Package 'MLRClust'

July 22, 2025

Type Package

Title Randomized spectral clustering for large-scale multi-layer networks

Version 2.0

Date 2025-07-20

Author Wenqing Su

Maintainer Wenqing Su <suwenqing21@gmail.com>

Description This package implements spectral clustering for large-scale multi-layer directed and undirected networks using randomization techniques.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp, stats, Matrix, irlba, gtools

LinkingTo Rcpp, RcppEigen

SystemRequirements C++11

RoxygenNote 7.3.2

NeedsCompilation yes

Suggests knitr,
rmarkdown

VignetteBuilder knitr

Contents

MathOverflow *MathOverflow multilayer network data*

Description

A directed multilayer network of interactions on the stack exchange website Math Overflow, with 3 layers corresponding to users' answers to questions(a2q), comments to questions(c2q), and comments to answers(c2a). The network includes 23,654 users, i.e., the multilayer network consists of 23,654 nodes.

Usage

```
data(MathOverflow)
```

Format

The `MathOverflow` object is a list of sparse matrices representing the adjacency matrices of the MathOverflow interaction network. All matrices share the same node set with consistent node IDs and dimnames.

References

Paranjape, A., Benson, A. R., and Leskovec, J. (2017). *Motifs in temporal networks.*, *In Proceedings of the tenth ACM International Conference on Web Search and Data Mining.*

Examples

```
data(MathOverflow)
mlA <- MathOverflow
timing <- system.time({
cluster <- mrcoclust(mlA, rank.r = 2, rank.c = 3, kr = 2, kc = 3, q = 4, p = 0.9)
})
print(timing)
```

mln_generator *Generate multilayer networks from multilayer stochastic block models and multilayer stochastic co-block models.*

Description

Generate adjacency matrices of multilayer undirected networks from multilayer stochastic block models and multilayer directed networks from multilayer stochastic co-block models.

Usage

```
mln_generator(row.label, col.label, probmat, directed = FALSE, sparse = TRUE)
```

Arguments

`row.label` The row community vector (from `1:kr`) with the numbers indicating which row community each node is assigned.

`col.label` The column community vector (from `1:kc`) with the numbers indicating which column community each node is assigned. `row.label` and `col.label` must have the same length but not necessarily indicate the same communities and community numbers. Only required when `directed = TRUE`.

`probmat` List of link probability matrices with dimension `c(kr, kc)`. `probmat[l][i,j]` is proportional to the probability of an edge from nodes in row community `i` to nodes in column community `j` in layer `l`.

`directed` If `TRUE`, the multilayer directed networks is generated from the multilayer stochastic co-block model. If `FALSE`, the multilayer undirected networks is generated from the multilayer stochastic block model. Default is `FALSE`.

`sparse` Logical indicating whether to return sparse matrices (`dgCMatrix`) for network adjacency matrices. Default is `TRUE`.

Details

This function generates multilayer undirected (directed) networks using multilayer stochastic block (co-block) models.In the multilayer stochastic block (co-block) models, all layers share common row and column communities but with possibly different edge densities. The layer-wise networks are generated independently from the stochastic block (co-block) model.

Value

A list of adjacency matrices representing the generated multilayer networks.

References

W. Su, X. Guo, X. Chang and Y. Yang. (2024) *Spectral co-clustering in multi-layer directed networks*, *Computational Statistics & Data Analysis, Vol. 198, 107987*
https://doi.org/10.1016/j.csda.2024.107987

Examples

```
# The multilayer stochastic block model
n <- 300
kr <- 2
row.label <- sample(rep(1:kr, each = n/kr), n)
probmat1 <- matrix(0.05, kr, kr) + diag(c(0.1, 0.05))
probmat2 <- matrix(0.01, kr, kr) + diag(c(0.05, 0.02))
probmats <- list(probmat1, probmat2, probmat2)
mln1 <- mln_generator(row.label = row.label, probmat = probmats, directed = FALSE)

# The multilayer stochastic co-block model
n <- 300
kr <- 2
kc <- 3
row.label <- sample(rep(1:kr, each = n/kr), n)
col.label <- sample(rep(1:kc, each = n/kc), n)
probmat1 <- matrix(0.05, kr, kc)
probmat1[1,1] <- 0.1
```

```
probmat1[2,2] <- 0.1
probmat2 <- matrix(0.01, kr, kc)
probmat1[1,1] <- 0.05
probmat1[2,3] <- 0.03
probmats <- list(probmat1, probmat2, probmat2)
mln2 <- mln_generator(row.label, col.label, probmats, directed = TRUE)
```

mrclust *Randomized spectral clustering for multilayer stochastic block models*

Description

Randomized spectral clustering for multilayer undirected networks. Randomized clustering uses both the random sampling strategy and the random projection strategy. Can deal with very large networks.

Usage

```
mrclust(
  mlA,
  rank.t,
  k,
  q = 2,
  p = 0.7,
  iter.max = 50,
  nstart = 10,
  nthread = 1
)
```

Arguments

`mlA` The adjacency matrices list of a multilayer undirected network consists of "`dgCMatrix`" types.

`rank.t` The target rank of the low-rank decomposition.

`k` The number of clusters.

`q` The power parameter. It need to be a positive integer number. Default is 2.

`p` The sampling probability. Should be between 0 and 1. Default is 0.7.

`iter.max` Maximum number of iterations in the `kmeans`. Default is 50.

`nstart` The number of random sets in `kmeans`. Default is 10.

`nthread` Maximum number of threads for specific computations which could be implemented in parallel. Default is 1.

Details

This function computes the clusters of multilayer undirected networks using randomized spectral clustering algorithms. Random sampling is first performed on the adjacency matrices, then the random projection-based eigendecomposition is performed on the aggregated matrix. The k-means is then performed on the randomized eigenvectors.

Value

`cluster` The cluster vector (from 1:k) with the numbers indicating which cluster each node is assigned.

Examples

```
n <- 500
k <- 2
label.true <- sample(rep(1:k, each = n/k), n)
probmat1 <- matrix(0.05, k, k) + diag(c(0.1, 0.05))
probmat2 <- matrix(0.01, k, k) + diag(c(0.05, 0.02))
probmats <- list(probmat1, probmat2, probmat2)
mlA <- mln_generator(row.label = label.true, probmat = probmats, directed = FALSE)
rank.t <- 2
q <- 2
p <- 0.7
mrclust(mlA, rank.t, k = k, q = q, p = p)

U <- matrix(c(1/2, 1/2, -sqrt(2)/2, 1/2, 1/2, sqrt(2)/2, sqrt(2)/2, -sqrt(2)/2, 0), nrow=3, byrow=TRUE)
Lambda1 <- diag(c(1.5, 0.2, 0.4))
Lambda2 <- diag(c(1.5, 0.2, -0.4))
B_1 <- U %*% Lambda1 %*% t(U)
B_2 <- U %*% Lambda2 %*% t(U)
layer <- 20
density <- 0.1
probmats <- lapply(1:layer, function(l) {if (l <= floor(layer / 2)) B_1 * density else B_2 * density})
k <- 3
# The effect of power parameter. (q = 2, 4, 6)
p <- 0.7 # sampling probability
nodes_list <- seq(400, 2000, 200)
mr.result <- matrix(0, length(nodes_list), 4) # misclassification rates
for (i in 1:length(nodes_list)) {
  n_nodes <- nodes_list[i]
  size_communities <- c(0.3, 0.4, 0.3) * n_nodes
  true_label <- sample(rep(seq_len(k), times=size_communities), n_nodes)
    for (j in 1:5){
      mlA <- mln_generator(row.label = true_label, probmat = probmats, directed = FALSE)
      # randomized spectral clustering q = 2
      cluster.r.q2 <- mrclust(mlA, rank.t = k, k = k, q = 2, p = p)
    mr.result[i, 1] <- mr.result[i, 1] + min_mis_error(true_label, cluster.r.q2, k_clusters = k) / 5
      # randomized spectral clustering q = 4
      cluster.r.q4 <- mrclust(mlA, rank.t = k, k = k, q = 4, p = p)
    mr.result[i, 2] <- mr.result[i, 2] + min_mis_error(true_label, cluster.r.q4, k_clusters = k) / 5
      # randomized spectral clustering q = 6
      cluster.r.q6 <- mrclust(mlA, rank.t = k, k = k, q = 6, p = p)
    mr.result[i, 3] <- mr.result[i, 3] + min_mis_error(true_label, cluster.r.q6, k_clusters = k) / 5
      # original spectral clustering
      cluster.sc <- dsos(mlA, rank.t = k, k = k)
    mr.result[i, 4] <- mr.result[i, 4] + min_mis_error(true_label, cluster.sc, k_clusters = k) / 5
  }
}
line_labels <- c("RSC q = 2", "RSC q = 4", "RSC q = 6", "SC")
matplot(nodes_list, mr.result, type = "l", lty = 1, col = 1:4, xlab = "Number of nodes (n)", ylab = "Misclassific
legend("topright", legend = line_labels, col = 1:4, lty = 1)
```

mrcoclust *Randomized spectral co-clustering for multilayer stochastic co-block models*

Description

Randomized spectral co-clustering for multilayer directed networks. Randomized clustering uses both the random sampling strategy and the random projection strategy. Can deal with very large networks.

Usage

```
mrcoclust(
  mlA,
  rank.r,
  rank.c,
  kr,
  kc,
  q = 2,
  p = 0.7,
  iter.max = 50,
  nstart = 10,
  nthread = 1
)
```

Arguments

`mlA`	The adjacency matrices list of a multilayer directed network consists of "`dgCMatrix`" types.
`rank.r`	The target row rank of the low-rank decomposition.
`rank.c`	The target column rank of the low-rank decomposition.
`kr`	The number of row clusters.
`kc`	The number of column clusters.
`q`	The power parameter. It need to be a positive integer number. Default is 2.
`p`	The sampling probability. Should be between 0 and 1. Default is 0.7.
`iter.max`	Maximum number of iterations in the `kmeans`. Default is 50.
`nstart`	The number of random sets in `kmeans`. Default is 10.
`nthread`	Maximum number of threads for specific computations which could be implemented in parallel. Default is 1.

Details

This function computes the clusters of multilayer directed networks using randomized spectral co-clustering algorithms. Random sampling is first performed on the adjacency matrices, then the random projection-based eigendecomposition is performed on the aggregated matrix. The k-means is then performed on the randomized eigenvectors.

Value

row.cluster	The row cluster vector (from 1:kr) with the numbers indicating which row cluster each node is assigned.
col.cluster	The column cluster vector (from 1:kc) with the numbers indicating which column cluster each node is assigned.

Examples

```
n <- 600
kr <- 2
kc <- 2
row.label.true <- sample(rep(1:kr, each = n/kr), n)
col.label.true <- sample(rep(1:kc, each = n/kc), n)
probmat1 <- matrix(0.05, kr, kc) + diag(c(0.1, 0.05))
probmat1[1,2] <- 0.08
probmat2 <- matrix(0.01, kr, kc) + diag(c(0.05, 0.02))
probmat2[2,1] <- 0.1
probmats <- list(probmat1, probmat2, probmat2)
mlA <- mln_generator(row.label.true, col.label.true, probmats, directed = TRUE)
rank.r <- 2
rank.c <- 2
q <- 2
p <- 0.7
cc <- mrcoclust(mlA, rank.r, rank.c, kr, kc, q = q, p = p)
```

Index