered (n + 1), this is because we allow a model with 0 factors and would like all models to be numbered with strictly positive integers for consistency. Models are naturally ordered by inclusion so that, for example, a model with 2 factors comes before a model with 3 or more factors.

Usage

```
FactorAnalyses(numCovariates = 1, maxNumFactors = 0)
```

Arguments

numCovariates	the number of covariates in all of the models.
maxNumFactors	the maximum number of factors allowed in a model, will create a hierarchy of all models with less than or equal to this number.

Value

An object representing the collection.

GaussianMixtures *Construct a poset of gaussian mixture models.*

Description

Creates an object representing a collection of gaussian mixture models. There is one model for each fixed number of components from 1 to some specified maximum. In particular each model is identified by a single number specifiying the number of components in the model. Models are naturally ordered by inclusion so that, for example, a model with 2 components comes before a model with 3 or more components.

Usage

```
GaussianMixtures(maxNumComponents = 1, dim = 1,
                 phi = "default", restarts = 50)
```

Arguments

maxNumComponents	the maximum number of gaussian components to consider in a mixture.
dim	the ambient dimension in which the gaussian mixtures reside. Default is 1, corresponding to gaussian mixtures on the real line.
phi	parameter controlling the strength of the sBIC penalty.
restarts	the number of random restarts to perform when computing the MLE.

Value

An object representing the collection.

---

generateAllBinaryTrees *Generate all non-isomorphic binary trees.*

Description

Generates all non-isomorphic binary trees with a given number of leaves where leaves are considered labeled and inner nodes are unlabeled. Takes as argument the number of leaves for which to produce the binary trees and returns a list of (n-1)x2 matrices where each row corresponds to a edge in the tree. These edge matrices will be in 'directed order,' i.e. will be so that if they are considered to be directed edges then the resulting graph will have exactly one source.

Usage

```
generateAllBinaryTrees(numLeaves)
```

Arguments

numLeaves	the number of leaves

---

getAllEdges *Edges representing the largest model.*

Description

When creating the LatentForests object a set of edges representing the largest model is required. This function returns those edges as a matrix. This matrix will have edges in the same order but may have flipped which node comes first in any particular edge. That is if edge (1,4) was the 5th edge then it will remain the 5th edge but may now be of the form (4,1).

Usage

```
getAllEdges(this, model)

## S3 method for class 'LatentForests'
getAllEdges(this, model)
```

Arguments

this	the LatentForests object.
model	the model number.

---

getCovMat *Create a covariance matrix.*

Description

Creates a covariance matrix for the latent forest model where edge correlations are given. Here edge correlations are given as a vector and correspond (in order) to the edges returned by this$getAllEdges().

Usage

```
getCovMat(this, edgeCorrelations)

## S3 method for class 'LatentForests'
getCovMat(this, edgeCorrelations)
```

Arguments

this	the LatentForests object.
edgeCorrelations	the edge correlations in a numeric vector.

---

getData *Return the set data.*

Description

If data has been set for the object using the setData() function then will return that data, otherwise will throw an error.

Usage

```
getData(this)

## S3 method for class 'BinomialMixtures'
getData(this)

## S3 method for class 'FactorAnalyses'
getData(this)

## S3 method for class 'GaussianMixtures'
getData(this)

## S3 method for class 'LCAs'
getData(this)

## S3 method for class 'LatentForests'
getData(this)

## S3 method for class 'ReducedRankRegressions'
getData(this)
```

Arguments

this	the object from which to get the data.

`getDimension` *Model dimension.*

Description

Computes the dimension of a model in the model poset.

Usage

```
getDimension(this, model)

## S3 method for class 'BinomialMixtures'
getDimension(this, model)

## S3 method for class 'FactorAnalyses'
getDimension(this, model)

## S3 method for class 'GaussianMixtures'
getDimension(this, model)

## S3 method for class 'LCAs'
getDimension(this, model)
```

```
## S3 method for class 'LatentForests'
getDimension(this, model)

## S3 method for class 'ReducedRankRegressions'
getDimension(this, model)
```

Arguments

this	the object representing the model poset.
model	the model for which the dimension should be computed.

---

getModelWithSupport *Get model with the given support.*

Description

Returns the model number corresponding to a given 0-1 vector representing the support of the model. This support should corresponds to the edges returned by this$getAllEdges()

Usage

```
getModelWithSupport(this, support)

## S3 method for class 'LatentForests'
getModelWithSupport(this, support)
```

Arguments

this	the LatentForests object.
support	the 0-1 vector representing the support.

---

getNumFactorsForModel *Number of factors for a model.*

Description

Given a model number returns the number of factors in that model

Usage

```
getNumFactorsForModel(this, model)

## S3 method for class 'FactorAnalyses'
getNumFactorsForModel(this, model)
```

Arguments

this	the FactorAnalyses object.
model	the model number.

---

getNumLeaves *Get number of leaves.*

Description

Gets the number of leaves in the latent forest models.

Usage

```
getNumLeaves(this)

## S3 method for class 'LatentForests'
getNumLeaves(this)
```

Arguments

this	the LatentForests object.

---

getNumModels *Number of models.*

Description

Returns the number of models in the collection.

Usage

```
getNumModels(this)

## S3 method for class 'BinomialMixtures'
getNumModels(this)

## S3 method for class 'FactorAnalyses'
getNumModels(this)

## S3 method for class 'GaussianMixtures'
getNumModels(this)

## S3 method for class 'LCAs'
getNumModels(this)
```

```
## S3 method for class 'LatentForests'
getNumModels(this)

## S3 method for class 'ReducedRankRegressions'
getNumModels(this)
```

Arguments

this	the model poset object.

`getNumSamples` *Number of samples in the set data.*

Description

If data has been set using the setData method then returns the number of samples in the data. Otherwise throws an error.

Usage

```
getNumSamples(this)

## S3 method for class 'BinomialMixtures'
getNumSamples(this)

## S3 method for class 'FactorAnalyses'
getNumSamples(this)

## S3 method for class 'GaussianMixtures'
getNumSamples(this)

## S3 method for class 'LCAs'
getNumSamples(this)

## S3 method for class 'LatentForests'
getNumSamples(this)

## S3 method for class 'ReducedRankRegressions'
getNumSamples(this)
```

Arguments

this	the object from which to get the number of samples.

getNumVertices *Maximum number of vertices.*

Description

A private method for LatentForests that computes the number of vertices a tree with this$getNumLeaves() number of leaves has.

Usage

```
getNumVertices(this)

## S3 method for class 'LatentForests'
getNumVertices(this)
```

Arguments

this	the LatentForests object.

getPhi *Get the phi parameter.*

Description

Gets the phi parameter controlling the strength of the sBIC penalty.

Usage

```
getPhi(this, phi)

## S3 method for class 'MixtureModels'
getPhi(this, phi)
```

Arguments

this	the MixtureModels object.
phi	the new phi value.

getPrior *The prior on the models.*

Description

Returns the unnormalized prior on the collection.

Usage

```
getPrior(this)

## S3 method for class 'BinomialMixtures'
getPrior(this)

## S3 method for class 'FactorAnalyses'
getPrior(this)

## S3 method for class 'GaussianMixtures'
getPrior(this)

## S3 method for class 'LCAs'
getPrior(this)

## S3 method for class 'LatentForests'
getPrior(this)

## S3 method for class 'ReducedRankRegressions'
getPrior(this)
```

Arguments

this	the model poset object.

getSamplingCovMat *Sampling covariance matrix.*

Description

Returns the sampling covariance matrix for the data set with setData().

Usage

```
getSamplingCovMat(this)

## S3 method for class 'LatentForests'
getSamplingCovMat(this)
```

Arguments

this	the LatentForests object.

---

getSupport *Get support for a given model.*

Description

Given a model number returns the support of the model. Let E by the matrix of edges returned by this$getAllEdges(), the support is represented by a 0-1 vector v where the ith entry of v is 1 if the ith edge in E is in the model and is 0 otherwise.

Usage

```
getSupport(this, model)

## S3 method for class 'LatentForests'
getSupport(this, model)
```

Arguments

this	the LatentForests object.
model	the model number.

---

getTopOrder *Topological ordering of models.*

Description

Returns a topological ordering of models in the collection.

Usage

```
getTopOrder(this)

## S3 method for class 'BinomialMixtures'
getTopOrder(this)

## S3 method for class 'FactorAnalyses'
getTopOrder(this)

## S3 method for class 'GaussianMixtures'
getTopOrder(this)

## S3 method for class 'LCAs'
```

```
getTopOrder(this)

## S3 method for class 'LatentForests'
getTopOrder(this)

## S3 method for class 'ReducedRankRegressions'
getTopOrder(this)
```

Arguments

this	the model poset object.

LatentForests *Construct a poset of gaussian latent forest models.*

Description

For a fixed binary forest where all leaves represent observed variables this function creates an object representing all gaussian latent forest models that are submodels of the given model. All models are enumerated from 1 to the total number of subforests, see the method `getSupport.LatentForests` for details on how to determine which model a particular model number corresponds to. Models are naturally ordered by inclusion so that, for example, the forest that has no edges is less than all other models.

Usage

```
LatentForests(numLeaves = 0, E = matrix(numeric(0), ncol = 2))
```

Arguments

numLeaves	the number of observed variables (these are the leaves of the model)
E	a 2xm matrix of edges corresponding to the edges of the 'super forest' f for which we compute all subforests. f should have nodes 1:numLeaves as leaves of the forest with no internal nodes as leaves.

Value

An object representing the collection.

LCAs *Construct a poset of latent class analysis models.*

Description

Creates an object representing a collection of latent class analysis models. There is one model for each fixed number of latent classes from 1 to some specified maximum. In particular each model is identified by a single number specifiying the number of latent classes in the model. Models are naturally ordered by inclusion so that, for example, a model with 2 latent classes comes before a model with 3 or more latent classes.

Usage

```
LCAs(maxNumClasses = 1, numVariables = 2,
     numStatesForVariables = 2, phi = "default")
```

Arguments

maxNumClasses	the number of classes in the largest LCA model to considered.
numVariables	the number of observed variables.
numStatesForVariables	the number of states for each observed variable, at the moment these must all be equal.
phi	parameter controlling the strength of the sBIC penalty.

Value

An object representing the collection.

learnCoef *Learning coefficient*

Description

Computes the learning coefficient for a model with respect to one of the model's submodels.

Usage

```
learnCoef(this, superModel, subModel)

## S3 method for class 'MixtureModels'
learnCoef(this, superModel, subModel)

## S3 method for class 'FactorAnalyses'
learnCoef(this, superModel, subModel)
```

```
## S3 method for class 'LatentForests'
learnCoef(this, superModel, subModel)

## S3 method for class 'ReducedRankRegressions'
learnCoef(this, superModel, subModel)
```

Arguments

this	the object representing the model poset.
superModel	the larger model of the two input models.
subModel	the submodel of the larger model.

---

logLike *Multivariate gaussian log-likelihood.*

Description

A private method that returns the log-likelihood of the data set with setData() under a multivariate gaussian model with a given covariance matrix and assumed 0 means.

Usage

```
logLike(this, covMat)

## S3 method for class 'LatentForests'
logLike(this, covMat)
```

Arguments

this	the LatentForests object.
covMat	a covariance matrix.

---

logLikeMle *Maximum likelihood for data.*

Description

Computes the maximum likelihood of a model in the model poset for the data set using the setData command.

Usage

```
logLikeMle(this, model, ...)

## S3 method for class 'BinomialMixtures'
logLikeMle(this, model, ...)

## S3 method for class 'FactorAnalyses'
logLikeMle(this, model, starts = 1, ...)

## S3 method for class 'GaussianMixtures'
logLikeMle(this, model, ...)

## S3 method for class 'LCAs'
logLikeMle(this, model, ...)

## S3 method for class 'LatentForests'
logLikeMle(this, model, ...)

## S3 method for class 'ReducedRankRegressions'
logLikeMle(this, model, ...)
```

Arguments

this	the object representing the model poset.
model	the model for which the maximum likelihood should be computed.
...	further parameters to be passed to methods
starts	The number of starting values to be tried

---

logLikeMleHelper *Help compute the MLE.*

Description

A private method that acts as a helper function for the logLikeMLE method in the ReducedRankRegressions class.

Usage

```
logLikeMleHelper(this, model)

## S3 method for class 'ReducedRankRegressions'
logLikeMleHelper(this, model)
```

Arguments

this	the ReducedRankRegressions object.
model	the model number.

MixtureModels *Linear collections of mixture models.*

Description

An abstract class representing a collection of mixture models that are linearly ordered by the number of components in the mixture. This class should not be instantiated, just extended.

Usage

```
MixtureModels(phi = "default")
```

Arguments

phi	parameter controlling the strength of the sBIC penalty.

Value

An object representing the collection.

See Also

GaussianMixtures, BinomialMixtures, LCAs

mle *Maximum likelihood estimator.*

Description

Computes the maximum likelihood estimator of the model parameters (for a given model in the collection) given the data set with setData.

Usage

```
mle(this, model)

## S3 method for class 'BinomialMixtures'
mle(this, model)

## S3 method for class 'FactorAnalyses'
mle(this, model)

## S3 method for class 'GaussianMixtures'
mle(this, model)

## S3 method for class 'LCAs'
```

```
mle(this, model)

## S3 method for class 'LatentForests'
mle(this, model)
```

Arguments

this	the object representing the model poset.
model	the model for which the maximum likelihood should be computed.

`parents` *Parents of a model.*

Description

Returns the immediate parents of a given model, i.e. those models M that are (in the poset ordering) less than the given model but for which there exists no other model M' such that M < M' < (given model).

Usage

```
parents(this, model)

## S3 method for class 'MixtureModels'
parents(this, model)

## S3 method for class 'FactorAnalyses'
parents(this, model)

## S3 method for class 'LatentForests'
parents(this, model)

## S3 method for class 'ReducedRankRegressions'
parents(this, model)
```

Arguments

this	the object representing the model poset.
model	the model for which the parents should be found.

ReducedRankRegressions *Construct a poset of reduced rank regression models.*

Description

Creates an object representing a collection of reduced rank regression models. There is one model for each fixed rank from 1 to some specified maximum. In particular each model is identified by a single number specifiying the rank of the regression matrix in the model. Models are naturally ordered by inclusion so that, for example, a model with 2 rank 2 comes before before a model with rank 3 or greater.

Usage

```
ReducedRankRegressions(numResponses, numCovariates, maxRank)
```

Arguments

numResponses	the number of response variables.
numCovariates	the number of covariates.
maxRank	the largest rank model to be included in the collection.

Value

An object representing the collection.

sBIC *Compute the sBIC.*

Description

Computes the sBIC for a given collection of models.

Usage

```
sBIC(X, mod)
```

Arguments

X	the data for which the maximum likelihood estimates will be computed for the given collection of models. To see how this data should be formatted check the documentation for setData.YourModelName (e.g. setData.LCAs). If X is NULL then it is assumed that data for the model has already been set, this can be useful if you want to compute the sBIC with the same model and data several times (perhaps after changing some parameter of the model) without having to reset the data and thus recompute maximum log-likelihoods.

`mod` an object representing a poset of models of the same type, e.g. a collection of binomial mixture models. The currently implemented models include:

- Binomial mixtures, see `BinomialMixtures`.
- Factor analysis, see `FactorAnalyses`.
- Latent class analysis, see `LCAs`.
- Latent gaussian forests, see `LatentForests`.
- Reduced rank regression, see `ReducedRankRegressions`.
- 1-dimensional gaussian mixtures, see `GaussianMixtures`.

Value

A named list containing the components

- logLike - the computed MLE log-likelihoods for each model.
- sBIC - the sBIC score for each model.
- BIC - the usual BIC score for each model.
- modelPoset - the input model poset mod.

`setData` *Set data for a model poset.*

Description

Sets the data to be used by a poset of models when computing MLEs.

Usage

```
setData(this, data)
```

Arguments

`this`	the model poset object.
`data`	the data to be set.

`setData.BinomialMixtures` *Set data for the binomial mixture models.*

Description

Sets the data to be used by the binomial mixture models when computing MLEs.

Usage

```
## S3 method for class 'BinomialMixtures'
setData(this, data)
```

Arguments

`this`	the BinomialMixtures object.
`data`	the data to be set, should be a numeric vector of non-negative integers.

`setData.FactorAnalyses` *Set data for the factor analysis models.*

Description

Sets the data to be used by the factor analysis models when computing MLEs.

Usage

```
## S3 method for class 'FactorAnalyses'
setData(this, data)
```

Arguments

`this`	the FactorAnalyses object.
`data`	the data to be set, should matrix of observed responses.

setData.GaussianMixtures *Set data for the gaussian mixture models.*

Description

Sets the data to be used by the gaussian mixture models when computing MLEs.

Usage

```
## S3 method for class 'GaussianMixtures'
setData(this, data)
```

Arguments

this	the GaussianMixtures object.
data	the data to be set, a matrix where each row corresponds to a single multivariate observation. If the corresponding GaussianMixtures object has ambient dimension 1, then data may be a numeric vector of observations.

setData.LatentForests *Set data for the latent forest models.*

Description

Sets the data to be used by the latent forest models models for computing MLEs.

Usage

```
## S3 method for class 'LatentForests'
setData(this, data)
```

Arguments

this	the LatentForests object.
data	the data to be set, should matrix of observed values where each row corresponds to a single sample.

setData.LCAs *Set data for the LCA models.*

Description

Sets the data to be used by the LCA models when computing MLEs.

Usage

```
## S3 method for class 'LCAs'
setData(this, data)
```

Arguments

this	the LCAs object.
data	the data to be set, should be an integer valued matrix where each row represents a single sample from the observed variables.

setData.ReducedRankRegressions *Set data for the reduced rank regression models.*

Description

Sets the data to be used by the reduced rank regression models when computing MLEs.

Usage

```
## S3 method for class 'ReducedRankRegressions'
setData(this, data)
```

Arguments

this	the ReducedRankRegressions object.
data	the data to be set, should be a named list with two components: • XA matrix containing the values of covariates for each sample. Here each COLUMN represents a single sample from all of the covariates. • YA matrix containing the values of the response variables for each sample. Again, each COLUMN is a single sample.

`setPhi` *Set phi parameter.*

Description

Set the phi parameter in a mixture model object to a different value.

Usage

```
setPhi(this, phi)

## S3 method for class 'MixtureModels'
setPhi(this, phi)
```

Arguments

`this`	the MixtureModels object.
`phi`	the new phi value.

Index