Package 'clustNet'

February 14, 2024

Type Package

Title Network-Based Clustering

Version 1.2.0

Description Network-
based clustering using a Bayesian network mixture model with optional covariate adjustment.

Depends R (>= 3.5.0)

Encoding UTF-8

License GPL-3

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, ggraph, ggpubr, ggplot2, grDevices,
reshape2, car, ks, testthat (>= 3.0.0)

VignetteBuilder knitr

Imports BiDAG (>= 2.0.2), pcalg, RBGL, parallel, clue, methods, graph,
igraph

Config/testthat/edition 3

NeedsCompilation no

Author Fritz Bayer [aut, cre, cph],
Jack Kuipers [ctb]

Maintainer Fritz Bayer <frbayer@ethz.ch>

Repository CRAN

Date/Publication 2024-02-14 15:20:02 UTC

R topics documented:

---

bestAICsearch   *bestAICsearch*

Description

best AIC search

Usage

```
bestAICsearch(
  binaryMatrix,
  minK = 2,
  maxK = 5,
  chiVec = c(0.001, 0.5, 1, 2, 3),
  startseed = 100,
  nIterations = 50,
  AICrange = 100,
  plot_heatmap = TRUE
)
```

Arguments

binaryMatrix	Data to be clustered
minK	Min number of clusters
maxK	Max number of clusters
chiVec	Vector of chi values
startseed	Seed
nIterations	Number of iterations
AICrange	AIC range
plot_heatmap	TRUE if plotting directly

Value

list of AIC scrores

density_plot *density_plot*

Description

Create 2d dimensionality reduction of sample fit to Bayesian network clusters

Usage

```
density_plot(cluster_results, var_selection = NULL, colourys = NULL)
```

Arguments

cluster_results

Cluster results from function get_clusters

var_selection   Selected variables to consider, e.g. c(1:5) for first five only

colourys   A vector specifying the colors of each cluster (optional)

Value

A density plot of class recordedplot.

Examples

```
# Simulate data
sampled_data <- sampleData(n_vars = 15, n_samples = c(200,200,200))$sampled_data
# Learn clusters
cluster_results <- get_clusters(sampled_data)
# Load additional pacakges to create a 2d dimensionality reduction
library(car)
library(ks)
library(ggplot2)
library(graphics)
library(stats)
# Plot a 2d dimensionality reduction
density_plot(cluster_results)
```

---

get_classification   *get_classification*

Description

Classification based on clustering

Usage

```
get_classification(cluster_results, data_classify)
```

Arguments

cluster_results

>   Output from get_clusters()

data_classify   Data that should be classified; colnames need to match the ones of cluster_results$data;
missing cols are allowed

Value

a list containing the classification as "clustermembership" and the probabilities of belonging to the
clusters as "allrelativeprobabs"

Examples

```
# choose data
sampled_data <- sampleData(n_vars = 15, n_samples = c(300,300,300))$sampled_data
# learn clusters
cluster_results <- get_clusters(sampled_data)
# visualize the networks
classification_results <- get_classification(cluster_results, sampled_data)
```

---

get_clusters   *get_clusters*

Description

Network-based clustering

Usage

```
get_clusters(
  myData,
  k_clust = 3,
  n_bg = 0,
  quick = TRUE,
  EMseeds = 1,
  edgepmat = NULL,
  blacklist = NULL,
  bdepar = list(chi = 0.5, edgepf = 8),
  newallrelativeprobabs = NULL
)
```

Arguments

myData	Data to be clustered, must be either binary (with levels "0"/"1") or categorical (with levels "0"/"1"/"2"/...)
k_clust	Number of clusters
n_bg	Number of covariates to be adjusted for; the position of the covariates must be in the last column of the myData matrix
quick	if TRUE, then the runtime is quick but accuracy is lower
EMseeds	Seeds
edgepmat	Matrix of penalized edges in the search space
blacklist	Matrix of forbidden edges in the search space
bdepar	Hyperparameters for structure learning (BDE score)
newallrelativeprobabs	
	relative probability of cluster assignment of each sample

Value

a list containing the clusterMemberships and "assignprogress"

Examples

```
# choose data
sampled_data <- sampleData(n_vars = 15, n_samples = c(300,300,300))$sampled_data
# learn clusters
cluster_results <- get_clusters(sampled_data)
# visualize the networks
library(ggplot2)
library(ggraph)
library(igraph)
library(ggpubr)
plot_clusters(cluster_results)
```

---

get_clusters_bernoulli

<div align="center">get_clusters_bernoulli</div>

Description

Categorical version of Bernoulli mixture model (binary clustering function BBMMclusterEM)

Usage

```
get_clusters_bernoulli(
  binaryMatrix,
  chi = 0.5,
  k_clust = 5,
  startseed = 100,
  nIterations = 10,
  verbose = FALSE
)
```

Arguments

binaryMatrix	Data to be clustered
chi	hyperparameter chi
k_clust	Number of clusters
startseed	Start seed
nIterations	number of iterations
verbose	set TRUE to display progress

Value

a list containing the clusterMemberships

---

nice_DAG_plot   *nice_DAG_plot*

Description

DAG visualization

Usage

```
nice_DAG_plot(
  my_DAG,
  print_direct = TRUE,
  node_size = NULL,
  CPDAG = TRUE,
  node_colours = "#fdae61",
  directed = TRUE
)
```

Arguments

`my_DAG`	DAG
`print_direct`	print DAG if TRUE
`node_size`	node size vector
`CPDAG`	if TRUE, then plot CPDAG instead of DAG
`node_colours`	node colours
`directed`	TRUE if nodes should be directed

Value

A plot of the DAG of class c("gg", "ggplot").

`plot_clusters`	*plot_clusters*

Description

Plot clusters

Usage

```
plot_clusters(
  cluster_results,
  node_colours = "#fdae61",
  scale_entropy = FALSE,
  directed = TRUE
)
```

Arguments

`cluster_results`	
	Cluster results
`node_colours`	node colours
`scale_entropy`	if true, entropy measure will be used to determine size of the nodes
`directed`	TRUE if nodes should be directed

Value

A summary plot of all cluster networks of class c("gg", "ggplot", "ggarrange").

Examples

```
# Simulate data
sampled_data <- sampleData(n_vars = 15, n_bg = 0)$sampled_data
# learn clusters
cluster_results <- get_clusters(sampled_data)
# Load additional pacakges to visualize the networks
library(ggplot2)
library(ggraph)
library(igraph)
library(ggpubr)
# Visualize networks
plot_clusters(cluster_results)
```

---

  sampleData   *sampleData*

Description

Sample binary data from different Bayes nets

Usage

```
sampleData(
  k_clust = 3,
  n_vars = 20,
  n_bg = 0,
  n_samples = NULL,
  bgedges = "different",
  equal_cpt_bg = TRUE
)
```

Arguments

k_clust	Number of clusters
n_vars	Number of variables
n_bg	number of conditioned covariates
n_samples	number of samples
bgedges	type of background edges
equal_cpt_bg	specify if conditional probability table of the background edges is constant across clusters

Value

sampled binary data

Examples

```
# sample data
simulation_data <- sampleData(k_clust = 3, n_vars = 15, n_samples = c(200,200,200))
sampled_data <- simulation_data$sampled_data
head(sampled_data)
```

Index