Package 'hbcm'

June 3, 2024

Type Package

Title HBCM model for cluster analysis

Version 0.0.0.9000

Date 2023-08-11

Author Xiang Li

Maintainer Xiang Li <xiangli2pro@gmail.com>

Description Package for heterogeneous block covariance model to conduct cluster analysis of network data.

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr,
ggplot2,
MASS,
Matrix,
MCMCpack,
Rcpp,
reshape2,
RSpectra,
stats

LinkingTo Rcpp

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

R topics documented:

`hbcm-package` *A short title line describing what the package does*

Description

A more detailed description of what the package does. A length of about one to five lines is recommended.

Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

Author(s)

Your Name, email optional.

Maintainer: Your Name <your@email.com>

References

This optional section can contain literature or other references for background information.

See Also

Optional links to other man pages

Examples

```
## Not run:
   ## Optional simple examples of the most important functions
   ## These can be in \dontrun{} and \donttest{} blocks.

## End(Not run)
```

`colMat_reorder` *Visulize the clustering results*

Description

'colMat_reorder( )' reorder the columns of affinity matrix by specified groups.

'colMat_heatMap( )' plots the heatmap of affMatrix ordered by groups.

'crossValid( )' 2-fold crossValidation selects the number of clusters.

Usage

```
colMat_reorder(affMatrix, centers, labels)

colMat_heatMap(
  affMatrix,
  centers,
  labels,
  margin = 0.5,
  midpoint = 0,
  limit = c(-1, 1),
  size = 0.2,
  legendName = "",
  title = ""
)

crossValid_func_adjR(x, centers, pt)
```

Arguments

affMatrix	a square affinity matrix.
centers	number of clusters.
labels	label assignment of the columns.
margin	a numeric value adjusting the position of group lines.
midpoint	a numeric value of the midpoint of heatmap range.
limit	a numeric range of the values.
size	size of the groupline.
legendName	a character name of the legend.
title	a character name of the heatmap.
x	an input matrix.
pt	partition times.

Value

matReorder	affinity matrix reordered by groups.
groupLens	a vector of group lengths.

---

data_gen *Generate simulation data and parameters*

Description

'data_gen( )' generates matrix data (x) and parameters (alpha, Lambda, Sigma) for experimenting with the HBCM model.

Usage

```
data_gen(n, p, centers, mu, omega, labels, size, hparam_func)
```

Arguments

n	an integer specifying the number of observations (rows) of data x.
p	an integer specifying the number of variables (columns) of data x.
centers	an integer specifying the number of clusters.
mu	a vector of size ‘centers‘ specifying the mean vector of the multivariate normal distribution of alpha.
omega	a matrix of size ‘centers x centers‘ specifying the covariance matrix of the multivariate normal distribution of alpha.
labels	a vector of size ‘p‘ specifying the cluster labels of the variables.
size	an integer specifying the number of simulation data sets.
hparam_func	a list of size two specifying the function of generating random numbers of parameter Lambda and Sigma.

Value

A list of lists.

x_list	a list of data x.
alpha_list	a list of alpha.
hlambda_list	a list of Lambda.
hsigma_list	a list of Sigma.

Examples

```
n <- 500
p <- 500
centers <- 3

mu <- rep(0, centers)

off_diag <- 0.5
omega <- diag(rep(1, centers))
for (i in 1:centers) {
 for (j in 1:centers) {
   if (i!=j){
     omega[i,j] = off_diag
   }
 }
}

ppi <- rep(1/centers, centers)
labels <- sample(c(1:centers), size = p, replace = TRUE, prob = ppi)

size <- 5
hparam_func <- list(
  lambda_func = function(p) stats::rnorm(p, 0, 1),
  sigma_func = function(p) stats::rchisq(p, 2) + 1
)
data_list <- data_gen(n, p, centers, mu, omega, labels, size, hparam_func)
```

data_gene_group *Transcript Factor (TF) regulators of yeast genes.*

Description

The raw data is downloaded from the supplementary file of paper https://www.tandfonline.com/doi/suppl/10.1080/15384
OR, see github https://github.com/xiangli2pro/hbcm/tree/main/inst/extdata/ to access the raw data.
Check github https://github.com/xiangli2pro/hbcm/tree/main/data-raw/yeast_gene_data.R to see how data is processed.

Usage

```
data_gene_group
```

Format

A data frame with three variables: `Probe`, `Gene`, `TF_regulator`.

data_gene_sample *Yeast gene expression from the experiment.*

Description

The raw data is downloaded the paper https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75694
OR, see github https://github.com/xiangli2pro/hbcm/tree/main/inst/extdata/ to access the raw data.
Check github https://github.com/xiangli2pro/hbcm/tree/main/data-raw/yeast_gene_data.R to see how data is processed.

Usage

```
data_gene_sample
```

Format

A data frame with 241 variables (genes).

heterogbcm *Estimate the optimal posterior distribution of the data column labels.*

Description

'heterogbcm( )' gives the optimal posterior distribution of the labels, which can be used to derive the optimal label assignment of the data columns.

'obj_logL' calculates the negative log-likelihood function of data fitted into the HBCM model

Usage

```
heterogbcm(x, centers, tol, iter, iter_init = 3, labels, verbose = FALSE)

obj_qc(x, centers, ppi, omega, qalpha, hlambda, hsigma)

obj_qalpha(x, centers, omega, qc, hlambda, hsigma)

obj_ppi(centers, qc)

obj_omega(centers, qalpha)

obj_hlambda(x, centers, qc, qalpha)

obj_hsigma(x, centers, qc, qalpha, hlambda)

obj_logL(x, centers, ppi, omega, qc, qalpha, hlambda, hsigma)
```

Arguments

x	a numeric matrix data.
centers	an integer specifying the number of clusters.
tol	numerical tolerance of the iteration updates.
iter	number of iterations.
iter_init	number of iterations of parameters initial estimation, default is 3.
labels	a vector specifying the cluster labels of the columns of x.
verbose	if TRUE, print iteration information.
ppi	probability of multi-nulli distribution.
omega	group-correlation matrix.
qalpha	posterior distribution of parameter vector alpha.
hlambda	heterogeneous parameter vector Lambda.
hsigma	heterogeneous parameter vector Sigma.
qc	posterior distribution of labels.

Value

A list of values.

omega	estimated optimal group-correlation matrix.
hlambda	estimated optimal heterogeneous parameter Lambda.
hsigma	estimated optimal heterogeneous parameter Sigma.
obj_logL_val	vector of -logL from each iteration.
qalpha	estimated optimal posterior distribution of the alpha.
qc	estimated optimal posterior distribution of the column labels.
cluster	a vector of integers (from 1:k) indicating the cluster to which each column is allocated.

---

heterogbcm_hparam *Estimate the optimal posterior distribution of the data column labels.*

Description

'heterogbcm_hparam( )' gives the optimal posterior distribution of the labels, use random initial values for hlambda and hsigma.

'heterogbcm_constant_hlambda_hsigma( )' experiment function to keep hlambda and hsigma constant across simulation

'heterogbcm_logL( )' experiment function to compare the initial logL and last logL.

'heterogbcm_iterStep( )' experiment function with different update algorithm first update q2 and parameters Theta, then update q1 and parameters Theta.

'heterogbcm_noInitLabel( )' experiment function with no initial guess of the labels.

'heterogbcm_qcDiscrete( )' experiment function to update qc with discrete values.

Usage

```
heterogbcm_hparam(
  x,
  centers,
  tol,
  iter,
  labels,
  verbose = FALSE,
  init_hlambda,
  init_hsigma
)

heterogbcm_constant_hlambda_hsigma(
  x,
  centers,
  tol,
  iter,
  iter_init = 3,
  labels,
```

```
  verbose = FALSE
)

heterogbcm_logL(x, centers, tol, iter, iter_init = 3, labels, verbose = FALSE)

heterogbcm_iterStep(
  x,
  centers,
  tol,
  iter,
  iter_init = 3,
  labels,
  verbose = FALSE
)

heterogbcm_noInitLabel(
  x,
  centers,
  tol,
  iter,
  iter_init = 3,
  labels = NA,
  verbose = FALSE
)

heterogbcm_qcDiscrete(
  x,
  centers,
  tol,
  iter,
  iter_init = 3,
  labels,
  verbose = FALSE
)
```

Arguments

x	a numeric matrix data.
centers	an integer specifying the number of clusters.
tol	numerical tolerance of the iteration updates.
iter	number of iterations.
labels	a vector specifying the cluster labels of the columns of x.
verbose	if TRUE, print iteration information.
init_hlambda	initial values for parameter vector Lambda.
init_hsigma	initial values for parameter vector Sigma.
iter_init	iteration times of initial parameter estimation

Value

A list of values.

omega	estimated optimal group-correlation matrix.
hlambda	estimated optimal heterogeneous parameter Lambda.
hsigma	estimated optimal heterogeneous parameter Sigma.
obj_logL_val	vector of -logL from each iteration.
qc	estimated optimal posterior distribution of the column labels.
cluster	a vector of integers (from 1:k) indicating the cluster to which each column is allocated.

`init_hparam` *Initial estimation of heterogeneous parameters Lambda and Sigma*

Description

'init_hparam( )' gives the initial estimation of parameters Lambda and Sigma.

calculate the number of pairs that hlambda and hsigma have the same sign

print iteration info

'init_omega( )' gives the initial estimation of group-correlation matrix omega.

Usage

```
init_hparam(x, centers, labels, tol, iter, verbose = FALSE)

init_hparam0(x, tol, iter, verbose = FALSE)

hparam_sign(vec1, vec2)

verbose_print_alpha(verbose, param_name, min_val, x, qalpha, hlambda, hsigma)

init_omega(x, centers, labels, hlambda, hsigma)

obj_init_qalpha(x, hlambda, hsigma)

obj_qalpha_logL(x, qalpha, hlambda, hsigma)

obj_init_hlambda(x, qalpha)

obj_init_hsigma(x, qalpha, hlambda)
```

Arguments

x	a numeric matrix data.
centers	an integer specifying the number of clusters.
labels	a vector specifying the cluster labels of the columns of x.
tol	numerical tolerance of the iteration updates.
iter	number of iterations.
verbose	if TRUE, print iteration information.
vec1	a vector of hlambda estimation.

vec2	a vector of hlambda estimation.
param_name	name of the parameter being updated.
min_val	minimum objective-value being calculated.
qalpha	distribution of parameter vector alpha.
hlambda	heterogeneous parameter vector Lambda.
hsigma	heterogeneous parameter vector Sigma.

Value

A list of values.

hlambda	optimal initial estimation of Lambda.
hsigma	optimal initial estimation of Sigma.

`matchLabel` *Calculate the rand-index of two label assignments*

Description

'matchLabel( )' calculate the rand-index and adjusted rand-index of estimated label assignment compared to true label assignment, which can be used to evaluate the performance of the estimated label. The metric takes value between 0 and 1, and higher value indicates better performance.

'calRand( )' calculates the rand-index from a confusion matrix.

'rSpecc( )' a customized spectral clustering model.

Usage

```
matchLabel(reference, label)

calRand(confmat)

rSpecc(x, centers, iter.max = 100, nstart = 10)
```

Arguments

reference	true label assignment.
label	estimated label assignment.
confmat	a 2-dimensional confusion matrix.
x	numeric matrix of data.
centers	the number of clusters.
iter.max	the maximum number of iterations allowed.
nstart	how many random sets in the kmeans step should be chosen? default is 10.

Value

`Rand`	rand-index
`adjRand`	adjusted rand-index
`.Data`	A vector of integers indicating the cluster to which each point is allocated.
`size`	The number of points in each cluster.
`totss`	The total sum of squares.
`withinss`	Vector of within-cluster sum of squares, one component per cluster.
`tot.withinss`	Total within-cluster sum of squares, i.e. ‘sum(withinss)‘.
`betweenss`	The between-cluster sum of squares, i.e. ‘totss-tot.withinss‘.

Index