Package 'greed'

May 10, 2021

Type Package

Title Clustering and Model Selection with the Integrated Classification Likelihood

Version 0.5.1

Date 2021-05-03

URL https://comeetie.github.io/greed/, https://github.com/comeetie/greed

BugReports https://github.com/comeetie/greed/issues

Maintainer Etienne Côme <etienne.come@univ-eiffel.fr>

Description An ensemble of algorithms that enable the clustering of networks and data matrix such as counts matrix with different type of generative models. Model selection and clustering is performed in combination by optimizing the Integrated Classification Likelihood (which is equivalent to minimizing the description length). Several models are available such as: Stochastic Block Model, degree corrected Stochastic Block Model, Mixtures of Multinomial, Latent Block Model. The optimization is performed thanks to a combination of greedy local search and a genetic algorithm (see <arXiv:2002:11577> for more details).

License GPL

Depends R (>= 2.10)

Imports Rcpp (>= 1.0.0), Matrix, future, listenv, ggplot2, graphics, methods, stats,RSpectra,ggpubr,GGally,cba

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, MASS, mclust, knitr, rmarkdown, igraph, dplyr, tibble, tidyr, spelling

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.1.1

Collate 'RcppExports.R' 'misc.R' 'models_classes.R' 'fit_classes.R' 'cleanpath.R' 'genetic_alg.R' 'hybrid_alg.R' 'alg_classes.R' 'dcsbm.R' 'coclust_dcsbm.R' 'data.R' 'diaggmm.R' 'generator.R' 'mvmreg.R' 'gmm.R' 'greed.R' 'sbm.R' 'misssbm.R' 'mm.R' 'multistart_alg.R' 'multsbm.R' 'plot.R'

Language en-US

NeedsCompilation yes

Author Etienne Côme [aut, cre],
Nicolas Jouvin [aut]

Repository CRAN

Date/Publication 2021-05-10 06:50:03 UTC

R topics documented:

`alg-class` *Abstract optimization algorithm class*

Description

An S4 class to represent an abstract optimization algorithm.

Slots

`name` algorithm name

`Blogs` *Political blogs network dataset*

Description

A directed network of hyperlinks between weblogs on US politics, recorded in 2005 by Adamic and Glance. Only the biggest connected component of the original graph is provided.

Usage

```
data(Blogs)
```

Format

An object of class `list` with two fields;

X network adjacency matrix as a `sparseMatrix` of size 1222x1222

label vector of political leaning of each blogs (size 1222) with the following encoding (0 = left or liberal,1 = right or conservative)

Source

M. E. J. Newman Network datasets

References

Lada A. Adamic and Natalie Glance, "The political blogosphere and the 2004 US Election", in Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem (2005) (ACM).

Examples

```
data(Blogs)
```

Books *Books about US politics network dataset*

Description

A network of books about US politics published around the time of the 2004 presidential election and sold by the online bookseller Amazon.com. Edges between books represent frequent co-purchasing of books by the same buyers. The network was compiled by V. Krebs and is unpublished, but can found on Krebs' web site. Thanks to Valdis Krebs for permission to post these data on this web site.

Usage

```
data(Books)
```

Format

An object of class `list` with two fields;

X network adjacency matrix as a `sparseMatrix` of size 105x105

label a factor of length (size 105) with levels "l", "n", or "c" to indicate whether the books are liberal, neutral, or conservative

Source

M. E. J. Newman Network datasets

Examples

```
data(Books)
```

coef,co_dcsbm_fit-method *Extract parameters from an* `co_dcsbm_fit-class` *object*

Description

Extract parameters from an `co_dcsbm_fit-class` object

Usage

```
## S4 method for signature 'co_dcsbm_fit'
coef(object)
```

Arguments

`object`	a `co_dcsbm_fit-class`

Value

a list with the model parameters estimates (MAP), the fields are:

- 'pirows': row cluster proportions
- 'picols': row cluster proportions
- 'thetakl': between clusters connection probabilities (matrix of size Krow x Kcol),
- 'gammarows': rows degree correction parameters (size Nrows),
- 'gammacols': cols degree correction parameters (size Ncols),

---

coef,dcsbm_fit-method *Extract parameters from an* dcsbm_fit-class *object*

Description

Extract parameters from an dcsbm_fit-class object

Usage

```
## S4 method for signature 'dcsbm_fit'
coef(object)
```

Arguments

object	a dcsbm_fit-class

Details

in case of undirected graph

Value

a list with the model parameters estimates (MAP), the fields are the following for "directed" models :

- 'pi': cluster proportions
- 'thetakl': between cluster normalized connection intensities (matrix of size K x K),
- gammain: node in-degree correction parameter
- gammaout: node out-degree correction parameter

And as follow for un-directed models : #'

- 'pi': cluster proportions
- 'thetakl': between cluster normalized connection intensities (matrix of size K x K),
- gamma: node degree correction parameter

---

coef,diaggmm_fit-method *Extract mixture parameters from* diaggmm_fit-class *object*

Description

Extract mixture parameters from diaggmm_fit-class object

Usage

```
## S4 method for signature 'diaggmm_fit'
coef(object)
```

Arguments

object	a diaggmm_fit-class

Value

a list with the mixture parameters estimates (MAP), the fields are:

- 'pi': cluster proportions
- 'muk': cluster means
- 'Sigmak': cluster co-variance matrices

---

coef,gmm_fit-method *Extract mixture parameters from* gmm_fit-class *object*

Description

Extract mixture parameters from gmm_fit-class object

Usage

```
## S4 method for signature 'gmm_fit'
coef(object)
```

Arguments

object	a gmm_fit-class

Value

a list with the mixture parameters estimates (MAP), the fields are:

- 'pi': cluster proportions
- 'muk': cluster means
- 'Sigmak': cluster co-variance matrices

---

coef,misssbm_fit-method *Extract parameters from an* misssbm_fit-class *object*

Description

Extract parameters from an misssbm_fit-class object

Usage

```
## S4 method for signature 'misssbm_fit'
coef(object)
```

Arguments

object	a misssbm_fit-class

Value

a list with the model parameters estimates (MAP), the fields are:

- 'pi': cluster proportions
- 'thetakl': between clusters connection probabilites (array of size K x K),
- 'epsilonkl': between clusters dyad observation probabilites (array of size K x K) for block-dyad sampling and double for dyad sampling,

---

coef,mm_fit-method *Extract parameters from an* mm_fit-class *object*

Description

Extract parameters from an mm_fit-class object

Usage

```
## S4 method for signature 'mm_fit'
coef(object)
```

Arguments

object	a mm_fit-class

Value

a list with the model parameters estimates (MAP), the fields are:

- 'pi': cluster proportions
- 'thetak': cluster profile probabilites (matrix of size K x D),

---

coef,multsbm_fit-method

Extract parameters from an multsbm_fit-class *object*

Description

Extract parameters from an multsbm_fit-class object

Usage

```
## S4 method for signature 'multsbm_fit'
coef(object)
```

Arguments

object	a multsbm_fit-class

Value

a list with the model parameters estimates (MAP), the fields are:

- 'pi': cluster proportions
- 'thetakl': cluster profile probabilities (array of size K x K x D),

---

coef,mvmreg_fit-method

Extract mixture parameters from mvmreg_fit-class *object*

Description

Extract mixture parameters from mvmreg_fit-class object

Usage

```
## S4 method for signature 'mvmreg_fit'
coef(object)
```

Arguments

object	a mvmreg_fit-class

Value

a list with the mixture parameters estimates (MAP), the fields are:

- 'pi': cluster proportions
- 'A': cluster regression matrix
- 'Sigmak': cluster noise co-variance matrices

coef,sbm_fit-method *Extract parameters from an sbm_fit-class object*

Description

Extract parameters from an sbm_fit-class object

Usage

```
## S4 method for signature 'sbm_fit'
coef(object)
```

Arguments

object	a sbm_fit-class

Value

a list with the model parameters estimates (MAP), the fields are:

- 'pi': cluster proportions
- 'thetakl': between clusters connections probabilities (matrix of size K x K)

co_dcsbm-class *Degree Corrected Stochastic Block Model for bipartite graph class*

Description

An S4 class to represent a degree corrected stochastic block model for co_clustering of bipartite graph, extends icl_model-class class. Such model can be used to cluster graph vertex, and model a bipartite graph adjacency matrix X with the following generative model :

$$\pi \sim Dirichlet(\alpha)$$
$$Z_i^r \sim \mathcal{M}(1, \pi^r)$$
$$Z_j^c \sim \mathcal{M}(1, \pi^c)$$
$$\theta_{kl} \sim Exponential(p)$$
$$\gamma_i^r \sim \mathcal{U}(S_k)$$
$$\gamma_i^c \sim \mathcal{U}(S_l)$$
$$X_{ij}|Z_{ik}^c Z_{jl}^r = 1 \sim \mathcal{P}(\gamma_i^r \theta_{kl} \gamma_j^c)$$

The individuals parameters γ_i^r, γ_j^c allow to take into account the node degree heterogeneity. These parameters have uniform priors over simplex S_k. This class mainly store the prior parameters value α of this generative model in the following slots (the prior parameter p is estimated from the data as the global average probability of connection between two nodes):

Slots

`alpha` Dirichlet parameters for the prior over clusters proportions (default to 1)

`p` Exponential prior parameter (default to Nan, in this case `p` will be estimated from data as the average intensities of X)

Examples

```
new("co_dcsbm")
new("co_dcsbm", p = 0.1)
```

`co_dcsbm_fit-class` *Degree corrected stochastic block model for bipartite graph fit results class*

Description

An S4 class to represent a fit of a degree corrected stochastic block model for co_clustering, extend `icl_fit-class`.

Slots

`model` a `co_dcsbm-class` object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`Krow` number of extracted row clusters

`Kcol` number of extracted column clusters

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- din: numeric vector of size K which store the sums of in-degrees for each clusters
- dout: numeric vector of size K which store the sums of out-degrees for each clusters
- x_counts: matrix of size K*K with the number of links between each pair of clusters
- co_x_counts: matrix of size Krow*Kcol with the number of links between each pair of row and column cluster

`clrow` a numeric vector with row cluster indexes

`clcol` a numeric vector with column cluster indexes

`Nrow` number of rows

`Ncol` number of columns

`move_mat` binary matrix which store move constraints

`train_hist` data.frame with training history information (details depends on the training procedure)

co_dcsbm_path-class *Degree corrected stochastic block model for bipartite graph hierarchical fit results class*

Description

An S4 class to represent a fit of a degree corrected stochastic block model for co_clustering, extend icl_path-class.

Slots

`model` a co_dcsbm-class object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`Krow` number of extracted row clusters

`Kcol` number of extracted column clusters

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- din: numeric vector of size K which store the sums of in-degrees for each clusters
- dout: numeric vector of size K which store the sums of out-degrees for each clusters
- x_counts: matrix of size K*K with the number of links between each pair of clusters
- co_x_counts: matrix of size Krow*Kcol with the number of links between each pair of row and column cluster

`clrow` a numeric vector with row cluster indexes

`clcol` a numeric vector with column cluster indexes

`Nrow` number of rows

`Ncol` number of columns

`path` a list of size K-1 with each part of the path described by:

- icl1: icl value reach with this solution for alpha=1
- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats: a list with the elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - din: numeric vector of size K which store the sums of in-degrees for each clusters
  - dout: numeric vector of size K which store the sums of out-degrees for each clusters

- x_counts: matrix of size K*K with the number of links between each pair of clusters
- co_x_counts: matrix of size Krow*Kcol with the number of links between each pair of row and column cluster

`logalpha` value of log(alpha)

`ggtree` data.frame with complete merge tree for easy plotting with `ggplot2`

`tree` numeric vector with merge tree `tree[i]` contains the index of `i` father

`ggtreerow` data.frame with complete merge tree of row clusters for easy plotting with `ggplot2`

`ggtreecol` data.frame with complete merge tree of column clusters for easy plotting with `ggplot2`

`train_hist` data.frame with training history information (details depends on the training procedure)

`cut,co_dcsbm_path-method`

method to cut a path solution to a desired number of cluster

Description

this method take a `co_dcsbm_path-class` and an integer K and return the solution from the path with K clusters

Usage

```
## S4 method for signature 'co_dcsbm_path'
cut(x, K)
```

Arguments

`x`	A an `co_dcsbm_path-class` solution
`K`	Desired number of cluster

Value

an icl_path object with the desired number of cluster

cut,icl_path-method *Method to cut a path solution to a desired number of cluster*

Description

This method take a icl_path-class object and an integer K and return the solution from the path with K clusters

Usage

```
## S4 method for signature 'icl_path'
cut(x, K)
```

Arguments

x	A an icl_path solution
K	Desired number of cluster

Value

an icl_path-class object with the desired number of cluster

dcsbm-class *Degree Corrected Stochastic Block Model class*

Description

An S4 class to represent a degree corrected stochastic block model, extend icl_model-class. Such model can be used to cluster graph vertex, and model a square adjacency matrix X with the following generative model :

$$\pi \sim Dirichlet(\alpha)$$

$$Z_i \sim \mathcal{M}(1, \pi)$$

$$\theta_{kl} \sim Exponential(p)$$

$$\gamma_i^+, \gamma_i^- \sim \mathcal{U}(S_k)$$

$$X_{ij} | Z_{ik} Z_{jl} = 1 \sim \mathcal{P}(\gamma_i^+ \theta_{kl} \gamma_j^-)$$

The individuals parameters γ_i^+, γ_i^- allow to take into account the node degree heterogeneity. These parameters have uniform priors over the simplex S_k ie. $\sum_{i:z_{ik}=1} \gamma_i^+ = 1$. This class mainly store the prior parameters value α of this generative model in the following slots (the prior parameter p is estimated from the data as the global average probability of connection between two nodes):

Slots

`name` name of the model

`alpha` Dirichlet over cluster proportions prior parameter (default to 10)

`p` Exponential prior parameter (default to NaN, in this case p will be estimated from data as the mean connection probability)

`type` define the type of networks (either "directed" or "undirected", default to "directed")

---

dcsbm_fit-class *Degree Corrected Stochastic Block Model fit results class*

Description

An S4 class to represent a fit of a degree corrected stochastic block model for co_clustering, extend `icl_fit-class`.

Slots

`model` a `dcsbm-class` object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- din: numeric vector of size K which store the sums of in-degrees for each clusters
- dout: numeric vector of size K which store the sums of out-degrees for each clusters
- x_counts: matrix of size K*K with the number of links between each pair of clusters

`obs_stats_cst` a list with the following elements:

- din_node: node in-degree, a vector of size N
- dout_node: node in-degree vector of size N

`move_mat` binary matrix which store move constraints

`train_hist` data.frame with training history information (details depends on the training procedure)

`dcsbm_path-class` *Degree Corrected Stochastic Block Model hierarchical fit results class*

Description

An S4 class to represent a hierarchical fit of a degree corrected stochastic block model, extend `icl_path-class`.

Slots

`model` a `dcsbm-class` object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- din: numeric vector of size K which store the sums of in-degrees for each clusters
- dout: numeric vector of size K which store the sums of out-degrees for each clusters
- x_counts: matrix of size K*K with the number of links between each pair of clusters

`path` a list of size K-1 with each part of the path described by:

- icl1: icl value reach with this solution for alpha=1
- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats: a list with the elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - din: numeric vector of size K which store the sums of in-degrees for each clusters
  - dout: numeric vector of size K which store the sums of out-degrees for each clusters
  - x_counts: matrix of size K*K with the number of links between each pair of clusters

`logalpha` value of log(alpha)

`ggtree` data.frame with complete merge tree for easy plotting with `ggplot2`

`tree` numeric vector with merge tree `tree[i]` contains the index of `i` father

`train_hist` data.frame with training history information (details depends on the training procedure)

diaggmm-class *Diagonal Gaussian mixture model description class*

Description

An S4 class to represent a multivariate diagonal Gaussian mixture model, extend icl_model-class. The model corresponds to the following generative model:

$$\pi \sim Dirichlet(\alpha)$$

$$Z_i \sim \mathcal{M}(1, \pi)$$

$$\lambda_k^{(d)} \sim \mathcal{G}(\kappa, \beta)$$

$$\mu_k^{(d)} \sim \mathcal{N}(\mu, (\tau\lambda_k)^{-1})$$

$$X_{i.}|Z_{ik} = 1 \sim \mathcal{N}(\mu_k, \lambda_k^{-1})$$

with $\mathcal{G}(\kappa, \beta)$ the Gamma distribution with shape parameter κ and rate parameter β.

Slots

`name` name of the model

`alpha` Dirichlet over cluster proportions prior parameter (default to 1)

`tau` Prior parameter (inverse variance), (default 0.01)

`kappa` Prior parameter (gamma shape), (default to 1)

`beta` Prior parameter (gamma rate), (default to NaN, in this case beta will be estimated from data as 0.1 time the mean of X columns variances)

`mu` Prior for the means (vector of size D), (default to NaN, in this case mu will be estimated from data as the mean of X)

References

Bertoletti, Marco & Friel, Nial & Rastelli, Riccardo. (2014). Choosing the number of clusters in a finite mixture model using an exact Integrated Completed Likelihood criterion. METRON. 73. 10.1007/s40300-015-0064-5. #'

Examples

```
new("diaggmm")
new("diaggmm",alpha=1,tau=0.1,beta=0.1)
```

diaggmm_fit-class *Diagonal Gaussian mixture model fit results class*

Description

An S4 class to represent a fit of a multivariate diagonal Gaussian mixture model, extend icl_fit-class.

Slots

model a diaggmm-class object to store the model fitted

name generative model name

icl icl value of the fitted model

K number of extracted clusters over row and columns

cl a numeric vector with row and columns cluster indexes

obs_stats a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- regs: list of size K with statistics for each clusters

move_mat binary matrix which store move constraints

train_hist data.frame with training history information (details depends on the training procedure)

diaggmm_path-class *Diagonal Gaussian mixture model hierarchical fit results class*

Description

An S4 class to represent a hierarchical fit of a diagonal gaussian mixture model, extend icl_path-class.

Slots

model a diaggmm-class object to store the model fitted

name generative model name

icl icl value of the fitted model

K number of extracted clusters over row and columns

cl a numeric vector with row and clolumns cluster indexes

obs_stats a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- regs: list of size K with statistics for each clusters

path a list of size K-1 with each part of the path described by:

- icl1: icl value reach with this solution for alpha=1

- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats: a list with the following elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - regs: list of size K with statistics for each clusters

`logalpha` value of log(alpha)

`ggtree` data.frame with complete merge tree for easy plotting with `ggplot2`

`tree` numeric vector with merge tree `tree[i]` contains the index of `i` father

`train_hist` data.frame with training history information (details depends on the training procedure)

`fashion` *Fashion mnist dataset*

Description

Zalando fashionmnist dataset, sample of 1 000 Zalando's article images from the test set.

Usage

```
data(fashion)
```

Format

An object of class `list` with two fields;

X pixels intensities values data.frame of size 1 000x784

labels vector of labels with the following encoding (0 = T-shirt/top, 1 = Trouser, 2 = Pullover, 3 = Dress, 4 = Coat, 5 = Sandal, 6 = Shirt, 7 = Sneaker, 8 = Bag 9 = Ankle boot)

Source

https://github.com/zalandoresearch/fashion-mnist

References

Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Han Xiao, Kashif Rasul, Roland Vollgraf (2017) (arXiv:1708.07747).

Examples

```
data(fashion)
```

Football *American College football network dataset*

Description

Network of American football games between Division IA colleges during regular season Fall 2000.

Usage

```
data(Football)
```

Format

An object of class `list` with two fields;

X network adjacency matrix as a `sparseMatrix` of size 115x115

label vector of teams conferences of size 115 with the following encoding (0 = Atlantic Coast, 1 = Big East, 2 = Big Ten, 3 = Big Twelve, 4 = Conference USA, 5 = Independents, 6 = Mid-American, 7 = Mountain West, 8 = Pacific Ten, 9 = Southeastern, 10 = Sun Belt, 11 = Western Athletic)

Source

M. E. J. Newman Network datasets

References

M. Girvan and M. E. J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002) (PNAS).

Examples

```
data(Football)
```

FrenchParliament *French Parliament votes dataset*

Description

French Parliament votes dataset

Usage

```
data(FrenchParliament)
```

Format

An object of class `list` with two fields;

X matrix of deputy votes a `sparseMatrix` of size 593x570

labels a data frame with deputy meta-data

Examples

```
data(FrenchParliament)
```

---

genetic-class *Genetic optimization algorithm*

Description

An S4 class to represent a genetic algorithm (extends `alg-class` class).

Slots

`pop_size` size of the solutions populations (default to 10)

`nb_max_gen` maximal number of generation to produce (default to 4)

`prob_mutation` probability of mutation (default to 0.25)

`sel_frac` fraction of best solutions selected for crossing (default to 0.75)

---

gmm-class *Gaussian mixture model description class*

Description

An S4 class to represent a multivariate Gaussian mixture model, extend `icl_model-class`. The model corresponds to the following generative model:

$$\pi \sim Dirichlet(\alpha)$$

$$Z_i \sim \mathcal{M}(1, \pi)$$

$$V_k \sim \mathcal{W}(\varepsilon^{-1}, n_0)$$

$$\mu_k \sim \mathcal{N}(\mu, (\tau V_k)^{-1})$$

$$X_i | Z_{ik} = 1 \sim \mathcal{N}(\mu_k, V_k^{-1})$$

with $\mathcal{W}(\varepsilon^{-1}, n_0)$ the Whishart distribution.

Slots

`name` name of the model

`alpha` Dirichlet over cluster proportions prior parameter (default to 1)

`tau` Prior parameter (inverse variance) default 0.01

`N0` Prior parameter (pseudo count) should be > number of features (default to NaN, in this case it will be estimated from data as the number of columns of X)

`epsilon` Prior parameter co-variance matrix prior (matrix of size D x D), (default to a matrix of NaN, in this case epsilon will be estimated from data and will corresponds to 0.1 times a diagonal matrix with the variances of the X columns)

`mu` Prior parameters for the means (vector of size D), (default to NaN, in this case mu will be estimated from the data and will be equal to the mean of X)

References

Bertoletti, Marco & Friel, Nial & Rastelli, Riccardo. (2014). Choosing the number of clusters in a finite mixture model using an exact Integrated Completed Likelihood criterion. METRON. 73. 10.1007/s40300-015-0064-5. #'

Examples

```
new("gmm")
new("gmm",alpha=1,tau=0.1,N0=15)
```

---

gmmpairs *Make a matrix of plots with a given data and gmm fitted parameters*

Description

Make a matrix of plots with a given data and gmm fitted parameters with ellipses.

Usage

```
gmmpairs(sol, X)
```

Arguments

`sol`	a `gmm_fit-class` or `diaggmm_fit-class`
`X`	the data used for the fit a data.frame or matrix.

Value

a `ggplot2` graphic

gmm_fit-class *Gaussian mixture model fit results class*

Description

An S4 class to represent a fit of a multivariate mixture of regression model, extend icl_fit-class.

Slots

`model` a gmm-class object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- regs: list of size K with statistics for each clusters

`move_mat` binary matrix which store move constraints

`train_hist` data.frame with training history information (details depends on the training procedure)

gmm_path-class *Gaussian mixture model hierarchical fit results class*

Description

An S4 class to represent a hierarchical fit of a gaussian mixture model, extend icl_path-class.

Slots

`model` a gmm-class object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`cl` a numeric vector with row and clolumns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- regs: list of size K with statistics for each clusters

`path` a list of size K-1 with each part of the path described by:

- icl1: icl value reach with this solution for alpha=1

- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats: a list with the following elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - regs: list of size K with statistics for each clusters

`logalpha` value of log(alpha)

`ggtree` data.frame with complete merge tree for easy plotting with `ggplot2`

`tree` numeric vector with merge tree `tree[i]` contains the index of `i` father

`train_hist` data.frame with training history information (details depends on the training procedure)

`graph_balance` *graph_balance*

Description

graph_balance

Usage

```
graph_balance(x)
```

Arguments

x	a `sbm_fit-class` object to be plot

Value

a ggplot2 graph

greed *Model based hierarchical clustering*

Description

Greed enables the clustering of networks and count data such as document-term matrix with different models. Model selection and clustering are performed in combination by optimizing the Integrated Classification Likelihood. Optimization is performed thanks to a combination of greedy local search and a genetic algorithm. The main entry point is the greed function to perform the clustering, which is documented below. The package also provides sampling functions for all the implemented DLVMs: Mixture of Multinomials (rmm), Stochastic Block Model (rsbm, rdcsbm) and Latent Block Model (rlbm).

Usage

```
greed(
  X,
  K = 20,
  model = find_model(X),
  alg = methods::new("hybrid"),
  verbose = FALSE
)
```

Arguments

X	data to cluster either a matrix,an array or a dgCMatrix-class
K	initial number of cluster
model	a generative model to fit sbm-class, dcsbm-class, co_dcsbm-class, mm-class,gmm-class, diaggmm-class or mvmreg-class
alg	an optimization algorithm of class hybrid-class (default), multistarts-class, seed-class or genetic-class
verbose	Boolean for verbose mode

Value

an icl_path-class object

greed_cond *Conditional model based hierarchical clustering*

Description

Conditional model based hierarchical clustering

Usage

```
greed_cond(
  X,
  Y,
  K = 20,
  model = find_model_cond(X, Y),
  alg = methods::new("hybrid"),
  verbose = FALSE
)
```

Arguments

X	design matrix
Y	target variables
K	Desired number of cluster
model	a conditional generative model mvmreg-class
alg	an optimization algorithm of class hybrid-class (default), multistarts-class, seed-class or genetic-class
verbose	Boolean for verbose mode

Value

an icl_path-class object

H *Compute the entropy of a discrete sample*

Description

Compute the entropy of a discrete sample

Usage

```
H(cl)
```

Arguments

cl	vector of discrete labels

Value

the entropy of the sample

Examples

```
cl =sample(2,500,replace=TRUE)
H(cl)
```

---

hybrid-class *Hybrid optimization algorithm*

Description

An S4 class to represent an hybrid genetic/greedy algorithm (extends alg-class class).

Slots

pop_size size of the solutions populations (default to 20)

nb_max_gen maximal number of generation to produce (default to 10)

prob_mutation mutation probability (default to 0.25)

Kmax maximum number of clusters (default to 100)

---

icl_fit-class *abstract class to represent a clustering result*

Description

An S4 abstract class to represent an icl fit of a clustering model.

Slots

K a numeric vector of length 1 which correspond to the number of clusters

icl a numeric vector of length 1 which store the the icl value

cl a numeric vector of length N which store the clusters labels

obs_stats a list to store the observed statistics of the model needed to compute ICL.

obs_stats_cst a list to store the observed statistics of the model that do not depend on the clustering.

move_mat binary matrix which store move constraints

train_hist a data.frame to store training history (format depends on the used algorithm used).

name generative model name

See Also

sbm_fit-class, dcsbm_fit-class, co_dcsbm_fit-class

`icl_model-class` *abstract class to represent a generative model An S4 class to represent an abstract generative model*

Description

abstract class to represent a generative model

An S4 class to represent an abstract generative model

Slots

`name` a character vector

`alpha` a numeric vector of length 1 which define the parameters of the Dirichlet over the cluster proportions (default to 1)

See Also

sbm-class, dcsbm-class,co_dcsbm-class,

`icl_path-class` *abstract class to represent a hierarchical clustering result*

Description

An S4 class to represent a hierarchical path of solution.

Slots

`path` a list of merge moves describing the hierarchy of merge followed to complete totally the merge path.

`tree` a tree representation of the merges.

`ggtree` a data.frame for easy plotting of the dendrogram

`logalpha` a numeric value which corresponds to the starting value of log(alpha).

See Also

sbm_path-class, dcsbm_path-class, co_dcsbm_path-class

Jazz *Jazz musicians network dataset*

Description

List of edges of the network of Jazz musicians.

Usage

```
data(Jazz)
```

Format

An object of class `sparseMatrix` with the network adjacency matrix.

Source

A. Arena Network datasets

References

P.Gleiser and L. Danon , Community Structure in jazz, Adv. Complex Syst.6, 565 (2003) (Arxiv)

Examples

```
data(Jazz)
```

Jazz_full *Jazz musicians / Bands relations*

Description

List Jazz musicians / Bands relationship

Usage

```
data(Jazz_full)
```

Format

A list with two fields:

X a `sparseMatrix` binary matrix of size 4475x965 with a one when one musicians played once in a given band.

col_meta a data.frame with columns (bands) meta_data

Source

A. Arena Network datasets

References

P.Gleiser and L. Danon , Community Structure in jazz, Adv. Complex Syst.6, 565 (2003) (Arxiv)

Examples

```
data(Jazz_full)
```

MI *Compute the mutual information of two discrete samples*

Description

Compute the mutual information of two discrete samples

Usage

```
MI(cl1, cl2)
```

Arguments

cl1	vector of discrete labels
cl2	vector of discrete labels

Value

the mutual information between the two discrete samples

Examples

```
cl1 =sample(2,500,replace=TRUE)
cl2 =sample(2,500,replace=TRUE)
MI(cl1,cl2)
```

misssbm-class *Stochastic Block Model with sampling scheme class*

Description

An S4 class to represent a Stochastic Block Model with a sampling scheme for missing data, extend icl_model-class. Such model can be used to cluster graph vertex, and model a square adjacency matrix X with the following generative model :

$$\pi \sim Dirichlet(\alpha)$$

$$Z_i \sim \mathcal{M}(1, \pi)$$

$$\theta_{kl} \sim Beta(a_0, b_0)$$

$$X_{ij}|Z_{ik}Z_{jl} = 1 \sim \mathcal{B}(\theta_{kl})$$

Missing value are supposed to be generated afterwards, the observation process correspond to a binary matrix R of the same size as X, with $R_{ij} = 1$ for observed entries and $R_{ij} = 0$ for missing ones. R may be supposed to be MAR:

$$\epsilon \sim Beta(a_{0obs}, b_{0obs})$$

$$R_{ij} \sim \mathcal{B}(\epsilon)$$

this correspond to the "dyad" sampling scheme. But the sampling scheme can also be NMAR:

$$\epsilon_{kl} \sim Beta(a_{0obs}, b_{0obs})$$

$$R_{ij}|Z_{ik}Z_{jl} = 1 \sim \mathcal{B}(\epsilon_{kl})$$

this correspond to the "block-dyad" sampling scheme. This class mainly store the prior parameters value $\alpha, a_0, b_0, a_{0obs}, b_{0obs}$ of this generative model in the following slots:

Slots

`name` name of the model

`alpha` Dirichlet over cluster proportions prior parameter (default to 1)

`a0` Beta prior parameter over links (default to 1)

`b0` Beta prior parameter over no-links (default to 1)

`type` define the type of networks (either "directed" or "undirected", default to "directed")

`sampling` define the sampling process (either "dyad" or "block-dyad")

`sampling_priors` define the sampling process priors parameters (list with `a0obs` and `b0obs` fields.)

References

Nowicki, Krzysztof and Tom A B Snijders (2001). "Estimation and prediction for stochastic block structures". In:Journal of the American statistical association 96.455, pp. 1077–1087

See Also

misssbm_fit-class,misssbm_path-class

greed

Examples

```
new("misssbm")
new("misssbm",a0=0.5, b0= 0.5,alpha=0.5,sampling="dyad",sampling_priors=list(a0obs = 2,b0obs = 1))
sbm = rsbm(100,c(0.5,0.5),diag(2)*0.1+0.01)
sbm$x[cbind(base::sample(1:100,50),base::sample(1:100,50))]=NA
sol = greed(sbm$x,model=new("misssbm",sampling="dyad"))
```

---

misssbm_fit-class *Stochastic Block Model with sampling scheme fit results class*

Description

An S4 class to represent a fit of a Stochastic Block Model, extend icl_fit-class.

Slots

`model` a misssbm-class object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over rows and columns

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- x_counts: matrix of size K*K with the number of observed links between each pair of clusters
- x_counts_obs: matrix of size K*K with the number of observed dyads between each pair of clusters

`move_mat` binary matrix which store move constraints

`train_hist` data.frame with training history information (details depends on the training procedure)

misssbm_path-class *Stochastic Block Model with sampling scheme hierarchical fit results class*

Description

An S4 class to represent a hierarchical fit of a stochastic block model, extend icl_path-class.

Slots

model a misssbm-class object to store the model fitted

name generative model name

icl icl value of the fitted model

K number of extracted clusters over row and columns

cl a numeric vector with row and columns cluster indexes

obs_stats a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- x_counts: matrix of size K*K with the number of observed links between each pair of clusters
- x_counts_obs: matrix of size K*K with the number of observed dyads between each pair of clusters

path a list of size K-1 with that store all the solutions along the path. Each element is a list with the following fields:

- icl1: icl value reach with this solution for alpha=1
- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats: a list with the following elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - x_counts: matrix of size K*K with the number of observed links between each pair of clusters
  - x_counts_obs: matrix of size K*K with the number of observed dyads between each pair of clusters

logalpha value of log(alpha)

ggtree data.frame with complete merge tree for easy plotting with ggplot2

tree numeric vector with merge tree tree[i] contains the index of i father

train_hist data.frame with training history information (details depends on the training procedure)

mm-class *Mixture of Multinomial model description class*

Description

An S4 class to represent a Multinomial model model, extends icl_model-class. Such model can be used to cluster a data matrix X with the following generative model :

$$\pi \sim Dirichlet(\alpha)$$

$$Z_i \sim \mathcal{M}(1, \pi)$$

$$\theta_k \sim Dirichlet(\beta)$$

$$X_{i.}|Z_{ik} = 1 \sim \mathcal{M}(L_i, \theta_k)$$

With $L_i = \sum_d = 1^D X_{id}$. This class mainly store the prior parameters value (α, β) of this generative model in the following slots:

Slots

`name` name of the model

`alpha` Dirichlet over cluster proportions prior parameter (default to 1)

`beta` Dirichlet over vocabulary prior parameter (default to 1)

Examples

```
new("mm")
new("mm",alpha=1,beta=1)
```

mm_fit-class *Mixture of Multinomial fit results class*

Description

An S4 class to represent a fit of a degree corrected stochastic block model for co_clustering, extend icl_fit-class.

Slots

`model` a mm-class object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- x_counts: matrix of size K*D with the number of occurrences of each modality for each clusters

`move_mat` binary matrix which store move constraints

`train_hist` data.frame with training history information (details depends on the training procedure)

`mm_path-class` *Mixture of Multinomial hierarchical fit results class*

Description

An S4 class to represent a fit of a stochastic block model, extend `icl_path-class`.

Slots

`model` a `mm-class` object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- x_counts: matrix of size K*D with the number of occurrence of modality word in each clusters

`path` a list of size K-1 with each part of the path described by:

- icl1: icl value reach with this solution for alpha=1
- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats a list with the following elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - x_counts: matrix of size K*D with the number of occurrence of modality word in each clusters

`logalpha` value of log(alpha)

`ggtree` data.frame with complete merge tree for easy plotting with `ggplot2`

`tree` numeric vector with merge tree `tree[i]` contains the index of `i` father

`train_hist` data.frame with training history information (details depends on the training procedure)

multistarts-class *Greedy algorithm with multiple start class*

Description

An S4 class to represent a greedy algorithm with multiple start (extends alg-class class).

Slots

nb_start number of random starts (default to 10)

multsbm-class *Multinomial Stochastic Block Model class*

Description

An S4 class to represent a Multinomial Stochastic Block Model, extends icl_model-class. Such model can be used to cluster multilayer-graph vertex, and model a square adjacency cube X of size NxNxM with the following generative model :

$$\pi \sim Dirichlet(\alpha)$$

$$Z_i \sim \mathcal{M}(1, \pi)$$

$$\theta_{kl} \sim Dirichlet(\beta)$$

$$X_{ij.}|Z_{ik}Z_{jl} = 1 \sim \mathcal{M}(L_{ij}, \theta_{kl})$$

With $L_{ij} = \sum_{m=1}^{M} X_{ijm}$. This class mainly store the prior parameters value α, β of this generative model in the following slots:

Slots

name name of the model

alpha Dirichlet over cluster proportions prior parameter

beta Dirichlet prior parameter over Multinomial links

type define the type of networks (either "directed" or "undirected", default to "directed")

See Also

multsbm_fit-class, multsbm_path-class

multsbm_fit-class *Multinomial Stochastic Block Model fit results class*

Description

An S4 class to represent a fit of a Multinomial Stochastic Block Model, extend icl_fit-class.

Slots

model a multsbm-class object to store the model fitted

name generative model name

icl icl value of the fitted model

K number of extracted clusters over row and columns

cl a numeric vector with row and columns cluster indexes

obs_stats a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- x_counts: cube of size KxKxM with the number of links between each pair of clusters

move_mat binary matrix which store move constraints

train_hist data.frame with training history information (details depends on the training procedure)

multsbm_path-class *Multinomial Stochastic Block Model hierachical fit results class*

Description

An S4 class to represent a hierarchical fit of a Multinomial Stochastic Block Model, extend icl_path-class.

Slots

model a multsbm-class object to store the model fitted

name generative model name

icl icl value of the fitted model

K number of extracted clusters over row and columns

cl a numeric vector with row and columns cluster indexes

obs_stats a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- x_counts: matrix of size KxKxM with the number of links between each pair of clusters

path a list of size K-1 with each part of the path described by:

- icl1: icl value reach with this solution for alpha=1

- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats: a list with the elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - x_counts: matrix of size KxKxM with the number of links between each pair of clusters

`logalpha` value of log(alpha)

`ggtree` data.frame with complete merge tree for easy plotting with `ggplot2`

`tree` numeric vector with merge tree `tree[i]` contains the index of `i` father

`train_hist` data.frame with training history information (details depends on the training procedure)

`mvmreg-class` *Multivariate mixture of regression model description class*

Description

An S4 class to represent a multivariate mixture of regression model, extends `icl_model-class`. The model follows minka-linear . The model corresponds to the following generative model:

$$\pi \sim Dirichlet(\alpha)$$
$$Z_i \sim \mathcal{M}(1, \pi)$$
$$V_k \sim \mathcal{W}(\varepsilon^{-1}, n_0)$$
$$A_k \sim \mathcal{MN}(0, (V_k)^{-1}, \tau X^t X)$$
$$Y_{i.}|X_{i.}Z_{ik} = 1 \sim \mathcal{N}(A_k x_{i.}, V_k^{-1})$$

with $\mathcal{W}(\epsilon^{-1}, n_0)$ the Whishart distribution and $\mathcal{MN}$ the matrix-normal distribution.

Slots

`name` name of the model

`alpha` Dirichlet over cluster proportions prior parameter (default to 1)

`tau` Prior parameter (inverse variance) default 0.01

`epsilon` Covariance matrix prior parameter (default to NaN, in this case epsilon will be fixed to a diagonal variance matrix equal to 0.1 time the variance of the regression residuals with only one cluster.)

`N0` Prior parameter (default to NaN, in this case N0 will be fixed equal to the number of columns of Y.)

Examples

```
new("mvmreg")
new("mvmreg",alpha=1,tau=0.1,N0=15)
```

mvmreg_fit-class — *Clustering with a multivariate mixture of regression model fit results class*

Description

An S4 class to represent a fit of a multivariate mixture of regression model, extend icl_fit-class.

Slots

`model` a mvmreg-class object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- regs: list of size K with statistics for each clusters

`move_mat` binary matrix which store move constraints

`train_hist` data.frame with training history information (details depends on the training procedure)

mvmreg_path-class — *Multivariate mixture of regression model hierarchical fit results class*

Description

An S4 class to represent a hierarchical fit of a multivariate mixture of regression model, extend icl_path-class.

Slots

`model` a mvmreg-class object to store the model fitted

`name` generative model name

`icl` icl value of the fitted model

`K` number of extracted clusters over row and columns

`cl` a numeric vector with row and columns cluster indexes

`obs_stats` a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- regs: list of size K with statistics for each clusters

`path` a list of size K-1 with each part of the path described by:

- icl1: icl value reach with this solution for alpha=1
- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats: a list with the following elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - regs: list of size K with statistics for each clusters

`logalpha` value of log(alpha)

`ggtree` data.frame with complete merge tree for easy plotting with `ggplot2`

`tree` numeric vector with merge tree `tree[i]` contains the index of `i` father

`train_hist` data.frame with training history information (details depends on the training procedure)

`NMI` *Compute the normalized mutual information of two discrete samples*

Description

Compute the normalized mutual information of two discrete samples

Usage

```
NMI(cl1, cl2)
```

Arguments

`cl1`	vector of discrete labels
`cl2`	vector of discrete labels

Value

the normalized mutual information between the two discrete samples

Examples

```
cl1 =sample(2,500,replace=TRUE)
cl2 =sample(2,500,replace=TRUE)
NMI(cl1,cl2)
```

nodelinklab *nodelinklab*

Description

nodelinklab

Usage

```
nodelinklab(sol, labels, s = 0)
```

Arguments

sol	mm_path-class object to be plot
labels	a vector of cluster labels
s	threshold for links

Value

a ggplot2 graph

plot,co_dcsbm_fit,missing-method *plot a co_dcsbm_fit-class*

Description

plot a co_dcsbm_fit-class

Usage

```
## S4 method for signature 'co_dcsbm_fit,missing'
plot(x, type = "blocks")
```

Arguments

x	a co_dcsbm_fit-class
type	a string which specify plot type: • 'blocks': plot a block matrix with summarizing connections between row and column clusters • 'nodelink': plot a nodelink diagram of the bipartite graph summarizing connections between row and column clusters

Value

a ggplot2 graphic

`plot,co_dcsbm_path,missing-method`
plot a `co_dcsbm_path-class`

Description

plot a `co_dcsbm_path-class`

Usage

```
## S4 method for signature 'co_dcsbm_path,missing'
plot(x, type = "blocks")
```

Arguments

x	a `co_dcsbm_path-class`
type	a string which specify plot type: • `'blocks'`: plot a block matrix with summarizing connections between row and column clusters • `'nodelink'`: plot a nodelink diagram of the bipartite graph summarizing connections between row and column clusters • `'front'`: plot the extracted front ICL, log(alpha) • `'path'`: plot the evolution of ICL with respect to K • `'tree'`: plot the associated dendrograms one for the row clusters and one for the column clusters

Value

a `ggplot2` graphic

`plot,dcsbm_fit,missing-method`
plot a `sbm_fit-class` *object*

Description

plot a `sbm_fit-class` object

Usage

```
## S4 method for signature 'dcsbm_fit,missing'
plot(x, type = "blocks")
```

Arguments

x	a sbm_fit-class
type	a string which specify plot type: • 'blocks': plot a block matrix with summarizing connections between clusters • 'nodelink': plot a nodelink diagram of the graph summarizing connections between clusters

Value

a ggplot2 graphic

---

plot,dcsbm_path,missing-method

plot a sbm_path-class *object*

Description

plot a sbm_path-class object

Usage

```
## S4 method for signature 'dcsbm_path,missing'
plot(x, type = "blocks")
```

Arguments

x	an sbm_path-class object
type	a string which specify plot type: • 'blocks': plot a block matrix with summarizing connections between clusters • 'nodelink': plot a nodelink diagram of the graph summarizing connections between clusters • 'front': plot the extracted front in the plane ICL, log(alpha) • 'path': plot the evolution of ICL with respect to K • 'tree': plot the associated dendrogram

Value

a ggplot2 graphic

---

plot,diaggmm_path,missing-method
plot a diaggmm_path-class *object*

Description

plot a diaggmm_path-class object

Usage

```
## S4 method for signature 'diaggmm_path,missing'
plot(x, type = "tree")
```

Arguments

x	a diaggmm_path-class
type	a string which specify plot type: • 'front': plot the extracted front ICL, log(alpha) • 'path': plot the evolution of ICL with respect to K • 'tree': plot the associated dendrogram

Value

a ggplot2 graphic

---

plot,gmm_path,missing-method
plot a gmm_path-class *object*

Description

plot a gmm_path-class object

Usage

```
## S4 method for signature 'gmm_path,missing'
plot(x, type = "tree")
```

Arguments

x	a gmm_path-class
type	a string which specify plot type: • 'front': plot the extracted front ICL, log(alpha) • 'path': plot the evolution of ICL with respect to K • 'tree': plot the associated dendrogram

Value

a ggplot2 graphic

`plot,misssbm_fit,missing-method`
plot a `misssbm_fit-class` *object*

Description

plot a `misssbm_fit-class` object

Usage

```
## S4 method for signature 'misssbm_fit,missing'
plot(x, type = "blocks")
```

Arguments

x	a `misssbm_fit-class`
type	a string which specify plot type: • `'blocks'`: plot a block matrix with summarizing connections between clusters • `'nodelink'`: plot a nodelink diagram of the graph summarizing connections between clusters

Value

a ggplot2 graphic

`plot,misssbm_path,missing-method`
plot a `misssbm_path-class` *object*

Description

plot a `misssbm_path-class` object

Usage

```
## S4 method for signature 'misssbm_path,missing'
plot(x, type = "blocks")
```

Arguments

x	an misssbm_path-class object
type	a string which specify plot type: • 'blocks': plot a block matrix with summarizing connections between clusters • 'nodelink': plot a nodelink diagram of the bipartite graph summarizing connections between clusters • 'front': plot the extracted front ICL, log(alpha) • 'path': plot the evolution of ICL with respect to K • 'tree': plot the dendrogram

Value

a ggplot2 graphic

---

plot,mm_fit,missing-method

plot a mm_fit-class *object*

Description

plot a mm_fit-class object

Usage

```
## S4 method for signature 'mm_fit,missing'
plot(x, type = "blocks")
```

Arguments

x	a mm_fit-class
type	a string which specify plot type: • 'blocks': plot a block matrix with summarizing connections between clusters

Value

a ggplot2 graphic

plot,mm_path,missing-method

plot a mm_path-class *object*

Description

plot a mm_path-class object

Usage

```
## S4 method for signature 'mm_path,missing'
plot(x, type = "blocks")
```

Arguments

x	an mm_path-class object
type	a string which specify plot type: • 'blocks': plot a block matrix with summarizing • 'nodelink': plot a nodelink diagram • 'front': plot the extracted front in the plane ICL, log(alpha) • 'path': plot the evolution of ICL with respect to K • 'tree': plot the associated dendrogram

Value

a ggplot2 graphic

plot,multsbm_fit,missing-method

plot a multsbm_fit-class *object*

Description

plot a multsbm_fit-class object

Usage

```
## S4 method for signature 'multsbm_fit,missing'
plot(x, type = "blocks")
```

Arguments

x	a multsbm_fit-class
type	a string which specify plot type: • 'blocks': plot a block matrix with summarizing connections between clusters • 'nodelink': plot a nodelink diagram of the graph summarizing connections between clusters

Value

a ggplot2 graphic

---

plot,multsbm_path,missing-method

plot a sbm_path-class *object*

Description

plot a sbm_path-class object

Usage

```
## S4 method for signature 'multsbm_path,missing'
plot(x, type = "blocks")
```

Arguments

x	an sbm_path-class object
type	a string which specify plot type: • 'blocks': plot a block matrix with summarizing connections between clusters • 'nodelink': plot a nodelink diagram of the bipartite graph summarizing connections between clusters • 'front': plot the extracted front ICL, log(alpha) • 'path': plot the evolution of ICL with respect to K • 'tree': plot the associated dendrogram

Value

a ggplot2 graphic

`plot,mvmreg_path,missing-method`
plot a `mvmreg_path-class` *object*

Description

plot a `mvmreg_path-class` object

Usage

```
## S4 method for signature 'mvmreg_path,missing'
plot(x, type = "tree")
```

Arguments

x	a `mvmreg_path-class`
type	a string which specify plot type: • `'front'`: plot the extracted front ICL, log(alpha) • `'path'`: plot the evolution of ICL with respect to K • `'tree'`: plot the associated dendrogram

Value

a `ggplot2` graphic

`plot,sbm_fit,missing-method`
plot a `sbm_fit-class` *object*

Description

plot a `sbm_fit-class` object

Usage

```
## S4 method for signature 'sbm_fit,missing'
plot(x, type = "blocks")
```

Arguments

x	a `sbm_fit-class`
type	a string which specify plot type: • `'blocks'`: plot a block matrix with summarizing connections between clusters • `'nodelink'`: plot a nodelink diagram of the graph summarizing connections between clusters

Value

a ggplot2 graphic

`plot,sbm_path,missing-method` *plot a* `sbm_path-class` *object*

Description

plot a `sbm_path-class` object

Usage

```
## S4 method for signature 'sbm_path,missing'
plot(x, type = "blocks")
```

Arguments

`x`	an `sbm_path-class` object
`type`	a string which specify plot type: • `'blocks'`: plot a block matrix with summarizing connections between clusters • `'nodelink'`: plot a nodelink diagram of the bipartite graph summarizing connections between clusters • `'front'`: plot the extracted front ICL, log(alpha) • `'path'`: plot the evolution of ICL with respect to K • `'tree'`: plot the associated dendrogram

Value

a ggplot2 graphic

`print,icl_path-method` *print an icl_path object*

Description

Print an `icl_path-class` object, model type and number of found clusters are provided.

Usage

```
## S4 method for signature 'icl_path'
print(x)
```

Arguments

x	icl_path-class object to print

Value

None (invisible NULL). No return value, called for side effects.

---

rdcsbm *Generates graph adjacency matrix using a degree corrected SBM*

Description

rdcsbm returns an adjacency matrix and the cluster labels generated randomly using a Degree Corrected Stochastic Block Model.

Usage

```
rdcsbm(N, pi, mu, betain, betaout)
```

Arguments

N	A numeric value the size of the graph to generate
pi	A numeric vector of length K with clusters proportions. Must sum up to 1.
mu	A numeric matrix of dim K x K with the connectivity pattern to generate, elements in [0,1].
betain	A numeric vector of length N which specify the in-degree correction will be normalized per cluster during the generation.
betaout	A numeric vector of length N which specify the out-degree correction will be normalized per cluster during the generation.

Details

It takes the sample size, cluster proportions and emission matrix, and as input and sample a graph accordingly together with the clusters labels.

Value

A list with fields:

- x: the count matrix as a dgCMatrix
- K: number of generated clusters
- N: number of vertex
- cl: vector of clusters labels
- pi: clusters proportions
- mu: connectivity matrix
- betain: normalized in-degree parameters
- betaout: normalized out-degree parameters

`rlbm` *Generate a data matrix using a Latent Block Model*

Description

`rlbm` returns the adjacency matrix and the cluster labels generated randomly with a Latent Block Model.

Usage

```
rlbm(Nr, Nc, pir, pic, mu)
```

Arguments

`Nr`	desired Number of rows
`Nc`	desired Number of column
`pir`	A numeric vector of length Kr with rows clusters proportions (will be normalized to sum up to 1).
`pic`	A numeric vector of length Kc with columns clusters proportions (will be normalized to sum up to 1).
`mu`	A numeric matrix of dim Kr x Kc with the connectivity pattern to generate. elements in [0,1].

Details

This function takes the desired graph size, cluster proportions and connectivity matrix as input and sample a graph accordingly together with the clusters labels.

Value

A list with fields:

- x: the generated data matrix as a `dgCMatrix`
- clr: vector of row clusters labels
- clc: vector of column clusters labels
- Kr: number of generated row clusters
- Kc: number of generated column clusters
- Nr: number of rows
- Nc: number of column
- pir: row clusters proportions
- pic: column clusters proportions
- mu: connectivity matrix

Examples

```
simu = rlbm(500,1000,rep(1/5,5),rep(1/10,10),matrix(runif(50),5,10))
```

`rmm` *Generate data using a Multinomial Mixture*

Description

`rmm` returns a count matrix and the cluster labels generated randomly with a Mixture of Multinomial model.

Usage

```
rmm(N, pi, mu, lambda)
```

Arguments

`N`	A numeric value the size of the graph to generate
`pi`	A numeric vector of length K with clusters proportions. Must sum up to 1.
`mu`	A numeric matrix of dim k x D with the clusters patterns to generate, all elements in [0,1].
`lambda`	A numeric value which specify the expectation for the row sums.

Details

It takes the sample size, cluster proportions and emission matrix, and as input and sample a graph accordingly together with the clusters labels.

Value

A list with fields:

- x: the count matrix as a `dgCMatrix`
- K: number of generated clusters
- N: number of vertex
- cl: vector of clusters labels
- pi: clusters proportions
- mu: connectivity matrix
- lambda: expectation of row sums

`rmreg` *Generate data from a mixture of regression model*

Description

`rmreg` returns an X matrix, a y vector and the cluster labels generated randomly with a Mixture of regression model.

Usage

```
rmreg(
  N,
  pi,
  mu,
  sigma,
  X = cbind(matrix(stats::rnorm(N * (nrow(mu) - 1)), N, nrow(mu) - 1), rep(1, N))
)
```

Arguments

`N`	A numeric value the size of the graph to generate
`pi`	A numeric vector of length K with clusters proportions (must sum up to 1)
`mu`	A numeric matrix of dim K x d with the regression parameters
`sigma`	A numeric of length 1 with the target conditional variance
`X`	A matrix of covariate

Details

It takes the sample size, cluster proportions and regression parameters matrix and variance as input accordingly

Value

A list with fields:

- X: the covariate matrix
- y: the target feature
- K: number of generated clusters
- N: sample size
- cl: vector of clusters labels
- pi: clusters proportions
- mu: regression parameters
- sigma: conditional variance

`rmultsbm` *Generate a graph adjacency matrix using a Stochastic Block Model*

Description

`rmultsbm` returns the multi-graph adjacency matrix and the cluster labels generated randomly with a Multinomial Stochastic Block Model.

Usage

```
rmultsbm(N, pi, mu, lambda)
```

Arguments

`N`	The size of the graph to generate
`pi`	A numeric vector of length K with clusters proportions (will be normalized to sum up to 1).
`mu`	A numeric array of dim K x K x M with the connectivity pattern to generate. elements in [0,1].
`lambda`	A double with the Poisson intensity to generate the total counts

Details

This function takes the desired graph size, cluster proportions and connectivity matrix as input and sample a graph accordingly together with the clusters labels.

Value

A list with fields:

- x: the multi-graph adjacency matrix as an `array`
- K: number of generated clusters
- N: number of vertex
- cl: vector of clusters labels
- pi: clusters proportions
- mu: connectivity matrix
- lambda:

Examples

```
simu = rsbm(100,rep(1/5,5),diag(rep(0.1,5))+0.001)
```

`rsbm` *Generate a graph adjacency matrix using a Stochastic Block Model*

Description

`rsbm` returns the adjacency matrix and the cluster labels generated randomly with a Stochastic Block Model.

Usage

```
rsbm(N, pi, mu)
```

Arguments

`N`	The size of the graph to generate
`pi`	A numeric vector of length K with clusters proportions (will be normalized to sum up to 1).
`mu`	A numeric matrix of dim K x K with the connectivity pattern to generate. elements in [0,1].

Details

This function takes the desired graph size, cluster proportions and connectivity matrix as input and sample a graph accordingly together with the clusters labels.

Value

A list with fields:

- x: the graph adjacency matrix as a `dgCMatrix`
- K: number of generated clusters
- N: number of vertex
- cl: vector of clusters labels
- pi: clusters proportions
- mu: connectivity matrix

Examples

```
simu = rsbm(100,rep(1/5,5),diag(rep(0.1,5))+0.001)
```

sbm-class *Stochastic Block Model class*

Description

An S4 class to represent a Stochastic Block Model, extends icl_model-class. Such model can be used to cluster graph vertex, and model a square adjacency matrix X with the following generative model :

$$\pi \sim Dirichlet(\alpha)$$

$$Z_i \sim \mathcal{M}(1, \pi)$$

$$\theta_{kl} \sim Beta(a_0, b_0)$$

$$X_{ij}|Z_{ik}Z_{jl} = 1 \sim \mathcal{B}(\theta_{kl})$$

This class mainly store the prior parameters value α, a_0, b_0 of this generative model in the following slots:

Slots

`name` name of the model

`alpha` Dirichlet over cluster proportions prior parameter (default to 1)

`a0` Beta prior parameter over links (default to 1)

`b0` Beta prior parameter over no-links (default to 1)

`type` define the type of networks (either "directed" or "undirected", default to "directed"), for undirected graphs the adjacency matrix is supposed to be symetric.

References

Nowicki, Krzysztof and Tom A B Snijders (2001). "Estimation and prediction for stochastic block structures". In:Journal of the American statistical association 96.455, pp. 1077–1087

See Also

sbm_fit-class,sbm_path-class

greed

Examples

```
new("sbm")
new("sbm",a0=0.5, b0= 0.5,alpha=0.5)
sbm = rsbm(100,c(0.5,0.5),diag(2)*0.1+0.01)
sol = greed(sbm$x,model=new("sbm",a0=0.5, b0= 0.5,alpha=0.5))
```

sbm_fit-class *Stochastic Block Model fit results class*

Description

An S4 class to represent a fit of a Stochastic Block Model, extend icl_fit-class.

Slots

model a sbm-class object to store the model fitted

name generative model name

icl icl value of the fitted model

K number of extracted clusters over rows and columns

cl a numeric vector with row and columns cluster indexes

obs_stats a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- x_counts: matrix of size K*K with the number of links between each pair of clusters

move_mat binary matrix which store move constraints

train_hist data.frame with training history information (details depends on the training procedure)

sbm_path-class *Stochastic Block Model hierarchical fit results class*

Description

An S4 class to represent a hierarchical fit of a stochastic block model, extend icl_path-class.

Slots

model a sbm-class object to store the model fitted

name generative model name

icl icl value of the fitted model

K number of extracted clusters over row and columns

cl a numeric vector with row and columns cluster indexes

obs_stats a list with the following elements:

- counts: numeric vector of size K with number of elements in each clusters
- x_counts: matrix of size K*K with the number of links between each pair of clusters

path a list of size K-1 with that store all the solutions along the path. Each element is a list with the following fields:

- icl1: icl value reach with this solution for alpha=1
- logalpha: log(alpha) value were this solution is better than its parent
- K: number of clusters
- cl: vector of cluster indexes
- k,l: index of the cluster that were merged at this step
- merge_mat: lower triangular matrix of delta icl values
- obs_stats: a list with the following elements:
  - counts: numeric vector of size K with number of elements in each clusters
  - x_counts: matrix of size K*K with the number of links between each pair of clusters

`logalpha` value of log(alpha)

`ggtree` data.frame with complete merge tree for easy plotting with `ggplot2`

`tree` numeric vector with merge tree `tree[i]` contains the index of `i` father

`train_hist` data.frame with training history information (details depends on the training procedure)

`seed-class` *Greedy algorithm with seeded initialization*

Description

An S4 class to represent a greedy algorithm with initialization from spectral clustering and or k-means (extends `alg-class` class).

`spectral` *Regularized spectral clustering*

Description

performs regularized spectral clustering of a sparse adjacency matrix

Usage

```
spectral(X, K)
```

Arguments

`X`	An adjacency matrix in sparse format (see the `Matrix` package)
`K`	Desired number of cluster

Value

cl Vector of cluster labels

References

Tai Qin, Karl Rohe. Regularized Spectral Clustering under the Degree-Corrected Stochastic Block Model. Nips 2013.

to_multinomial *Convert a binary adjacency matrix with missing value to a cube*

Description

Convert a binary adjacency matrix with missing value to a cube

Usage

```
to_multinomial(X)
```

Arguments

X	A binary adjacency matrix with NA

Value

a cube

Xvlegislature *French Parliament votes dataset*

Description

French Parliament votes dataset

Usage

```
data(Xvlegislature)
```

Format

An object of class `list` with three fields;

X matrix of deputy votes a `sparseMatrix` of size 593x1839

rowmeta a data frame with deputy meta-data

colmeta a data frame with votes meta data

Examples

```
data(Xvlegislature)
```

Index